A DEEP LEARNING MODEL TO RECOGNIZE COMMUNICATION-ORIENTED ENTITY OF ICT IN CONSTRUCTION

WU HENGQIN

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and Harbin Institute of Technology

2021

The Hong Kong Polytechnic University

Department of Building and Real Estate

Harbin Institute of Technology

School of Management

A Deep Learning Model to Recognize Communication-oriented Entity of ICT in Construction

WU Hengqin

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August, 2019

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

Signed

WU Hengqin

Abstract

Due to the fragmented nature and complexity of the industry, effective communication is increasingly recognized as a key factor to enable real-time transfer of information and to achieve the success of construction projects. Information and communication technology (ICT) has been recognized as an important determinant to enable and enhance the communication. A large volume of unformatted textual data is available in digital forms in the current web 2.0 era, and it seems no exception for the field of ICT in construction. There is a number of experts who are skilled in the ICT and familiar with the construction industry, leaving a large volume of technical documents (such as handbooks, patents, literature, and reports) embodied with professional and skilled knowledge. This motivates this study to investigate those textual data of ICT in construction.

The key component of ICT in construction is the communication functionality, forming the communicating process whereby construction data is coordinated during the whole life cycle of construction projects. Most of the functionalities' specifications are archived as written language in the technical documents of ICT in construction, mainly by referring three types of communication-oriented entity (CE): transferred information, communication models and communication subjects. This study seeks to develop an entity recognition approach that can automatically identify the CEs from raw text and categorize them into pre-defined entity types.

Entity recognition (sometimes called concept extraction, term identification or event extraction) has become an important approach in recent informatics studies of the Construction Engineering and Management (CEM) domain, aiming to automatically extract all the entities (sometimes called

concepts, textual elements, textual items, or terminologies) from raw text that are relevant to a specific domain. Recently, a small number of CEM studies utilized rule-based and pre-defined vocabularies to enable automatic entity recognition to create digital dictionaries. The problems of these approaches are two-fold. First, these methods require expertise knowledge and enormous efforts for the establishment of rules and vocabularies, which are extremely time-consuming and labor-intensive. In addition, pre-defined vocabularies exclude unknown entities (the entities exist but are not known by the developer), resulting in the missing of numerous relevant entities. Second, those methods always lead to lower accuracy in discerning ambiguous entities (the spellings can appear as an entity at one position and common noun at another position, or appear as different entity types). Those methods typically utilized external lexical databases such as WordNet to discern ambiguous entities, but the performance is still unsatisfactory due to the limited coverage of the lexical databases and ignoring contextual information.

To remedy these problems, the proposed approach resorts to the techniques from the realm of deep learning that can utilize the contextual information within raw texts. In recent years, deep learning techniques have been recognized as a powerful tool to aid human beings in solving complex tasks to explore and utilize the unstructured text data in a robust way, in which the embedded information can be extracted in a readable manner. The key merit of deep learning is its capacity to utilize the contextual information within raw texts, rather than the external linguistic resources that are required in previous studies. The proposed deep learning model is developed through the transformer, a novel neural network structure with self-attention mechanism, which enables parallelly computing among the input sequence in memorizing contextual information. In this way, the whole model improves performance compared with traditional RNN in which the contextual information is computed sequentially.

Specifically, the objectives of this study are as follows:

(1) To develop a deep learning model for entity recognition which can automatically recognize CEs from raw text through utilizing the contextual information.

(2) To compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER).

(3) To train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings.

This study first reviewed relevant research about ICT in construction and entity recognition to identify the research gaps. The basic concepts of CER of ICT in construction were clarified and the technical problems of CER were highlighted. Considering the technical problems embodied in recognizing CEs from the technical documents of ICT in construction, a deep learning approach based on transformer was developed to recognize CEs from the technical documents of ICT in construction, which enables the whole model to understand the contextual meanings. The patents of ICT in construction were compiled as an initial dataset, and the training and testing instances for CER were achieved by manually tagging CEs in each sentence of the database. The validation

was carried in two perspectives: (1) The accuracy of the deep learning model in recognizing the CEs was validated, compared to the traditional RNN-based deep learning model; (2) The practical value of recognized CEs was validated by applying the CEs as features to achieve a classification scheme that categorizes patents of ICT in construction into different communication modes.

The key findings achieved from this study are as follows. First, a deep learning model was developed based on the transformer for entity recognition that was tailored to CER of ICT in construction. The validation results indicated that it outperformed the traditional RNN-based deep learning model, yielding better performance by 15%. Second, through the training process, the developed model acquired intelligence that is able to recognize a CE by addressing the contextual information that surrounds the CE. Thirdly, a database of patents of ICT in construction was compiled, and the CEs of each patent was recognized by the deep learning model. The recognized CEs are more valuable and informative than common words in indicating the communication patterns of ICT in construction, resulting in better performance when they were used as features for classifying the patents into different communication modes.

This study contributes to knowledge in three aspects. **First**, a deep learning model was developed for entity recognition, which can identify and classify the entities out of common words by utilizing the contextual meanings of the surrounding text, rather than the word-level features and syntactic rules that were used in previous construction informatics studies. In this way, the proposed approach can recognize unknown entities (entities beyond a pre-defined vocabulary) and improve the performance in recognizing ambiguous entities. **Second**, the developed model utilizes the “self-attention” mechanism to capture the contextual meanings in recognizing the CEs through the training process, leading to better performance compared to the traditional RNN-based model. The trained deep learning model acquires the intelligence like human beings to recognize the CEs by understanding the contextual meanings rather than the word-level meanings. **Third**, this study contributes an effective NLP approach for the practitioners to access and perceive the communication functionality underlying the patents of ICT in construction. This can help acknowledge how the up-to-data inventions employ and utilize devices to scheme the information flows.

Publications

Refereed Journal Papers

1. **Hengqin Wu**, Geoffrey Qiping Shen, Xue Lin*, Minglei Li and Boyu Zhang. Screening Patents of ICT in Construction Using Deep Learning and NLP Techniques. *Engineering, Construction and Architectural Management*, 2020, 27(8), 1891-1912. (**SCI/SSCI**, **IF: 2.160**)
2. **Hengqin Wu**, Xiaolong Xue, Zebin Zhao*, Zeyu Wang and Geoffrey Qiping Shen. Major Knowledge Diffusion Paths of Mega-Project Management: A Citation-Based Analysis. *Project Management Journal*, 2020, *51*(3), 242-261. (**SSCI, IF: 2.506**)
3. **Hengqin Wu**, Zebin Zhao, Xiaolong Xue*, Geoffrey Qiping Shen, Rebecca Jing Yang and Luqi Wang. An Integrated Scientometric and SNA Approach to Explore the Classics in CEM Research. *Journal of Civil Engineering and Management*, 2020, 26(5), 459-474. (**SCI, IF: 2.338**)
4. Xue Lin, Haibo Zhang, **Hengqin Wu*** and Dongjin Cui. Mapping the knowledge development and frontier areas of public risk governance research. *International Journal of Disaster Risk Reduction*, 2020, 43, 101365. (**SCI, IF: 2.896**)
5. Xiaolong Xue*, **Hengqin Wu**, Xiaoling Zhang, Jason Dai, Chang Su. Measuring energy consumption efficiency of the construction industry: the case of China. *Journal of Cleaner Production*, 2015, 107: 509-515. (**SCI, IF: 5.651**)
6. Xiao Li, Geoffrey Qiping Shen, Peng Wu, Hongqin Fan, **Hengqin Wu** and Yue Teng. RBL-PHP: Simulation of Lean Construction and Information Technologies for Prefabrication Housing Production. *Journal of Management in Engineering*, 34(2), 04017053,2018.
7. Tengfei Huo, Hong Ren, Weiguang Cai, Geoffrey Qiping Shen, Bingsheng Liu, Minglei Zhu and **Hengqin Wu**. Measurement and Dependence Analysis of Cost Overruns in Megatransport Infrastructure Projects: Case Study in Hong Kong. *Journal of Construction Engineering and Management*, 2018, 144 (3).

8. Xiaozhi Ma, Albert P. C. Chan, **Hengqin Wu**, Feng Xiong, Na Dong. Achieving Leanness with Bim-Based Integrated Data Management in a Built Environment Project. *Construction Innovation-England*, 2018, 18 (4): 469-487

Conference Papers

1. **Hengqin WU**, Xiaolong Xue, Geoffrey Qiping Shen and Yazhuo Luo. Mapping the Knowledge Structure in Megaproject Management Research Using Complex Network Analysis, 2017 International Conference on Construction and Real Estate Management: Project Management and Construction Technology, ICCREM 2017, November 10, 2017 - November 12, 2017. American Society of Civil Engineers (ASCE), Guangzhou, China, pp. 82-88, 2017.
2. Rui Liu, Xiaolong Xue, **Hengqin Wu**. Identifying Knowledge Structures in Construction Innovation Research Using the Mapping Knowledge Domain Method[C]. Proceedings Of 2014 International Conference on Construction & Real Estate Management, ASCE,2014, 922-930
3. Ruixue Zhang, Xiaolong Xue, **Hengqin Wu**, Liu Rui. Research on the Synergetic Evolution of Collaborative Innovation Systems in Construction[C]. Proceedings Of 2014 International Conference on Construction & Real Estate Management, ASCE,2014, 265-272

Refereed Journal Papers – Under Review

1. **Hengqin Wu**, Geoffrey Qiping Shen and Xue LIN. Communication-oriented entity recognition of ICT in construction: A deep learning approach modelling contextual representations. *Automation in Construction.*

Acknowledgements

I would like to take this opportunity to express my deepest and sincere appreciation to all those people who helped me go through the Ph.D. journey.

First and foremost, I would like to show my most sincere gratitude to my supervisor Prof. Geoffrey Q.P. Shen. He contributed his time and resources to guide my study without reservation. He generously provided valuable advice during my topic choice, methods design, and data collection and analysis. He cared for my study progress and helped me to resolve the difficulties. Thanks for his continuously encouragements and supports along with the study.

My sincere gratitude also goes to Prof. Xiaolong Xue (the Dean of School of Management, Guangzhou University), for his guidance in this thesis. I also wish to express my appreciation to Dr. Hongqin Fan, Dr. Jack Chin Pang CHENG and Prof. H. David JEONG, for their kind comments and support in my research.

My thanks are extended as well to many persons that have given supports to cover the challenge, especially the collogues in Sustainable Construction Lab, ZN 1004 and ZN 710, Dr. Jingke Hong, Dr. Fan Xue, Dr. Wei Zheng, Dr. Shan Guo, Dr. Zhengdao Li, Dr. Shanshan Bu, Dr. Bingxia Sun, Dr. Lizi Luo, Jin Xue, Emma Liqun Xiang, and Xin Jin. I am also grateful for the help of CIB members, Dr. Ma Xiaozhi and Ming Luo.

I wish to extend my thanks to Joint Ph.D students from Harbin Institute of Technology, Dr. Yukuan Xu, Dr. Ying Xia, Dr. Peng Luo, Dr. Fei Liu and Yu Jin. In addition, I wish to thank the members

of YiRen badminton club, Dr. Zhijian He, Dr. Xue Pang, Dr. Fuliang Li, Dr. Guang Xiao, Dr. Dong Yang, Dr. Xu Xu, and Dr. Zejia Zhao.

Getting through my dissertation required more than technical and academic support, and I have to express my special appreciation for the patience of many, many people. Dr. Minglei Li, Prof. Han Lin, Dr. Wenjing Yang, and Dr. Xue Lin have been constant in their personal and professional support without any hesitation during my Ph.D. career. I also have to thank many people for listening to and, at times, having to tolerate me over the past four years, including Dr. Xiao Li, Dr. Pengkun Wu, Dr. Juan Huang, Boyu Zhang and Fang He.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction

This chapter introduces the research proposition of this study. First, this chapter clarifies the research background in terms of why communication-oriented entity is the concern, why entity recognition approaches are important for the CEM domain, and why a deep learning model is needed. Second, this chapter describes the research scope. Third, the overall design is described. Finally, research significance is highlighted.

1.2 Research Background

1.2.1 Why Communication-oriented Entity is the Concern

Due to the fragmented nature and complexity of the industry, effective communication is increasingly recognized as a key factor to achieve the success of construction projects (Cheng Eddie et al., 2001), enabling real-time transfer of information (Heinzerling and Strube, 2018; Sardroud, 2015). Targeting at forming and enhancing such communication, ICT in construction enables access, store, transmission, and manipulation of information throughout life-cycle of construction projects (Heinzerling and Strube, 2018; Hosseini et al., 2013; Sardroud, 2015; Van Slyke and Belanger, 2003).

ICT is an extensional concept that incorporates a wide range of technical approaches, mainly concentrating on communication functionality (Mathur, 2017). In recent years, the value of successful applications of ICT has been increasingly recognized as a key factor to enable effective communication in the construction industry (Cheng Eddie et al., 2001).

However, it is challenging for correct adoption and proper implementation of ICT to solve the problems in the communication process. In the construction industry, adoption and innovation of ICT highly depend on trial-and-error experiments or personal judgment to raise ideas to shape or enhance the communication and conceive solutions for the problems in the communication process of construction projects. These traditional methods lacked effective analysis of the communication functionality of previous innovations and overlooked the existing knowledge underlying raw texts in the technical documents (especially the patents) of ICT in construction (Tan, 2007).

A large volume of unformatted textual data exists in the current web 2.0 era (Ittoo et al., 2016). Most technical information is expressed as written language in patents, because they are regarded as the most prominent and up-to-date technology source (Gredel et al., 2012; Kim et al., 2016b; Li et al., 2012), and up to 95% of all inventions can be found in patents (Nave, 2010). It seems not an exception for the information of the communication functionality of ICT in construction. Typically, in the patents of ICT in construction, the detailed specifications of the communication functionality are conveyed through the mentions of how the construction data was transmitted by virtual or physical models and how the data was coordinated between construction sites and users or among the stakeholders (Alsafouri and Ayer, 2018). Examples of such mentions include "installation information was transferred from an RFID tag to a construction item" in an RFID patent (Costin et al., 2012; El Ghazali et al., 2012), and "the technology conveys geographic data to display devices that users could manipulate" in a GIS patent (Deng et al., 2016). These mentions are mainly formed by three types of communication-oriented entities (CEs):

transferred information (TI), communication models (CM), and communication subjects (CS) (Table 1.1 shows the examples of CEs). This study seeks to develop a computer-aided system for proposing an entity recognition approach that can automatically identify the CEs from raw text and categorize them into the pre-defined types.

Table 1.1 Examples of the communication-oriented entity classes

	ICT in construction	**Communication-oriented entities**		
		Transferred information	**Communication models**	**Communication subjects**
1	RFID	Installation information	RFID tag	*Not given*
2	GIS	Geographic data	Display devices	User
3	CAD applications	3D building model	Electronic device	Viewer

1.2.2 Why Entity Recognition Approaches are Important for the CEM domain

From a general perspective, entity recognition (sometimes called concept extraction, term identification, or event extraction) aims to automatically extract all entities that are relevant to a specific domain from raw texts (Zhang and El-Gohary, 2016). The so-called entities (sometimes called concepts, textual elements, textual items, or terminologies) are the names of things for a specific domain, which are the basis in constituting digital dictionaries (such as ontologies, taxonomies, and knowledge graphs). A digital dictionary is fundamentally a claim on knowledge of a domain formed by taxonomies of entities and their relations that can be performed by computer-aided programs (Poli et al., 2010).

Due to its formal representation and interoperability, digital dictionaries, especially ontologies and taxonomies are fast becoming a key instrument for processing textual data and information representation, and an increasing number of CEM studies applied them to address various management issues (i.e. (Lin and Soibelman, 2006; Rezgui, 2006; Staub-French et al., 2003; Zhou et al., 2016)). Despite their successful applications, most of these studies directly applied existing digital dictionaries or developed them using manual methods (El-Diraby et al., 2005; El-Diraby and Kashif, 2005; Seedah et al., 2016; Wetherill et al., 2002), which are typically conducted based on professional experience. To develop more specific digital dictionaries according to varied research objectives, automatic entity recognition approaches have been raised by a small number of CEM studies, identifying entities from raw texts and classifying them into existing concept classes of digital dictionaries of the CEM domain (Le and Jeong, 2017; Zhang and El-Gohary, 2016).

1.2.3 Why Deep Learning

The existing automatic entity recognition approaches were mainly developed by manual rules and pre-defined vocabularies, which are still unsatisfactory for real-world cases. First, these methods require professional knowledge and the construction process of man-made rules is expensive. In addition, the pre-defined vocabulary excludes unknown entities (the entities exist but are not known by the developer). Second, these methods always lead to lower accuracy in discerning ambiguous entities (the spellings can appear as an entity at one position and common noun at another position, or appear as different entity types). Those methods typically utilized external lexical databases such as WordNet to discern ambiguous entities (Eftimov et al., 2017; Gaizauskas et al., 1995; Grishman and Sundheim, 1996b; Ravin and Wacholder, 1997), but the performance is not acceptable due to the limited coverage of the lexical databases and ignoring contextual information (Le and

Jeong, 2017). In the case of communication-oriented entities, the ambiguous entities can be numerous, because the most of mentions of CE are common technical terms, such as “sensor”, “database”, “display device”, “map information”, “location information” etc.

To remedy these problems, the proposed approach resorts to the techniques from the realm of deep learning that can utilize contextual information within raw texts for CE recognition (CER). Deep learning techniques have been widely used in NLP tasks, fast becoming necessary tools to explore and utilize unstructured text data in a robust way, in which the information in unstructured data could be represented in a readable way. Prior studies that investigated the linguistic materials in the construction industry have merely employed techniques from traditional machine learning or text mining methods (Al Qady and Kandil, 2014; Fan et al., 2015a; Gao et al., 2015; Le and Jeong, 2017; Tixier et al., 2016; Zhou and El-Gohary, 2016; Zou et al., 2017), while the increasingly advanced techniques of deep learning have been overlooked, regardless the improvements gained in many NLP studies by the application of deep learning models (Araque et al., 2017; Dong et al., 2018; Guan et al., 2019; Kratzwald et al., 2018; Mahmoudi et al., 2018).

This study developed the deep learning model for constituting the communication-oriented entity recognition approach based on the transformer, which was proposed in 2017 by Google AI team (Vaswani et al., 2017), and has gained increased attention from researchers in the field of NLP. The core function lies in the deep learning model is the capacity to learn and utilize the contextual information. This means the proposed model, rather than utilizing the word-level features and hard rules that were used in previous studies, identify and classify the entities out of common words by utilizing the contextual information of surrounding text, which is recognized as an important ability for computer-aided systems to achieve intelligence to perform NLP tasks (Zhu and Iglesias, 2018).

1.3 Research Scope

This study focuses on ICT in construction. According to the definition, ICT is the information technologies that primarily focus on the communication functionality (Akarowhe, 2017). As a result, this study concerns the information technologies that are mainly utilized to measure, monitor, transfer, access, and display construction information in the whole life cycle of a construction project.

This study focuses on the expressions of the communication functionality of ICT in construction embedded in raw text. On one hand, the communication functionality plays an important role in ICT applications, forming the process in which the construction data is coordinated during the whole life cycle of construction projects. On the other hand, most technical information is expressed as written language in patents, and it seems not an exception for the information of communication functionality of ICT in construction.

This study focuses on automatic approaches of entity recognition. These approaches have gained increasing attention because of their efficiency in the extraction of entities from raw text, thus facilitating the creation of digital dictionaries of CEM domain.

This study focuses on approaches based on deep learning techniques for entity recognition. Deep learning is fast becoming a vital instrument in assisting human to deal with complex tasks. Recently, in the field of NLP, many deep learning models were proposed to extract useful information from textual data. Therefore, this study develops a deep learning model with neural network and NLP techniques and algorithms to recognize the CEs from technical documents of ICT in construction.

1.4 Research Aim and Objectives

The primary aim of this study is to develop a deep learning model to recognize the CEs from the technical documents of ICT in construction. Specifically, the objectives of this study are as follows:

(1) To develop a deep learning model for entity recognition which can automatically recognize CEs from raw text through utilizing the contextual information.

(2) To compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER).

(3) To train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings.

1.5 Research Design

Figure 1-1 illustrates the overall framework of this study.

(1) Through critical review about the ICT in construction and entity recognition (ER), the research gaps are highlighted.

(2) The basic concepts of CER and the embodied technical problems are clarified. Taking the technical problems into consideration, a deep learning model with a complex neural network architecture based on the transformer is developed, involving various neural networks and NLP techniques, such as byte-pair encoding, wordpiece tokenization, position embedding, self-attention algorithm, etc.

(3) NLP techniques, such as part-of-speech (PoS), tokenization and lemmatization are employed integrating the Multi-Layer Perceptron (MLP) model to screen the patents of ICT in construction for compiling a patent database. Based on the database, the annotated data for training the deep learning model of CER are achieved by manually labeling.

(4) The performance of the proposed deep learning model is validated. The so-called 10-fold cross-validation method is employed to evaluate the performance. To shed lights on advances of the developed model in overcoming the technical problems in CER of ICT in construction, the empirical validation would be implemented compared with Bi-directional LSTM+CNN (BLC), another deep learning model that was reported as one of the most advanced deep learning models for ER.

(5) The practical value of recognized CEs is validated by using the CEs as features to achieve a classification scheme that categorizes the patents of ICT in construction into different communication modes.

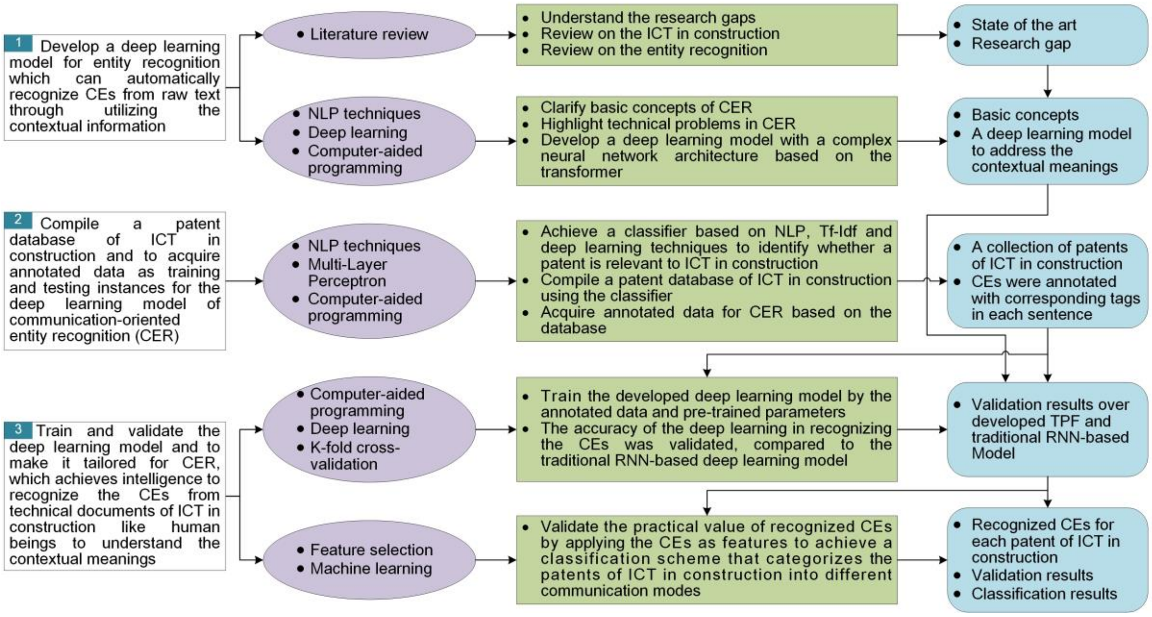

1 Develop a deep learning model for entity recognition which can automatically recognize CEs from raw text through utilizing the contextual information
• Literature review
• NLP techniques
• Deep learning
• Computer-aided programming
• Understand the research gaps
• Review on the ICT in construction
• Review on the entity recognition
• Clarify basic concepts of CER
• Highlight technical problems in CER
• Develop a deep learning model with a complex neural network architecture based on the transformer
• State of the art
• Research gap
• Basic concepts
• A deep learning model to address the contextual meanings
2 Compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER)
• NLP techniques
• Multi-Layer Perceptron
• Computer-aided programming
• Achieve a classifier based on NLP, Tf-Idf and deep learning techniques to identify whether a patent is relevant to ICT in construction
• Compile a patent database of ICT in construction using the classifier
• Acquire annotated data for CER based on the database
• A collection of patents of ICT in construction
• CEs were annotated with corresponding tags in each sentence
3 Train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings
• Computer-aided programming
• Deep learning
• K-fold cross-validation
• Train the developed deep learning model by the annotated data and pre-trained parameters
• The accuracy of the deep learning in recognizing the CEs was validated, compared to the traditional RNN-based deep learning model
• Validation results over developed TPF and traditional RNN-based Model
• Feature selection
• Machine learning
• Validate the practical value of recognized CEs by applying the CEs as features to achieve a classification scheme that categorizes the patents of ICT in construction into different communication modes
• Recognized CEs for each patent of ICT in construction
• Validation results
• Classification results

Figure 1-1 Overall research framework of the thesis

1.6 Research Significance

(1) The proposed deep learning model contributes an entity recognition approach based on a deep learning model by utilizing the contextual meanings of the surrounding text, rather than utilizing word-level features and syntactic rules that were used in previous construction informatics studies. Automatic approaches of entity recognition forms a key component to establish customized digital dictionaries with specific purposes of the CEM domain. To establish digital dictionaries of the CEM domain, existing entity recognition approaches were mainly developed by manual rules and pre-defined vocabularies. Although these methods enable an automatic process in which entities can be recognized from raw text, they suffer two main drawbacks: (1) these methods are time-consuming and labor-intensive, which are conducted based on personal knowledge and experience. Moreover, pre-defined vocabularies cannot recognize the unknown entities, which may result in the missing of numerous related entities; and (2) lower accuracy in discerning

ambiguous entities and unknown entities. Alternatively, the proposed entity recognition approach, through the developed deep learning model, utilize contextual information to recognize entities from raw texts. In this way, the proposed approach, after a supervised learning process, obtain the intelligence to understand the contextual meaning underlying the raw text, and thus can recognize entities out of raw text merely based on the surrounding text rather than the external linguistic resources that are required in previous studies.

(2) The deep learning model utilizes the "self-attention" mechanism to capture the contextual meanings in recognizing the CEs, leading to better performance compared to the traditional RNN-based models. Through the training process, the deep learning model acquires the intelligence like human beings to recognize CEs by understanding the contextual meanings rather than the word-level meanings. Recognizing a word or phrase as CEs from a text highly depends on the surrounding information. The technical documents of ICT in construction incorporate a number of general technical entities, such as "location information", "building data", "camera device", "mobile device", etc. These entities may be expressed as different meanings and usages according to the surrounding written language, and only a part of them may be the CEs that are used for communication-information coordination and transmission.

(3) In practice, this study contributes a high-level and effective NLP approach for the practitioners in construction projects to extract essential information of the communication functionality in the patents of ICT in construction. Based on the compiled database of patents of ICT in construction and the recognized CEs, a communication-oriented retrieval system allows users to access the patents of ICT in construction according to the

communication scenarios they face in practice by querying CEs, providing an efficient and smart method to understand the communication functionality about how the up-to-date inventions of ICT employ and utilize devices to scheme the information flows. Successful implementation of ICT requires professional experience for linking existing inventions to the communication needs in practice. Due to the fragment and knowledge-intensive nature of the construction industry, it is a challenging work for the managers in the construction projects to correctly adapt and properly implement ICT applications to enable and enhance the communication. This retrieval system provides an efficient method for users to access the core information of the communication functionality embedded in the raw text, forming a more effective approach to connect the potential inventions of ICT in construction to the communication problems to be solved.

1.7 Structure of the Thesis

The study incorporates nine chapters.

Chapter 1 introduces the whole research, by indicating why CER is performed (research background section), summarizing the scope of relevant concepts, proposing the main objectives to be achieved in this study (research aim and objectives section), describing the methodology route of this study (research design section), highlighting the significance to the knowledge (research significance section), as well as the structure of the study.

Chapter 2 reviews relevant literature of ICT in construction to examine how the ICT in construction was investigated in the prior research. Totally, three perspectives were reviewed: categories ICT in construction, the current use of ICT, and barriers of ICT adoption in the construction industry. This chapter also summarizes the research gaps to stress the significance of this study.

Chapter 3 reviews relevant CEM and NLP literature that developed automatic entity recognition approaches. Then, the disadvantages of existing approaches are proposed.

Chapter 4 presents the methodologies employed in this study. This chapter summarizes the techniques corresponding to each research objective. Each of the employed research technique is illustrated, such as NLP techniques, annotation techniques, deep learning models, and validation methods.

Chapter 5 describes the architecture of the developed deep learning model for CER of ICT in construction. First, this chapter clarifies the basic definitions of CE and the defined three CE classes. In addition, technical problems of CER of ICT in construction are described. Second, this chapter illustrates the motivations for developing the whole structure of neural networks based on the transformer. After that, the basic components and in each layer of a single transformer are described. The training techniques are illustrated at the end of this chapter.

Chapter 6 compiles a patent database of ICT in construction and annotates the data for training the deep learning model. This chapter screens a corpus of patents that are relevant to ICT in construction by using Multi-Layer Perceptron. Furthermore, based on the database, annotated data for the CER task were labeled manually.

Chapter 7 presents the experiments, validation results and the implications from the results. This chapter first introduces the training process in terms of the training and testing instances and the empirical setup. After that, the validation results were reported to validate the deep learning model for CER task compared to Bi-directional LSTM+CNN (BLC) model. Finally, the advantages of the deep learning model were described through discussing the model's abilities to cope with the aforementioned difficulties in CER task in the context of the patents of ICT in construction.

Chapter 8 describes the applications of the deep learning model of ICT in construction. This chapter firstly describes the potential applications of recognized CEs in two perspectives: direct and indirect applications. Secondly, the chapter presents a specific application, utilizing the recognized CEs as features to train a classifier to categorize the patents of ICT in construction into pre-defined communication modes. This study assumes that the recognized CEs are more informative than common words in discerning communication modes. Therefore, this chapter validates the classification based on CEs as features compared to common words as features.

Chapter 9 reviews the main research findings covered in this study. The contributions to the body of knowledge and practice in the construction industry are highlighted. After that, the limitations and recommendations for future research are discussed.

1.8 Summary of the Chapter

This chapter outlines the overall research proposition, in terms of introducing the research background, presenting the basic motivations to initial this study, and describing the research aim and objectives, scope of the study, research design and significance.

Chapter 2 Literature Review: ICT in Construction

2.1 Introduction

This chapter critically reviews past research on ICT in construction from perspectives of classification of ICT in construction, the current use of ICT in construction, benefits and barriers of ICT adoption in the construction industry.

2.2 Classification of ICT in Construction

In the past decade, several attempts have been made to classify the applications of ICT in construction into categories from users' perspectives. Table 2.1 provides a summarization of these classifications in terms of the features to differentiate the ICTs. The first type of classification categorizes ICT applications based on the information exchange among external or internal organizations. Such classification mainly focused on categorizing web-based ICT applications that were adopted to facilitate the information coordination for members in organizations or between organizations. The second type of classification categorizes ICT applications based on the problems to be solved in construction projects. Under such a classification, the ICTs in the same class would be implemented for similar objectives. The third type of classification focuses on communication modes. The implementations of ICTs that enable similar communication processes would be classified into the same class. The details are described in the following Subsections.

Table 2.1 Summary of classifications of ICT in Construction

	Dimension	Authors	Classifications	Features	Items to be classified
1	Internal and External Exchange of Information	Lu et al. (2015)	Web; VR/AR; Wireless; EDI/EDMS/BIM	The factors that impact on the adoption and diffusion	None
2	Solutions to Problems in Practice	Rimmimgton et al. (2015)	Data Storage Mediums; Data Storage Software; Digital Data; Recording Mediums; Computer Devices; Internet; …	Communication interfaces	None
3	Solutions to Problems in Practice	Lam et al. (2009)	Intranets; Extranets	Information exchange	EDMS

4	Communication mode	Alsafouri and Ayer (2018)	Automated bidirectional;
			Unidirectional-site;
			Non-automated Bidirectional
				Flows of ICTs	RFID; 3D laser scanning; quack response; NFC; Mobile computing

2.2.1 Classification Based on Information Exchange

ICT has been introduced into the construction industry for a long time, and some of them, such as internet of thing and electronic document management systems became the normal tools to enhance the electronic communications for stakeholders in construction projects such as design consultants, contractors, owners, site managers and suppliers (Rojas and Songer, 1999; Ruddock, 2006). To this regard, those applied ICTs could be mainly divided into two groups: intranets and extranets. The intranets related to an electronic document management system that frequently used inner organization, whereas web-based collaboration portals are proliferating fast to cater for the transfer of information and instigate work flow processes among multidisciplinary team members (Lam Patrick et al., 2010). Intranets incorporate some software products such as EDMS, Microsoft SharePoint and ERP that are primarily designed for the users within the organization through the internal network (Van Slyke and Belanger, 2003), which has enabled the e-mail and electronic templates functions for information transformation, knowledge sharing and documents delivering. These ICTs can be easily used and thus have been widely applied in small and medium enterprises (Hillier, 2007). ICT of extranet offers the potential to improve communication between inner and internet networks (Finch, 2000).

Such a classification seems simple to some extent, and nowadays the boundary between the ICT of internal and external exchange forms became vague because organizations applied platforms integrates them.

2.2.2 Classification Based on Solutions to Problems in Practice

The most common knowledge about ICT is that they have specific functions to solve problems or facilitate manage processes. By reviewing 145 articles, Lu et al. (2015) proposed a taxonomy system with five categories of ICT that have been mainly applied in the construction industry: Web; VR/AR; Wireless; EDI/EDMS; BIM. The ICT within the same category commonly provide similar communication patterns whereby information can be transferred among parties in a project. Rimmimgton et al. (2015) highlight the communication interface as the key factors to differentiate ICT into certain categories: Data Storage Mediums; Data Storage Software; Digital Data; Recording Mediums; Computer Devices; and Internet.

2.2.3 Classification Based on Communication Modes of ICTs

Integrating certain of the applications of ICT in construction can coordinate information across different project phases and on-site and off-site, improving the communication between project stakeholders (Rimmimgton et al., 2015). To this regard, applications of ICT in construction could be categorized into certain groups according to communication modes. The categories include Unidirectional-Model Coordination, Unidirectional-Site Coordination, Non-automated Bidirectional Coordination, and Automated Bidirectional Coordination. This classification system is complicated in two perspectives. First, ICTs are not divided into disjoint groups. When ICTs can be integrated to achieve the given information coordination pattern, they would be categorized into the given group. Second, the features of each category are complicated, with advanced logics that indicate how the technology combined to impact information coordination.

2.3 Current Use of ICT in Construction

2.3.1 Intranets

Intranets is one of the most frequently used ICT in the construction industry, especially at the organization and individual level. An intranet is built as a private network using local serves that could only be assessed by the members of the organization (Van Slyke and

Belanger, 2003). Typically, its main function is to send and receive electronic documents based on an inter e-mail system. Moreover, it may provide a knowledge base that can enhance knowledge sharing within the organization. In the construction industry, despite these general intranets, an extended electronic document management system (EDMS) was increasingly used to orient a database with an easy assess for users to retrieve relevant documents and figures. This actually forms an efficient communication platform for small and medium companies, especially when standard software applications were adopted such as Microsoft (Hillier, 2007).

Enterprise resource planning (ERP) is another type of intranet that was commonly used in the construction industry. ERP provides a comprehensive solution with a packaged suite of software applications for managing organizational activities, including human resources management, finance and material procurement. ERP significantly improves the efficiency in business affairs in linking the individual events into an all-in-one system (Voordijk et al., 2003).

2.3.2 Extranets

The extranet is usually used between stakeholders in the construction industry by providing access to an intranet of an organization. The shared intranet in an organization allows access through virtual private networks that ensure safety when linking all the stakeholders (Wong, 2007). Another extranet is through the external platforms, which could be easily accessed by registered users, sharing the needed information for project management. This extranet could be accessed regardless of the limitations of users' locations and the installed software products.

2.3.3 BIM

Building information modelling is a platform in which the physical characteristics and functions of any part of the construction industry can be represented in a three-dimensional (3D) model. All the information of relevant parties during the whole life-cycle construction project embodied in the model (such as building components, geographic information, geometry, etc.) can be easily accessed and retrieved by users. BIM can be used to support coordination, construction monitoring, installation methods, constructability, automation, and robot control systems, integration of subcontractor and supplier data as well as worker safety (Vähä et al., 2013). BIM integrates various building information and converts the traditional 3D and 4D models into an n-D model (Aouad et al., 2009). Most of the beneficial effects of BIM are taken place in the design phase, in which BIM aids in the analysis of the conflicts in the design, providing virtual prototyping, and revising and storing the building data. In addition, many ICT in construction are implemented based on BIM. Even though no one can deny the benefits of using BIM in the construction industry, its adoption is still limited, because the computing capacity of current hardware could not meet the need of information computing when using BIM.

2.3.4 Web

The application of web technologies to the construction industry dates back to the very beginning of ICT adoption in the industry. Currently, most of the frequently used ICT inventions were employed based on web technologies (Lam et al., 2009). Web technologies provide a collaborative working environment in which construction-related information could be shared. The advantage of using web technologies in the construction industry comprises of facilitating the access to common documents and removing a sheer volume of papers for contractors (Nitithamyong and Skibniewski, 2004). The major limitation is that web technologies only improve the share of electronic documents.

2.3.5 Geographic Information System

GIS is an information system that embraces a number of functions to receive, form, store, manage and display all types of geographical data. GIS plays an important role in the construction industry, as more than 80% information in the industry is geographically related (Kim et al., 2012). GIS is most often used as a complement to aid the visualization of CAD and BIM in construction. The advantages of the combination of GIS and BIM have been reported in several studies (Irizarry et al., 2013; Karan and Irizarry, 2015). When the data of GIS and BIM interact effectively, the integrated system could provide information of building objects not only in the indoor environments such as geometry but also in outdoor environments (Deng et al., 2016).

2.3.6 Global Positioning System

GPS is one of the tracking and location technologies, which has been widely applied in the construction industry. GPS receives and displays the outdoor position of the objects in 3D and navigates these positions based on triangulation information obtained by a GPS receiver from a number of satellites and ground control stations. Its major application environment is at outdoor, which limits, to some extent, the application in the construction industry in which the materials and products have trajectories between indoor and outdoor locations (Ergen et al., 2007; Martínez-Rojas et al., 2015). In addition, the accuracy of the position information is sensitive to weather conditions.

2.3.7 Radio Frequency Identification

RFID comprises of a collection of identification technologies that can monitor, detect and transmit relevant data from an RFID tag to another or some other reader equipment through radio frequencies. RFID technologies obtain relevant data in an efficient mode, in which the tag could be easily set on any building components and the reader equipment can scan and read more than one RFID tags at one time. RFID techniques have been successfully employed in the construction industry in terms of prefabrication (Li et al., 2014), quality control (Ergen et al., 2007), and life-cycle management (Motamedi and Hammad, 2009).

2.3.8 3D Laser Scanning

The 3D laser scanning technology was firstly developed for surveying engineering (Kim et al., 2015). The major component is a scanner that measures the surface of a physical object represented by point clouds. The scanner emits a laser beam detecting the arrival time of the beam reaching the text object (Yoon et al., 2018). All the detected information is stored as point clouds reflecting the actual spatial performance of the target object (Liu et al., 2018). In the construction industry, 3D laser scanning has been used in a wide range of applications, such as in 3D representation (Kim et al., 2015), assessment of construction structures (Lubowiecka et al., 2009), monitoring the progress of construction (Bernardini and Rushmeier, 2002; Bosché and Guenet, 2014), assessment of precast concrete elements (Kim et al., 2016a; Kim et al., 2015).

2.3.9 Augmented Reality

The application of AR in the construction industry emerged at least a decade (Shin and Dunston, 2010). AR refers to a collection of software, computing devices, input and output devices and algorithms that enable users to interact with the augmented world (Bosché et al., 2016). A computing device manipulates the interaction between the users and AR scenarios by continually receiving the users' reaction and computing the corresponding AR world that is displayed in an interface (Hou et al., 2017). At the beginning, the application AR in the construction industry was mainly embodied in the design and plan domain, such as architectural design (Shin and Dunston, 2010) and urban planning (Anagnostou and Vlamos, 2011). As time goes by, the AR application was extended to facility management (Baek et al., 2019), safety inspection (Sacks et al., 2015) and instruction and construction education (Teizer et al., 2013).

2.4 Summary of the Chapter

Nowadays, due to the sheer scale of unstructured textual data available in digital forms, it is obviously a tough work to manually analyze these data, and it seems no exception for the construction industry. The former studies investigated the ICT in construction in terms of the application of ICT to the construction industry, how the ICT improve the communication and management efficiencies. However, rare studies focused on extracting knowledge from the large corpus of documents of ICT in construction. In addition, a major factor that may block the adoption of ICT in the construction industry is the shortage of relevant knowledge. The technical documents of ICT in construction embody a wide range of knowledge in terms of the technical components, the utilization of relevant inventions, and the problems to be solved. Prior studies extract relevant knowledge and made certain achievements, but the deep learning techniques have not been used in the field of ICT in construction, despite the wide application in many fields.

Chapter 3 Literature Review: Entity Recognition

3.1 Introduction

Entity recognition (ER) is an NLP task that aims to identify and classifies the mentions of entities that can be defined with domain interest. A large and growing body of NLP studies have investigated methods for entity recognition, focusing on introducing NLP and machine learning techniques to improve performance. Entity recognition has been increasingly regarded as an important approach in the CEM domain. A few number of informatics studies developed automatic approaches for entity recognition to create specific digital dictionaries. This chapter introduces the definition and motivation of entity recognition, and reviews entity recognition approaches in CEM studies. Then recent NLP studies about entity recognition approaches are reviewed. Finally, the advantages and disadvantages of current approaches are discussed.

3.1 Definition and Motivation of Entity Recognition

The "Entity" was an NLP concept that was first introduced in 1996 (Grishman and Sundheim, 1996b), and its goal is to automatically identify the names of all people, locations, and organizations using information extraction techniques. At the beginning, such a task was called as "named entity recognition" (NER). As time goes by, entities were no longer limited to the categories above. More generally, an entity is a category of phrases that have similar properties, which are always rigid designators or members of the semantic classes (Goyal et al., 2018). Entities could be defined by domain specifications and interests. For instances, in dietary research, entities of food and nutrient gained increasing interest (Eftimov et al., 2017); in the chemical and gene domain, gene and protein are important entities (Hemati and Mehler, 2019); in general domains, entities normally incorporate location, person, organization and time.

ER task has gained increasing attention after it was proposed, with major embodiments in information extraction of company activities and military messages, by automatically identifying the entities (person, location, time, currency, etc.) from structured data and unstructured data. During the subsequent development, many efforts have been devoted to introducing suited machine learning approaches and algorithms. As ER highly depends on the language features of specific domains, the developed approaches have different performance. The major factors that affect the performance are textual genres and *entity types*. On one hand, currently, many attempts have been made in various domains. The textual genres archived in the unstructured data in those domains varied significantly, which makes ER tasks have varied performance. For example, general domains have a higher performance than biomedicine ER, which entities showed a complicated structure (Shen et al., 2003). On the other hand, entity types can have an important effect on ER performance. Person, location and organization are frequently used for general ER tasks. According to the objectives of tasks, scholars have to set some specific entities types for the target domain, such as miscellaneous weapons, vehicles and facilities for military messages and drug names and disease names for the medical domain.

3.2 Entity Recognition in CEM Studies

From a general perspective, entity refers to concepts, textual elements, textual items, or terminologies that represent information elements in raw text. In the CEM domain, many entity types were developed according to the specific objectives for subfields.

In the CEM domain, entity recognition is regarded as an important approach to extract entities for the creation of digital dictionaries. Digital dictionaries, especially ontologies and taxonomies are fast becoming a key instrument for processing textual data and information representation, and an increasing number of CEM studies applied them to address various management issues (i.e. (Lin and Soibelman, 2006; Rezgui, 2006; Staub-French et al., 2003; Zhou et al., 2016)). Despite their successful conductions, the establishment of digital dictionaries suffered from a major drawback - limited coverage of entities (Le and Jeong, 2017). To create a digital dictionary, conventionally, a developer has to resort to experts to determine which entities are important for constituting the digital dictionary. Therefore, most of the previous studies directly applied existing digital dictionaries or developed them using manual methods (El-Diraby et al., 2005; El-Diraby and Kashif, 2005; Seedah et al., 2016; Wetherill et al., 2002) that are labor-intensive and time-consuming. To address that concern, a small number of approaches have been developed in the CEM domain to automatically recognize entities from raw texts (Le and Jeong, 2017; Zhang and El-Gohary, 2016).

Zhang and El-Gohary (2016) developed their own digital dictionary of building design information to advance the understanding of BIM modeling requirements for compliance checking, thus enabling automated compliance checking. An automatic entity recognition method was developed to extract BIM entities from documents that are relevant to compliance checking, generating an extended digital dictionary with existing industry foundation classes (IFC). The proposed approach automatically identifies entities related to compliance checking (e.g. building codes) and classify them into IFC classes. To enable such an automatic recognition process, the approach integrates manual rules and pre-defined vocabularies. Specifically, part-of-speech (POS) patterns and an exclusion word vocabulary were used to extract the entities, and a semantic-based match method based on WordNet (a lexical database) was used to classify the extracted entities into existing IFC concept classes. Similarly, Le and Jeong (2017) also used rule-based and pre-defined vocabularies to recognize transportation items from documents.

3.3 Entity Recognition in NLP Studies

In 1996, Grishman and Sundheim (1996b) proposed the task MUC-6 that was recognized as the first effort that served entity recognition as an NLP task. This study, from a general aspect, reviews the entity recognition approaches that have been used to analyze texts in the last two decades. According to the functions the systems perform, entity recognition models based on NLP-based techniques could be mainly categorized into three groups: rule-based approaches and *learning-based* approaches. In recent years, scholars prefer to apply deep learning technologies to perform ER tasks, which is actually under the scope of the *learning-based* approaches.

3.3.1 Corpus-based Approaches

Corpus-based models are also called terminology-driven or dictionary-based ER models (Eftimov et al., 2017). This kind of models perform the ER tasks through matching relevant mentions in the text with a pre-defined entity hierarchy consisting of a wide range of terminologies and their relations (Aronson, 2001). The earlier studies prefer to directly find out whether an entity in the pre-defined corpus occurs or not in a sentence (Miller et al., 1992). Such a direct matching approach always leads to poor performance because of ignoring the synonyms. Currently, the corpus-based approaches concern the disambiguation issue by measuring the similarities between the potential entities and those in the corpus, taking the relations and ontologies into consideration (Kumar and Muruganantham, 2016).

The corpus-based approaches are not without disadvantages, and the major one comes from the pre-defined corpus, which was always defined by experts who have professional knowledge for a specific domain. As a result, the performance highly depends on the capability of those experts and is sensitive to the domain's features. This may cause missing some important entities that were not listed in the pre-defined corpus (Eftimov et al., 2017).

3.3.2 Linguistic Rules

At the early stage of the development of entity recognition, the approaches are most build upon the corpus-based and rule-based techniques. The rule-based models recognize entities based on the man-made linguistic rules, (Nadeau and Sekine, 2007), including lexical attributes (i.e., digit patterns (Yu et al., 1998) and morphological attributes (Bick, 2004)), corpus attributes (i.e., coreference and alias attributes (Gaizauskas et al., 1995) and domain-specific attributes (Zhu et al., 2005)). The performance of rule-based models could be excellent when the rules are well established. However, well-established rules may consume a lot of time and energy of rule makers who always are the experts, leading to a much more expensive construction process for the linguistic rules (Nadeau and Sekine, 2007).

3.3.2.1 Lexical attributes

People developed a number of ways to write words in English, resulting in various features in the written language. A simple example is whether a word is capitalized, represented as a Boolean value. Also, a word may have a numeric value denoting the length (the number of the characters). The other word-level attributes are as follows: (1) Digit patterns. Digit patterns may convey a lot of useful information such as data, intervals, and statistical numbers. Some special patterns should be concerned. For example, four digits always stand for an expression of a year, but one or two digits are usually a day or a month (Yu et al., 1998). (2) Morphological attributes. Some words with special ends may reflect a class for entities. For example, some language entities end with “ish”, such as Spanish and Danish (Bick, 2004).

3.3.2.2 Corpus attributes

(1) Coreference and alias. Coreference refers to a word appearing more than once in a document. Alias means the different written versions for an entity. For example, there are several ways to write English names. Some scholars designed certain rules according to the coreference and alias to identify and classify entities (Gaizauskas et al., 1995).

(2) Domain-specific attributes. The document types may provide additional information. For example, the headers in the type of email documents are always good indicators for person and location entities (Zhu et al., 2005).

3.3.2.3 Disadvantages of rule-based models

The performance of rule-based models could be excellent when the rules are well established. However, well-established rules may consume a lot of time and energy of rule makers who always are the experts. This leads to a much more expensive construction process for the rule-based approaches. In addition, the rule makers have to gain better background knowledge about the target domain, which cannot always be satisfied in practice.

3.3.3 Machine Learning

Recent studies often use learning-based approaches to perform entity recognition rather than rule-based models. Learning-based approaches utilize machine learning algorithms to learn the relations between the input features and output entities and their classification. In a general perspective, deep learning methods are a type of machine learning methods. With the rapid development of neural networks, a variety of deep learning models were proposed. To make the clarity of relevant mentions, "machine learning" in this study refers to machine learning methods without the use of deep neural networks. This study reviews the two branches of methods respectively.

3.3.3.1 Supervised learning

Supervised learning approaches model the input features and output labels based on pre-annotated data. For each training instance, the associated features should be represented by measurable indicators, usually quantitated by vectors. These features are fed into the machine learning algorithms to learn the difference between the true and negative labels by recognizing the quantitated features.

The general learning process was highlight by Goyal et al. (2018) and Nadeau and Sekine (2007). Annotated data is necessary for supervised learning. The training instances are typically separate sentences, in which entities are tagged with corresponding labels, recognized manually by experts. The tagged labels, in the learning-based models, serve as the outputs that to be predicted. With respect to inputs, features about the textual objects were selected and further converted to computational forms. According to the input features and output true labels, machine learning algorithms were selected to learn mathematical logic between the inputs and outputs. After the training process, a learned model was achieved and used to perform entity recognition automatically.

The core of the supervised learning models is machine learning algorithms. In the past two decades a number of studies have investigated entity recognition by using certain machine learning algorithms, including: Support Vector Machine (SVM) algorithm (Saha et al., 2010), Conditional Random Field (CRF) algorithm (Majumder et al., 2012), Hidden Markov Model (HMM) algorithm (Wang et al., 2014), Maximum Entropy Markov Model (MEMM) (Saha et al., 2009), etc.

3.3.3.2 Semi-supervised learning

The Semi-Supervised Learning (SSL) methods were derived from the combination of supervised and unsupervised learning, which considers the unlabeled data in the training. The supervised learning approach requires sufficient labeled data to obtain a better-trained model. However, in most cases, especially in speech recognition and textual data, labeled data is always limited. Even though the labeled data can be achieved by manually labeling, but it would consume a considerable amount of time and energy (Pavlinek and Podgorelec, 2017; Zhao et al., 2012).

Semi-supervised learning methods concern the improvement of supervised learning by considering the unlabeled samples. On the other hand, semi-supervised entity recognition was developed to achieve better-defined entities, taking the background knowledge into consideration when using the unlabeled data to aid the unsupervised learning methods (Zhao et al., 2012). Thenmalar et al. (2015) utilized the semi-supervised learning manner to carry out an ER task.

3.3.4 Deep Learning

Deep learning performs entity recognition in a supervised learning manner. The deep learning models distinguished with supervised learning methods in the application of deep neural networks instead of machine learning algorithms. The traditional machine learning algorithms need human intervention to help the system to learn the statistical associations between input and outputs. Deep learning models are the most state-of-the-art approach with a significant improvement in the performance without consideration of the manually made rules. A deep learning model is always used with a feed-forward-based architecture and back-propagation learning process (Goodfellow et al., 2016). There is a number of neurons in the deep learning model, and each of them receives signals from the former layer and passes transformed singles by activation functions to the subsequent layer (Riedmiller, 1994). Deep learning models may have different architectures. In the field of NLP, most of the deep learning models are designed based on Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN).

3.3.4.1 Basic architecture of deep learning models

Similar to most of the current NLP studies, deep learning models for entity recognition are performed based on RNN and CNN. Figure 3-1 plots the general architecture of the neural networks of deep learning models of entity recognition. Although deep learning models varied with each other, they have common input and output layers. Most of the NLP deep learning models adopt the word embeddings as input features to be fed into the main neural network because word embeddings are easy to train and take the contextual and surrounding information into consideration. The output layer in deep learning models always uses Conditional Random Fields (CRF), because CRF concerns the dependencies between the entities in a sentence.

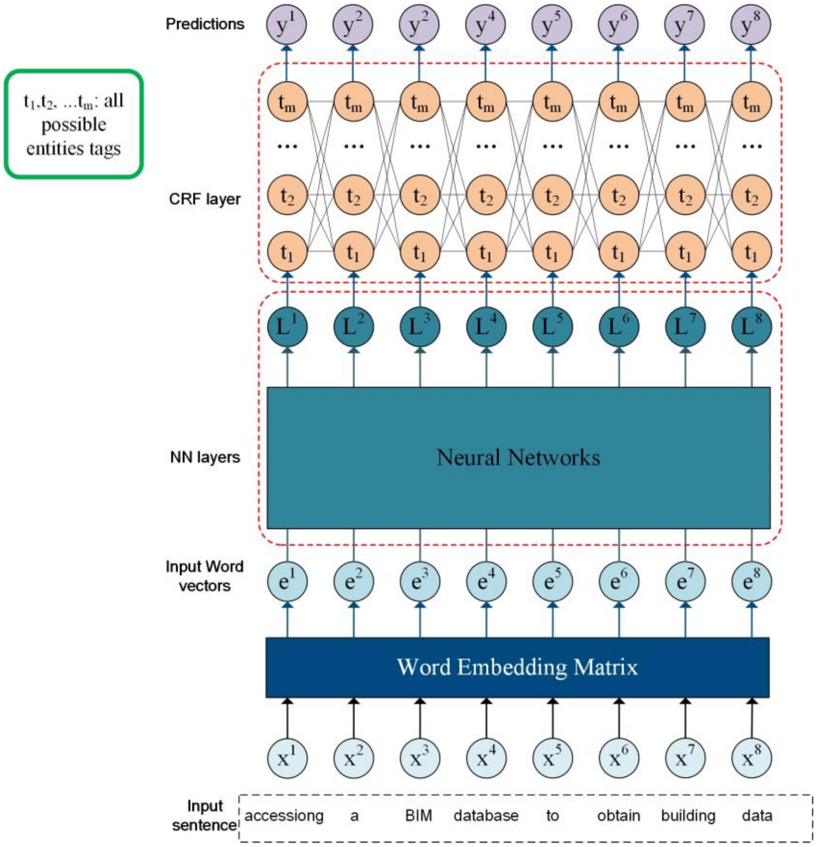


Figure 3-1 Neural network structure of deep learning models of ER

(1) Word embedding as input features

The neural network architecture requires feature vectors to be fed into the input layer. In order to obtain such “feature vectors”, several methods have been developed to transform the raw words into feature vectors. Almost all the relevant studies applied word embedding models to transform words into feature vectors. Word embedding models (word representation, or word vectors) use neural networks to compute distributed word representations from a large corpus, which could be used as dense vectors with numeric values that could be implemented in a computer-aided program (Qiu et al., 2019). In 2003, the word embedding model was first used to represented words by using a feedforward neural network-based model (Bengio et al., 2003). In most recent studies, two types of word embedding model adopted: word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b) and GloVe (Pennington et al., 2014), developed by Google AI team and Stanford University respectively.

Word2vec model was proposed to obtain word vectors considering the surrounding and contextual information to represent words from a large number of texts (Mikolov et al., 2013b). Two main neural network structures were proposed: continues bag-of-words model and continues skip-gram model. The basic idea is that a word meaning is determined by the surrounding context.

GloVe is another word embedding model and is increasingly applied in many studies. The basic structure is similar to word2vec, but GloVe considers the global information in a corpus and it follows the unsupervised learning manner (Pennington et al., 2014).

(2) CRF as the output layer

Conditional random fields (CRF) was first proposed as an improvement of Markov models in 2001 (Lafferty et al., 2001). The application of Markov models lies in the basic assumption that the words in the input sentence are independent. However, across dependencies always exist among the output of the sequence labeling (ER is a specific task of sequence labeling) (Lample et al., 2016). English language obeys the syntactic and grammatical rules that make the independencies among the words in a sentence impossible.

This study set a matrix H ∈ $\mathbb{R}^{n\times m}$ as the output of the hidden layers of the whole NN, where m is the number of distinct entities tags. The input sentence is set as X = $(x_1, x_2, \dots, x_n)$, where x_j denotes a single word at the j position in the sequence. The output is y = $(y_1, y_2, \dots, y_n)$, where y_i denotes an entity tag. Therefore, an input sentence has m^n possible Y, and the score of Y based on the input X is defined as:

$$s(X,Y) = \sum_{i=0}^{n} A_{y_i,y_{i+1}} + \sum_{i=1}^{n} P_{i,y_i} \quad (3.1)$$

where $A_{y_i,y_{i+1}}$ is the transition score from entity y_i to y_{i+1} and the whole matrix A is the parameter to be trained. According to the score for each possible y, the probability for the entity tags Y based on the input sentence X is as follows:

$$P(y|X) = \frac{\exp(s(X,y))}{\sum_{\tilde{y}\in Y} \exp(s(X,\tilde{y}))} \quad (3.2)$$

Given the input sentence and all possible entity tags, the value of $\sum_{\tilde{y}\in Y} \exp(s(X,\tilde{y}))$ is fixed. Therefore, to achieve the highest value of $P(y|X)$, CRF actually finds a tagging sequence of entities Y that make s(X,y) highest.

3.3.4.2 RNN-based models

Recurrent neural networks have been widely and frequently used in NLP tasks because its architecture concerns the sequence information which meets the basic characteristic of language writing: the words have to appear in order from start to end and such a sequence cannot convey correct meanings when the order is wrong.

Before RNN was developed, traditional neural networks always treat the inputs and outputs independently, regardless of the interrelations and sequence that always exist in real-world data. The basic architecture of RNN was first proposed by Rumelhart et al. (1986), and its main motivation is to remedy the problems of traditional neural network models. Because RNN follows a sequence-to-sequence (Seq2Seq) learning fashion, a sequence is fed into the input layer and all the data in the sequence would be computed simultaneously. Since RNN concern the order information in a sequence, it is an ideal algorithm to model time-series and language data (Panakkat and Adeli, 2009).

Recurrent neural networks are particularly suited to model time-series data such as temporal magnitude recordings because they incorporate a time delay in their operations through a feedback connection between the output layer and the hidden layer(s). Figure 3-2 illustrates the basic structure of RNNs (kindly note that the input and out layers are the same in Figure 3-1). An RNN (R^t) receives signals both from the input word vector (e^t) and the past input information in the hidden state (h^t), and the RNNs are listed as the same order as the input word vectors. Each of the RNN shares the same parameters which could be updated by the gradients in the back-propagation process.

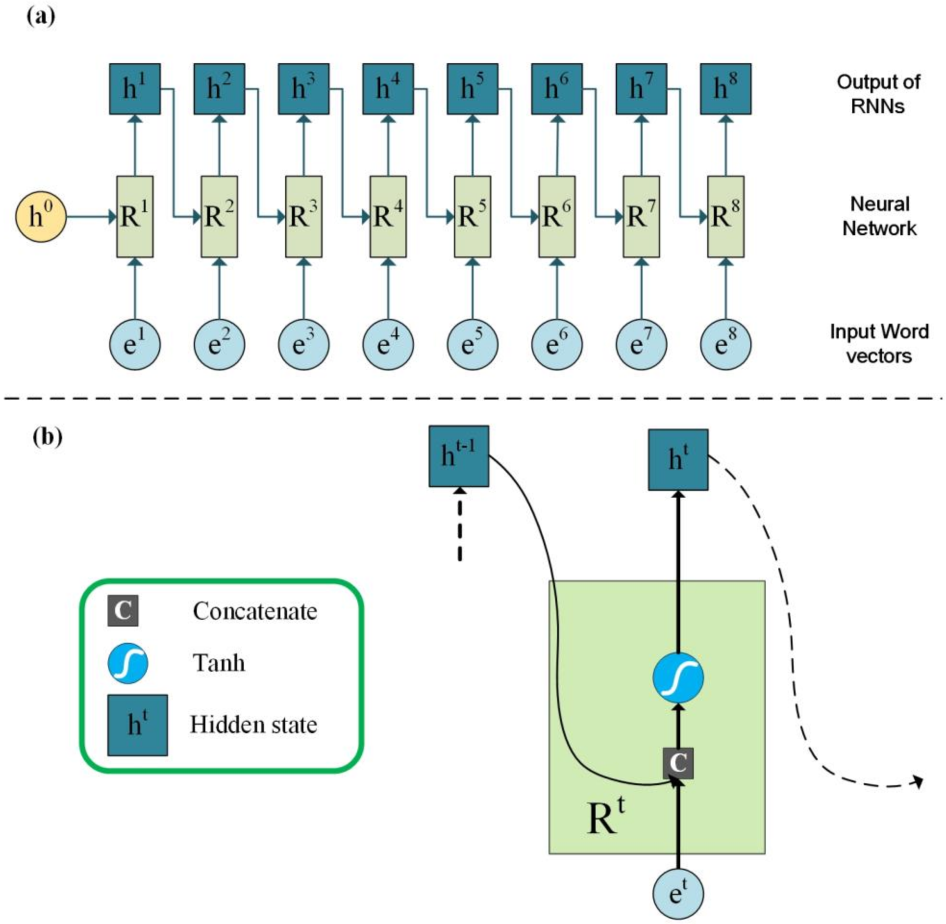


Figure 3-2 Basic structure of RNN: (a) Neural networks of RNN; (b) Plot of a recurrence

However, due to the shared parameters for each RNN, the gradients, during the back-propagation process more often vanish at the beginning and explode at the end of the sequence. In the field of ER, an advanced version of RNN, long-short transform memory (LSTM) has been widely applied (Bekoulis et al., 2018; Huang et al., 2015; Lample et al., 2016; Luo et al., 2018; Ma, 2016). LSTM was proposed to solve the gradient vanish problems in RNNs, which considers the long-term dependencies of RNNs (Hochreiter and Schmidhuber, 1997). Figure 3-3 plots the special structure of LSTM. The major difference between LSTM and general RNN is the so-called *cell state*, the horizontal line on the top of Figure 3-3. The cell state only has two linear transformations, which is easy to be conveyed from past to next. The function of the cell state is to decide how much information should be discarded or remained. This function could not be implemented without the four gates in the figure. Gates decide whether the input information should be passed through. The sigmoid activation functions compress the values into the range from 0 to 1, determining how much information in each lane should be passed through by a point multiplication operation. The first gate is "forget gate layer", determining what information should be kept from the last cell state.

$$f^t = sigmoid\ (W_f[h^{t-1},\ x^t] + b_f) \quad (3.4)$$

The second gate is "input gate layer", which decides what new information should be kept in the cell state. A temp cell state $\tilde{C}^t$ (the same as hidden state h^t in RNNs) is first calculated, and what information of the temp cell state would be conveyed into the new cell state is determined by the "input gate layer" i^t, a sigmoid function of the concatenate of h^{t-1} and e^t.

$$\tilde{C}^t = \tanh\ (W_c\ [h^{t-1}, x^t] + b_c) \qquad (3.5)$$

$$i^t = \mathrm{sigmoid}\ (W_i\ [h^{t-1}, x^t] + b_i) \qquad (3.6)$$

Then, $\tilde{C}^t$ would be combined with the kept last cell state by a pointwise addition operation, generating the new cell state value C^t ($C^t = f^t * C^{t-1} + i^t * \tilde{C}^t$). The last gate is "output gate layer", determining what should output. A tanh function compresses C^t between 0 and 1, and a sigmoid function o^t would be used to decide what information in the C^t should be kept.

$$o^t = \text{sigmoid}\,(W_o\,[h^{t-1}, x^t] + b_o) \quad (3.7)$$

$$h^t = o^t * \tanh\,(C^t) \quad (3.8)$$

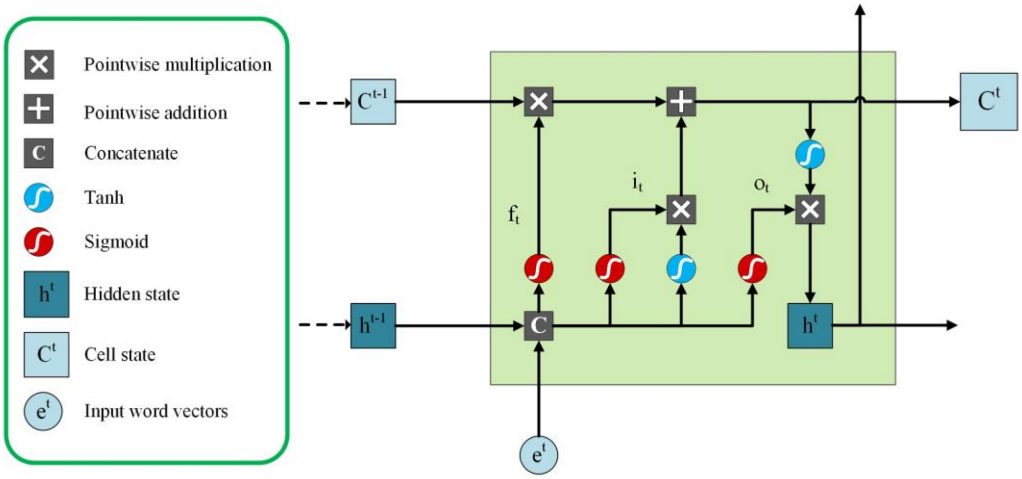


Figure 3-3 Neural networks of LSTM

LSTM, although aims to capture the long-term dependencies, it's basic recurrence architecture limits the efficiency. In the context of ER, a bidirectional structure of LSTM was frequently used to capture both of the left-to-right and right-to-left encoding information (Bekoulis et al., 2018; Huang et al., 2015; Lample et al., 2016; Luo et al., 2018; Ma, 2016). As Figure 3-4 shown, the output h^t is the concatenate of $\overrightarrow{h}^t$ and $\overleftarrow{h}^t$, capturing the both direction information in the context(Dyer et al., 2015).

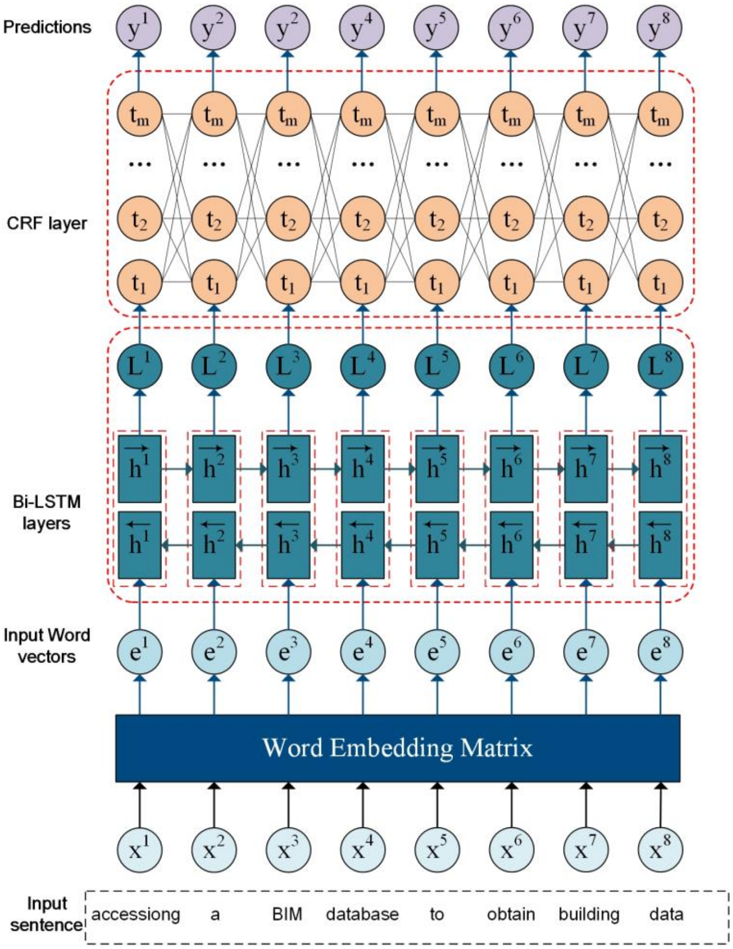

Predictions
y^1 y^2 y^2 y^4 y^5 y^6 y^7 y^8
CRF layer
t_m
...
t_2
t_1
L^1 L^2 L^3 L^4 L^5 L^6 L^7 L^8
Bi-LSTM layers
$\overrightarrow{h}^1$ $\overrightarrow{h}^2$ $\overrightarrow{h}^3$ $\overrightarrow{h}^4$ $\overrightarrow{h}^5$ $\overrightarrow{h}^6$ $\overrightarrow{h}^7$ $\overrightarrow{h}^8$
$\overleftarrow{h}^1$ $\overleftarrow{h}^2$ $\overleftarrow{h}^3$ $\overleftarrow{h}^4$ $\overleftarrow{h}^5$ $\overleftarrow{h}^6$ $\overleftarrow{h}^7$ $\overleftarrow{h}^8$
Input Word vectors
e^1 e^2 e^3 e^4 e^5 e^6 e^7 e^8
Word Embedding Matrix
x^1 x^2 x^3 x^4 x^5 x^6 x^7 x^8
Input sentence
accessiong a BIM database to obtain building data

Figure 3-4 Neural network architecture of Bi-LSTM

3.3.4.3 CNN-based models

RNN-based models improve the performance of entity recognition and have been applied in a wide range of studies. However, because of the architecture, RNN is not without limitations. For example, RNN treats each of the words as an independent input, neglecting the phrases that are combinations of a sequence of words (i.e., “build information modelling”). This leads to additional prefixes or punctuations to “connect” these singles words (i.e., “build-information-modelling”). Such additional information could only be implemented manually in RNN-based models, which consumes a lot of time. In this case, there is a variety of communication entities that are constituted by more than one word, such as *RFID tag*, *CAD drawings*, *underground facilities*, etc. In addition, the last words always have more weights because of the sequence and the direction.

Against the limitations mentioned above, CNN-based models were proposed in ER tasks. CNN-based models consider every possible sub-sequence of a sentence and compute a representation for it. CNN is a typical neural network to capture vision features, extracting features in different positions of the image regardless of where it appears. The basic idea of CNN is illustrated in Figure 3-5, a filter is used to compute a dot product with a window that has the same size as the patch. All the possible windows in the target are extracted to make dot products, generating a convoluted matrix. The filter is set according to the patterns that are expected to identify in the images, such as the eye, nose, mouth, etc.

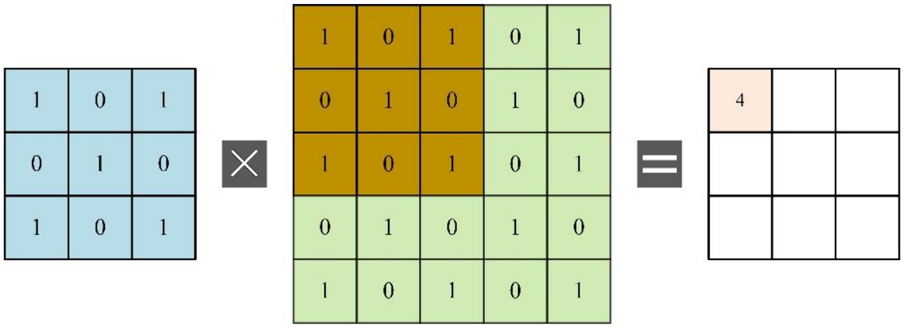


Figure 3-5 General computation process of CNN

With respect to NLP, 1D convolution is frequently used, and the sentence is usually with padding. Set a sentence "accessing a BIM database to obtain building data" as an example, shown in Figure 3-6, a pointwise multiplication operation is taken to combine three words in a sequence, and the outputs are the three-gram features which would be fed into the next layer.

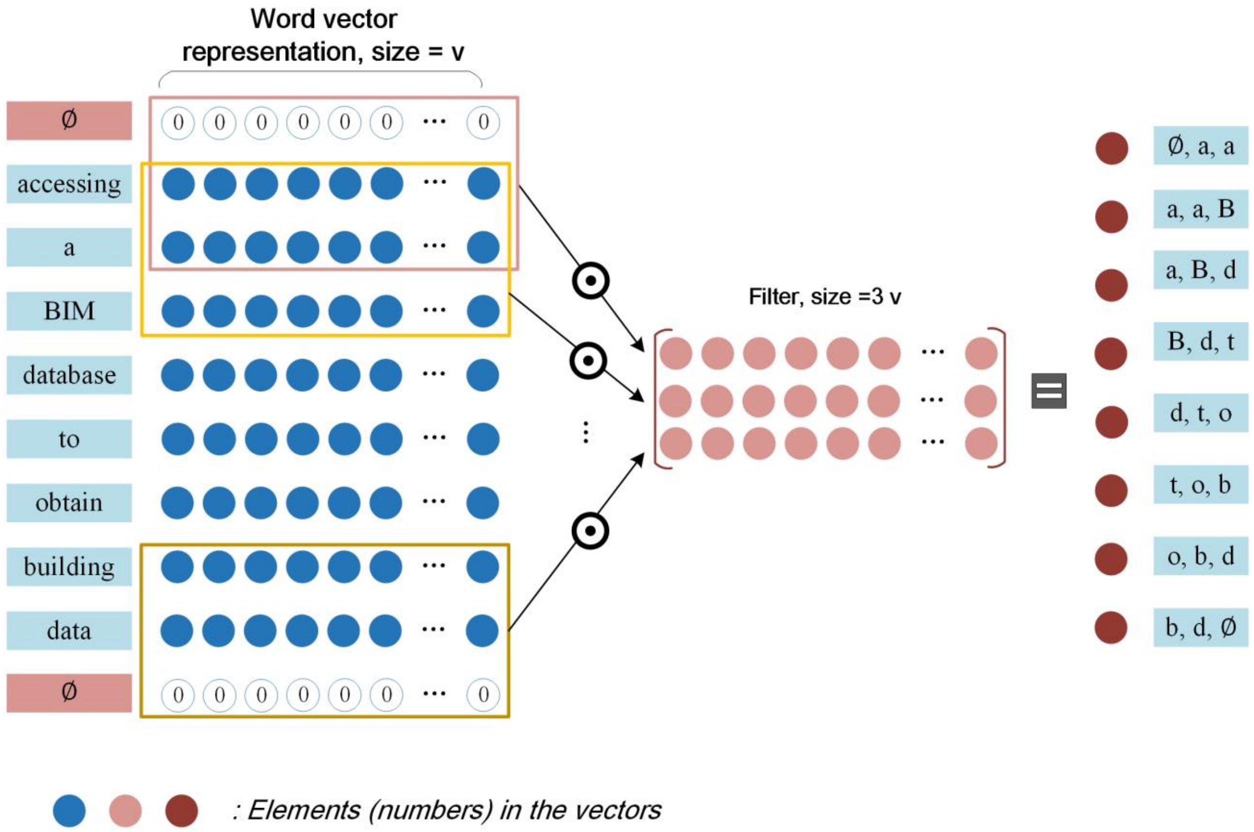


Figure 3-6 Operation of CNN for ER tasks

In the field of entity recognition, CNN does not attract the same attention as RNN-based models. The main reason may be that CNN does not concern the global information of a sentence that the RNN-based models capture. The performance partly depended on the applied models, is highly affected by the domain. In the domain of biological entities recognition, CNN has been proved a better model than RNN (Bekoulis et al., 2018), imposing the local context features into the CRF layer.

3.3.4.4 Bi-LSTM with CNN

CNN and RNN varied from each other according to the neural network structures, and several studies used both of the deep learning models for ER. For example, Korvigo et al.

(2018) developed an RNN-CNN model chemical entities recognition, which basic architecture is shown in Figure 3-7. In the integrated model, a 1D CNN layer was used to extract the features, then a bi-LSTM received the outputs of CNN and operated them as usual. Another study integrated RNN and CNN structures by using a convolutional neural network to embed the character-level features before the word embedding representations being fed into the neural network. Its main body employed a bi-directional LSTM architecture.

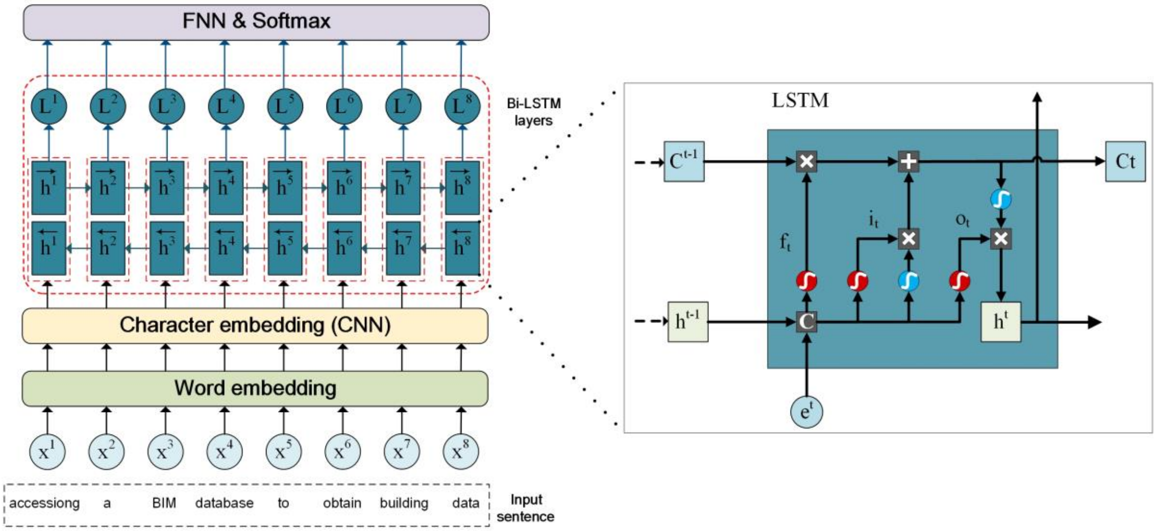


Figure 3-7 The neural network architecture of Bi-directional LSTM with CNN for ER tasks

3.4 Summary of the Chapter

Entity recognition was viewed differently in CEM and NLP studies. On one hand, CEM studies regarded entity recognition as automatic approaches to identify related entities and classified them into existing entity classes in digital dictionaries. On the other hand, NLP studies regarded entity recognition as a normal task, aiming to develop techniques to improve performance.

The existing entity recognition approaches in CEM studies have two major flaws. First, these approaches are extremely time-consuming and labor-intensive, because hand-code rules and vocabularies require professional experience for word selection and rule establishment. Moreover, directly matching entities with pre-defined vocabularies cannot recognize the unknown entities. Second, these methods always lead to lower accuracy in discerning the ambiguous entities (the spellings can appear as an entity at one position and common noun at another position, or appear as different entity types). Those methods typically utilized external lexical databases such as WordNet to discern ambiguous entities, but the performance is still unsatisfactory due to the limited coverage of the lexical databases.

Some NLP studies utilized deep learning techniques to discern ambiguous entities, because deep learning could learn non-linearity by the complex computational functions in the deep layers of neurons. Until 2018, RNN-based deep learning models have been widely recognized as the state of the art model in NLP tasks, including ER, text classification, sentiment analysis, machine translation, etc (Bahdanau et al., 2014; Cho et al., 2014). The basic nature of RNN-based models is the sequential computation, generating a sequence of hidden state values h^t according to the value of previous hidden value h^{t-1} and the input value at position t (e^t). This sequential computation prevents parallelization in the training process, and thus prevents further utilization of advanced hardware such as TPU and GPU. Without parallelization, the computing for long sequence takes a considerable amount of time due to the limited use of batching across examples (Vaswani et al., 2017). Although some improvements in RNN-based models have been achieved through conditional computation and factorization tricks, the aforementioned limitation remains due to the nature of sequential computation in RNN.

Chapter 4 Research Methodology

4.1 Introduction

This chapter describes the research framework of this study. The methodologies are then presented corresponding to the objectives, followed by detailed discussions of the employed techniques to each of the methodologies.

4.2 Research Framework

As shown in Table 4.1, the detailed research methodologies and techniques were described corresponding to the objectives. This study mainly employs deep learning techniques, NLP techniques, and computer-aided programs to achieve these research objectives.

(1) To develop a deep learning model for entity recognition which can automatically recognize CEs from raw text through utilizing the contextual information.

(2) To compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER).

(3) To train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings.

Table 4.1 Adopted techniques for research objectives

Research objectives	Research methodologies	Techniques
(1) To develop a deep learning model for entity recognition which can automatically recognize CEs from raw text through utilizing the contextual information	1. Document analysis 2. NLP techniques 3. Deep learning 4. Computer-aided programming	1. Literature review 2. Transformer 3. Wordpiece embedding 4. IOB tagging system 5. TensorFlow
(2) To compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER).	1. NLP techniques for text pre-processing 2. Data annotation 3. Multi-Layer Perceptron 4. Computer-aided programming	1. Tokenization 2. Part of speech 3. lemmatization and stop-words removal 4. Text crawling from web
(3) To train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings	1. Validation 2. Bi-directional LSTM+CNN 3. Comparative analysis 4. Empirical analysis 5. Feature selection	1. K-fold cross-validation

Figure 4-1 illustrates the logic flows between the methods and research objectives. Literature review and document analysis were conducted to highlight the technical problems in recognizing communication-oriented entities. According to the problems, this study selected NLP techniques and neural networks to structure the deep learning model. Transformer was used as basic neural networks to enable the so-called “self-attention” mechanism. This lets the whole model for understanding contextual information underlying the raw text. In addition, the wordpiece tokenization is used to represent infinite words by a limited scope of wordpieces, enabling the model to recognize unknown entities. These techniques, in combination with normal neural networks such as feed-forward networks, shape the architecture of the deep learning model. Then, to achieve objective 2, this study applied NLP techniques and MLP model to build up a classifier to compile a database of patents of ICT in construction. The training and testing instances were obtained by annotation of the screened patents. Based on the structured deep learning model and training and testing instances, this study applied pre-training and fine-tuning techniques to train the model. K-fold cross-validation was used to validate the model and compare the validation with the traditional RNN-based model.

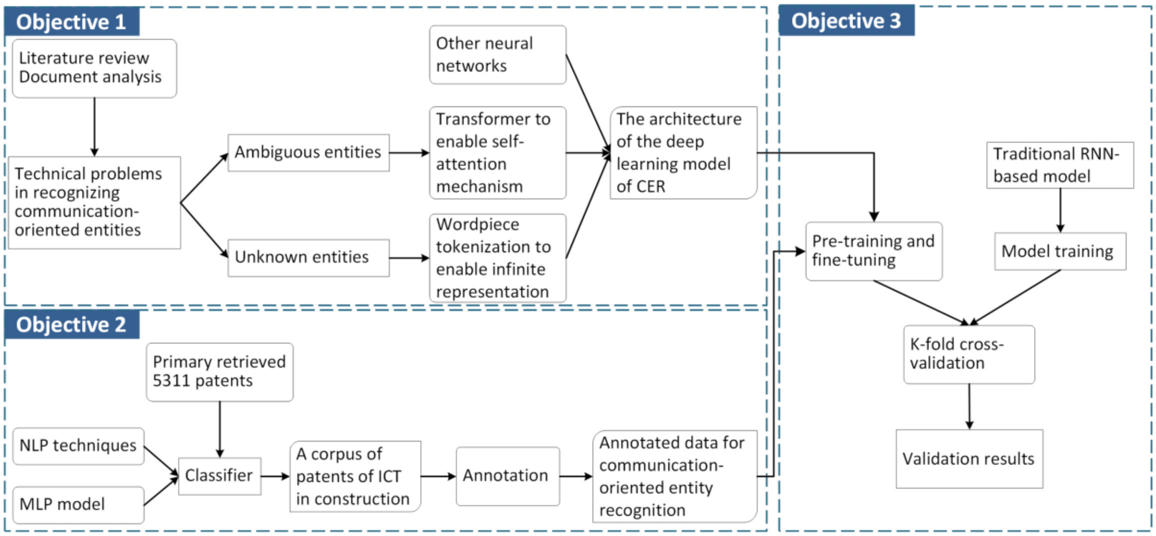

Objective 1
Literature review
Document analysis
Technical problems in recognizing communication-oriented entities
Ambiguous entities
Unknown entities
Other neural networks
Transformer to enable self-attention mechanism
Wordpiece tokenization to enable infinite representation
The architecture of the deep learning model of CER
Objective 2
Primary retrieved 5311 patents
NLP techniques
MLP model
Classifier
A corpus of patents of ICT in construction
Annotation
Annotated data for communication-oriented entity recognition
Objective 3
Traditional RNN-based model
Pre-training and fine-tuning
Model training
K-fold cross-validation
Validation results

Figure 4-1 The flow chart of the methods used for objectives

4.3 NLP Techniques for Text Pre-processing

This study utilizes text pre-processing as a necessary process to compile a patent database of ICT in construction (chapter 6). NLP, a sub-field of computer science, is a branch of computational techniques such as linguistics and machine learning methods for the automatic processing and representation of human language (Cambria and White, 2014). Human language is extremely complex, integrating a catalog of words, called a lexicon, and a grammar system with a set of structural rules (Manning, 1999). The main concern of

NLP is to make a bridge between human and computer, in which they can interact efficiently. Recently, scholars prefer using machine learning algorithms and statistical methods when developing NLP tools. The processing tools with machine learning enable a machine to learn the semantic meaning from human language input and give feedback based on pre-defined rules (Jain et al., 2018). A large and growing body of research has investigated certain machine learning techniques with NLP, such as clustering algorithms (k-means, naive Bayes, and support vector machines) (Hindle, 1989; Karatzoglou and Feinerer, 2010; Li and Wu, 2010).

Processing the unstructured data is one of the most time-consuming phases in the NLP tasks (Munková et al., 2013), which aims to clean and format the raw data, eliminating noisy features as many as possible (Haddi et al., 2013). Many techniques have been introduced to process unstructured textual data, such as stop and common words removal, tokenization, lemmatization and stemming (Aggarwal and Reddy, 2013). Recently, NLP techniques have received intense attention in processing textual data. NLP techniques could analyze the natural language and process them by a computer. This study employs four NLP techniques to for pre-processing, including (1) *tokenization*, which can split a sentence into tokens; (2) *POS tagging* that recognizes and processes syntax information (grammatical meanings) (Collobert et al., 2011; Gimpel et al., 2011); (3) *lemmatization and stop-words removal* that match the words with different forms and remove stop-words.

4.4 Deep Learning Models

Deep learning techniques have been widely used in NLP tasks, fast becoming necessary tools to explore and utilize the unstructured text data in a robust way, in which the information in the unstructured data could be extracted and represented in a readable way. Prior studies that processed the textual data in the construction industry have merely employed techniques from traditional machine learning or text mining, while the increasingly advanced techniques from deep learning have been widely overlooked. In the CEM domain, deep learning models have been majorly developed to process images and video data. However, with respect to the textual data, deep learning models have rarely been developed regardless of the outstanding performance that has been achieved in NLP tasks. This study resorts to the realm of deep learning for entity recognition.

4.4.1 Transformer

In the recent five years, state-of-art deep learning models for ER are RNN or CNN based models with word2vec or Glove embeddings as inputs. At the end of 2018, a novel deep learning model - transformer was proposed by Google AI team (Vaswani et al., 2017). The transformer is fast applied in developing deep learning models for NLP tasks, many of which have gained better performance (Devlin et al., 2018).

Transformer's structure enables pre-training deep bidirectional representations by only using plain text corpus. The pre-trained transformers could perform several NLP tasks, even though the training instances are limited. The RNN and CNN based models that were applied to specific NLP tasks require a large amount of annotated data to keep a better performance. This limitation could be largely offset by the pre-training phase, in which specific NLP tasks could be fine-tuned based on a small collection of data.

4.4.2 Multi-Layer Perceptron

To effectively screen the patents of ICT in construction, this study proposes a deep learning model based on the Multi-Layer Perceptron (MLP). The structure of MLP is typically fully connected, which is an ideal model to learn and train the complex relations between inputs and outputs. MLP is always used with a feed-forward-based architecture and back-propagation learning process (Goodfellow et al., 2016). There is a number of neurons in the MLP, and each of them receives signals from the former layer and passes a transformed single by an activation function to the subsequent layer (Riedmiller, 1994). Although it is a general wisdom that deep NN model is better than machine learning models, NN design and hyperparameters choice are more important than the deep NN model itself (Levy et al., 2015).

4.5 Computer-aided Tools for Programming

This study uses self-developed python programs to perform most of the techniques. In specific, this study employs Beautiful soup and Regular Expression to crawl the patent texts of ICT in construction from the website of the United States Patent and Trademark Office (USPTO); employs NLTK to perform NLP techniques to process the text data; uses Keras and Tensorflow to build up deep learning model and perform training process. In addition, this study implements the validation based on self-developed programs and draws the figures relevant to results by using Seaborn and Matplotlib. The most important computer-aided tools applied in this study are NLTK, Keras and Tensorflow.

4.5.1 NLTK

NLTK is a suite of toolkits for statistical natural language processing (NLP). It was developed by the University of Pennsylvania mainly English language. NLTK incorporates various sub-programs for NLP techniques, such as word tokenization, sentence tokenization, part-of-speech tagging, and lemmatization. This study adopts NLTK to pre-process the achieve raw text data.

4.5.2 Computer-aided Programming for Deep Learning

4.5.2.1 Keras

Keras is a Python package for neural networks design and tune. Keras could be performed on top of TensorFlow. Keras is developed to provide easy used and modular neural network API to fast implement deep learning models. This study uses Keras to build the MLP model (chapter 5) to train a binary classifier, which could screen patents of ICT in construction.

4.5.2.2 Tensorflow

Tensorflow is the most frequently used Python package for deep learning models. Therefore, Tensorflow has the largest community for sharing knowledge of deep learning. Tensorflow, compared with Keras, was harder for use. This study uses Tensorflow to train the developed deep learning model for CER based on the transformer.

4.6 Validation Methods

Traditionally, the performance has been measured for machine learning methods through various statistical indicators, such as precision (P), recall (R), F-score (F1), accurate, error rate, etc (Sokolova and Lapalme, 2009). This study, similar to most of the deep learning studies, utilizes precision, recall, F-score to validate the deep learning model based on the number of true positives (TP), false positives (FP) and false negatives (FN). Generally, TP is the number of instances the model correctly predicted. FP denotes the number of instances the model incorrectly predicted. FN reflects the number of instances the model failed to predict. Based on the count of TP, FP and FN, the precision, recall, and F-score can be computed by

$$\mathrm{P} = \frac{TP}{TP+FP}\ ,\mathrm{R} = \frac{TP}{TP+FN}\ ,\mathrm{F1} = \frac{2\times P\times R}{P+R} \tag{4.1}$$

The counting methods of TP, TF, and FP are varied in different tasks. This study proposed two deep learning models and their performances were evaluated based on P, R, and F1.

4.7 Summary of the Chapter

This chapter first illustrated the overall research framework. After that, the employed techniques were listed and described. NLP techniques are mainly used to pre-process the textual data; deep learning methods and algorithms are used to establish a learning model for CER of ICT in construction; computer-aided programs are used to perform the NLP and deep learning techniques.

Chapter 5 The Architecture of the Deep Learning Model for CER of ICT in Construction

5.1 Introduction

This chapter describes the architecture of the developed deep learning model for CER of ICT in construction, illustrating how the neural networks are structured according to CER, as well as the basic components in each layer of the neural networks. This chapter firstly clarifies the basic concepts of CER. Secondly, this chapter describes the motivations for developing the deep learning model based on the transformer for CER, as well as the overview of the architecture neural network of the developed deep learning model. After that, the basic components and in each layer of a single transformer are described. The training techniques are illustrated at the end of this chapter.

5.2 Basic Concepts of CER and the Embodied Technical Problems

5.2.1 Basic Concepts of CER

From a general perspective, entity refers to concepts, textual elements, textual items, or terminologies that represent information elements in raw text. An entity is a category of phrases that have similar properties, which are always rigid designators or members of the semantic classes (Goyal et al., 2018). Entities could be defined by domain specifications and interests. For instances, in dietary research, entities of food and nutrient gained increasing interest (Eftimov et al., 2017); in the chemical and gene domain, gene and protein are important entities (Hemati and Mehler, 2019); in general domains, entities normally incorporate location, person, organization and time.

In the CEM domain, particular entity entities were developed according to the specific objectives for subfields. A number of studies applied existing entity types of Industry Foundation Classes (IFC). The IFC entity types are designed for representing objects of BIM-based projects (Zhang and El-Gohary, 2016). Besides the existing IFC entity types, some CEM studies defined their own entity types. For example, Liu and El-Gohary (2016) and Liu and El-Gohary (2017) recognized entities related to bridge deterioration information to effectively extract relevant information from bridge inspection reports and convert them into structured and easy-to-perceive units. They categorized the entities into several types, including bridge conditions, maintenance actions, corrosion, cracking, decay, delamination, efflorescence, scaling and spalling, scour, and so on.

With respect to the communication functionality of ICT in construction, this study, following the previous studies, define the CEs as the information units that describe the communication functionality in the technical documents of ICT in construction. The information related to communication functionality is typically expressed by the descriptions of how the construction data was transmitted through virtual or physical models and how the data was coordinated between construction sites and users or among the stakeholders (Alsafouri and Ayer, 2018).

For example, RFID was applied to facilitate timely delivery of construction materials and its communication functionality is expressed by the sentence “sensing the material information through RFID tags” (Cho et al., 2011). The communication functionality majorly involves two basic units: “material information” and “RFID tags”. The “material information” is the information to be transferred, and the “RFID tags” is the device to receive or send the information.

Based on the critical review of relevant literature (Alsafouri and Ayer, 2018) and the patents of ICT in construction, this study defines three CE types that always appear in the patents of ICT in construction for the descriptions of the communication functionality, including transferred information (TI), communication models (CM) and communication subjects (CS). (Table 5.1 illustrates the detailed definitions and examples).

TI refers to the type of information the ICT mainly conveys, such as geographic data for GIS. TI is always transferred in digital forms. CM refers to the software or equipment used to convey the transferred information. CM could be virtual or physical. For example, in the mention of a BIM invention "accessing a BIM database to obtain building data", the BIM database is a virtual CM to store, receive and send building data, while an RFID tag in the mention "sensing the material information through RFID tags" is a physical CM to store and send the install information of the building components. CS is the people or organization that utilizes ICT in the context of construction. Most of the patents of ICT in construction express the way people or organizations use ICT and how they participated in the communication functionality, such as the way they receive, generate or send information.

Table 5.1 Definitions and examples of CE classes for ICT in construction

CE classes	Description & Examples
Transferred information (TI)	Information the ICT mainly conveys, transmits, manipulates, or receive and always in a digital form

	• The apparatus for editing the 3D building data includes an input unit configured to obtain **3D scan data of a building** • The building's developer sends a **design order** for the building to a construction design office • An estimation engine processes the **aerial image** at a plurality of angles to automatically identify a plurality • A first **hierarchical data structure** generated by the mobile device is received at the first machine • The construction operation system comprises a photodetection sensor for receiving **light beams** from the rotary laser irradiating systems
Communication models (**CM**)	Software or equipment that is used to convey the transferred information. CM could be virtual or physical, which could be accessed and manipulated remotely. • The construction operation system comprises a **photodetection sensor** for receiving light beams from the rotary laser irradiating systems • A first hierarchical data structure generated by the mobile device is received at the **first machine** • Exemplary systems and methods include **marking devices** that generate, store and/or transmit electronic records of marking information • Embodiments of the present invention relate to receiving radio frequency (RF) data stored in an RF tag using an **RF reader** • A first communication channel is established between a **first machin**e and a **mobile device**
Communication subjects (**CS**)	People or organizations that participate in communication activities. CS is always the people the transferred information would be delivered to in the context of construction. • The **developer** of the building sends a design order for the building to a construction design office • The design environment supports multi-modal input, side-by-side layout of the stored documents, access permissions for **users** of the design environment • An environmental sensor assembly associated with the panel and configured to detect an environmental condition of the panel and wirelessly communicate data on the environmental condition to a **reader** • The method comprising: (a) receiving, into the computing device, an input from a **user** (either a **person** or an **automated program interface (API)**) • Plurality of project functions comprising means for collaborating on bids between **owners**, **architects**, **general contractors**, **subcontractors**, **suppliers**, **wholesalers** and **building product manufacturers**

5.2.2 Technical Problems of CER

Compared to the aforementioned other types of entities, CEs may be written in a complex genre in technical documents, because: (1) food, disease, and drug entities may have a more fixed vocabulary and less novel words than CEs that incorporate a lot of unknown phrases because the construction industry is applying techniques from many other fields. (2) The implementation conditions of ICT in construction vary with each other due to the complex and fragmented nature of the construction industry and each technology may have a specific usage and implement, leading to various expression for the communication patterns. Therefore, technical problems exist in CER of ICT in construction.

5.2.2.1 Limited size of annotated data

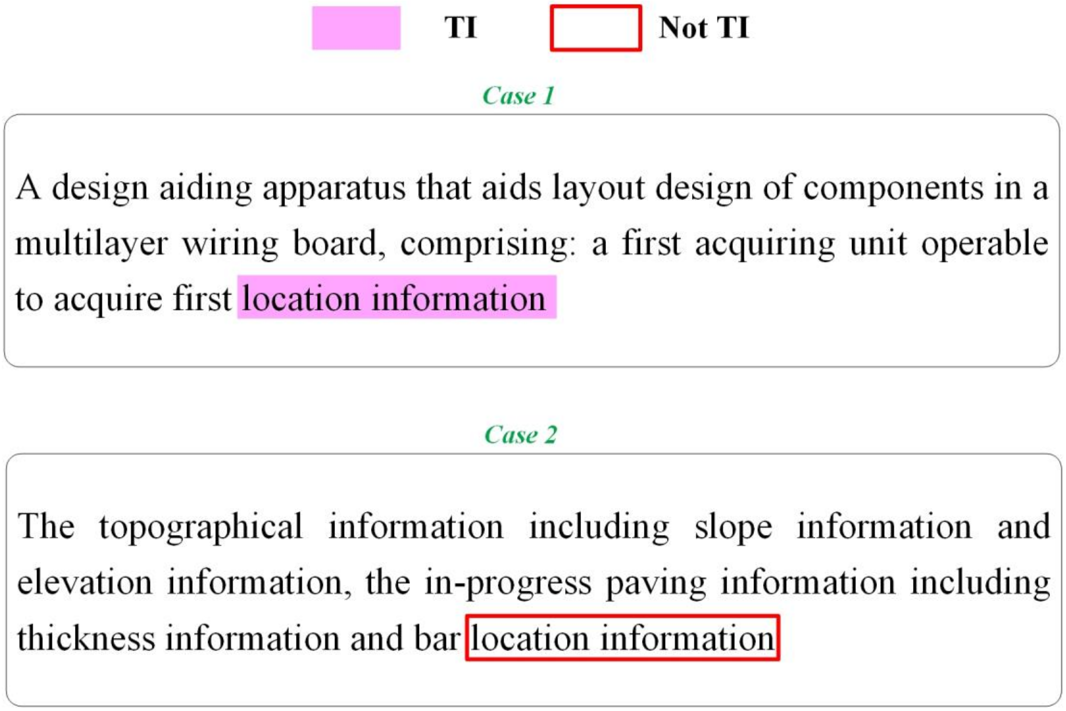
One of the technical problems with CER task is that annotated data is not available. Most of the models developed for ER tasks are trained with annotated data that is available online, in which the entity types are tagged in each of the sentences. The aim of these studies is to provide an advanced entity recognition model with better accuracy and less computation cost, but the developed models are specific for the fields in which the annotated data is available, such as dietary, biomedical and chemical fields.

This overlooks the opportunities to advance the understanding of the real world in the fields that researchers desired. To this regard, this study has to annotate data for training manually, which consumes a large amount of time and energy, and such consumption would increase when considering the diverse expression for the mentions of communication patterns in the ICTCs. Therefore, the annotated data for training is limited, which poses a challenge for the developed deep learning model.

5.2.2.2 Plenty of ambiguous entities

Since ICT incorporates a variety of information techniques, some technical mentions may or may not appear as CEs. The major difference between CEs and other types of entities is that CEs incorporate a number of ambiguous entities.

In the NLP domain, the entities are typically the mentions of persons and organizations, which could be discerned by word-level features. For example, mentions of people and organizations are always appeared with capitalized of the first character, such as the "Henry Ford" in the sentence "automotive company created by Henry Ford". However, in the context of CE, there are many entities could be referred to as CEs or non-CEs, because the most of mentions of CE are common technical concepts of information technology, such as "sensor", "database", "display device", "map information", "location information" etc. Figure 5-1 plots an example for ambiguous CE of "location information". In the first sentence, "location information" refers to TI, because it is articulated as digital data to be transferred. However, "location information" in the second sentence is not a TI because it is not used for communication according to the surrounding context. Therefore, the developed deep learning model needs to consider the contextual information to determine the CEs.



🟪 TI   □ Not TI

Case 1

A design aiding apparatus that aids layout design of components in a multilayer wiring board, comprising: a first acquiring unit operable to acquire first location information

Case 2

The topographical information including slope information and elevation information, the in-progress paving information including thickness information and bar location information

Figure 5-1 An example of ambiguous entities

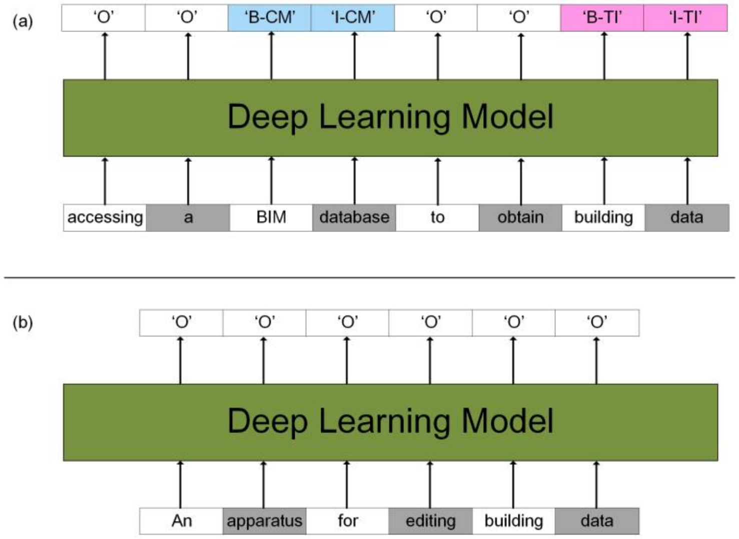
5.2.2.3 Unknown CEs

Another problem is the unknown entities that have not be seen in the training dataset but may be seen in the testing dataset and future applications. The construction industry holds the potential for harnessing novel ICTs (Hosseini et al., 2013) and thus new patents of ICT in construction may arise with a lot of unknown CEs that are expected to be recognized by the proposed model.

5.3 Motivation and the Overview of the Architecture of the Deep Learning Model

5.3.1 Motivation of Structuring the Deep Learning Model

Against the aforementioned problems of ambiguous and unknown entities, the developed deep learning model is expected to understand the contextual information of the whole sentence for discerning whether a noun word or phrase is a CE. Figure 5-2 plots the inputs and outputs of the desired learning-based model for CE recognition, which aims at predicting the labels of CEs (especially the ambiguous CEs) with accuracy as high as possible. Figure 5-2 (a) and (b) illustrate the phrase “building data” in two different sentences that would be fed into the model. In Figure 5-2 (a), according to the contextual information of the entire sentence, the phrase “building data” is essentially a type of information that can be transferred remotely and thus should be labeled as “TI”, whereas the contextual information in the input sequence in Figure 5-2 (b) indicates that “building data” is not used for communication and thus should be labeled as “O”. This study chooses deep learning rather than machine learning models for CE recognition due to the two main factors that can impede the performance: (1) deep learning can draw representations from unstructured text data based on the architecture of neural networks without any need for enhanced pre-engineered features that are needed for machine learning models (Kalchbrenner et al., 2014; Mahmoudi et al., 2018); and (2) it can address highly non-linear associations between the representations and the outputs through the neurons and activation functions in each layer of neural networks (Wang et al., 2016), whereas the machine learning algorithms typically need human intervention to help the system to learn the statistical associations between inputs and outputs.



Notes: Label 'O' denotes non-CEs; For CEs, the labels 'TI', 'CM' and 'CS' denote the three CE classes of transferred information, communication models, and communication subjects respectively. In the front of CE classes labels, 'I' denotes the inside of a CE, and B denotes a start of a start of a CE. For example, a word is labeled as: 'B-TI' if it is the first word of entity of communication information; 'I-TI' if it is the inside but not first word of entity of communication information.

Figure 5-2 The desired inputs and outputs of a deep learning model for CER

5.3.2 Overview of the Architecture of the Deep Learning Model

This study draws upon the transformer as the basic neural network rather than RNN since it was reported as the most effective neural network structure for addressing contextual information (Chen et al., 2019; Devlin et al., 2018; Vaswani et al., 2017). The transformer was proposed in 2017 by Google AI team (Vaswani et al., 2017), and has gained increasing attention from NLP researchers. A number of NLP tasks have been reported with improved performance by employing the transformer. Figure 5-3 plots the overall structure of the deep learning model. Unlike the RNN-based models which were typically developed with "end-to-end" structures, the deep learning model follows a parallel structure, comprising various sublayers of neural networks. The first step is to split the raw text into tokens using wordpiece tokenization, which would be fed into the sublayers of neural networks. The first sublayer is token embedding, converting the tokens into token vectors by look-up operation. The subsequent sublayer is position embedding, encoding the position information of the corresponding tokens in the input sequence. The output of the position encoding sublayer would be fed into the stacked transformers, utilizing the multi-head self-attention and feed-forward to extract features representing the contextual information. The outputs of the transformers would be connected with a linear and a softmax (the softmax would be computed over the possible CE labels) neural network to generate the possibilities for each of the labels. In the training process, the cross-entropy is used as the loss-function to compute the gross gradients for the back-propagation process.

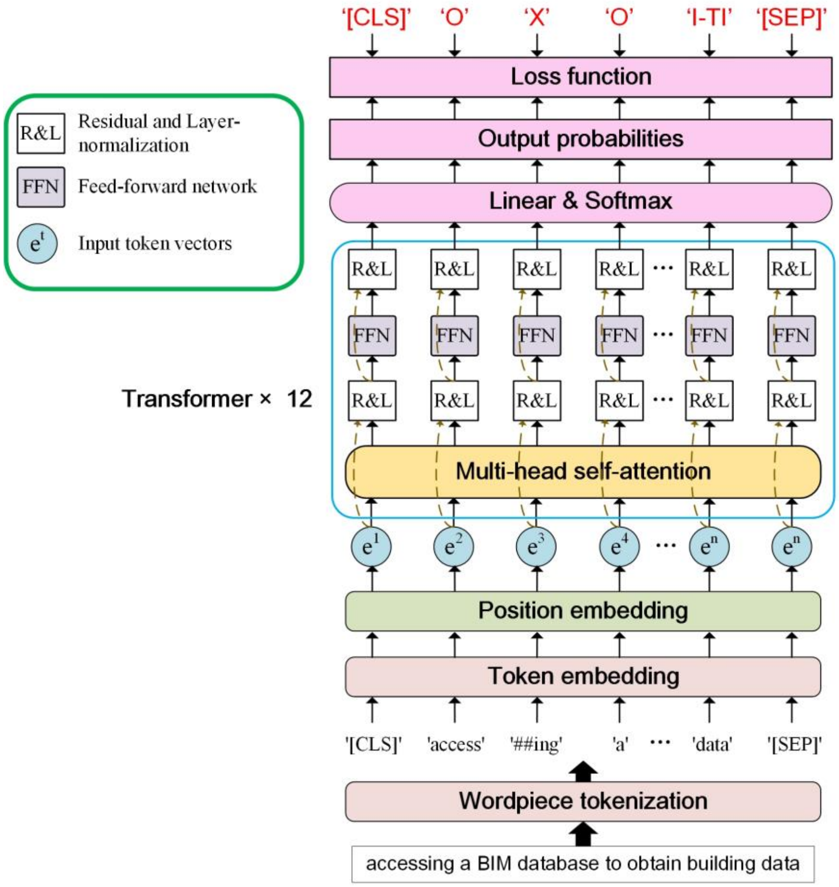


Figure 5-3 The overall neural network structure of the deep learning model

This study applies the so-called "transformer" - a deep neural network structure that is developed by Vaswani et al. (2017). The transformers in the pre-training and fine-tuning share the same architecture, incorporating two sub-layers: multi-head self-attention and point-wise feed-forward network (Figure 5-4). The key component in the deep learning model is the self-attention sublayer, which addresses the dependencies between the input works. Multi-head self-attention is a linear combination of several self-attention sub-layers. Each of the sublayers is connected with a residual connection and layer-normalization. The deep learning model is mainly structured based on self-attention, as well as some other sub-layers and neural network operations.

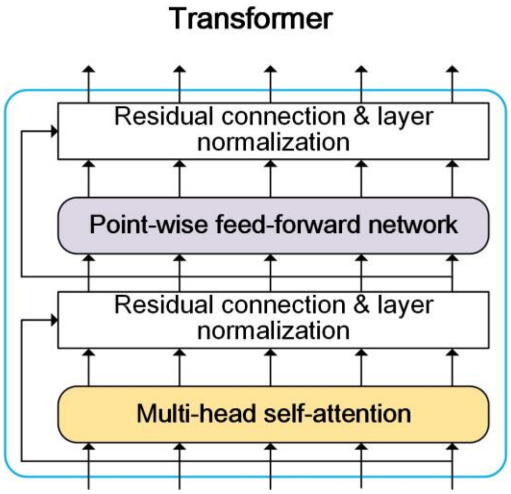

Transformer
Residual connection & layer normalization
Point-wise feed-forward network
Residual connection & layer normalization
Multi-head self-attention

Figure 5-4 Neural networks in a transformer

5.4 Sub-layers of the Deep Learning Model

5.4.1 Tokenization and Embedding

Word embeddings (i.e., word2vec and GloVe) are dense vectors that represent words in lower-dimensional space. These word vectors are pre-trained in an unsupervised fashion over a sheer volume of texts, making significant improvements for certain NLP tasks (Araque et al., 2017; Camacho-Collados et al., 2016; Fu et al., 2017; Ren et al., 2016). However, word2vec and GloVe failed to discern the unknown words. The word vectors are obtained by word embedding models based on a large volume of texts, and there is still a number of unknown words that always appear in the testing data of ER tasks. When there is a word that does not exist in the vocabulary of pre-trained word vectors, a possible way is to get the word vector of that word in the training text and use it (Dhingra et al., 2017). Nevertheless, two main problems still exist in word vectors obtained by word2vec and GloVe models: (1) The pre-trained word vectors always convey the contextual meaning of a word based on the pre-trained context, but not the context in which the word appears; (2) The pre-trained word vectors only have one representation for a word, which may have different perspectives including contextual meanings, syntactic behavior and word meaning. For example, the word building could have different contextual meaning in the context of “building a model” and “a building model”, where the former means to build up a model, and the latter means an architecture model.

In this study, the token representations that to be fed into the stacked transformers are combinations of byte-pair and position encoding, which concern the information of the tokens’ identities and positions in the input sentence. The overall process is shown in Figure 5-5. The input sequence would be firstly tokenized by the wordpiece. After that, each of the tokens would be embedded by byte-pair and position encoding, independently creating two sequences of vectors that have the same dimension. The combinations of the token and position embedding vectors are fed into the stacked transformers. The detailed algorithms for the token and position embedding are described in the following sections.

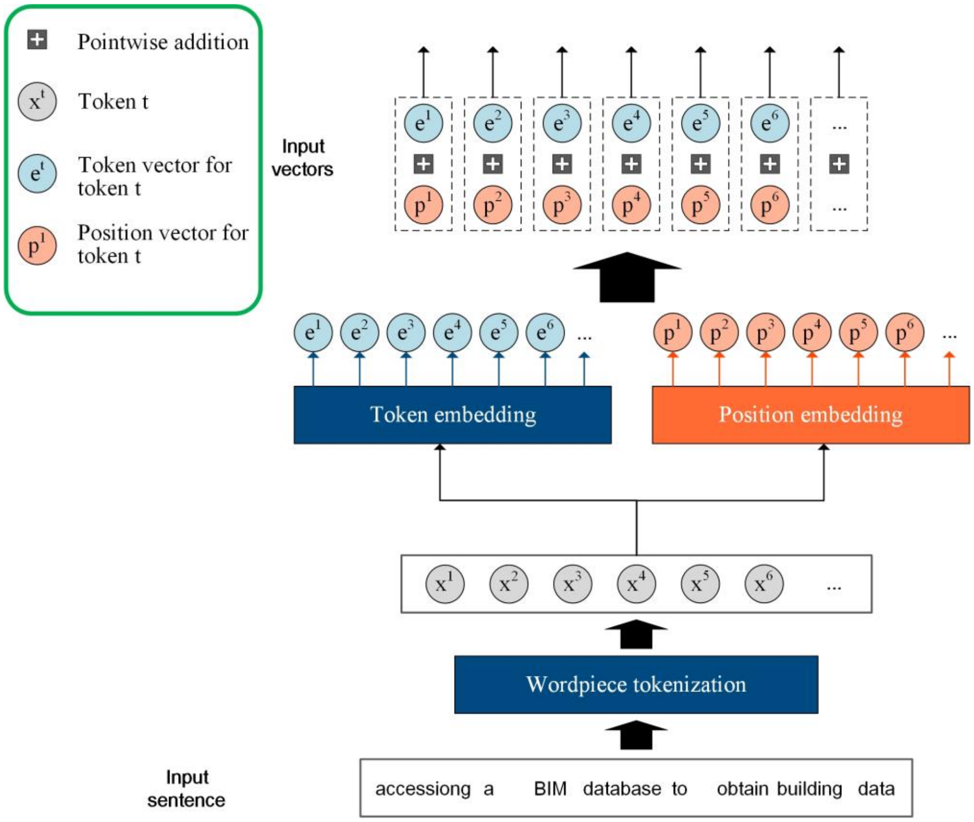


Figure 5-5 Tokenization and embedding process of the deep learning model

5.4.1.1 Wordpiece tokenization and token embedding

This study uses wordpiece to tokenize the input text based on byte-pair encoding. Unlike other tokenization methods such as NLTK tokenization (Bird and Loper, 2004), wordpiece tokenization is a simple compression algorithm and has been successfully used in several NLP tasks, representing infinite vocabulary of words based on a medium size corpus of texts for pre-training. Given the desired tokens V, the motivation, according to the selected algorithm, is to select minimal segmented wordpieces that can make combinations of the tokens V (Qiu et al., 2019). The algorithm is shown in Figure 5-6, which is implemented in Python programming.

Algorithm 1: Wordpiece tokenization

```
def statistic(vocabulary):
    pairs_words = coll.defaultdict(int)
    for token, count in vocabulary.items():
        characters = token.split()
        for i in range(len(characters) - 1):
            pairs_words[characters[i], characters[i + 1]] += count
    return pairs_words

def merge_pair(pair, input_vocabulary):
    output_vocabulary = dict()
    Byte_encoding = re.escape(' '.join(pair))

    p = re.compile(r'(?<!\S)' + Byte_encoding + r'(?!\S)')
    for word in input_vocabulary:
        w_out = p.sub(".join(pair), word)
        output_vocabulary[w_out] = input_vocabulary[word]
    return output_vocabulary

Input: a token vocabulary V and the corresponding occurrence times, and the
number for merge times.
Begin
    for i in range(num_merges):
        pairs = statistic(vocabulary)
        best = max(pairs, key=pairs.get)
        vocabulary = merge_pair(best, vocabulary)
        a = a + 1
End
```

Figure 5-6 Wordpiece embedding algorithm

The core idea lies in the byte-pair encoding, the core algorithm of wordpiece tokenization, is that any word could be represented by parts, which is inspired by the so-called morphology, a linguistic methodology indicating how the words are made by morphological forms. Figure 5-7 plots the morphology tree for word “independently”, which splits the word by morphological forms from top to bottom. Most of the English words could be represented by morphological forms which always have important meanings.

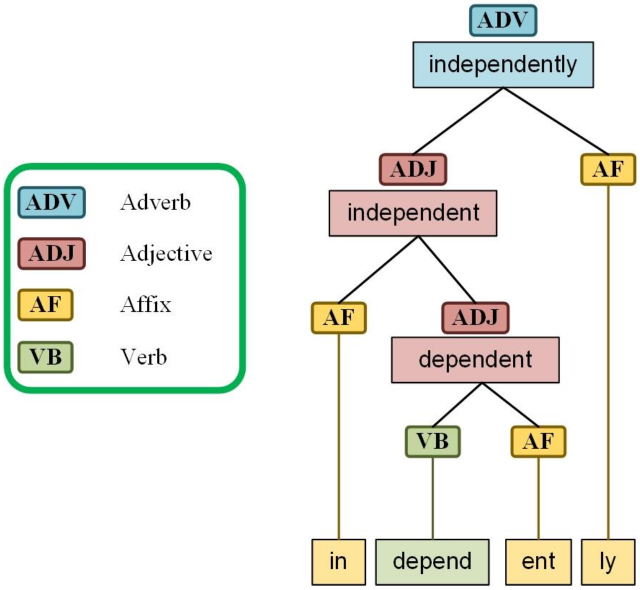


Figure 5-7 An example of morphology tree

Given a corpus of texts which has a vocabulary of ['transmitting': 35, 'bidding': 30, 'screen': 31, 'camera': 31, 'accessing': 40, 'building': 420, 'receiving': 168]. The algorithm first splits all the words as single characters and adds an end symbol (</w>) after the last character of each word : ['t r a n s m i t t i n g </w>': 35, 'b i d d i n g </w>': 30, 's c r e e n </w>': 31, 'c a m e r a </w>': 31, 'a c c e s s i n g </w>': 40, 'b u i l d i n g </w>': 420, 'r e c e i v i n g </w>': 168], in which the numbers denotes the occurrence times for the words. All the split characters are viewed as symbols, and the basic idea is to iteratively integrate the most frequent pair of symbols into a new symbol (Heinzerling and Strube, 2018).

The deep learning model uses a pre-trained word embedding of wordpiese, containing 30,522 tokens. It has been manifested that 8k - 30k could work with any sequence of symbols, including the out of vocabulary words and special symbols (Heinzerling and Strube, 2018; Schnabel et al., 2015; Wu et al., 2016). Wordpiece uses “##” to denote split pieces. Figure 5-8 displays an example of tokenization process of a sentence. The input sentence “accessing a BIM database to obtain building data” is tokenized as “['[CLS]', 'access', '##ing', 'a', 'bi', '##m', 'database', 'to', 'obtain', 'building', 'data', '[SEP]']”. The result tokenized sequence consists of 12 tokens, in which two words “accessing” and “BIM” are represented by part-of-words and [CLS] and [SEP] are added to denote the start and end of the input sequence. This step results in a vocabulary ID for each token.

According to the IDs, each word is converted into a 512 dimensions vector representation, through a pre-defined token embedding matrix $\mathrm{D} \in \mathbb{R}^{|\mathrm{v}|\times|d|}$ (in this study, |v| = 30,522, $|d|$= 512). Similar to the word2vec and GloVe, the pre-defined token embeddings were set as parameters that are fully connected to the whole model and could be updated by the gradients through the back-propagation process.

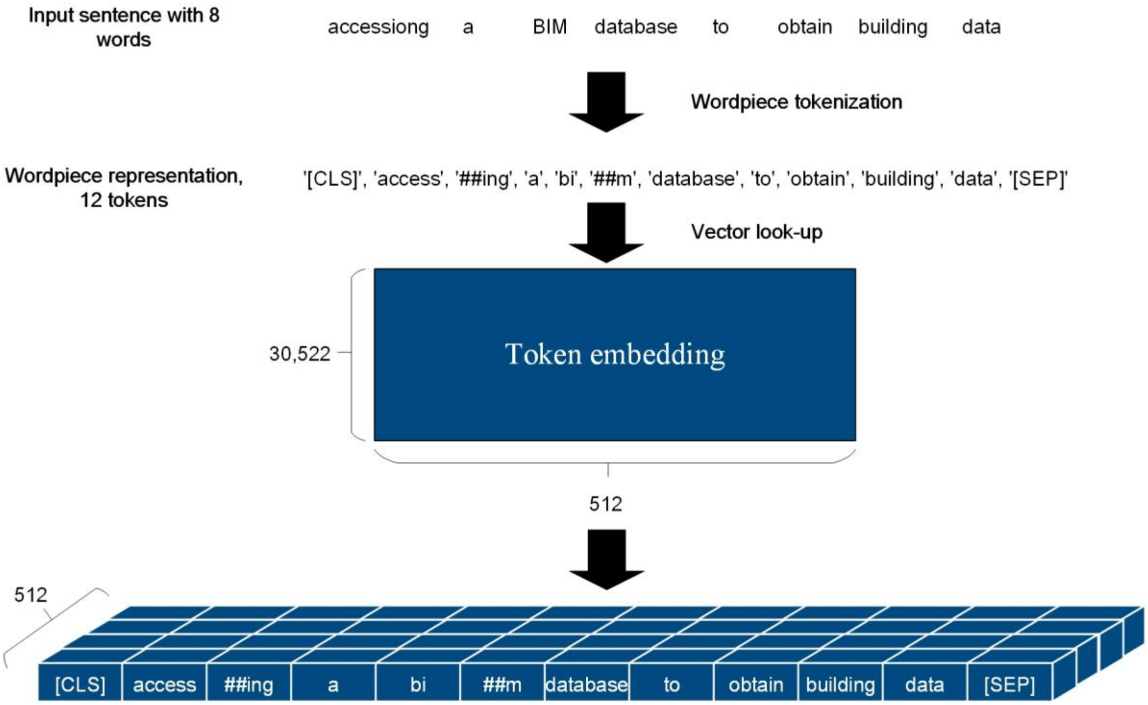

Input sentence with 8 words
accessiong a BIM database to obtain building data
Wordpiece tokenization
Wordpiece representation, 12 tokens
'[CLS]', 'access', '##ing', 'a', 'bi', '##m', 'database', 'to', 'obtain', 'building', 'data', '[SEP]'
Vector look-up
30,522
Token embedding
512
512
[CLS]
access
##ing
a
bi
##m
database
to
obtain
building
data
[SEP]

Figure 5-8 The process of token embedding

5.4.1.2 Position embedding

The deep learning model discards the RNN structure which naturally fits modeling language, because the neural network in RNN is recurrent in a sequence, crabbing the order information of a sentence. Therefore, the deep learning model has to encode the position information in the inputs. It utilizes a position embedding method at the input layer to encode the sequential property for each word in the input sentence. For example, the token "building" in "building a model" and "a building model" would have different word embeddings in this step. To achieve this object, each word should have a positional vector, which could encode the position information of the word in the sequence. The generated positional vector for each word should have the same embedding dimension with the token vector to enable the pointwise addition operation. Little modification of adding the position embeddings to token vectors is preferred because the semantic and syntactic meaning for a word highly depends on the token vectors. The model applies a sinusoidal function as the positional embedding model (Vaswani et al., 2017), presenting as follows:

$$\mathrm{PE}_{i,j} = \begin{cases} \sin\left(\frac{i}{10000^{\frac{j}{d_{ed}}}}\right), if\ j\ is\ even \\ \cos\left(\frac{i}{10000^{\frac{j-1}{d_{ed}}}}\right), if\ j\ is\ odd \end{cases} \tag{5.1}$$

where i denotes the position for the token to be embedded, and j denotes the dimension of the word embedding (in this study, j = 768). As shown in Table 5.2, each token is embedded with a position vector, which is determined by the position in the tokenized sequence but not the token itself. By using Eq. (5.1), position embedding not only encodes the position information for each word but also makes all the values in the generated positional vectors in the range [-1,1]. In addition, the sinusoidal function allows a simple linear representation from $PE_{i,j}$ to $PE_{i,j+k}$, providing an easy way for neural networks to learn the position relations between tokens in different positions. In the training process, the sinusoidal algorithm has to two main advantages: (1) it can memorize the position information for a much longer sequence compared to other position embedding models, and (2) the sinusoidal function involves fewer parameters, leading to less computation resource in the training process (Vaswani et al., 2017).

Table 5.2 Positional vectors based on sinusoidal function

Tokens	Position	position vector				
		d_1	d_2	d_3	…	d_{512}
[CLS]	1	$\cos\left(\frac{0}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{0}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{0}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{0}{10000^{\frac{512}{512}}}\right)$
access	2	$\cos\left(\frac{1}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{1}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{1}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{1}{10000^{\frac{512}{512}}}\right)$
##ing	3	$\cos\left(\frac{2}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{2}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{2}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{2}{10000^{\frac{512}{512}}}\right)$
a	4	$\cos\left(\frac{3}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{3}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{3}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{3}{10000^{\frac{512}{512}}}\right)$
bi	5	$\cos\left(\frac{4}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{4}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{4}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{4}{10000^{\frac{512}{512}}}\right)$
…	…	…	…	…	…	…
[SEP]	12	$\cos\left(\frac{12}{10000^{\frac{0}{512}}}\right)$	$\sin\left(\frac{12}{10000^{\frac{0}{512}}}\right)$	$\cos\left(\frac{12}{10000^{\frac{2}{512}}}\right)$	…	$\sin\left(\frac{12}{10000^{\frac{512}{512}}}\right)$

5.4.2 Self-attention

In a general perspective, the concept of attention in neural networks refers to choices, which are made by the neural networks about which feature should be paid attention to, and how attention should be paid. The ability of attention exists in human brains, through occasionally perceiving the input signals (i.e., visions, voices, and sensations), choosing to focus on parts of the signals and to ignore others. Attention plays a central role for human to conceive and understand the world, because the bandwidth in human brains to process the input signals is narrow, and some inputs are more valuable in determining the outputs according to the targets. Attention could happen either from inputs to outputs or in inputs themselves, namely global attention and intra-attention (or self-attention) respectively. For example, Buddhism channels his attention of others to preach, and he can also channel his own attention to attain enlightenment.

In NLP, self-attention is a neural network that addressing the relations between the words in a sentence when computing a corresponding output sequence. The self-attention, as an alternative to the logic of comprehension of the sentence by human beings, provides a process to address attention on interdependencies among the words in the sentence. The frequently used word embeddings such as word2vec and GloVe were obtained by training a large corpus of documents (always the Wikipedia articles), representing the words' meaning by a vector. Such a word vector indeed reflects the average contextual meaning overall the sentences it appears, which is fixed after training by word2vec or GloVe model. However, the meaning of a word may vary according to the surrounding texts. For example, the phrase "building data" may convey two different meanings, architecture model data or process of creating data, and one can get the meaning it exactly reflects unless after getting to know the context surrounding it.

This study, similar to the examples above, uses the sentence “accessing a BIM database to obtain building data” as an illustration for self-attention (Figure 5-9). After tokenization by wordpiece, the sentence splits into 12 tokens. For a word that is split into Wordpiece tokens in the tokenization process, this study keeps the word’s label for the first sub-token and annotates the others as “X”.

The task is to train a model to predict the annotated communication entity classifications with higher accuracy. Self-attention refers to the function that pays attention to tokens in the same sentence that may affect the prediction. The “building data” alone does not convince people that this phrase is a communication-oriented entity and should be categorized into “transferred information”, unless comprehension of the whole sentence provides a clue that (1) the verb “obtain” is ahead of “building data”, indicating that the building data is an object consisting of two nouns that denote architecture model data, but not a combination of verb and noun that denotes the process of creating data; (2) “accessing”, “BIM database”, “obtain”, “building data” and their position information could let a reader be aware that “building data” is a TI, reflecting the building data stored in the BIM database and can be transmitted through accessing the BIM database.

The self-attention is actually a neural network that integrates key, query, value memory networks and an intra-decoder attention algorithm. To support a better understanding of the self-attention, this study first introduces the general attention and intra-attention mechanism in RNN-based models, then describes the details of the self-attention.

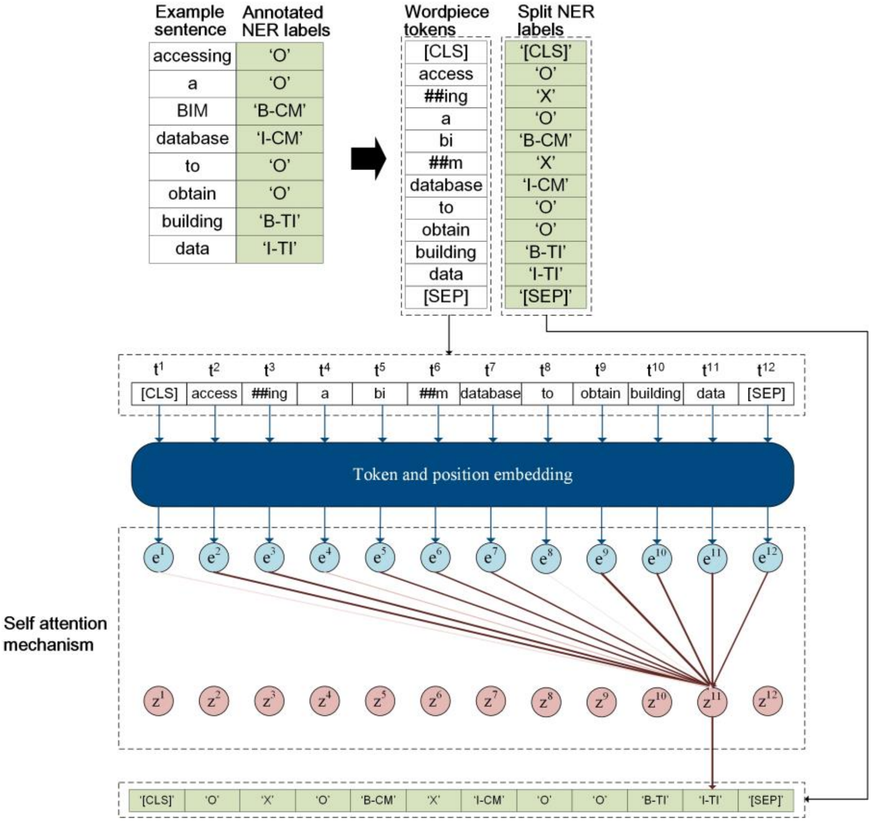


Figure 5-9 A simple example of the self-attention mechanism in modeling a sentence

5.4.2.1 General attention mechanism in RNN-based models

The attention mechanism was firstly used in machine translation tasks based on RNN architecture (Bahdanau et al., 2014), always incorporating an encoder and a decoder. Normally, the first hidden state value in the decoder is the last hidden state in the encoder. In this way, the decoder has to treat all the information of the inputs as a whole in each step. Attention mechanism offsets the burden of the decoder, by paying attention to part of the inputs that may affect the outputs. Figure 5-10 plots the attention mechanism that has been addressed in Seq2Seq models.

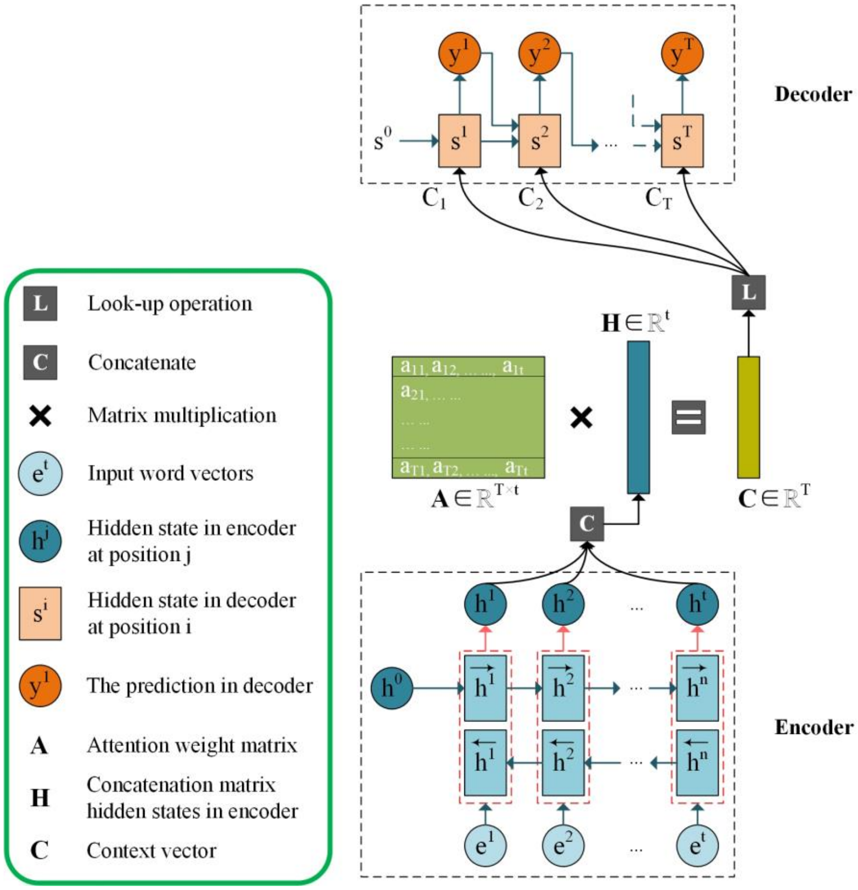


Figure 5-10 Attention mechanism in RNN

In Seq2Seq models, the hidden states in decoder normally receive the encoder signals at S^0, which is equal to h^t. The encoder signals transit in decoder from a state to the next. Attention mechanism transit encoder signals relying on a context matrix C, in which each element denotes a combination of hidden states, being input as singles into corresponding hidden states in a decoder. Equation 5.2 denotes the computation of context matrix C.

$$\mathrm{C} = \mathrm{AH} \tag{5.2}$$

Where A denotes the attention weight matrix and H is a matrix concatenating the hidden states of an encoder. The attention weight matrix consists of alignment scores computed by an alignment model. Those scores determine how much of the inputs before position j in the encoder and outputs before i in the decoder related. The alignment model is actually a feedforward neural network, connecting the hidden S_{i-1} and h_j. The alignment model computes the scores by the following equation:

$$\mathrm{a}_{ij} = \frac{\exp(\mathrm{r}_{i,j})}{\sum_{j=1}^{nx} \exp{(\mathrm{r}_{ij})}} \tag{5.3}$$

where T_x is the length of the input sequence, and e_{ij} is computed by a content-based score function:

$$\mathrm{r}_{ij} = \begin{cases} S^{(i-1)}\left(h^{(i)}\right)^T, \; dot \\ S^{i-1}W_{ag}\left(h^{(i)}\right)^T, \; general \\ V_{ac}tanh\;(W_{ac}\left(\left[S^{(i-1)}h^{(i)}\right]\right)^T \; + b_a), \; concat \end{cases} \tag{5.4}$$

From the equation we can see that the score function, whatever dot, general or concatenate operation, is actually a feedforward neural which is a component in the deep learning model, updating the parameters in the back-propagation according to the gradients (Bahdanau et al., 2014). According to Eq. (5.2), a_{ij} is the normalization process (the same normalization process of softmax) that transforms score r_{ij} into probabilities (sum to 1), reflecting in what a combination of hidden states $h^{(i)}$ in the decoder in determining the hidden state $S^{(i-1)}$ and the output y_i.

5.4.2.2 Intra-decoder attention

The attention mechanism allows the decoder to pay more attention to useful inputs, but repeated phrases have been reported in long sequences due to the neglect of the attention on previous hidden states in the decoder. The intra attention, therefore, was introduced to address varied attention on previous hidden states in a decoder, avoiding too much focus on repeated information that inherently exists in hidden states (Paulus et al., 2017). The computation is similar to general attention:

$$\mathrm{a}_{t,t'} = \frac{\exp\left(\mathrm{r}_{t,t'}\right)}{\sum_{t'=1}^{n} \exp\left(\mathrm{r}_{t,t'}\right)} \tag{5.5}$$

$$C = \mathrm{AS} \tag{5.6}$$

$$\mathrm{r}_{t,t'} = \begin{cases} S^t\left(S^{t'}\right)^T, \ dot \\ S^t W_{ag}\left(S^{t'}\right)^T, \ general \\ V_{ac} tanh\ \left(W_{ac}\left(\left[S^t S^{t'}\right]\right)^T + b_a\right), concat \end{cases} \tag{5.7}$$

where $W_{ag} \in \mathbb{R}^{\mathrm{dm} \times dm}$, $W_{ac} \in \mathbb{R}^{\mathrm{dc} \times 2dm}$ and $V_{ac} \in \mathbb{R}^{1 \times \mathrm{dc}}$. Therefore, to compute the energy $\mathrm{r}_{t,t'}$, the dot, general and concatenate operation add 0, dm^2 and dc(1+2dm) parameters respectively.

5.4.2.3 Self-attention in transformer

The RNN-based deep learning model is built upon an encoder-decoder architecture. The computation in each layer is a regressive process, whereby the output value in the previous position t-1 makes of additional input of the computation process in position t (Graves, 2013). Similar to the RNN-based deep learning model, the transformer was designed with a decoder-encoder structure, which discards the RNN neurons and relies almost on the self-attention algorithm. The proposed self-attention has a similar computation process with intra-attention, achieved by Query-Key-Value memory networks.

Self-attention is similar to intra-decoder attention with dot operation in Eq. (5.7). The goal is to compute a combination of all the inputs for an input token at position #t by a context vector c^t, in which each element represents a weight of an input. The bottom of the self-attention is the inputs E = (e^1,…,e^n), incorporating n tokens of the input sentence where each token is represented by an embedding $e^t \in \mathbb{R}^{1\times dm}$computed by contextual word representation. The outputs of self-attention are the context matrix Z, in which each element is computed as follows:

$$z^{(t)} = \sum_{t'=1}^{n} \mathrm{a}_{t,t'}(\mathrm{x}_{t'}W^{v}) \tag{5.8}$$

$$\mathrm{a}_{t,t'} = \frac{\exp\left(\mathrm{r}_{t,t'}\right)}{\sum_{t'=1}^{n} \exp\left(\mathrm{r}_{t,t'}\right)} \tag{5.9}$$

$$\mathrm{r}_{t,t'} = \frac{e^{(t)}W^{Q}\left(e^{(t')}W^{K}\right)^{T}}{\sqrt{dc}} \tag{5.10}$$

where:

- t is the target position which is intended to compute an output by self-attention, which aims to compute an attention matrix A in which each row consists of coefficients (sum up to 1) representing the normalized attention weights.

- t' denotes the position and self-attention aims to compute how much attention is addressed from t' to t.

- $c^{(t)} \in \mathbb{R}^{1\times dc}$. In this study, dc is set as 64 to speed up computation. $c^{(t)}$ is the attention vector

- $W^Q, W^K, W^V \in \mathbb{R}^{\mathrm{dm} \times dc}$. $W^Q, W^K\ and\ W^V$ are query, key and value memory matrix, fully connected with the whole deep learning model and the elements in the matrices are parameters to be estimated during the feedforward and back-propagation process via stochastic gradient descent.

- W^Q W^K and $W^V$are used to compute the query vector, key vector and value for an input $e^{(t)}$.

- $\mathrm{r}_{tt'}$ is an energy score from $e^{(t')}$ to $e^{(t)}$, achieved by a scaled dot product operation that facilitates computation. $\mathrm{r}_{tt'}$ reflects how much of the input $e^{(t')}$ with respect to $e^{(t)}$.

To illustrate the computing process of self-attention, Figure 5-11 plots its vector computation from e^{11} to z^{11}. The first step is to compute energy scores each of which denotes how much attention should the tokens in the sentence pay to "data" (represented by e^{11}) according to Eq. (5.10). Each energy score $\mathrm{r}_{11,t'}$ would be computed by the dot production between the query vector of "data" and key vector of token t' in the sentence. Once the energy scores are computed, they are fed into a feedforward network with a softmax activation function, transforming the energy scores into probabilities that sum up to 1, constituting the attention vector. By multiplying weights in the attention vector to corresponding vectors and sum them together, the output z^{11} is achieved.

Compare to the intra-attention algorithm mentioned above, self-attention adds the operation of Query-Key-Value memory networks in which elements are all parameters.

The added Query-Key-Value memory networks have three main benefits: (1) Effective transformation manner would be achieved for the whole system due to the computation process for energy $\mathrm{r}_{t,t'}$ (Eq. (5.10)). Dot production generates a greater score if the two vectors are more similar and thus being used as a directed way to measure the similarity between two vectors. Without query and key matrices, the dot production (same as the dot operation in intra-attention, see Eq. (5.7)) would be operated between the two token vectors, and the generated score reflects no more than the word-level similarity. (2) Compared to the *general* and *concatenate* operation in Eq. (5.7), self-attention would alleviate the computation burden because fewer parameters would be achieved. As mentioned above, the *general* and *concatenate* operation for energy in intra-attention would add dm^2 and dc(1+2dm) parameters. Self-attention, with the memory matrices, would add dc × dm or 2dc × dm parameters that are all fewer than the number in intra-attention, depends on conditions that the self-attention to be applied (these conditions are described in the following sections). Moreover, the output of self-attention has a shorter dimension dc (dc is always set much less than dm. In this study dc and dm are set as 64 and 768 respectively), significantly reducing the computation cost in the next layers. (3) The self-attention provides an advanced architecture to replace the recurrences in the RNN-based models. The advancement relays on the more efficient memory system and much fewer parameters used in a simplified neural network structure. On the one hand, RNN-based models predict the output based on the previous stream of words, and each recurrence could only remember information in the very nearby tokens. The intra-attention mechanism, as well as LSTM structure and CNN are applied in RNN-based models to rectify poor memory capacity in RNNs, but the memory performance is still unsatisfied (Weston et al., 2014). The Query-Key-Value memory networks embodied in self-attention provides more efficient and flexible memories than recurrences or LSTM. Specifically, LSTM and CNN are used to replace the recurrence to provide more memory about the long and short dependencies, and intra-attention is used to address the dependencies between the hidden states. Self-attention, however, with the query-key-value memory matrices, provides a much easier way to remember the dependencies between inputs. All the dependencies would be stored in the memory matrices, and the lookup operation provides flexibility for the transformation from input to outputs. Moreover, the self-attention has a simple structure compared to an RNN-based model with *LSTM+CNN+intra-attention* structure, leading to an efficient training process through the feed-forward and back-propagation. On the other hand, the recurrence structure is no longer used in the self-attention layer, in which the hidden states are discarded, and all the input neurons and connected memory matrices are parallelly connected, not in a sequence such as RNN. This not only avoids the gradients explosion and vanish in training that would happen in RNN-based models, but also reduces a large number of parameters due to the absence of complicated LSTM or RNN always involving many parameters.

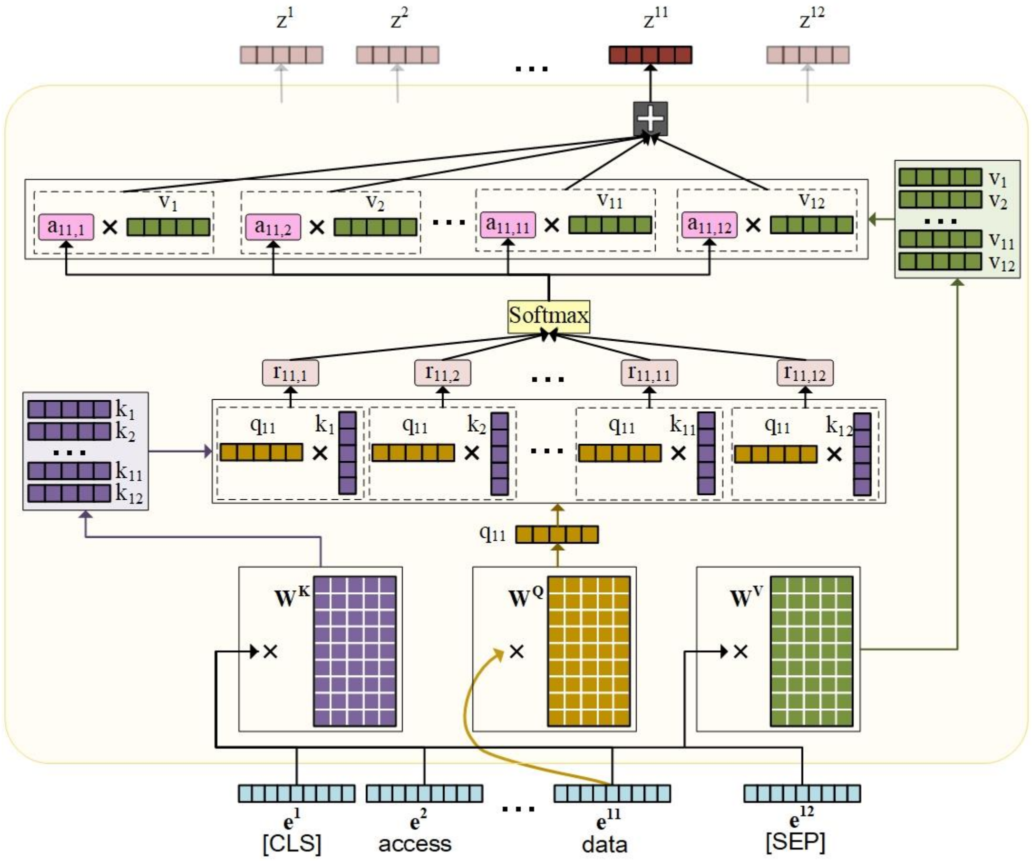


Figure 5-11 Computation process of z_{11} by self-attention

The transformer is a fully connected network which computes the self-attention by stacking the single vectors into a matrix. Therefore, self-attention could be represented by equations using matrices:

$$E = \left\{\begin{matrix} (e^1)^T \\ \dots \\ \dots \\ \dots \\ (e^n)^T \end{matrix}\right. ,Z = \left\{\begin{matrix} (z^1)^T \\ \dots \\ \dots \\ \dots \\ (z^n)^T \end{matrix}\right. ,\mathrm{A} = \begin{pmatrix} \mathrm{a}_{11} & \dots & \mathrm{a}_{1n} \\ \dots & \dots & \dots \\ \mathrm{a}_{n1} & \dots & \mathrm{a}_{nn} \end{pmatrix} \tag{5.11}$$

Attention matrix A:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{dc}}\right) \tag{5.12}$$

Output matrix Z could be express as an alignment equation $\text{Alig}(\mathrm{Q}, \mathrm{K}, \mathrm{V})$

$$Z = \text{Alig}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{dc}}\right) V \tag{5.13}$$

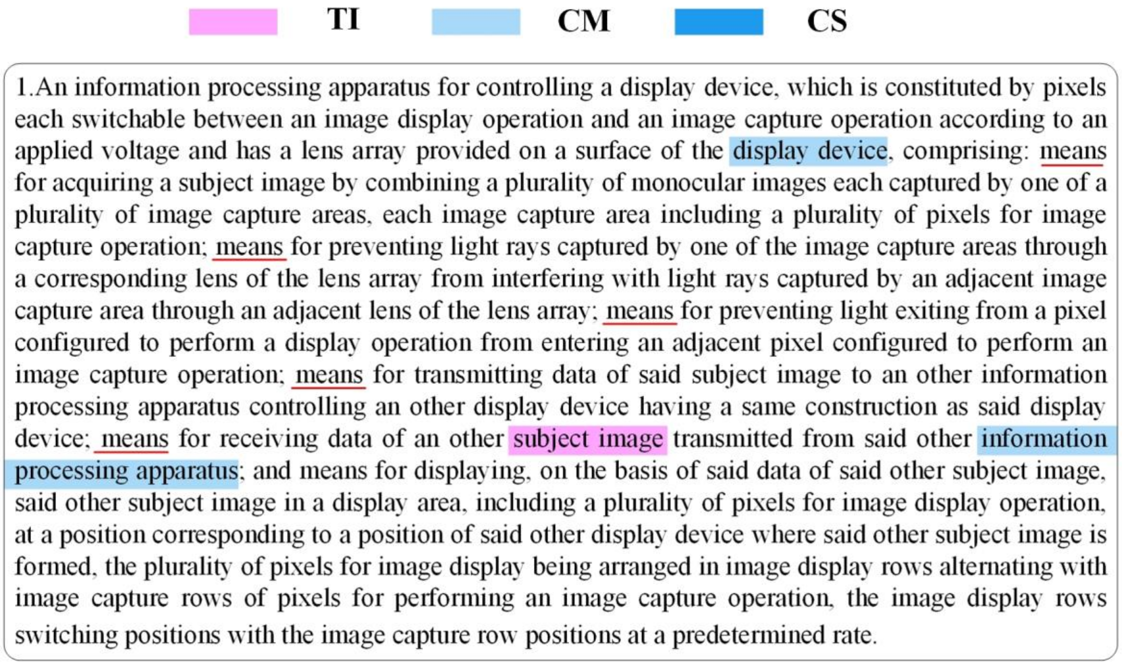
where V = EW^{V}, K = EW^{K} and Q = EW^{Q}. $W^Q, W^K, W^V \in \mathbb{R}^{\mathrm{dm} \times dc}$. Normally, dc is set as dm/h, where h could be set arbitrarily. This study follows the setting of Google AI team's research (Devlin et al., 2018; Vaswani et al., 2017), sets h as 8 and thus dc = dm/h = 64.

5.4.2.4 Benefits of Self-attention

Compared with RNN, CNN and fully connect feed-forward networks, the merits of self-attention are based on the following two perspectives:

(1) The ability to capture long-term dependencies

Figure 5-12 illustrates a long sentence containing 285 words and punctuations. Such long sentences are very common in technical documents, especially patents. The mentions of "display device" (dashed with blue background color in Figure 5-12) and the forward words could not provide sufficient information to make a judgment that the "display device" is a communication model, because these mentions do not indicate this "display device" is used for communication. However, after the appearance of "display device", a number of mentions have relations with this phrase, including the "subject" (dashed with pink background color in Figure 5-12). By reading the sub-sentence where the "subject image" appears, it could easily indicate that the "subject image" is digital data that can be transferred from "information processing apparatus" to the "means", which represents a technical component of the aforementioned "display device". All these local information and the long-term dependencies could help to recognize that the "display device" is a communication model. Effectively addressing these long-term dependencies within a sentence poses a challenge for neural network.



1.An information processing apparatus for controlling a display device, which is constituted by pixels each switchable between an image display operation and an image capture operation according to an applied voltage and has a lens array provided on a surface of the display device, comprising: means for acquiring a subject image by combining a plurality of monocular images each captured by one of a plurality of image capture areas, each image capture area including a plurality of pixels for image capture operation; means for preventing light rays captured by one of the image capture areas through a corresponding lens of the lens array from interfering with light rays captured by an adjacent image capture area through an adjacent lens of the lens array; means for preventing light exiting from a pixel configured to perform a display operation from entering an adjacent pixel configured to perform an image capture operation; means for transmitting data of said subject image to an other information processing apparatus controlling an other display device having a same construction as said display device; means for receiving data of an other subject image transmitted from said other information processing apparatus; and means for displaying, on the basis of said data of said other subject image, said other subject image in a display area, including a plurality of pixels for image display operation, at a position corresponding to a position of said other display device where said other subject image is formed, the plurality of pixels for image display being arranged in image display rows alternating with image capture rows of pixels for performing an image capture operation, the image display rows switching positions with the image capture row positions at a predetermined rate.

Figure 5-12 An example of a lengthy sentence of ICT in construction

Figure 5-13 plots the encoding manners of RNN, CNN, fully connect FNN and self-attention. In NLP tasks, an encoder models the dependencies of the input features in hidden states (z^1~z^n in Figure 5-13), by assigning the importance of the weights of the NNs. A deep learning model, consisting of several connected NNs, aims to accurately determine that importance that by updating the quantities of the weights based on gradients. Then, the summed importance of the input features embodied in hidden states, predicts the output by learning what a combination of dependencies of the input features has a high correlation with outputs. RNN receives input from the present position, as well as input from hidden states in the previous position. However, each hidden state could only remember information in the very nearby tokens because of the explosion and vanish of gradients. CNN is designed to capture short-term dependencies. Fully connect FNN could learn the long-term dependencies, but it can hardly work when the sequence is long because each neuron has a unique weight to be learned and the parameters increase exponentially when the number of inputs increases. Self-attention, provides a different structure, encoding all the dependency information in external memory networks. Through the look-up operation, queries, keys, and values can be easily obtained and used to compute the combination of dependencies for each position.

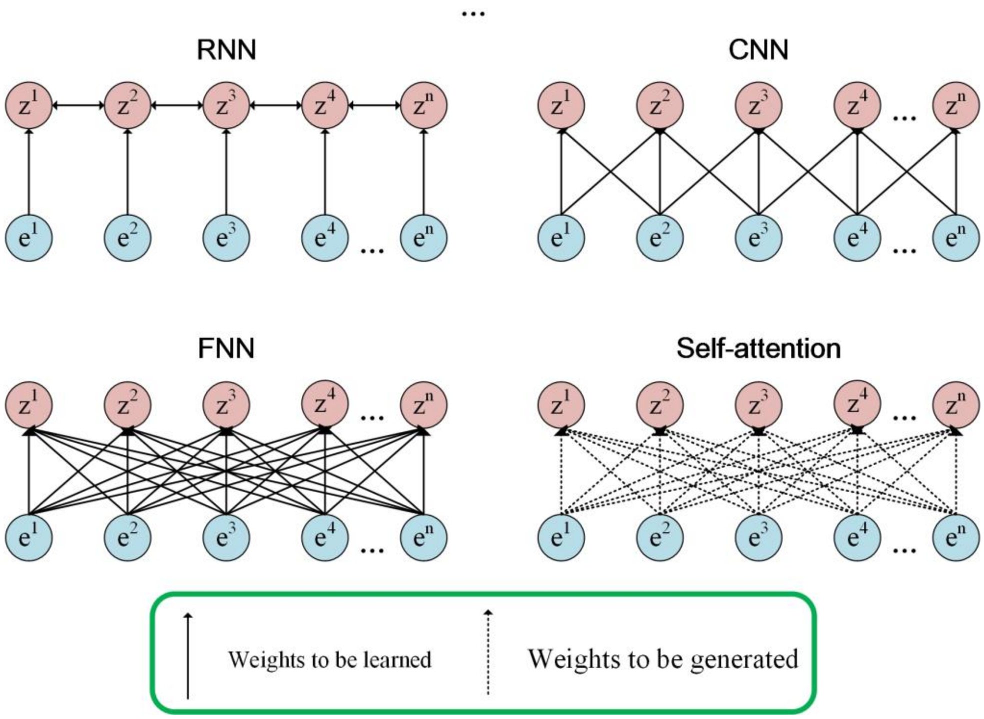


Figure 5-13 The encoding manners of RNN, CNN, FNN, and self-attention

(2) More efficient in feed-forward and back-propagation

The self-attention provides an advanced architecture to replace the recurrences in the RNN-based models.

The intra-attention mechanism, as well as LSTM structure and CNN, are applied in RNN-based models to rectify poor memory capacity in RNNs, but the memory performance is still unsatisfied (Weston et al., 2014). Self-attention, however, with the Query-Key-Value memory matrices, provides a much easier way to remember the dependencies between inputs. All the dependencies would be stored in the memory matrices, and the lookup operation provides flexibility for the transformation from input to outputs. Moreover, the self-attention has a simple structure compared to an RNN-based model with *LSTM+CNN+intra-attention* structure, leading to an efficient training process through the feed-forward and back-propagation. On the other hand, the recurrence structure is no longer used in the self-attention layer, in which the hidden states are discarded and all the inputs neurons and connected memory matrices are parallelly connected, not in a sequence such as RNN. This not only avoids the gradients explosion and vanish in training that would happen in RNN-based models, but also reduces a large number of parameters due to the absence of complicated LSTM or RNN always involving many parameters. In addition, compared with fully connected networks which draw all the dependencies without consideration of focus, the self-attention based model has much fewer parameters to be learned.

5.4.3 Multi-head Self-attention

Rather than making outputs with |d| dimension, it is reported that linearly projecting the queries, keys, and values h times with dim/h dimensions (dc) would achieve better performance (Vaswani et al., 2017).

$$Multi_{head(Q,K,V)} = \text{Concat}(\text{Z}_1, \dots, \text{Z}_h)W^o \tag{5.14}$$

where

$$Z_i = \text{softmax}\left(\frac{\left(QW_i^Q\right)\left(K^T W_i^K\right)}{\sqrt{dc}}\right)\left(VW_i^V\right) \tag{5.15}$$

where the projection is computed upon the parameter matrices $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{\text{dim}\times dc}$ and $W^o \in \mathbb{R}^{\text{dim}\times\text{dim}}$.

Figure 5-14 plots the process of multi-head attention. The computation cost of the multi-head attention (each attention results in a $z \in \mathbb{R}^{1\times dc}$) is similar to a single self-attention that yields a $Z \in \mathbb{R}^{\text{dim}\times(\text{h}\times dc)}$, but it is empirically found that multi-head attention would achieve much better performance (Vaswani et al., 2017). The parameter matrices $W_i^Q, W_i^K\ and\ W_i^V$ are initialed with different values, which allows the model to capture different representations for different inputs, while the single attention can overlook some different representations. This is mainly due to the nature of deep neural networks, which typically learn the relations from inputs to outputs with linear transformation and non-linear activation functions, amplifying the updates of the parameters by the gradients. Simply enlarge the dimensions of the memory network which always leads to problems with overfitting (Blei et al., 2003).

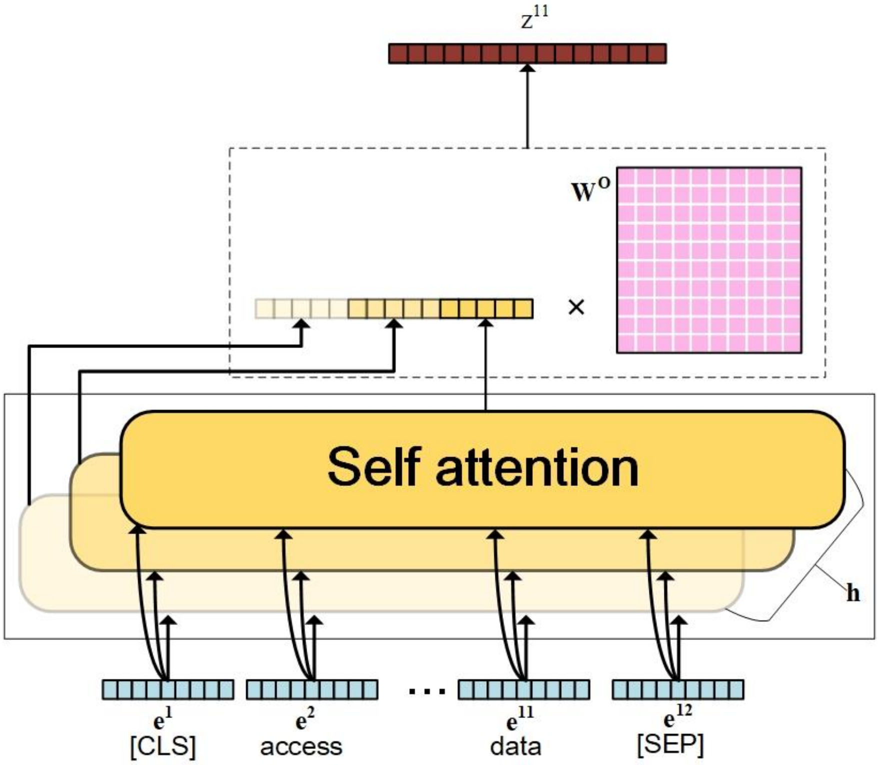


Figure 5-14 Neural networks of multi-head attention

The author, according to the own intuition, explains multi-head attention using the following example: Let two reader A and B predicts the entity tags of the sentence “accessing a BIM database to obtain building data”. Reader A has an extraordinary memory and he is allowed to read the sentence once before prediction; Reader B has a normal memory and he is required to read the sentence several times and each time he has to focus on different parts of the sentences. Once, he may get to know the BIM database can be accessed by focusing on the first three words; twice, by focusing on last three words, he may be aware that building data is something that could be obtained; third time, he may get to know that the something could be obtained through accessing something; Finally, reader B makes the predictions based on connecting all the three attentions.

For example, a Ph.D. student may not firmly indicate that “building data” is an entity of transferred information after reading once the sentence “accessing a BIM database to obtain building data”. However, it would be easy work to indicate after reading the sentence several times, and each time the sentence is read from a different perspective or focus.

5.4.4 The Sub-layer of Point-wise Feed-forward

The outputs of the self-attention sub-layer would be fed into a fully connected sub-layer of point-wise Feed-forward networks (FFN), which treats each position independently and identically. The feed-forward network is consisting of two linear transformations and a ReLU activation function.

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_1 \tag{5.16}$$

A transformer consists of two sub-layers: multi-head self-attention and FFN. Each of the sublayers is connected with a residual connection and layer-normalization. It can be expressed by a combined equation:

$$h^{(t)} = f_{Layer\_nor}(e^{(t)} + \, z^{(t)}) \tag{5.17}$$

The residual connection is a typical technique in deep learning models, which makes deep neural networks easier to train by adding the input to the output of the last layer (He et al., 2016). In addition, it carries positional information to higher layers. Normalization is a regular operation in deep learning models from one layer to another before the non-linear activation function, leading to a faster converge with stochastic gradient descent. For an input vector $x^{(t)} = (x_1^{(t)}, \dots, x_{dim}^{(t)})$, layer-normalization computes the mean and variance value across the elements:

$$f_{Layer\_nor}\left(x^{(t)}\right) = (\hat{x}_1^{(t)}, \dots, \hat{x}_{dim}^{(t)}) \tag{5.18}$$

where

$$\hat{x}_i^{(t)} = \frac{x_i^{(t)} - \mu^{(t)}}{\sigma^{(t)}} \tag{5.19}$$

5.5 Summary of this Chapter

This chapter illustrates the architecture of the neural networks of the developed deep learning model. The basic concepts of CER and technical problems were clarified. According to the objective of CER and the technical problems, a deep learning model based on the transformer was developed. The overall structure of the deep learning model was illustrated in detail.

The pre-training phase aims to make the model acquire the ability to indicate a token by the contextual representations. The masked language model was used as the NLP task in the pre-training phase following a semi-supervised, in which the model learned the contextual representations for each word from the BooksCorpus and Wikipedia. The model, after the pre-training phase, has the ability to indicate what a combination of other tokens as well as positions may have more effects on determining the meaning of a token in a given sentence.

The fine-tuning phase, based on the pre-trained parameters, continues to train the model with the annotated data: tokens of a sentence as inputs and the predicted probabilities of entity labels as outputs. After fine-tuning, the model is built completely, with the ability to predict an entity classification of a given word based on the contextual representation.

Chapter 6 Data Acquisition and Annotation for Training the Deep Learning Model

6.1 Introduction

To train the deep learning model for CER of ICT in construction, a corpus of texts relevant to ICT in construction with CE labels are required. This study compiles a database of patents that are relevant to ICT in construction. Furthermore, based on the patents of ICT in construction, annotated data were labeled manually.

6.1.1 Motivation for Data Selection

Patent offices (such as World Intellectual Property Organization (WIPO), USPTO and European Patent Office (EPO), and USPTO) provide plenty of structured data, such as citation information, but it has been widely accepted that most of the technical information was archived as unstructured textual data in the patent documents. In the construction industry, the patent documents do provide valuable information, but they are no more than raw data without perceiving and understanding. For example, for a specific patent of ICT in construction, one cannot even exactly recognize the topic until he or she read the title, and cannot perceive problems it may solve, technological components and processes it has. For a pool of patents related to ICT, one cannot recognize which patents are relevant to the construction industry and which patents share similar topics. However, it would be an extremely difficult task to manually analyze a large number of patent documents. Regardless of the formal and concise language, the context in patents incorporates a lot of technological terminologies, and it would consume a lot of time and lead to inevitable personal bias especially for non-users (Rezaeian et al., 2017). Recently, approaches that integrate techniques of natural language processing (NLP), machine learning methods are widely used to analyze the natural language to convert raw information into knowledge (Souili et al., 2015). This offers an opportunity to analyze the ICT in construction from the perspective of patent documents. Because of the complexity and diversity of construction operations and processes, understanding how different ICT in construction would allow implementing the right technology to the right place (Alsafouri and Ayer, 2018)(Lu et al., 2014).

Patents could provide technological knowledge (Li et al., 2012), especially in high-tech fields (Gredel et al., 2012). They are recognized as the most prominent and up-to-date technology source (Kim et al., 2016b), comprising scientific and technological information of machines, approaches, functions, processes, and solutions to problems (El-Ghandour and Al-Hussein, 2004). Up to 80% of all technical information can be found in patent documents (Nave, 2010). Therefore, a patent database that widely covers the inventions of ICT in construction is a valuable source for understanding relevant technologies, not only providing a dictionary for searching relevant technologies of ICT in construction, but also identifying problems to be solved by the state of art inventions, and all possible specific embodiments of relevant technologies (El-Ghandour and Al-Hussein, 2004). Such a corpus can help link the construction context with ICTs, assisting users in understanding the degree to which their requirements could be met by the current ICTs.

Another reason why this chapter uses patents as training data is due to the vague language embodied in patent documents. The purpose of this chapter is to propose a trained communication-oriented entity recognizer that could recognize and classify the entities in a given sentence. The proposed recognizer should have the desired performance, not only with higher accuracy in recognizing and classifying the entities but also with a general application to a wide range of text types that are written with different English levels. Therefore, patents are ideal training resources as the patents are always written in a vague manner and the content is difficult to understand (Bertram and Mandl, 2017), and the nature in patents are hard to model by NLP techniques.

All the patents of ICT in construction are intended to collect from USPTO, because (1) USPTO is the largest international patent grant office, and (2) USPTO is recognized as the most representative database to analyze the technological knowledge, providing patents that are well written and structured according to its requirement (Wang, 2018). However, retrieving patents of ICT in construction through traditional methods, such as search engines and patent retrieval techniques cannot ensure accurate and comprehensive results returned.

6.1.2 The Obstacle in Retrieving Patents of ICT in Construction

The difficulty for accurately obtaining a corpus of patents of ICT in construction poses a challenging task for non-ICT-professional users, although most of the patents provided by patenting authorities are easy to access. The challenge is mainly resulting from the lack of a specific searchable category for ICT in construction in the classification scheme of the patent offices. Table 6.1 provides the existing patent classes for the construction industry. It can be observed that two authorities provide specific categories that are relevant to the construction industry. However, these two classes, E and D25, focus on inventions about building materials and fixed construction rather than information and communication technologies.

Table 6.1 Existing classification schemes in the three major patent offices

Classification Scheme	Organizations	The specific classification of patents in construction
International Patent Classification (IPC)	World Intellectual Property Organization (WIPO)	E: Fixed Constructions E01. Construction of roads, railways, or bridges E02. Hydraulic engineering; foundations; soil-shifting E03. Water supply; sewerage E04. Building E05. Locks; keys; window or door fittings; safes E06. Doors, windows, shutters, or roller blinds, in general; ladders E21. Earth or rock drilling; mining E99. Subject matter not otherwise provided for in this section
Cooperative Patent Classification (CPC)	USPTO and European Patent Office (EPO)	None
United States Patent Classification (USPC)	USPTO	D25: Building units and construction elements 1. Structure 2. Prefabricated unit 3. Stair, ladder, scaffold, or similar support 4. Trellis or treillage unit 5. Architectural stock material

Because of the absence of a specific category for ICT in construction, searching engines become a possible way for users to find the required patents of ICT in construction. Moreover, a number of studies have developed advanced methods for patent retrieval, such as query reduction techniques (Bouadjenek et al., 2015; Mahdabi et al., 2011), expanding the queries by external dictionary and corpus (Tannebaum and Rauber, 2014), metadata-based methods (citations and classification) (Giachanou et al., 2015; Mahdabi and Crestani, 2014).

However, these query-based methods (including searching engines and patent retrieval tools) cannot return desired and corrected results in retrieving patents of ICT in construction. The main process of the query-based methods is to effectively search relevant patent documents in response to a given search request. This requires sufficient and specific input information to query desired patents, which highly relies on the knowledge of users. In addition, ICT in construction, standing for a set of information and communication technologies that are invented with major embodiments in the construction industry (Ahuja et al., 2009; Alsafouri and Ayer, 2018), incorporates a variety of features that cannot be described by specific queries. These methods tend to return a large volume of irrelevant patents and miss relevant patents, and cannot satisfy the information needs of the decision-makers. Table 6.2 shows two trials for retrieving collections of patents of ICT in construction from the USPTO website (USPTO, 2007) based on complex searching strategies. For each of the collections, 50 patents were randomly selected to manually check and compute the proportion of patents of ICT in construction in the collections. The low accuracy indicates that querying a searching engine could not return a collection that broadly and precisely covers patents of ICT in construction, even though fined and complicated strategies were applied. Moreover, most of the latent users in the construction practice are non-experts, who may not be able to perform such a searching task that is complex and time-consuming (Liu et al., 2011).

Table 6.2 Searching results by using the search engine in USPTO

Search strategies	Matched results	Accurac y

Strategy 1	**Search items:** CPC Classification Class and topic (matching input keywords within patent titles, abstracts, and descriptions) CPC Classification Class: ICT-related classes, including **H04** - *electric communication technique;* **G06** - *computing, calculating or counting;* **H01P** - *waveguides, resonators, lines, or other devices of the waveguide type;* **H01Q** - *antennas, i.e. radio aerials;* **G01S** - *radio direction-finding, radio navigation, determining distance or velocity by use of radio waves, locating or presence-detecting by use of the reflection or reradiation of radio waves or analogous arrangements using other waves;* **G08B** - *signalling or calling systems, order telegraphs or alarm systems;* **G08C** - *transmission systems for measured values, control or similar signals;* **G11B** - *information storage based on relative movement between record carrier and transducer.* Keywords: Construction Engineering and Management domain terms, including *construction project, project management, infrastructure project, civil engineering,* and *transportation project.* (Flyvbjerg, 2014; Greiman, 2013; Levitt, 2007; Mok et al., 2015; Zidane et al., 2013)	Collection 1: 5311 patents	7%
Strategy 2	**Search item:** Topic Keywords: ICT-related terms (*Radio frequency identification (RFID), 3D laser scanning, quick response, NFC, Augmented reality (AR), mobile computing, wireless connection (Wi-Fi) and robotics(drones)*) (Ahuja et al., 2009; Alsafouri and Ayer, 2018; Li et al., 2016) and Construction Engineering and Management domain terms	Collection 2: 922 patents	12%

6.1.3 The Purpose of the Chapter

Given the aforementioned constraints of existing patent retrieval tools, this chapter proposes an automatic approach that could automatically screen patents of ICT in construction. The present approach trains a classifier based on a neural network model that can automatically identify whether a piece of a patent is relevant to ICT in construction or not, and thus gathering a corpus that covers patents of ICT in construction. The resulting corpus consists of the primary source of training and testing data for the communication-orient entity recognition.

To date, a variety of approaches were developed and introduced to automatically classify the patents into existing classes (Chakrabarti et al., 1998; Smith, 2002; Venugopalan and Rai, 2015), but real-world cases were often overlooked (Li et al., 2018). The novelty of this study bases on the integration of NLP techniques and neural network model. On one side, NLP techniques provide a smart way to process unstructured textual data (Kurdi, 2017). Several studies have shown the importance of using NLP and machine learning techniques to convert information embodied in patent documents into knowledge (Agrawal and Henderson, 2002; Cassetta et al., 2017; Choi et al., 2012; Gwak and Sohn, 2018). This approach can save time and avoid personal bias in analysis processes (Bell et al., 2009), especially when a patent volume is large (Shekarpour et al., 2015; Silva et al., 2016). On the other hand, a neural network model, Multi-Layer Perceptron (MLP) was proposed to generate classification rules based on automatic learning from the input features and output class labels. In patent classification tasks, most studies used traditional machine learning methods (Li et al., 2012; Wu et al., 2010). The most prominent advantage of MLP, comparing with traditional machine learning methods, lies in its structure of neural network layers that have the capability to learn the non-linear relations between the inputs and outputs (Maaløe et al., 2016). This study also validates the model (several machine learning models were also validated for comparison) within different databases.

6.2 Proposed Approach Integrating NLP and MLP

To screen patents of ICT in construction from a patent collection, a binary classifier is developed to classify pieces of patents into two classes: relevant to ICT in construction or not. Figure 6-1 shows the overall procedure of the approach to achieve the learned classifier. The first step is to collect a database for training, incorporating the full texts of the instances annotated with target labels (the two classes). Then, NLP tools were used to process the textual data to achieve clean data. Based on the processed texts, N-gram and Tf-Idf algorithms are employed for the vectorization to represent each of the patents as a numerical vector that could be fed into the MLP that is trained by gradient descent and the hyperparameters were optimized. At last, a validation experiment is conducted by means of k fold cross-validation in two datasets.

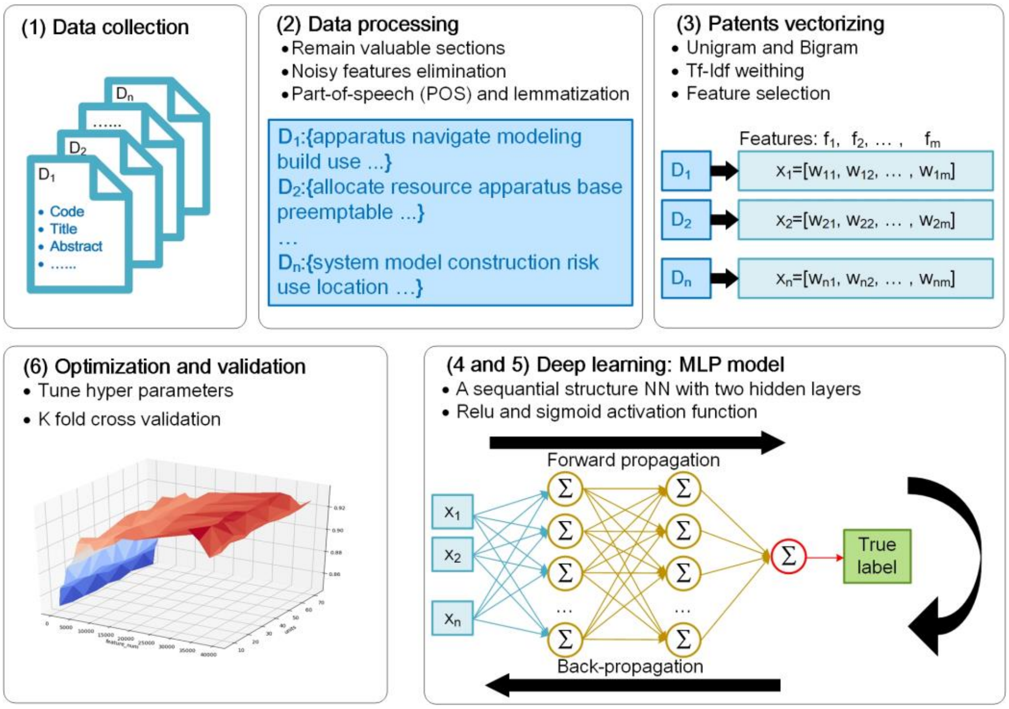

(1) Data collection
Dn
……
D2
D1
• Code
• Title
• Abstract
• ……
(2) Data processing
•Remain valuable sections
•Noisy features elimination
•Part-of-speech (POS) and lemmatization
D1:{apparatus navigate modeling build use ...}
D2:{allocate resource apparatus base preemptable ...}
…
Dn:{system model construction risk use location ...}
(3) Patents vectorizing
• Unigram and Bigram
• Tf-Idf weithing
• Feature selection
Features: f1, f2, … , fm
D1
x1=[w11, w12, … , w1m]
D2
x2=[w21, w22, … , w2m]
Dn
xn=[wn1, wn2, … , wnm]
(6) Optimization and validation
• Tune hyper parameters
• K fold cross validation
0.92
0.90
0.88
0.86
0
5000
10000
15000
20000
25000
30000
35000
40000
feature_num
10
20
30
40
50
60
70
units
(4 and 5) Deep learning: MLP model
• A sequantial structure NN with two hidden layers
• Relu and sigmoid activation function
Forward propagation
x1
x2
xn
Σ
…
True label
Back-propagation

Figure 6-1 Overall procedure of screening the patents of ICT in construction

6.2.1 Training Instances

The target of this step is to obtain labeled data as training instances for the MLP. The patents that are labeled with the class of relevant to ICT in construction (labeled as *ICTC)* or class of not relevant to ICT in construction (labeled as *non-ICTC)*. All the required patents were retrieved and crawled from USPTO (a self-developed Python program was used to perform crawling texts, see Appendix I for details), because (1) USPTO is the largest international patent grant office, and (2) USPTO is recognized as the most representative database to analyze the technological knowledge, providing patents that are well written and structured according to its requirement (Wang, 2018). The authors retrieved the patents on July 30, 2018. Totally, we have collected and annotated 348 patents as the dataset for further training and testing. The detailed processes are described in the following two paragraphs.

Figure 6-2 depicts the data collection process and annotation process, whereby patents in each of the two classes were collected and annotated respectively. As for the *ICTC class*, patents were gathered in the following steps: (1) By querying searching strategy 1 in Table 6.1 (ICT classes and CEM domain terms), 5311 patents were obtained in collection 1; (2) total 1500 patents were randomly selected from the 5311 patents. (3) Through the process of manually checking[1], 174 patents were obtained as *ICTC class* from the 1500 patents.

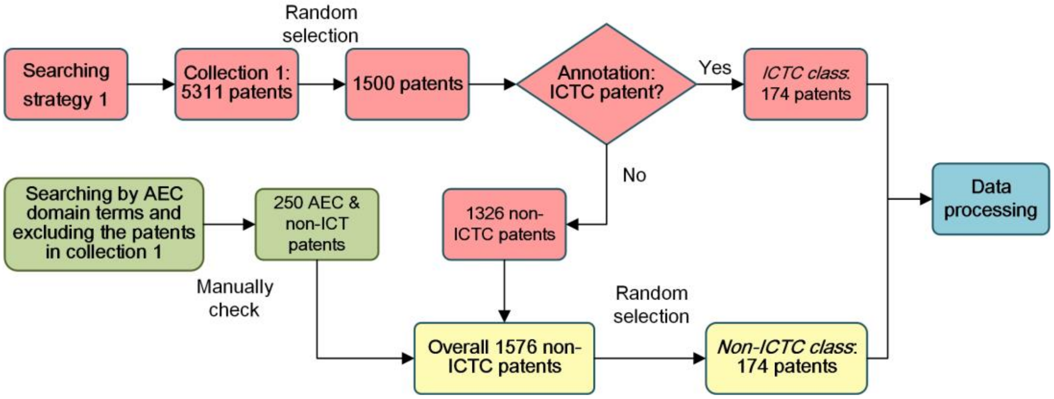

Searching strategy 1
Collection 1: 5311 patents
Random selection
1500 patents
Annotation: ICTC patent?
Yes
ICTC class: 174 patents
No
1326 non-ICTC patents
Data processing
Searching by AEC domain terms and excluding the patents in collection 1
250 AEC & non-ICT patents
Manually check
Overall 1576 non-ICTC patents
Random selection
Non-ICTC class: 174 patents

Figure 6-2 The process of obtaining the training instances for the two classes

As for the non-ICTC class, the patents were collected from two different sources. One was through the annotation process mentioned above, in which 1326 patents were identified as non-patents of ICT in construction. The other was obtained from another collection of patents that was retrieved by searching AEC domain terms and excluding patents in collection 1. This results in a combined collection of 1576 non-ICTC patents and 174 instances were randomly selected as training instances for the *non-ICTC class*. The complex collection process has two advantages: (1) The non-ICTC class contains not only

[1] The process of manually checking labels a patent as either *ICTC* or *non-ICTC*, performed by three Ph.D. students (their research directions are related to the AEC area) through in-depth reviewing of the title, abstract, claim and description. A patent can be labeled as *ICTC* class if the content expresses that the essence of the technology application is under the ICT scope and the AEC industry is a major embodiment in which the technology application can be implemented. To prevent mistakes as much as possible, two students annotated the patents independently, and the third student would make a judgment when the labels of a patent are inconsistent.

ICTs that are not developed for the AEC industry, but also the technologies of the AEC industry that are not relevant to ICT scope. This could prevent the data over-fitting and generate a more generalized model that is able to distinguish patents of ICTC class from patents of ICT, as well as from other type patents of the construction; (2) This study uses the negative sampling to make the two classes have the same size, because the balanced size for each class for training is proven as a key factor to achieve high accuracy (Brown and Mues, 2012; Zhao et al., 2015).

6.2.2 Data Processing by NLP Techniques

The raw text of each patent contains several sections (i.e., code, title, abstract, CPC classes, inventors and countries, and description). Among them, *title*, *abstract* and *claim* are frequently utilized and remained for further analysis in this study, because they were recognized as useful items providing basic technological information (Niemann et al., 2017; Venugopalan and Rai, 2015). *Title* and *abstract* convey the essence about the technology, which are always written in a restricted pattern within short content(Lee et al., 2013). In addition, the *claim* defines the protection right of the invention in professional expressions, always providing articulated expressions about the technical boundaries and specifications (Niemann et al., 2017).

The selected text of patents is raw data, which is pre-processed by NLP techniques for further analysis. Without pre-processing, the texts would contain a lot of noisy features (in a typical case, the number of features can be close to the number of words in the dictionary of the training instances), and thus creating higher-dimension vectors. Using NLP techniques can make the computer-aided program to learn the semantic meaning from the raw text give feedback based on pre-defined rules (Jain et al., 2018). To process the selected raw text, this study employs three NLP techniques (Figure 6-3 plots the pre-processing procedure using these techniques): (1) Tokenization. For each raw sentence in the texts, tokenization is utilized to split the sentence into words. In addition, all the words are converted to lowercase and punctuations are removed. Through this step, all the raw sentences would be replaced with sequenced and lowercase words. (2) POS tagging. In this step, each word is tagged with POS tag that indicates its syntactic role (i.e., noun, adverb) according to the surrounding context. POS tagging plays a central role in text processing, which could increase the accuracy for lemmatization and stemming (Habash et al., 2009). (3) Lemmatization and stop-words removal. The purpose of this step is to correctly match the words with different forms, such as plural forms for nouns and presenting and past forms for verbs. Lemmatization transforms the different forms into the stem forms (root words). However, lemmatization may generate a number of mistakes without POS tagging. For example, "modeling" could be a present participle of a verb (with lemma "model") or a noun (with lemma "modeling") according to the context, and the lemma of noun "modeling" would be wrongly identified as "model" without POS information (Vlachidis and Tudhope, 2016). This study utilizes NLTK toolkits to perform POS tagging and lemmatization (Bird and Loper, 2004). Moreover, stop-words (i.e., a, an, of, one, two, three, etc.) are removed, because they are non-descriptive and do not convey any semantic meanings.

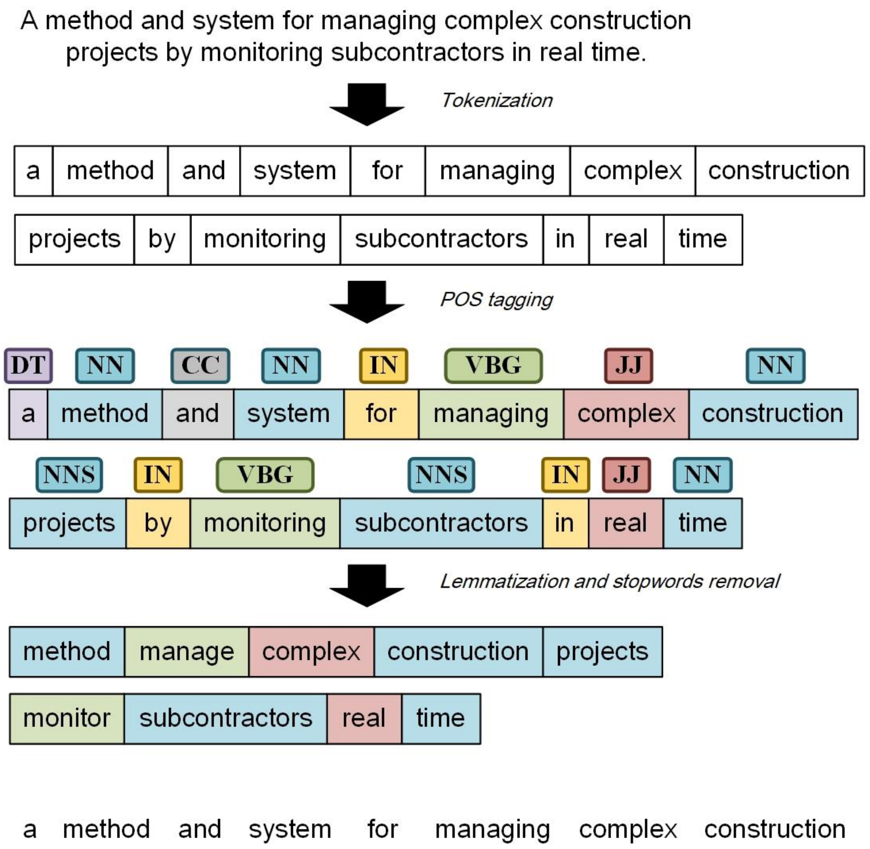


Figure 6-3 The proposed processing procedure for textual data in patent documents

6.2.3 Vectorizing Patent Documents

The processed patent documents have to be converted into numerical vectors that can be fed into MLP. With regard to the vectorization, a number of algorithms have been developed to convert the textual data into vectors. Bag-of-words (BOW), topic models and subject–action–object (SAO) have been used in recent patent classification studies (Li et al., 2018; Venugopalan and Rai, 2015). Traditional BOW models typically construct the feature space vectors in which each position is occupied by a term or a phrase (Forman, 2002). Its measurements include n-grams, bi-grams, and word frequency to identify phrases from the texts (Onan et al., 2016), depending on how the phrases were counted. Although BOW models are simple and may generate a large number of features, they remain the most effective feature selection method (Mirończuk and Protasiewicz, 2018; Onan et al., 2016). Topic model and subject–action–object (SAO) were mainly developed to solve the high dimension problem, replacing the BOW features by latent topics (Kaplan and Vakili, 2013) or SAO structures (Gerken, 2012).

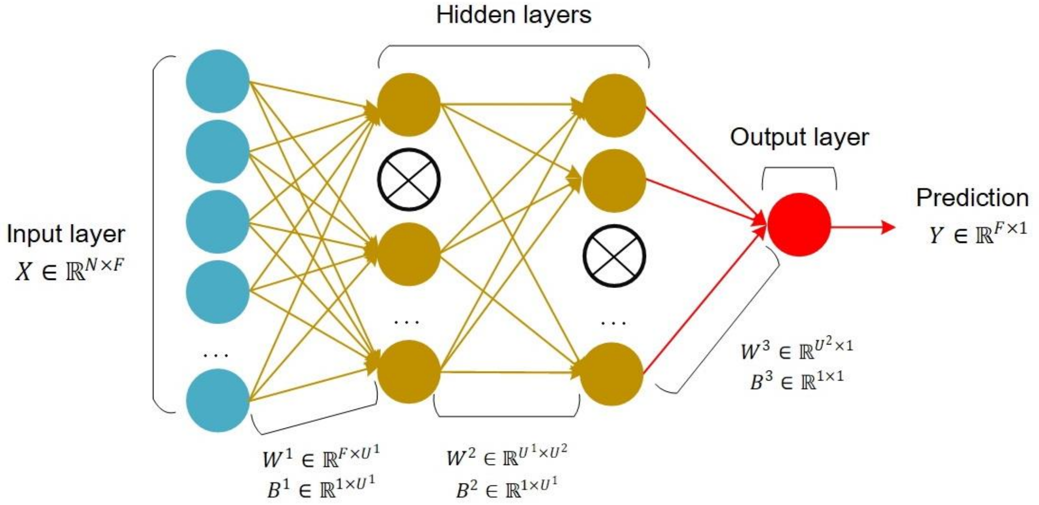
This study adopts the N-gram model with Tf-Idf weighting algorithm to vectorize the patent. N-gram considers the N words in a sequence as a feature, which has been proposed in the 1940s (Shannon, 1948) and has been employed in a large and growing body of literature (Bengio et al., 2003; Benson and Magee, 2013). In this case, two typical N-gram models, N = 1 (unigram) and 2 (bigram) are used to extract unigrams and bigrams as from the patent documents, constituting of the vocabulary with size v (overall v unigrams and bigrams are identified from the patent texts). A vector with v-dimension in which each position is the Tf-Idf (term frequency & inverse document frequency, see Sparck Jones (1972)). Another necessary step is to filter useful features because many of the features do not contribute to the training and prediction. This study, according to the Tf-Idf vectors, uses ANOVA F-value to select top features (number = K). In this study, K is set as a hyperparameter that would be tuned in the optimization step.

This study adopts N-gram and Tf-Idf as the vectorizing approach but not the topic models or SAO structures, because (1) BOW and Tf-Idf have been widely used in NLP studies and have been proven as the prominent vectorizing algorithm due to the simplicity and effectiveness (García Adeva et al., 2014; Mirończuk and Protasiewicz, 2018; Pavlinek and Podgorelec, 2017); (2) Topic models and SAO structures are suitable in clustering or classification tasks that have more than two classes to be distinguished (Blei et al., 2003; Choi et al., 2012); (3) Topic models and SAO structures replace the N-grams with latent topics or subject-active-objective structures. This would generate vectors with much lower dimensions which is not necessary in this case, because the proposed MLP model may get better performance when the number of input features is large (Cakir and Yilmaz, 2014).

6.2.4 The Structure of Proposed MLP

To improve the prediction precision and take the non-linear factors into consideration, this study proposes a neural network model - MLP - to learn and train the complex relations between inputs and outputs. MLP is always used with a feed-forward-based architecture and back-propagation learning process (Goodfellow et al., 2016). There is a number of neurons in the MLP, and each of them receives signals from the former layer and passes a transformed single by an activation function to the subsequent layer (Riedmiller, 1994). Although it is a general wisdom that neural network models are better than machine learning models, neural network design and hyperparameters choice are more important than the neural network model itself (Levy et al., 2015). This section describes the proposed MLP architecture.

After vectorizing, the input matrix in this study is $X \in \mathbb{R}^{N \times F}$, in which N and represent the number of instances and features respectively. Features are set as columns, and thus each patent is reflected as a row vector $x_i \in \mathbb{R}^{1\times F}$. The output is a column vector $Y \in \mathbb{R}^{N\times 1}$. The main target of the proposed MLP model is to learn a deep and complex neural network structure that could predict from X to Y. Figure 6-4 illustrates the architecture of the proposed MLP consisting of four layers: one input layer, two hidden layers and a prediction layer, labeled from layer 0 to layer 3. The weigh matrices connect the layers in sequence, and the neurons in the hidden and prediction layers are processing units, embodied with activation functions to transform the input to outputs. The number of the neurons in the input and output layers are set as N and 1, which are subject to the dimensions of the input data and prediction. The number of neurons in the hidden layers is set as a hyperparameter.


Hidden layers
Output layer
Prediction
$Y \in \mathbb{R}^{F\times 1}$
Input layer
$X \in \mathbb{R}^{N\times F}$
...
...
...
$W^3 \in \mathbb{R}^{U^2\times 1}$
$B^3 \in \mathbb{R}^{1\times 1}$
$W^1 \in \mathbb{R}^{F\times U^1}$
$B^1 \in \mathbb{R}^{1\times U^1}$
$W^2 \in \mathbb{R}^{U^1\times U^2}$
$B^2 \in \mathbb{R}^{1\times U^1}$

Notes: (1) The features are set as columns in the input matrix, therefore each neuron in the input layer is a F (the number of features) dimension vector, representing a patent.(2)⊗ denotes the dropouts of the hidden layers in the training.

Figure 6-4 Neural network architecture of proposed MLP

The entire MLP predicts the outputs based on the connection weights and the activation functions. In specific, the j-th neuron in l-th layer transforms an output based on the following equations:

$$\begin{cases} h_i^l = f^l\left(\sum_{i=1}^{U^{l-1}} h_i^{l-1} w_{ij}^l + b^l\right), l = 1,2 \\ h^l = h^3 = f^3\left(\sum_{i=1}^{U^2}(h_i^2 w_i^3 + b_i^3)\right), l = 3 \end{cases} \tag{6.1}$$

where l represents the layer sequence, U^{l-1} indicates the number of neurons in the $(l-1)$-th layer, x_i^{l-1} denotes the output of i-th neuron it receives, w_{ij} is the weight connecting x_i^{l-1} and j-th neuron in l-th layer, and b is the bias function for this neuron. f^l is the activation function in l-th layer. In this case, the two hidden layers (layer 1 and layer 2) and the output layer (layer 3) use *Rectified Linear Unit* (ReLU) and *Sigmoid* functions as the activation functions respectively.

With the back-propagation process, the neurons in hidden and output layers can be trained with unique weight matrix and bias, producing different outputs according to the distinct tasks (Garcia-Laencina et al., 2013). Moreover, to prevent the overfitting, dropout functions were adopted in the hidden layers. In Srivastava et al. (2014), it is shown that dropout can improve the performance by preventing the over-fitting over the text classification tasks.

6.2.5 Parameters Training by Gradient and Dropout

As mentioned above, the main task of MLP is to make the neurons to be learned, which could predict Y from X. The learning process is achieved by certain iterations, each of which is a loop consisting of a feed-forward and a back-propagation process (Haykin, 1999; Riedmiller, 1994). In the feed-forward process, the weights and bias in the hidden and output layers are randomly generated and propelled forward, calculating the output value h^3 from input X. Since a sigmoid function is selected as the activation function in the output layer, the errors follow a logistic distribution between the predictions (with values between 0 and 1) and true labels (with values are only 0 or 1). The loss function is following:

$$J = -\sum_{n=1}^{N} y_n \log(h_n^3) + (1 - y_n) \log(1 - h_n^3) \tag{6.2}$$

In the back-propagation, the parameters θ (including all the weights and bias in hidden and output layers) would be updated by stochastic gradient descent. Two types of signals constitute the gradients: (1) global signals that can be computed from the derivatives, which transform the errors from the loss function; (2) local signals that are the inputs from the former layer. The θ would be updated from back to front, as the gradients are computed from the loss value to the former layers, one by one. For the clarity of the back-propagation process, this study illustrates the updating process of w_{ij}^2 and w_j^3 in layer 2 and 3. Figure 6-5 shows the functions of the neurons in layer 2 and 3. The gradient of w_j^3 (∇w_j^3) is defined as the derivative from J to w_j^3, which could be computed by the chain rule of derivatives:

$$\nabla w_j^3 = \frac{dJ}{dw_j^3} = \left(\frac{dJ}{dh^3} \times \frac{dh^3}{dN^3}\right) \times \frac{dN^3}{dw_j^3} = f'(N^3) \times h_j^2 \tag{6.3}$$

$$w_j^3 new = f(w_j^3 old, \nabla w_j^3) \tag{6.4}$$

where the $f'(N^3)$ is the global signal that could be computed by the derivative with loss value, h_j^2 is the local signal (the output of the j-th neuron in layer 2), and a is the learning rate that is pre-defined.

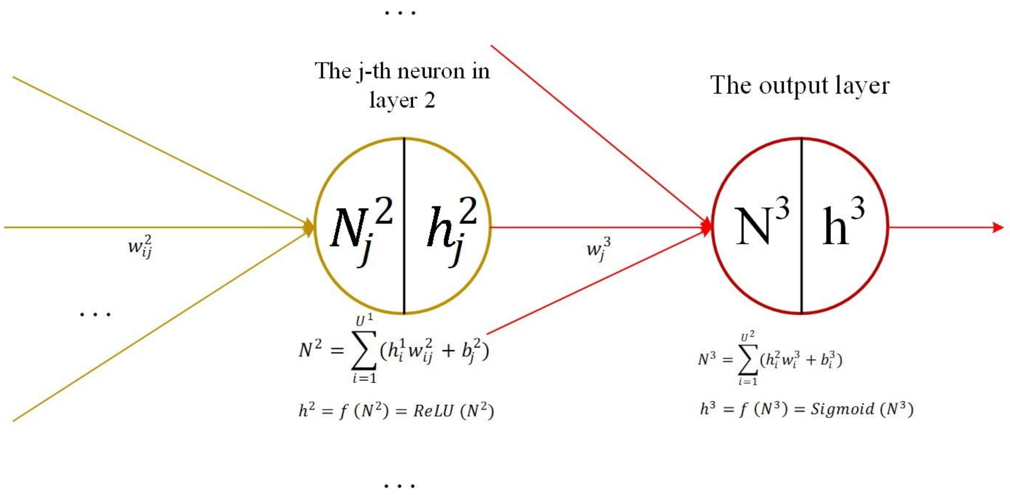

. . .
The j-th neuron in layer 2
The output layer
w_{ij}^2
N_j^2
h_j^2
w_j^3
N^3
h^3
. . .
$N^2 = \sum_{i=1}^{U^1}(h_i^1 w_{ij}^2 + b_j^2)$
$h^2 = f\ (N^2) = ReLU\ (N^2)$
$N^3 = \sum_{i=1}^{U^2}(h_i^2 w_i^3 + b_i^3)$
$h^3 = f\ (N^3) = Sigmoid\ (N^3)$
. . .

Figure 6-5 The neurons with input and activation functions in the last two layers

Similar to layer 3, ∇w_{ij}^2 could be computed by the following:

$$\nabla w_{ij}^2 = \frac{dJ}{dw_j^3} = \frac{dJ}{dh^2} \times \frac{dh^2}{dN^2} \times \frac{dN^2}{dw_{ij}^2} = f'(N^2)w_j^3 f'(N^3) \times h_i^1 \tag{6.5}$$

$$w_{ij}^2 new = f(w_{ij}^2 old, \nabla w_{ij}^2) \tag{6.6}$$

where $f'(N^2)w_j^3 f'(N^3)$ is the global signal that is propagated from the loss value, and h_i^1 is the local signal. The computations of other parameters, such as w and b are similar to equation (6.3) and (6.5). According to the gradients, the parameters could be updated by optimization algorithms (equation (6.4) and (6.6)). Typical algorithms include Stochastic Gradient descent (Robbins and Monro, 1985), AdaGrad (Duchi et al., 2011), RMSProp (Tieleman and Hinton, 2012), and Adam (Kingma and Ba, 2014). This study applies the

Adam algorithm as the optimizer for gradients, as it has been recognized as the most effective in most cases with less computation time.

The applied MPL model includes dropout operation (Figure 6-4). “Dropout” refers to temporarily eliminating some neurons and their incoming and outgoing connections in the neural network. The dropped neurons are selected randomly based on a pre-defined ratio a (a=0.2 in this case). In the back-propagation of a training loop, a new thinned neural network is achieved with the proportion of (1 - a) neurons remained. The parameters updating process would be implemented within the thinned neural network. In the feed-forward process of the subsequent loop, the removed neurons would turn on, which parameters are obtained from the remaining neurons by a scale of 1/a. Therefore, training MLP with dropout can be regarded as training a larger number of thinned NNs which share the same parameters. Such a training fashion effectively prevents neurons from co-adapting and thus preventing overfitting issues (Al Rahhal et al., 2018). As for the details of dropout, please see Al Rahhal et al. (2018).

After updating θ, a loop with a feed-forward and back-propagation finishes, which would be iterated in training. In this case, the maximum of epochs is set as 1000, and the consecutive tries of loss value without decrease is set two. The training process would iterate the loops until any of the above stop conditions is met.

6.2.6 Optimization and Validation

The purpose of hyperparameters tuning is to achieve an MLP model with the best performance, thought tuning the hyperparameters. As mentioned above, the number of features and neurons of the two hidden layers are set as hyperparameters. This study validates the proposed MLP model based on two different databases. On one hand, validation is carried out along with the training process to test the performance over the collected data that were retrieved from USPTO. On the other hand, the universality of MLP is tested through randomly selected samples from patents that were retrieved by another database. The k-fold cross-validation is adopted to test the performance of the model, in which k is the number of folds for splitting the training and testing data. In the training process, all the dumping data would be randomly split into k folds with the same size, and one of them is set as a test instance and others are used for training. Such a training process is performed in k times, each of which has a different fold for testing and a different composition of k-1 folds for training, resulting in a validation value. The final validation value is the average value of k validation values. In this way, K-fold cross-validation prevents the bias in data selection and ensures the measures of the performances with objections (Friedman et al., 2001).

6.2.6.1 Optimization

The process of hyperparameters tuning aims to achieve the MLP model with the best performance through selecting the number of features (F) and units (U). The range of the features is from 1000 to 40000, with step of 1000 and 2500 for F in (1000, 10000) and (10000, 40000) respectively. With regard to the number of units, this study adopts the measurement proposed by Fan et al. (2015b), which proposed a range around $\sqrt{N+1}$ (N denotes the number of neurons in input layer). The resulting range of number of units is from 5 to 69 and the step is set as 8. Figure 6-6 shows the optimization process. The proposed MLP model reaches the highest precision (0.959) when F is 30000 and U is 13.

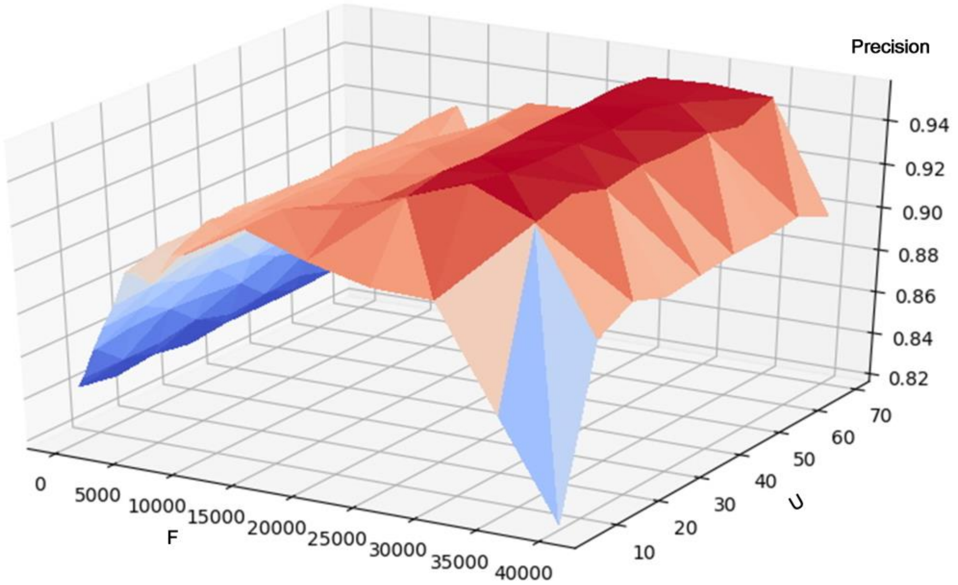


Figure 6-6 Hyperparameters tuning process

6.2.6.2 Validation

In this evaluation, the goal is to verify if the proposed MLP has better screening accuracy than the traditional machine learning models. The performance of the proposed MLP is compared against existing machine learning models, including Gaussian Naive Bayes (GNB), SVM and Bernoulli Naive Bayes (BNB). This study validates the performance through precision, recall, and F-score. Table 6.3 shows the confusion matrix for the evaluation of the precision, where "TP" denotes the number of true positive classifications, "FP" denotes the number of false positive classifications, "TN" denotes the true negative classifications, and "FN" denotes the false negative classification. Normally, scholars prefer the precision and recall to evaluate accuracy. The recall is defined as: Recall = $\frac{TP}{TP+FN}$, which is the proportion of the true positives against the sum of true positives and false negatives. Precision is the value that the true positives divide the true positives and false positives: Precision = $\frac{TP}{TP+FP}$.

The value of F-measure is defined as: F-measure = $\frac{2\times Precision\times Recall}{Precision + Recall}$, and F-measure should be in the range between 0 and 1.

Table 6.3 Confusion matrix for TP, FP, FN, and TN for the validation of MLP

		Predicted outputs	
		C_i	Not C_i
Paired classes of the patents	C_i	TP	FN
	Not C_i	FP	TN

Figure 6-7 shows the accuracy of the model over the different feature numbers. The highest precision values for each model are marked above the lines. By examining the figure, we can verify that the MLP model is superior to those machine learning models over all the features except K = 1000 and K = 40000. From Figure 6-7 we can also observe that the MLP model is more sensitive to the number features, with the highest standard deviation value (0.032) over the models. This is consistent with one of the major differences between deep NN model and traditional machine learning models: the traditional machine learning models are not capable of adjusting the model complexity according to the inputs, whereas the deep NN could tune the structure (number of layers and neurons) that is most suitable for input features (Moraes et al., 2013).

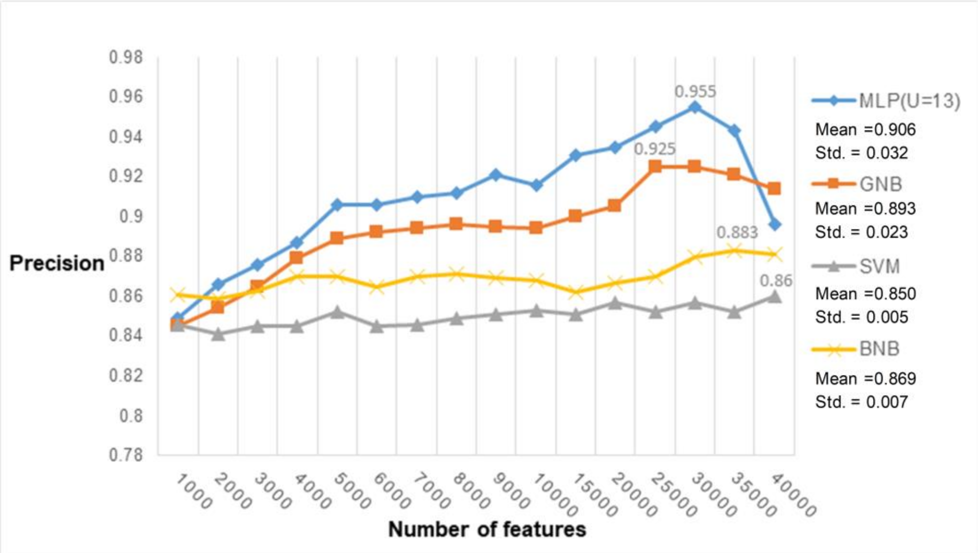


Figure 6-7 The precision values for MLP and machine learning models over the features

Table 6.4 illustrates the cross-validation results over the optimized MLP (K= 30000, U = 13) and the machine learning models that have the highest accuracy over the feature numbers. As was mentioned above, 5 folds cross-validation is used to measure the performance of the trained model, in which 80% of the annotated data is randomly selected as training data, and the rest 20% is used to test the performance. It can be observed that MLP (K= 30000, U = 13) has the best performance overall the three indexes (precision, recall and F1 score). In this study, precision and recall are equally important to evaluate the MLP. On the one hand, this chapter aims to compile a database of patents of ICT in construction with fewer irrelevant instances, which can be evaluated by precision. On the other hand, the classifier is expected to screen patents that are relevant to ICT in construction as many as possible, which can be evaluated by recall. Therefore, all the measurements are used to evaluate the models from different perspectives, and F-score is used as a mediate validation.

Table 6.4 Cross-validation results over MLP, GNB, SVM, and BNB in the initial dataset

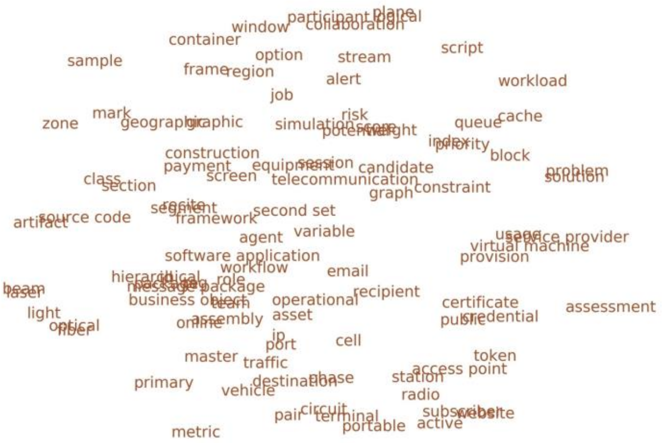
	Precision	Recall	F1 score
MLP (K=30000,U=13)	0.955	0.954	0.954
GNB (K=25000)	0.925	0.919	0.918
SVM (K=40000)	0.86	0.852	0.848
BNB (K=35000)	0.883	0.86	0.853

6.3 The Resulting Patent Database of ICT in Construction

Although the proposed screening approach is validated, some important implications should be further discussed. The authors use the proposed approach to automatically screen patents of ICT in construction from collection 1, resulting in a collection of 392 patents of ICT in construction. To compare the topic distribution of the patents in the corpus, as well as the patents in collection 1 and 2 (Table 6.2), this study plots the figures of feature space for each of the collections (Figure 6-8). The t-Distributed Stochastic Neighbor Embedding (TSNE) algorithm (Czerniawski et al., 2018; Maaten and Hinton, 2008) is adopted to project the high dimensional feature vectors into 2D plot, in which the physical distance between two features roughly represents the degree of association of them in the corresponding collection.

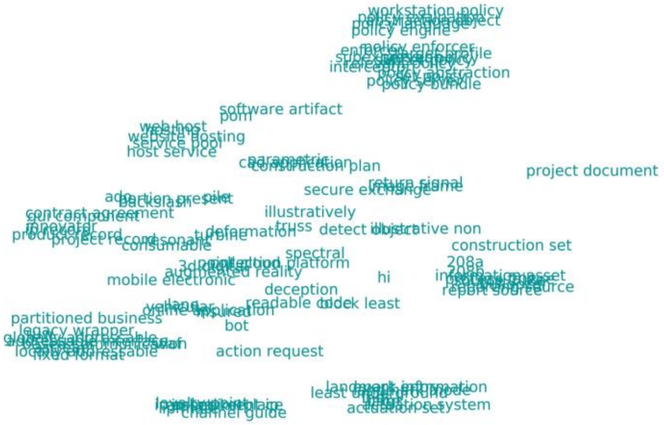
(a)

No.	Term	No.	Term
1	Asset	16	Score
2	Virtual machine	17	Class
3	Package	18	Light
4	Recite	19	Optical
5	Station	20	Terminal
6	Workflow	21	Circuit
7	Agent	22	Block
8	Risk	23	Participant
9	Session	24	Stream
10	Logical	25	Subscriber
11	Metric	26	Phase
12	Construction	27	Business object
13	Vehicle	28	Role
14	Tag	29	Certificate
15	Segment	30	Section



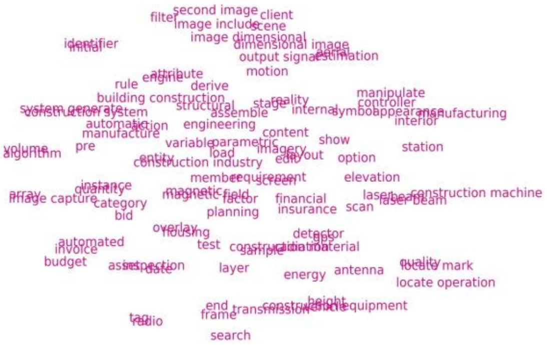
(b)

No.	Term	No.	Term
1	Enforcer	16	Embodiment IP
2	Policy enforcer	17	Process failure
3	IP asset	18	IP mate
4	IP marketplace	19	Lane
5	Policy abstraction	20	Return signal
6	Reporting source	21	Policy engine
7	Report source	22	Hosting
8	Policy server	23	Web host
9	WAF	24	Parametric
10	Event entry	25	Secure exchange
11	Addressable interface	26	Actuation system
12	LUW	27	Actuation set
13	Block least	28	Software artifact
14	Information asset	29	Point cloud
15	GUI component	30	Policy language



(c)

No.	Term	No.	Term
1	laser	16	radio
2	antenna	17	reality
3	parametric	18	bid
4	asset	19	quality
5	construction machine	20	scan
6	structural	21	stage
7	aerial	22	beam
8	vehicle	23	layer
9	layout	24	inspection
10	construction equipmen	25	search
11	construction material	26	end
12	magnetic	27	action
13	controller	28	rule
14	symbol	29	radiation
15	tag	30	transmission



Notes: (2) In each sub-figure, top 100 features with highest average Tf-Idf value are plotted, and top 30 are list at left for clarity. (2) As for strategy 1 and 2, please see Table 6.2 for details.

Figure 6-8 TSNE plots of feature spaces

As explained in the introduction, the searching engines for patents has two major flaws: (1) the searching engines can only perform "match" logic based on structured data; and (2) searching by keywords cannot avoid personal preference, and thus the results highly depend on users' knowledge. Figure 6-8 (a) depicts the feature space of collection 1, in which patents were searched by ICT classes and Construction Engineering and Management domain keywords. The features in this figure are averagely distributed, incorporating a large number of ICT-related features, but some typical terms that are relevant to the scope of ICT in construction do not appear. Such feature distribution indicates that the patents in this collection are mainly relevant to ICT, but not ICT in construction. A possible explanation for this might be that the Construction Engineering and Management domain keywords are not capable of discerning patents of ICT in construction from ICT patents using query-based methods. For example, the keyword "construction project" may match patents related to construction projects, but it can also match patents of "software project" containing sayings about "construction project" which means to build up a project.

Despite the mismatching problem, strategy 2 can lead to short coverage of ICT techniques. As Figure 6-8 (b) shown, the features are agglomerated into clusters, indicating an unbalanced distribution of topics. The features, not surprised, are mainly related to the searching keywords, such as wireless and mobile. The proposed method is advanced than traditional methods. The features in Figure 6-8 (C) are distributed averagely, incorporating a wide range of terminologies related to ICT in construction, such as CAD, 3D, BIM, and RFID.

6.4 Data Annotation

The aim of data annotation in NLP tasks is to manually label the language with the corresponding tags. With respect to CER task, the data annotation aims to label each of the tokens as the corresponding CE type for every single sentence of patents of ICT in construction. Basically, the annotation process is to manually label the expressions of CE as corresponding class tags through a web-based tool Doccano[2]. This study, based on the resulting patent database of ICT in construction, randomly selects 180 patents to for further annotation. The texts to be labeled incorporate titles, abstracts, and the first claim. The claim section of a patent always contains several points to define the protection right of the invention in professional expressions, and the first one always describes the overall technical boundaries and specifications for the proposed invention (Niemann et al., 2017).

This study follows the annotation specifications of IOB (Ramshaw and Marcus, 1999). IOB consists of a tag set [I,O,B], in which B denotes a start of an entity tag, “I” denotes the inside of an entity tag, and “O” denotes outside of any entities. Given the three types of CEs, namely transferred information (TI), communication models (CM), and communication subjects (CS), there are totally seven CE tags: 'B-TI', 'I-TI', 'B-CM', 'I-CM', 'B-CS', 'I-CS' and 'O'. For example, a word is labeled as: 'B-TI' if it is the first word of entity of TI; 'I-TI' if it is the inside but not the first word of entity of TI .

[2] https://github.com/chakki-works/doccano

This study includes here different annotation rules to decide whether a word or phrase is a type of CE of ICT in construction. Specifically, these rules are as follows:

(1) The annotation task was carried out exclusively for CEs of ICT in construction. This implies that other types of expression were not labeled, such as the construction embodiments and the basic techniques.

(2) The labels are tagged based on contextual information in a given sentence, rather than depending on a vocabulary of pre-defined CEs. In the field of ICT in construction, the technical documents incorporate a wide range of mentions to describe the communication patterns in varied embodiments. One of the objectives of the deep learning model is to predict unknown CE based on the contextual information. Therefore, the annotation should be implemented not based on the prior knowledge of specific mention but on the contextual meaning of the whole sentence. This is mainly due to the contextual information is the first priority to recognize CE in the context of ICT in construction. Traditionally, entity recognition tasks were always performed for person, organization, chemical entities. These entities could be recognized by the word-level features that are more often used exclusively, with specialized meanings when they appear. However, the CEs in the context of ICT in construction comprise plenty of expressions which are always combined with common words, such as “receiver antenna”, “position information”, “cost feedback”, et al. In this way, recognizing CEs highly depends on the contextual information. In addition, word-level meanings do not independently determine the type of CE. For example, one could easily indicate *building data* is a TI in a sentence “An apparatus for editing three-dimensional (3D) building data”, but this study does not label that because this sentence

alone does not express building data is used for communication. However, *building data* would be labeled as TI in the sentence "accessing a Building Information Modeling (BIM) database to obtain building data for a building component", as the sentence independently express that the *building data* is something that can be obtained from a database, indicating the usage for communication. Moreover, all the sentences are labeled independently, and the identified CE could not be used as prior knowledge in the next sentence.

(3) The information, such as the annotated CEs and words, would not be delivered across sentences. For example, two sentences have mentions of "aerial images" in the same patent named "Concurrent display systems and methods for aerial roof estimation":

- "displaying a plurality of aerial images of a roof at the same time on a single display, each of the aerial images …"
- "the respective line drawings overlying a first and a second aerial image of the plurality of aerial images of the roof"

The first sentence is in front of the second in the same patent document. In the first sentence, the second "aerial images" was annotated as a CE of transformed information, because the first appearance indicates that it is a type of data to be displayed and this information could be copied backward. However, in the second sentence, the mention of "aerial images" was not annotated, even though the former sentence has identified "aerial images" as a CE. This rule is due to the basic task of CER. Similar to sequence labeling and ER tasks, CER is an NLP task focusing on the sentence level.

(4) Common words about entities were not labeled, such as data, information, model, etc. because these general expressions do not provide specific information. For example, in the sentence “obtaining information associated with a project, said project comprising a plurality of tasks”, the information would not be labeled as TI due to it is a common word.

(5) The annotation task in this study does not highly depend on the background knowledge of the annotators. This is mainly due to the annotate process in which the contextual mentions play a central role in determining CE. Despite the less requirement of the annotators, three Ph.D. students worked as annotators.

After the annotation, 414 sentences were annotated with CE tags in a total of 171 patent documents. This study uses negative sampling for selecting training instances, by randomly extract the same size of sentences in the non-tagged contents in the annotated 171 patent documents. The resulting collection for training comprising 824 sentences. The detailed information about the training collection can be seen in Table 6.5. On average, each patent of ICT in construction has 2.36 sentences in which at least one word is tagged as a CE class. Each annotated sentence averagely contains 10.66 CE labels, and 5.16 appearances of 2.5 CEs. The most frequently appearing CE class is TI, because TI could be mentioned independently. For example, in the sentence “The present invention relates to an environmental load assessment system, which is capable of efficiently and simply assessing an environmental load of a building in all stages”, the phrase *environmental load* was articulated as a TI due to its ability to be accessed, and one could determine it even there is no other class CEs in the sentence.

Table 6.5 Descriptive statistics of different items of the training instances

ID	Item	Number	Per annotated sentence	Percent of all CEs
1	Total sentences	824	/	/
2	Annotated sentences	412	/	/
3	Total words	63765	/	/
4	Total labels	4392	10.66	/
5	Total occurrence of CE	2191	5.16	/
6	Total number of CE	1028	2.50	/
7	Occurrence of TI	1043	2.53	49.11%
8	Occurrence of CM	822	2.00	38.70%
9	Occurrence of CS	259	0.63	12.19%
10	Number of TI	571	1.39	55.54%
11	Number of CM	375	0.91	36.48%
12	Number of CS	82	0.20	7.98%

For each CE class, Figure 6-9 indicates the top 10 CEs that frequents occur in the annotated sentences. Most of the frequently appeared CEs are common technical concepts related to information technology, such as image, position, laser beams, sensor, display device, etc.

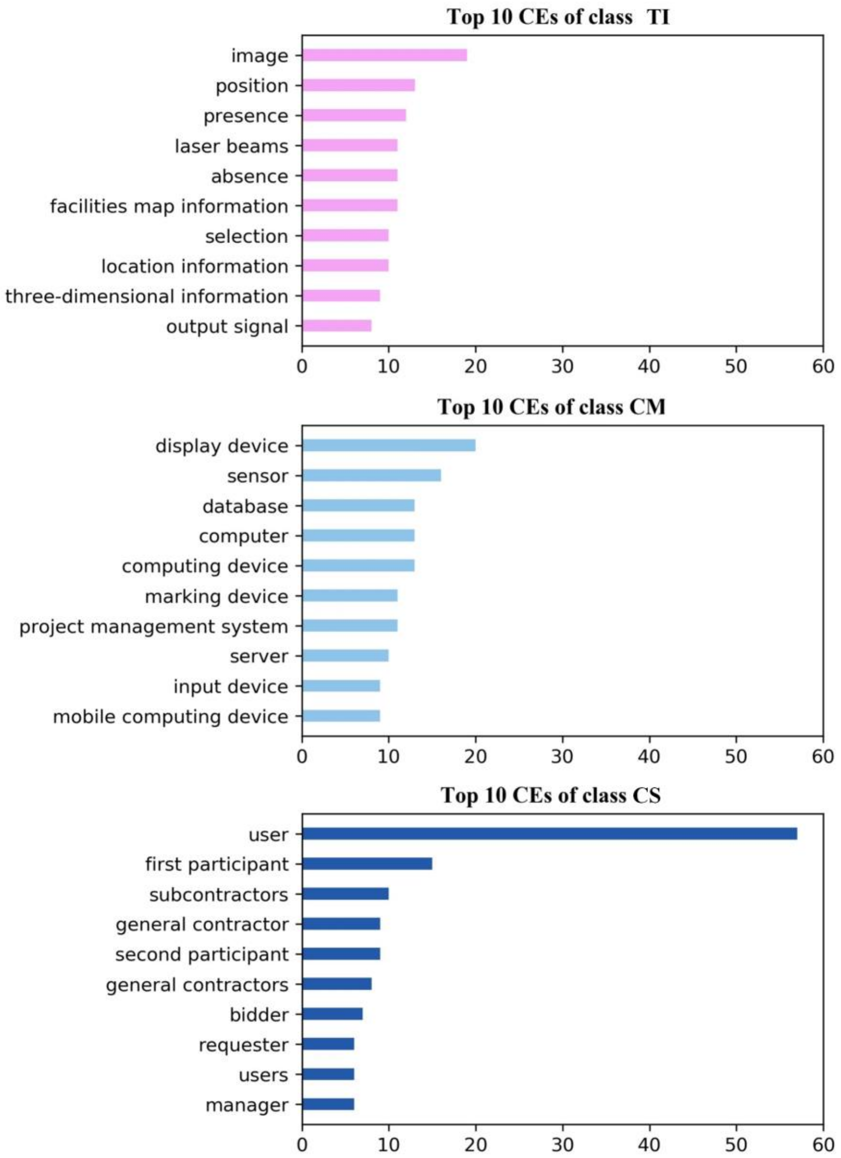


Figure 6-9 The most frequent CEs

6.5 Summary of this Chapter

This chapter compiles a patent database of ICT in construction by proposing a classifier integrated NLP and MLP. The proposed approach could automatically screen patents that are relevant to ICT in construction from a collection of patents with high accuracy. Such a screening task could not be performed by current patent retrieval tools. Because these methods are query-based, advanced and complicated domains like ICT in construction could not be easily described by specific queries. This chapter contributes an alternative way in which a classifier is trained in a supervised fashion and could screen a more complete and accurate collection of patents. The validation results indicate that the proposed approach outperforms within different databases (WoS and USPTO), which enables users, even without specific expertise, to obtain patents of ICT in construction accurately from various databases. In addition, this study has made the following contributions: (1) a deep learning model with two layers of neurons is trained to learn the non-linear relations between the input features and output classes, providing better performance than traditional ML models. (2) This study uses NLP-based techniques to automatically extract features from the textual data of the related patent documents. This study also adds valuable implications to ICT and automation in engineering, as well as current studies of patent classification.

Chapter 7 Training and Validation of the Deep Learning Model for CER of ICT in Construction

7.1 Introduction

In this chapter, this study presents the training process of the deep learning model, as well as the validation results and the implications. Firstly, pre-training techniques are described. Secondly, based on the predicted labels on testing instances, this study validates the deep learning model. Moreover, to make a necessary comparison, another RNN-based deep learning model, Bi-LSTM-CNN (BLC) was trained and validated with the same training and testing instances. This study utilizes the k-fold cross-validation to validate the two models. Finally, the implications of the validation results were discussed.

7.2 Training Techniques

7.2.1 Pre-trained Parameters

Pre-training technique was developed in recent years, and has increasingly been investigated in several studies (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018). The pre-training technique has been proved as an effective approach to improve the performance of NLP tasks. The main idea is to pre-design a training process in which RNN-based models such as LSTM is used to perform the language task by using a large corpus of materials. As Figure 7-1 shown, the language task predicts the next words based on the surrounding words. The language task follows an unsupervised manner, in which the data is not required to be labeled because the true labels are the input sentence itself in the next position. In this way, a large corpus of data could be fed into the pre-training model, and the aim is to acquire the contextual information embodied in the parameters, both in the LSTM and word embedding layer. The downstream task has to train their model depending on the same LSTM and word embedding layers, with an alternative output layer according to the NLP tasks.

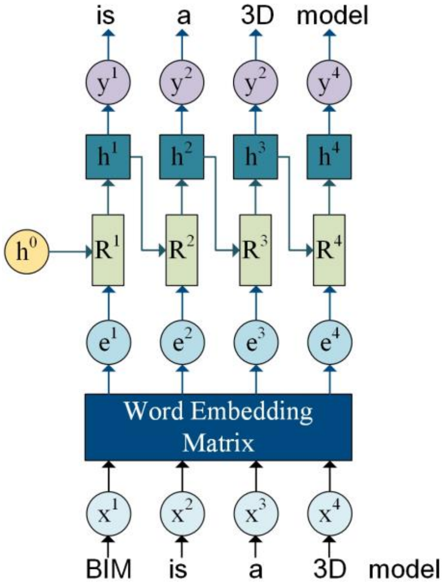


Figure 7-1 Language model performed on the RNN-based model

The major problem is the previous pre-training models is that the bidirectional information is only concatenated in a high-level layer, which makes the tokens meet themselves through leftward and rightward encoding. The deep learning model codes the bidirectional information within the self-attention sublayer. Figure 7-2 shows the masked language model performed in the pre-training process. A percentage of 15% of the input tokens are masked randomly, and the aim is to predict the masked tokens in the outputs. The elements of the token embedding matrix are parameters that would be learned in the pre-training phase. "MASK" is also included in the wordpiece tokens and each token embedding vector is initialed randomly. In the top part of the model, only the hidden states corresponding to the masked tokens are connected to an output softmax overall the vocabulary to generate a probability vector $\hat{y}^{(t)} \in \mathbb{R}^{|\mathrm{v}|}$ (t denotes the position, |v| denotes the vocabulary size of the wordpiece).

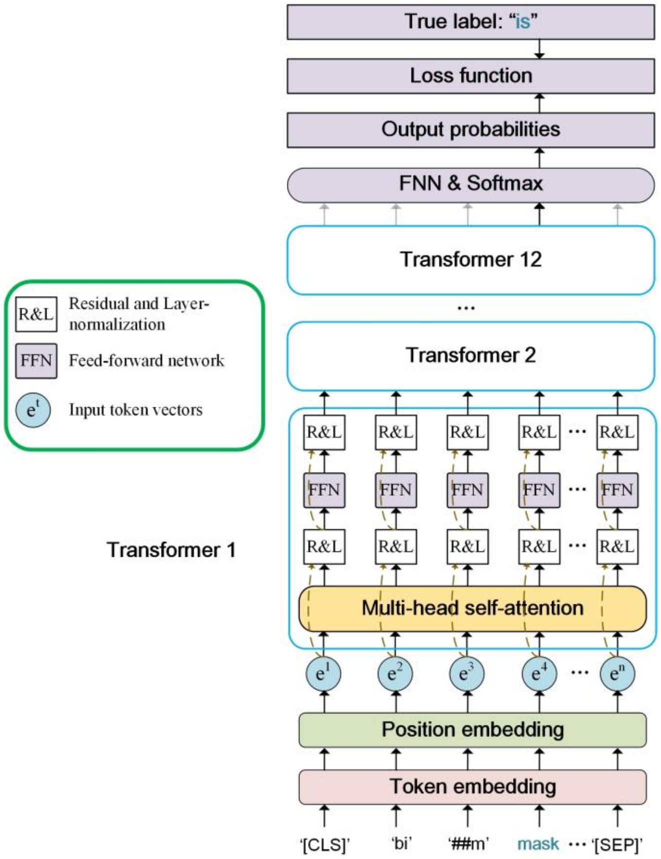


Figure 7-2 Masked language model performed in the pre-training process

According to the value in the hidden state,

$$\hat{y}^{(t)} = softmax\left(O^{(t)}\right) \tag{7.1}$$

where softmax function could be expressed as:

$$\hat{y}^{(t)} = \left(\hat{y}_1^{(t)}, \hat{y}_2^{(t)}, \dots, \hat{y}_{|\mathrm{v}|}^{(t)}\right) \qquad \hat{y}_i^{(t)} = \frac{\exp\left(\hat{y}_i^{(t)}\right)}{\sum_{\mathrm{j}=1}^{|\mathrm{v}|} \exp\left(\hat{y}_j^{(t)}\right)} \tag{7.2}$$

where $\hat{y}_i^{(t)}$ is the predicted probability of the t^th token in the input sentence (i denotes the position of this token in the vocabulary), and $\hat{y}_1^{(t)}, \hat{y}_2^{(t)}, \ldots, \hat{y}_{|\mathrm{v}|}^{(t)}$ sum up to 1. $O^{(t)} \in \mathbb{R}^{|\mathrm{v}|}$, computed by a feed-forward network:

$$O^{(t)} = h^{(t)}U + b \tag{7.3}$$

where $h^{(t)} \in \mathbb{R}^{1 \times \dim}$, $U \in \mathbb{R}^{\dim \times |\mathrm{v}|}$, $b \in \mathbb{R}^{1 \times |\mathrm{v}|}$. $h^{(t)}$ denotes the value of hidden state at t position, U is a linear transformation matrix shared by all the hidden states in the last transformer, and its elements are parameters to be learned. According to the predicted probabilities and the true labels, the loss function is defined by

$$J^{(t)}(\theta) = CE\left(y^{(t)}, \hat{y}^{(t)}\right) = -\sum_{i \in |\mathrm{v}|} y_i^{(t)} \log \hat{y}_i^{(t)} = -\log \hat{y}_i^{(t)} \tag{7.4}$$

where $y^{(t)}$ is the true label vector created by one-hot encoding, in which the element $y_w^{(t)}$ representing the masked word is 1 (w denotes that the masked word occupies m^th position in the vocabulary) and other elements are 0, and CE denotes the cross-entropy loss. The gradients are generated according to the loss function.

This study uses one of the pre-trained parameters - BERT$_{\text{base}}$ trained by Google AI team, which consumed four days based on 4 Cloud TPUs in Pod configuration (16 TPU chips total)[3]. BERT$_{\text{base}}$ contains more than 110M parameters, and the hyperparameters are set as

[3] https://github.com/google-research/bert

following: L = 6 (the number of transformers stacked together), dim = 512, |v|= 30,522, A = 12 (the number of attention heads), max_hidden = 768 (the maximum number of the hidden states).

7.2.2 The Back-propagation of the Deep Learning Model

After the pre-trained parameters were put into the token embedding matrix and transformers of the deep learning model, the training process could be implemented based on the back-propagation from the loss function to the bottom. Such a training process is called fine-tuning. The outputs of the last transformer are fed into a standard linear and softmax over the type numbers of the pre-defined communication-oriented entities:

$$\hat{c}^{(t)} = softmax\left(k^{(t)}\right) \tag{7.5}$$

where softmax function could be expressed as:

$$\hat{c}^{(t)} = \left(\hat{c}_1^{(t)}, \hat{c}_2^{(t)}, \dots, \hat{c}_{|\mathrm{v}|}^{(t)}\right) \qquad \hat{c}_i^{(t)} = \frac{\exp\left(\hat{c}_i^{(t)}\right)}{\sum_{\mathrm{j}=1}^{|\mathrm{v}|} \exp\left(\hat{c}_j^{(t)}\right)} \tag{7.6}$$

where $\hat{c}_i^{(t)}$ is the predicted probability of the entity classification of the token in the t$^{\text{th}}$ of input sentence (i denotes the position of this token in the vocabulary), and $\hat{c}_1^{(t)}, \hat{c}_2^{(t)}, \dots, \hat{c}_{|\mathrm{v}|}^{(t)}$ sum up to 1. $k^{(t)} \in \mathbb{R}^{|\mathrm{v}|}$, computed by a feed-forward network:

$$k^{(t)} = h^{(t)}U^f + b^f \tag{7.7}$$

where $h^{(t)} \in \mathbb{R}^{1 \times \mathrm{dim}}$, $U \in \mathbb{R}^{\mathrm{dim} \times |\mathrm{E}|}$, $b \in \mathbb{R}^{1 \times |\mathrm{E}|}$. |E| is the number of pre-defined entity labels. According to the annotated entity and the predicted probabilities, the loss of a single token could be computed by

$$J^{(t)}(\theta) = CE\left(y^{(t)}, \hat{y}^{(t)}\right) = -\sum_{i\in|\mathrm{v}|} y_i^{(t)} \log \hat{y}_i^{(t)} = -\log \hat{y}_i^{(t)} \tag{7.8}$$

and the overall loss function is defined as:

$$\frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) = \frac{1}{T}\sum_{t=1}^{T} -\log \hat{y}_i^{(t)} \tag{7.9}$$

The maximum number of tokens of the input sentence is set as 512, and each batch contains 256 sentences. Adam optimization is adopted with the learning rate of 1e-4.

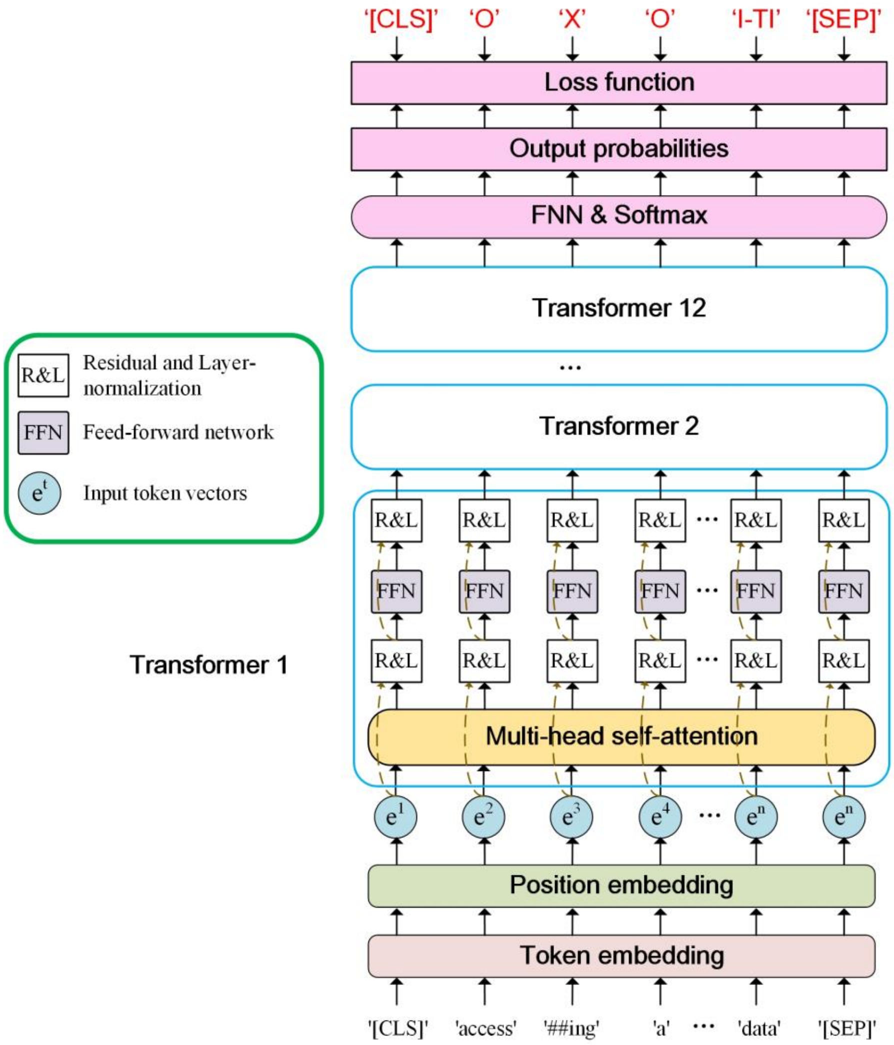


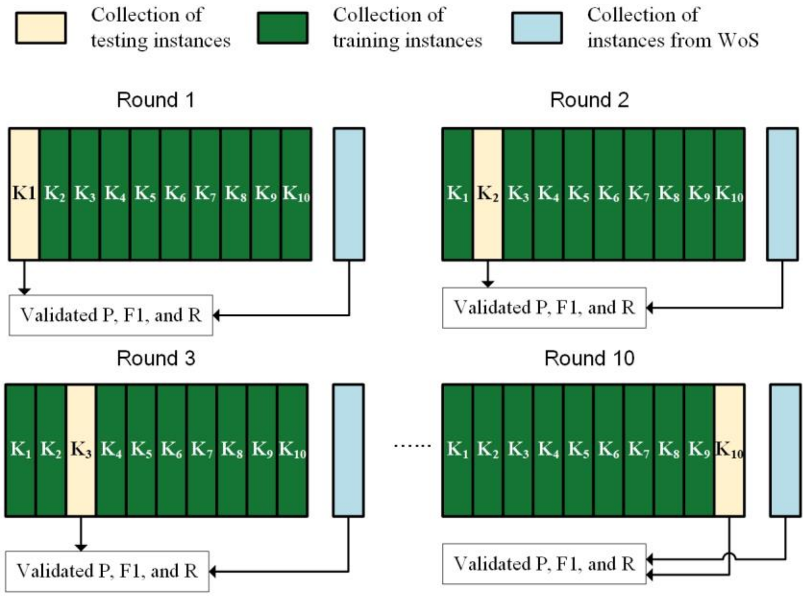
Figure 7-3 Fine-tuning towards CER

7.3 Training Process

7.3.1 Training and Testing Instances

This study, similar to many NLP studies, uses k-fold cross-validation to evaluate the performance by setting k as 10. As mentioned in Table 6.5, the collection of annotated instances incorporates 824 sentences. All the instances were randomly divided into 10 folds. For each training round, nine folds consist of the training collection and the rest one consists of the testing collection. Table 7.1 gives detailed statistics about the sentences, words, number of CEs for each class in each training round.

In addition, this study also retrieves literature relevant to ICT in construction from Web of Science (WoS) and annotate the abstracts of the literature as another testing collection. This can validate the application of the deep learning model to another text source of ICT in construction. The WoS collection was used as an additional validation collection of instances for each round, as shown in Figure 7-4.


Collection of testing instances
Collection of training instances
Collection of instances from WoS
Round 1
K1 K2 K3 K4 K5 K6 K7 K8 K9 K10
Validated P, F1, and R
Round 2
K1 K2 K3 K4 K5 K6 K7 K8 K9 K10
Validated P, F1, and R
Round 3
K1 K2 K3 K4 K5 K6 K7 K8 K9 K10
Validated P, F1, and R
Round 10
K1 K2 K3 K4 K5 K6 K7 K8 K9 K10
Validated P, F1, and R

Final P, F1, and R = average P, F1, and R of round1,2,…,10

Figure 7-4 Illustration for 10-fold validation

This study identifies relevant concepts of ICT in construction for searching by referring to some review papers, (Alsafouri and Ayer, 2018; Lu et al., 2015; Rimmimgton et al., 2015). In addition, we selected four well-known journals (*Advanced Engineering Informatics, Automation in Construction, Computer-Aided Civil and Infrastructure Engineering*, and *Journal of Computing in Civil Engineering*) in the field of ICT in construction as the data source based on two criteria: (1) a journal needs to be included in the ***Web of Science*** or

Scopus databases which are recognized as authoritative (Meho and Yang, 2007); (2) a journal needs to have an important impact and unanimous recognition in the research community of CEM (Chan et al., 2004; Lin and Shen, 2007). The final search strategy is: (*"Topic" =("radio frequency identification" or "rfid" or "3D laser" or "quick response" or "augmented reality" or "mobile computing" or "mobile computing" or "wireless connection" or "robotics")) and ("Publication Name" = ("Advanced Engineering Informatics", "Automation in Construction", "Computer-Aided Civil and Infrastructure Engineering"*, and *"Journal of Computing in Civil Engineering"))*. This study randomly selected 100 literature of the total 322 retrieval results and annotated the abstracts. Finally, 44 sentences from 25 literature were annotated with CEs. The fewer annotated sentences of WoS is due to little communication-related expression in the content of the abstract. This may be reasonable, as many of the literature focus on the management issue and do not mention technical components of the proposed ICT in construction. Overall, the detailed statistics of training and testing collections were described in Table 7.1.

Table 7.1 The statistics for training and testing collections

Round	Purpose	Sentences	Words	TIs	CMs	CSs	Total CEs
R1	Train	741	56360	913	679	222	1814
	Test	83	7405	131	155	38	324
R2	Train	741	57311	942	740	239	1921
	Test	83	6454	105	86	21	212
R3	Train	741	58008	972	770	247	1989
	Test	83	5757	68	52	12	132
R4	Train	741	56653	921	729	235	1885
	Test	83	7112	125	101	24	250
R5	Train	742	58114	944	748	238	1930
	Test	82	5651	102	74	22	198
R6	Train	742	58270	931	749	243	1923
	Test	82	5495	114	75	19	208

R7	Train	742	53701	906	703	210	1819
	Test	82	10064	131	116	51	298
R8	Train	742	59533	977	777	242	1996
	Test	82	4232	71	45	18	134
R9	Train	742	57750	933	764	229	1926
	Test	82	6015	112	64	37	213
R10	Train	742	58185	948	739	226	1913
	Test	82	5580	96	89	37	222
WoS	Test	88	2687	45	34	6	85

7.3.2 Experiment Setup for Training

Based on the split training and testing instances, the hyperparameters were set as shown in Table 7.2. The training programs were implemented based on a workstation with the CPU: Intel(R) Core(TM) i7-7700HQ CPU @2.80Hz 2.81GHz and 16.0G RAM, the GPU: NVIDIA Quadro P4000, 8G. GPU plays a major role in TPF fine-tuning process. Because of the limitation of the GPU (only with 8G RAM), the batch size was set as 2. Each batch size comprises a sentence which maximum length is set as 512. Certain self-developed Python programs were used to perform the training and testing tasks.

Table 7.2 Hyperparameters setting for fine-tuning of TPF

Model settings	
Number of transformers	12
Dimension of WordPiece tokens	512
Number of attention heads	12
Maximum number of hidden states	768
Training settings	
Batch size	2
Max sequence length	512
Drop rate	0.1
learning rate	5e-5
Training epochs	3
Optimizer	Adma

7.4 Validation Methods

7.4.1 Basic Validation Measurement

The validation is carried out according to the aforementioned P, R, and F1. This study has three types of CEs to be identified and classified. Therefore, the counts of TP, FN, and FP were based on the instances of the three types of CEs respectively, expressed by the following equations:

$$\begin{cases} TP_{Total} = & TP_{TI} + TP_{CM} + TP_{CS} \\ FN_{Total} = & FN_{TI} + FN_{CM} + FN_{CS} \\ FP_{Total} = & FP_{TI} + FP_{CM} + FP_{CS} \end{cases} \tag{7.10}$$

Setting TI as an example, Table 7.3 illustrates the counting process of TP$_{TI}$, FN$_{TI,}$ and FP$_{TI}$. "TP$_{TI}$" denotes the number of TIs correctly labeled by the deep learning model, "FP" denotes the number of TIs incorrectly labeled, and "FN" denotes the number of TIs the deep learning model failed to predict. Based on the counted TP, FN and FP for TIs, CMs and CSs, the computations of P, R, and F1 follow Eq. (4.1).

Table 7.3 TI-specific contingency table

		Predicted outputs	
		TI	not TI
True labels	TI	TP$_{TI}$	FN$_{TI}$
	Not TI	FP$_{TI}$	TN$_{TI}$

7.4.2 Hard and Soft Rules

Since IOB annotation system was used for labeling the CE tags, a CE integrating more than one words may have more than one labels. This study presents two types of evaluation according to the process for identifying TPs, FPs and FNs (Table 7.4): (1) Hard: a predict

CE class is considered as a correct prediction if all the labels of sub-words and split tokens are correct (Goyal et al., 2018; Sang and Meulder, 2003); (2) Soft: if a sub-word or sub-token of a CE is correctly predicted this study considers this prediction of the CE is correct (Grishman and Sundheim, 1996a). The soft rule may be different with the hard rule for the same CE when the CE contains more than one tokens. This study uses the soft rule for validation because the mentions about CEs are always written in complex genres, incorporating possessive cases and modifiers.

Table 7.4 Examples of strict and soft evaluation

		Evaluation	
		Strict	**Soft**
Original text with true labels	BIM database (communication model)		
Tokenization	‘b’, ‘##im’, ‘database’		
True labels	*B-CM, X, I-CM*		
Prediction 1	*B-CM, X, I-CM*	√	√
Prediction 2	*B-CM, X, O*	×	√
Prediction 3	*O, X, I-CM*	×	√
Prediction 4	*B-CM, X, B-TI*	×	×
Prediction 5	*O, X, O*	×	×

7.4.3 K-fold Cross-Validation

Cross-validation is a technique that splits the annotated instances into two collections, each of which is used for training and validation respectively. The so-called k-fold validation technique is proposed at 1995, and has been increasingly used to evaluate the performance for NLP tasks (Kohavi, 1995). K-fold validation partitions the collection of annotated data into k subsets namely folds K_1 ,…,K_k. Basically, the training process comprises k rounds, each of which has a unique combination of k-1 folds for training and one fold for validation. The final validation results (P, R, and F1) is the average value of the results of the k rounds.

7.5 Validation Results

This study evaluates the performance of the deep learning model based on the transformer over pre-training and fine-tuning (TPF) and Bi-LSTM-CNN (BLC) based on the 10-fold instances. BLC is built upon RNN-based architecture, combining a bi-directional LSTM layer, a CNN layer, and a CFR layer. BLC is selected to compare with TPF because it is regarded as one of the most state-of-art and outperformed models (Hofer et al., 2018).

The validation values of F-score, precision, and recall of BLC and TPF over the two testing collections for each of the training rounds are described in Table 7.5. Apparently, it can be observed that the proposed model TPF is superior to BLC. It could be further noted that TPF appears to outperform BLC in all the training rounds over the two different testing collections. The CEs were recognized highly depending on the contextual meanings within a sentence rather than word-level meanings, because the digital data, technical models or apparatus, and organization or peoples would be determined as TI, CM, and CS respectively as long as they are recognized they play communication-oriented roles in the sentence. The performance evaluations indicate that TPF is more accurate, and the main reason is the better capacity to utilize the contextual information to recognize the CEs.

On one hand, TPF has a deeper and thinner neural structure where all the tokens are fed parallelly and the dependencies among the input tokens were addressed only by the self-attention mechanism, while the BLC is built with one or two layers of neural networks (Figure 3-7). In addition, the transformer-based structure is more effective in transmitting gradients, leading to a better learning ability than RNN-based model, because TPF draws the dependencies parallelly, while BLC transfers the dependencies from a recurrence to the next.

Table 7.5 Performances of TPF against BLC over the 10 training rounds

	Baseline model: BLC						**Proposed model: TPF**					
	USPTO			WoS			USPTO			WoS		
Round	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**
1	0.670	0.737	0.614	0.400	0.579	0.306	0.856	**0.884**	0.830	0.706	0.814	0.624
2	0.713	0.739	0.689	0.389	0.600	0.287	0.854	0.836	0.873	0.702	0.875	0.586
3	0.593	0.532	0.669	0.421	0.523	0.353	0.798	0.793	0.803	0.689	0.808	0.600
4	0.667	0.687	0.648	0.393	0.594	0.294	0.859	0.855	0.864	0.732	0.842	0.647
5	0.661	0.616	0.712	0.423	0.590	0.329	0.834	0.851	0.818	0.702	0.870	0.588
6	0.693	0.675	0.712	0.455	0.526	0.400	0.860	0.874	0.846	0.739	0.842	0.659
7	0.657	0.644	0.672	0.434	0.595	0.341	0.793	0.775	0.812	0.704	0.851	0.600
8	0.644	0.585	0.716	0.455	0.492	0.424	0.850	0.864	0.836	**0.862**	**0.933**	**0.800**
9	0.676	0.645	0.709	0.482	0.582	0.412	**0.882**	0.864	**0.901**	0.669	0.848	0.553
10	0.657	0.665	0.649	0.341	0.500	0.259	0.823	0.822	0.824	0.766	0.873	0.682

Notes: The highest values for each model over both testing collection are highlighted

On the other hand, TPF provides a pre-training phase taking the context-level representations into consideration, whereas the BLC only has a word-level pre-training. Figure 7-5 shows the pre-training phase of BLC. Word2Vec embedding technique was used to pre-train a word embedding matrix based on the language model through a large corpus of Wikipedia materials. In the ER task, each input word would be extracted as a word vector from the achieved word embedding matrix by look-up operation, and then the word vectors were fed into the neural network. Therefore, the BLC model has to learn the inter-dependencies from scratch to implement the CER task. On the contrary, the TPF performs the pre-training with its own neural network, providing learned neural networks with pre-trained parameters and token embeddings, forming the ability to utilize contextual information. Based on such ability, the fine-tuning phase could train the model to perform CER task in a more effective way. Given the relatively small size of the collection of annotated data for ICT in construction, the advantage of the pre-training phase of TPF may widen because it could focus on the CER task in the fine-tuning, while the BLC has to learn the inter-dependencies and their relation with CEs simultaneously.

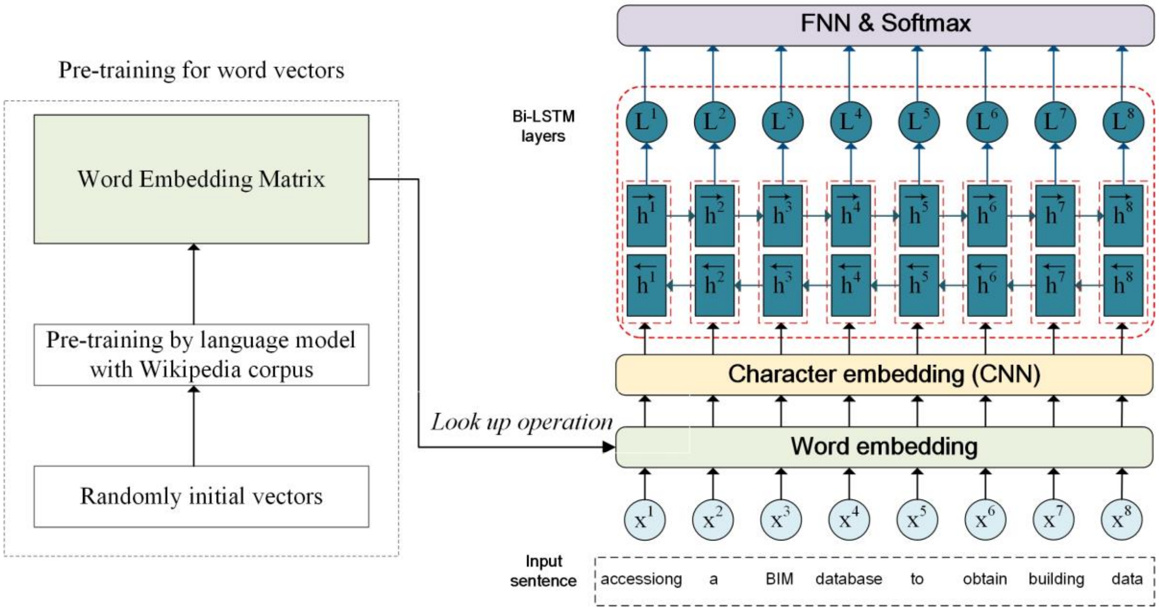


Figure 7-5 The pre-training for word vectors in BLC

Table 7.6 compares the proposed TPF to BLC over the average values of the validation. All the validation indexes of TPF yielded at least 15% higher accuracy than BLC. It could also be found from the validation results that the TPF is more compatible with WoS testing collection in which the language is written in a varied style to the language in patents. Although the recall value of TPF over WoS collection is 0.634 (compared with 0.841 of USPTO collection), the gap is not as large as BLC, which validation indexes drop significantly. With respect to TPF's lower value of recall but higher precision over WoS, it indicates that the TPF is more careful to identify a CE by utilizing the contextual information.

Table 7.6 Comparison of BLC and TPF over the average performance value over the 10 training rounds

	Baseline model: BLC			Proposed model: TPF		
	F1	P	R	F1	P	R
USPTO	0.654	0.631	0.683	0.841 **(+18.7%)**	0.842 **(+21.1%)**	0.841 **(+15.8%)**
WoS	0.42	0.558	0.341	0.727 **(+30.7%)**	0.855 **(+29.7%)**	0.634 **(+29.3%)**

Besides the average performance, it is worth to project the results over the 10 training rounds, as shown in Figure 7-6. The green boxes represent the TPF, and blue boxes represent BLC. As can be observed, the performance of the training round of TPF that has the lowest accuracy is higher than the BLC's best training round. In addition, the TPF has a shorter boxes' hight indicating that TPF is more stable for training, resulting in more similar validation results over the different training rounds.

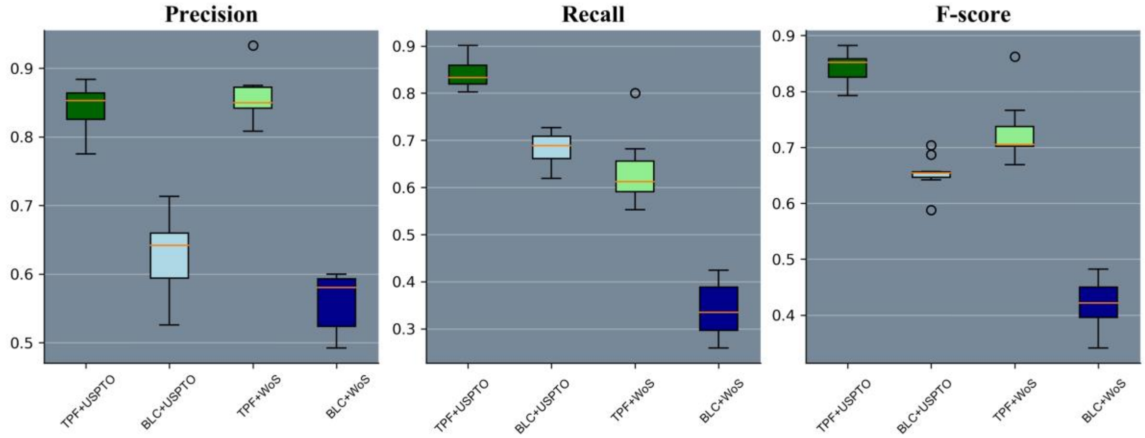
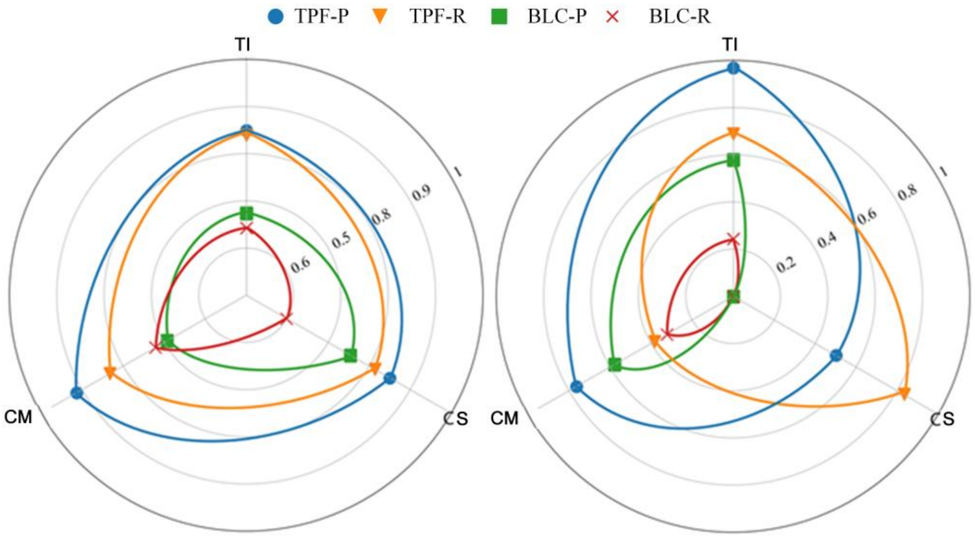


Figure 7-6 The performance of TPF versus BLC over different testing collections

Figure 7-7 plots the radar map to compare the performance of precision and recall for the three CE classes over TPF and BLC. First of all, all the green lines are inside in the blue ones, and all the red lines are inside of the orange lines, indicating that TPF outperforms BLC in each of the CE classes.



(a) USPTO (b) WoS

Figure 7-7 Performance of three CE classes over two testing collections

In specific, Figure 7-8 plots the heat maps for the confusion matrixes for the two models. The numbers are obtained by combining all the testing instances in the ten training rounds. The heat maps plot the comprehensive information about the number of correctly predicted CEs, as well as the number of misclassification information. The sum of the row indicates the number of true labels of the row's CE class (TP+FN), and the diagonal elements are the number of correctly predict instances of the corresponding CE class. Counting all the instances of the three CE classes, TPF correctly recognizes 300 CEs more than BLC.

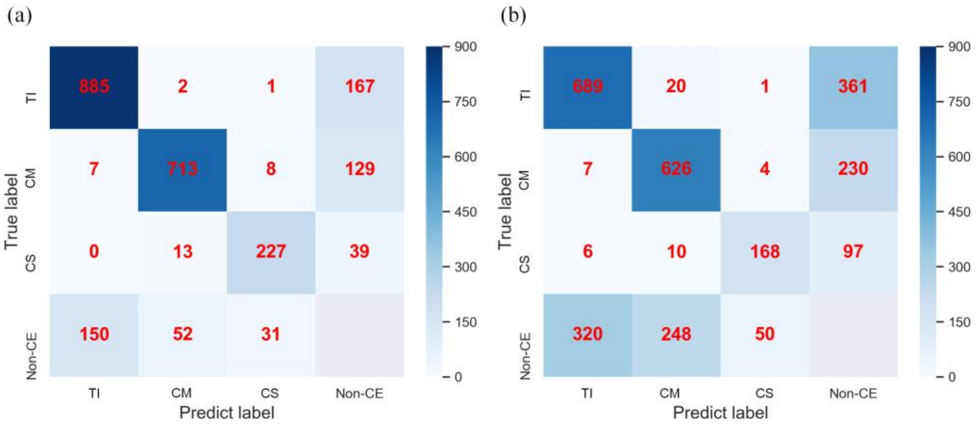


Figure 7-8 Heat maps for confusion matrixes: (a) Proposed TPF, (b) BLC. The value v_{ij} corresponds to the number of CE class i that were predicted as CE class j.

7.6 Implications

Although the results have indicated that the TPF outperforms RNN-based model for the CER of ICT in construction, some implications have been drawn from the results.

7.6.1 The Ability to Recognize Ambiguous Entities

One of the technical problems of CER is that plenty of ambiguous entities exist in the technical documents of ICT in construction. Confused entities are the technical entities that they have the same spellings but different usage in different contexts. Table 7.7 describes the validation values for all the ambiguous entities, indicating that TPF is the better model to predict ambiguous entities, which precision and recall value are 0.875 and 0.844 respectively, whereas BLC has lower accuracy in terms of precision and recall.

Table 7.7 Performance TPF and BLC for ambiguous entities

Precision		**Recall**		**Number of instances of ambiguous entities**
TPF	BLC	TPF	BLC	
0.875	0.753	0.844	0.717	1219

Figure 7-9 illustrates the performance of the models on the ambiguous entities over the different rate of CEA/TA. The CEA denotes the appearance times of the ambiguous entities as CEs, and TA denotes the total appearance times of the ambiguous entities. The bar plots were ordered as the cumulative percentage of CEA/TA, and the red line represents the number of instances of CE. It can be observed that the smaller CEA/TA leads to a lower accuracy of the two models.

A smaller CEA/TA for a specific CE indicates the less opportunity for the deep learning model to learn from training instances. For example, the word "image" appears 291 times in the annotate data, but 19 (less than 1%) of them are transferred information (TI). Such a low percentage of TI gives the deep learning model much more information about how to decide "image" is not a TI but reducing the chances to learn how to utilize the surrounding information to determine "image" as a TI. It can also be observed that the gap between the two models' performance decrease as the CEA/TA increases. This may indicate that the TPF is much better at learning when the number of training instances of CEs is limited.

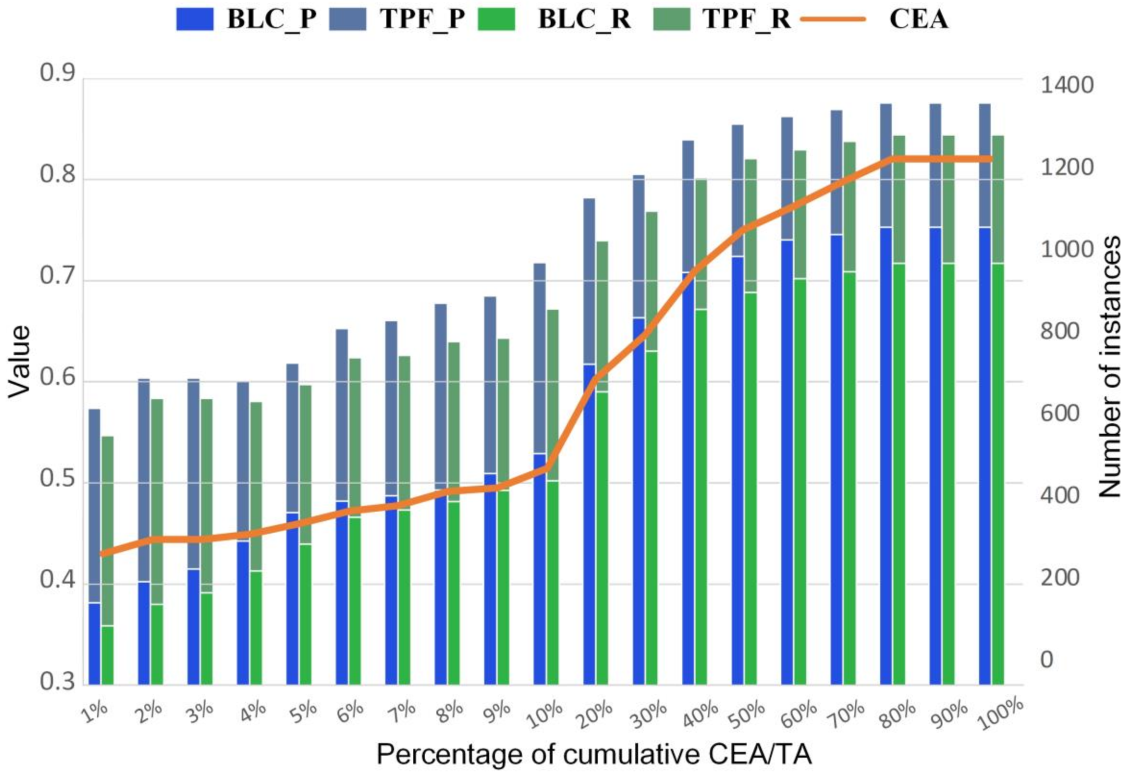
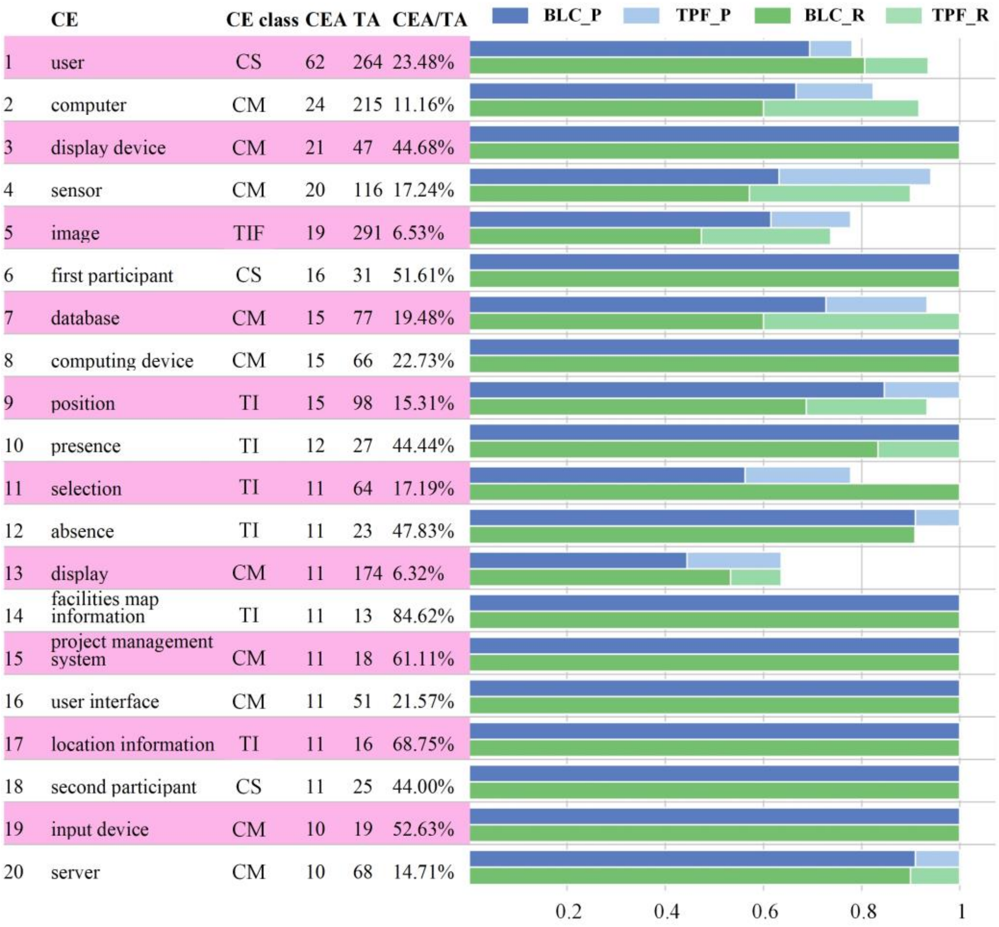


Figure 7-9 Performance of TPF and BLC for ambiguous entities towards different percentages of CEA/TA

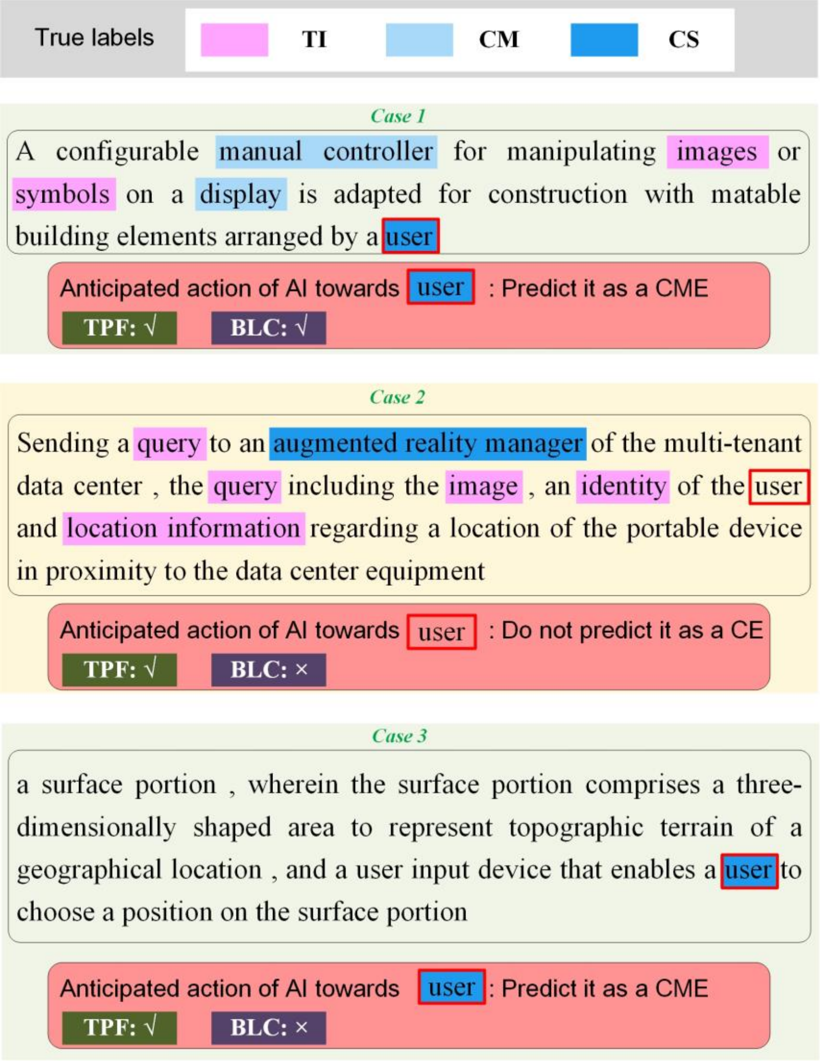
Figure 7-10 plots the performance of the specific ambiguous entities. The figure shows three trends: (1) the ambiguous entities with lower CEA/TA (i.e. “computer”, “sensor”, “image”, and “display”) tend to make the deep learning models predict incorrectly. This may be due to the ambiguous entities could be expressed in a wide range of expressions in the technical documents of ICT in construction, increasing the burden for the deep learning models to discern the relations between CEs and the contextual information; (2) the CEs with more specific expressions (i.e. “display device”, “facilities map information”, “project management system”, “user interface”, and “location information”) tend to experience higher accuracy of the two models. More specific expression conveys more specific word-level information and surrounding context; (3) TPF is much better for recognizing ambiguous entities, with higher accuracy in terms of precision and recall.



	CE	CE class	CEA	TA	CEA/TA
1	user	CS	62	264	23.48%
2	computer	CM	24	215	11.16%
3	display device	CM	21	47	44.68%
4	sensor	CM	20	116	17.24%
5	image	TIF	19	291	6.53%
6	first participant	CS	16	31	51.61%
7	database	CM	15	77	19.48%
8	computing device	CM	15	66	22.73%
9	position	TI	15	98	15.31%
10	presence	TI	12	27	44.44%
11	selection	TI	11	64	17.19%
12	absence	TI	11	23	47.83%
13	display	CM	11	174	6.32%
14	facilities map information	TI	11	13	84.62%
15	project management system	CM	11	18	61.11%
16	user interface	CM	11	51	21.57%
17	location information	TI	11	16	68.75%
18	second participant	CS	11	25	44.00%
19	input device	CM	10	19	52.63%
20	server	CM	10	68	14.71%

Figure 7-10 Performance of TPF and BLC over the ambiguous entities

To better illustrate the difference between the recognition process of the two models, Figure 7-11 plots three sentences containing the word "user", which may be or not be a CS depending on the contextual information. In case 1, it is not a hard work to indicate that "user" is a CS, because the former part of this sentence expresses a communication activity involving transferred digital data and communication apparatus. With this information, the "user" could be recognized as a CS when the deep learning sees the "arranged by" before it, which indicates that the "user" participants in the communication activity mentioned in the former of the sentence. Both models correctly recognized it. Case 2 and case 3 are more complicated, in which the TPF correctly predicts but BLC does not. The word "user" in case 2 is not a CS, because it is no more than a common word. When the surrounding context incorporates many CEs, the BLC tends to predict the ambiguous entities as CEs that are actually not. Case 3 expresses a communication environment where the user participants. The difficulty lies in the vague expression of the communication environment. The sentence does not articulate specific information for the transformation patterns, nor it does not contain any other type of CEs. This misleads the BLC to make an incorrect judgment that no recognition of CS towards the "user".



Case 1

A configurable manual controller for manipulating images or symbols on a display is adapted for construction with matable building elements arranged by a user

Anticipated action of AI towards user : Predict it as a CME

TPF: √ BLC: √

Case 2

Sending a query to an augmented reality manager of the multi-tenant data center , the query including the image , an identity of the user and location information regarding a location of the portable device in proximity to the data center equipment

Anticipated action of AI towards user : Do not predict it as a CE

TPF: √ BLC: ×

Case 3

a surface portion , wherein the surface portion comprises a three-dimensionally shaped area to represent topographic terrain of a geographical location , and a user input device that enables a user to choose a position on the surface portion

Anticipated action of AI towards user : Predict it as a CME

TPF: √ BLC: ×

Figure 7-11 Examples of recognition of ambiguous entities

7.6.2 The Ability to Predict Unknown CEs

One of the main motivations to utilize the contextual information is due to the widespread unknown CEs in the technical documents of ICT in construction. ICT in construction evolved fast, involving various information techniques. Therefore, the deep learning model has to be able to predict the CEs which are not met during the training process. Figure 7-12 plots the performance over the unknown CEs of the two models. As the 10-fold cross-validation separates the annotated data into training and testing collections for a single training round, the unknown CEs are the labeled CEs in the testing collection but not in training collection. Table 7.8 shows the recall value (as unknown CEs only appear in the testing collection, it could only be validated by recall value) of unknown CEs and total CEs. The performance of both models decrease for predicting the unknown CEs, but TPF's recall value remains as 0.741, almost 20% larger than BLC.

Table 7.8 Recall value of TPF and BLC for unknown CEs

TPF		**BLC**		**Number of instances of unknown CEs**
Total	Unknown CEs	Total	Unknown CEs	
0.841	0.741071	0.683	0.544642857	112

With regard to the three CE classes, a similar trend could be observed. Figure 7-12 depicts the radar map to illustrate the performance of the three CE classes. It can be observed that all of the three classes of unknown CEs experience reduced recall values. However, TPF has a similar ability to predict unknown TI and CM, which have a small decrease in terms of recall value compared with total CEs. The difficulty lies in the recognition of unknown CSs, which performance is less than the total CSs by a half.

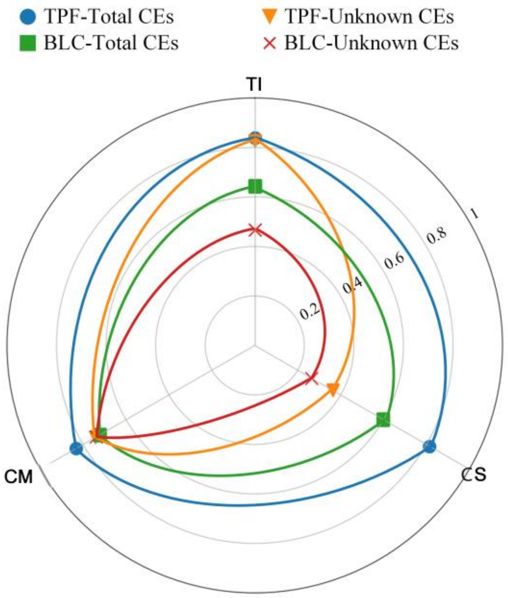
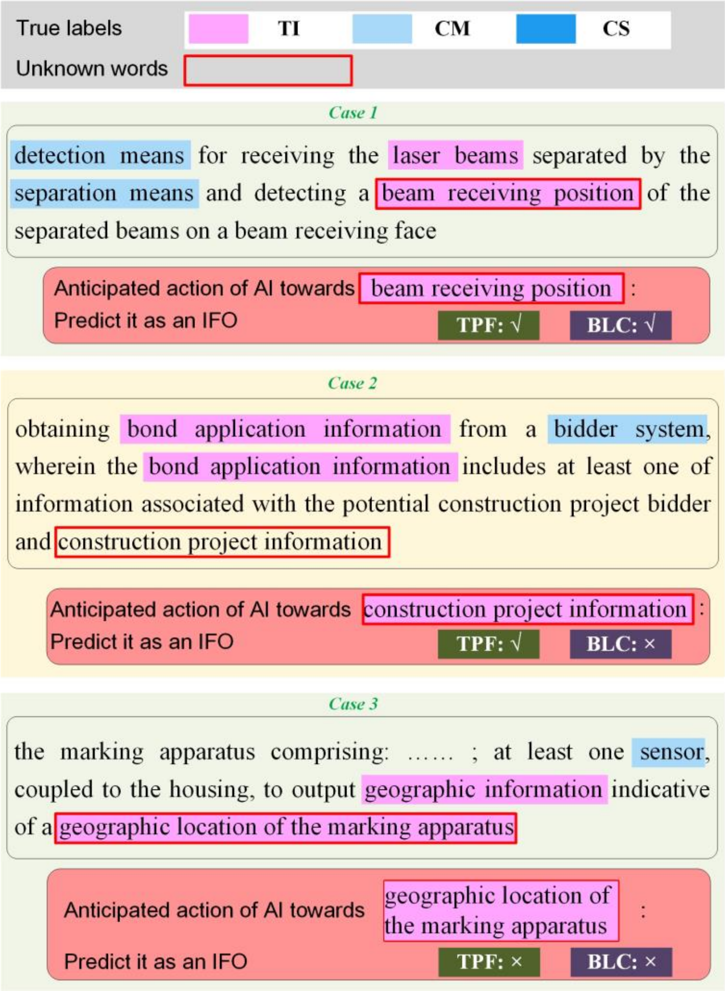


Figure 7-12 Performance of TPF and BLC of unknown TI, CM, and CS

This study sets some examples to further illustrate the difference in the ability to recognize unknown CEs between the two models (Figure 7-13). Case 1 describes a sentence containing an unknown TI "beam receiving position". The contextual information gives both models sufficient clues that "beam receiving position" could be a TI although the two models have not seen this phrase during the training process. Case 2 poses a more complex context, in which the unknown TI "construction project information" is far away from the former part, and it was not directly expressed that it can be transferred or conveyed. The phrase "construction project information" in the sentence constitutes the aforementioned "bond application information", which could be obtained by a "bidder system", indicating that "construction project information" is a TI. Understanding such a complex and indirect expression needs to draw the interdependencies among the words and phrases. For such a complex context, TPF correctly recognized the unknown TI "construction project information" but BLC fails. In case 3 the unknown TI "the geographic location of the marking apparatus” could be recognized by understanding that “geographic location of the marking apparatus” is the further explanation of the aforementioned TI “geographic information”. Both of the models failed to recognize the unknown TI. This may result from the adjective “indicative of” that makes the model hesitate to link the “geographic location of the marking apparatus” to the former TI “geographic information”



True labels TI CM CS

Unknown words

Case 1

detection means for receiving the laser beams separated by the separation means and detecting a beam receiving position of the separated beams on a beam receiving face

Anticipated action of AI towards beam receiving position : Predict it as an IFO

TPF: √ BLC: √

Case 2

obtaining bond application information from a bidder system, wherein the bond application information includes at least one of information associated with the potential construction project bidder and construction project information

Anticipated action of AI towards construction project information : Predict it as an IFO

TPF: √ BLC: ×

Case 3

the marking apparatus comprising: …… ; at least one sensor, coupled to the housing, to output geographic information indicative of a geographic location of the marking apparatus

Anticipated action of AI towards geographic location of the marking apparatus : Predict it as an IFO

TPF: × BLC: ×

Figure 7-13 Examples of recognition of unknown CEs

7.7 Summary of this Chapter

This chapter employs k-fold cross-validation to validate the deep learning model, compared to the RNN-based model. The training and validation techniques were illustrated. The results show that when the size of the annotated data collection for CER is small, the developed deep learning model significantly outperforms the RNN-based model. Furthermore, the performance of the two models over ambiguous entities and unknown CEs were computed. The results also indicate that the developed deep learning model is better for addressing the contextual information to recognize the CEs.

Chapter 8 Applications of the Deep Learning Model for CER of ICT in Construction

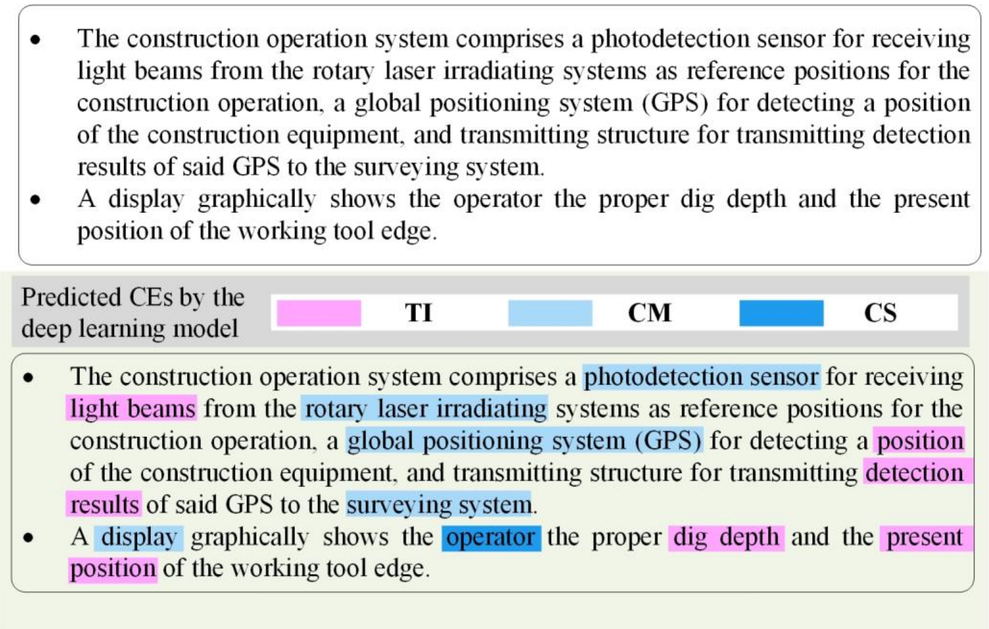
8.1 Introduction

The deep learning model for CER of ICT in construction can be used for further NLP applications. This chapter firstly describes the potential applications in two perspectives: direct and indirect applications. Secondly, the chapter presents two specific applications, utilizing the recognized CEs as features to train a classifier to categorize the patents of ICT in construction into pre-defined communication modes. This study assumes that the recognized CEs are more informative than common words in discerning communication modes. Therefore, this chapter validates the classification based on CEs as features compared to common words as features. Furthermore, the resulting classification scheme reveals the trends in communication modes embodied in the patents of ICT in construction.

8.2 Potential Applications of the Deep Learning Model for CER of ICT in Construction

8.2.1 Assistance in Understanding Communication Patterns

The direct application is to automatically identify CEs and classify them into pre-defined classes, as shown in Figure 8-1. As for those who are familiar with the field of ICT in construction, the deep learning model could save their time to tag all the CEs into the pre-defined classes: TI, CM, and CS. The technical documents are boring and long, consuming a large amount of time and energy for users. The deep learning model could identify and label all the CEs into classes, assisting users to quickly and correctly get to know the communication patterns from the raw texts.



- The construction operation system comprises a photodetection sensor for receiving light beams from the rotary laser irradiating systems as reference positions for the construction operation, a global positioning system (GPS) for detecting a position of the construction equipment, and transmitting structure for transmitting detection results of said GPS to the surveying system.
- A display graphically shows the operator the proper dig depth and the present position of the working tool edge.

Predicted CEs by the deep learning model 🟪 TI 🟦 CM 🟦 CS

- The construction operation system comprises a photodetection sensor for receiving light beams from the rotary laser irradiating systems as reference positions for the construction operation, a global positioning system (GPS) for detecting a position of the construction equipment, and transmitting structure for transmitting detection results of said GPS to the surveying system.
- A display graphically shows the operator the proper dig depth and the present position of the working tool edge.

Figure 8-1 Recognition of CEs in a text by the deep learning model

8.2.2 Down Streaming Applications

8.2.2.1 Indexing system for the technical documents of ICT in construction

Indexing is a necessary step before retrieval. The deep learning model provides an effective way to index the technical documents of ICT in construction by CEs. Traditionally, the indexing techniques utilized the features extracting from the text by n-grams and Tf-Idf, which relying on the term frequencies to extract features. This does not bring valuable information for specific knowledge. Through the deep learning model, all the CEs and their classes would be recognized. This enables an alternative way to index these technical documents with emphasis on communication functionality. Table 8.1 shows the recognized CEs for 10 patents of ICT in construction. Unlike the general words many of which bring nothing valuable information, the recognized CEs would indicate the communication patterns for the ICT in construction.

Table 8.1 Indexing 10 patents of ICT in construction by recognized CEs compared with n-grams

Code	N-grams (N=1 and 2)	Communication-oriented entities in terms of Transferred information (TI), Communicated models (CM) and Communicated subjects (CS)
7593751	['data', 23], ['field', 20], ['management', 13], ['handheld', 12], ['data management', 12], ['remote', 11], ['handheld data', 8], ['access', 7], ['related', 7], ['user', 7], ['the field', 7], ['using', 6], ['field location,', 6], ['access to', 5], ['related to', 5], ['at least', 4], ['field data', 4]	**TI**: ['industry-specific programs', 1], ['instructions', 1], ['field location', 2], ['remote resources', 3], ['field related data', 1], ['information', 1], ['field operation instructions', 1], ['real - time guidance', 1], ['third party information', 1], ['location', 1] **CM**: ['server', 1], ['computer workstation', 1], ['remote computers', 1], ['handheld data management devices', 1], ['digital', 1], ['digital assistants', 1], ['way page', 1], ['handheld device', 3], ['remote resource', 1], ['internet', 1], ['wireless communication module', 1], ['handheld', 1], ['user interface', 1] **CS**: ['user', 5], ['remote personnel', 1]
7911344	['data', 17], ['radio', 15], ['construction', 11], ['transceiver', 9], ['tag', 8], ['low', 5], ['identification', 5], ['radio tag', 5], [' construction', 5], [' radio', 5], ['visibility', 4], ['material', 4], ['construction material', 4], ['radio frequency', 4], ['said transceiver', 4], ['data storage', 4], ['identification data', 4]	**TI**: ['tag', 2], ['frequency', 2], ['transceiver radio', 1], ['transceiver radio radio tag', 1], ['data signals', 1], ['identification signal', 1], ['identification data', 1], ['frequency interrogation', 1], ['frequency interrogation signals', 1] **CM**: ['transceiver', 6], ['tag antenna', 2], ['antenna', 1], ['data processor', 1], ['data storage device', 1], ['energy source', 1], ['radio tags', 1] **CS**: *Not given*
8001160	['information', 45], ['said', 41], ['least', 21], ['one', 20], ['at least', 20], ['least one', 20], ['contractor', 18], ['construction', 16], ['particular', 16], ['job', 14], ['information for', 14], ['contractor information', 12], ['select', 11], ['construction information', 10], ['plurality of', 10], ['particular contractor', 9], ['information; ', 8], ['particular construction', 7]	**TI**: ['contractor information', 5], ['manager information', 2], ['job information', 4], ['construction information', 2], ['information', 2], ['building code information', 1], ['permit', 1], ['material information', 1], ['name information', 1], ['contact information', 1], ['license information', 2], ['experience information', 1], ['insurance information', 1] **CM**: ['processor', 5], ['input device', 1], ['database', 3], ['kiosk', 1], ['devices', 1] **CS**: ['contractor', 3], ['manager', 2], ['contractors', 1]

8024094	['maintenance', 14], ['machine', 11], ['history', 10], ['construction', 10], ['said', 10], ['information', 9], ['data', 9], ['time', 9], ['maintenance history', 9], ['construction machine', 8], ['operating', 6], ['date', 6], ['operating time', 6], ['time', 6], ['date and', 6], [' maintenance', 6], ['history information', 5], ['mobile terminal', 5], ['information management', 4]	**TI**: ['construction machine maintenance items', 2], ['operating time', 2], ['date', 2], ['serial number', 1], ['machine information', 1], ['time', 1] **CM**: ['monitor', 1], ['mobile terminal', 4], ['data recording device', 1], ['communication cable', 3], ['communication connector', 1], ['control means', 1], ['display means', 2], ['input means', 1], ['means', 1], ['communication connectors', 2], ['data control device', 2], ['connector', 1] **CS**: ['worker', 2]
8244606	['lien', 14], ['waiver', 14], ['lien waiver', 14], ['payee', 11], ['signed', 11], ['signed lien', 11], ['document', 10], ['waiver document', 10], [' signed', 9], [' payee', 8], ['payment', 7], ['transmitting', 6], ['document to', 6], ['construction', 5], ['device', 5], ['payor', 5], ['a construction', 5], ['payee device', 5]	**TI**: ['lien waiver document', 2], ['signed lien waiver document', 10], ['payment', 4] **CM**: ['payee device', 4], ['construction project management server', 2] **CS**: ['payee', 7], ['payor', 8]
9311614	['locate', 32], ['electronic', 17], ['of ', 13], ['access', 10], ['record', 9], ['electronic record', 9], ['record of', 8], [' electronic', 8], ['by ', 8], ['to ', 7], ['mechanism', 6], ['operation,', 6], ['technician', 6], ['access mechanism', 6], ['jobsite', 5]	**TI**: ['location', 1], ['electronic record', 3], ['mark', 1], ['electronic data', 2], ['digital image', 1], ['electronic marking', 1] **CM**: ['locate equipment', 4], ['data repository', 2], ['gps apparatus)', 1], ['wireless communications system', 1], ['wifi', 1], ['locate technician', 5], ['excavator', 1] **CS**: ['person', 1], ['technician', 1], ['authorized person', 1]
9460561	['physical', 11], ['drawing', 10], ['structure', 9], ['physical structure', 9], ['view', 8], ['2-D', 8], ['2-D drawing', 8], ['within', 7], ['internal', 7], ['portion', 6], ['features', 6], [' 2-D', 6], ['internal features', 6], ['displaying', 5], ['features', 5]	**TI**: ['two-dimensional', 1], ['2 - d drawing', 1], ['2-d drawing', 3], ['two-dimensional (2 - d) drawing', 1], ['3-d', 1] **CM**: ['display screen', 2], ['electronic device', 1], ['camera', 1] **CS**: ['user', 1]
9582946	['breakdown', 15], ['construction', 15], ['control', 10], [' construction', 9], ['equipment', 8], ['current', 8], ['workable', 8], ['diagnosis', 8], ['information', 8], ['construction equipment', 8], ['current workable', 8], ['a breakdown', 7], ['equipment,', 6], ['state', 6], ['construction equipment,', 6], ['plurality of', 6], ['of vehicle', 6], ['vehicle control', 6]	**TI**: ['state information', 4], ['breakdown information', 2], ['breakdown part', 4], ['workable range', 1], ['construction equipment', 2] **CM**: ['diagnosis device', 8], ['equipment', 1], ['vehicle control devices', 4], ['peripheral communication terminal', 2], ['control server', 4], ['communication terminal', 2], ['construction equipment', 1] **CS**: *Not given*

9817922	['3D', 35], ['object', 22], ['3D object', 19], ['new', 18], ['model', 17], ['object model', 17], ['modeling', 16], ['type', 14], ['type of', 14], ['3D modeling', 13], ['first', 12], ['application', 12], ['selected', 12], ['server', 11], [' new', 10], [' selected', 10], ['final type', 10], ['first server', 9], ['BIM 3D', 9]	**TI**: ['2d electronic', 1], ['2d electronic data', 3] **CM**: ['server application', 2], ['server network device', 2], ['processors', 1], ['library application', 2], ['server network', 1], ['application', 1], ['communications network', 1] **CS**: *Not given*
10094654	['said', 43], ['movable', 21], ['mechanical', 21], ['movable mechanical', 21], ['laser', 19], ['member', 18], ['distance', 15], ['second', 14], ['sensing', 13], ['mechanical member', 11], ['sensing device', 10], ['second movable', 10], ['working tool', 9], ['first movable', 9], ['said first', 9], ['laser receiver', 8]	**TI**: ['jobsite', 1], ['elevation', 2], ['angle', 1], ['laser plane of light', 1], ['jobsite elevation', 1], ['laser light energy', 1], ['position of incoming laser light', 1] **CM**: [['gravity sensor', 3], ['laser receiver', 6], ['sensors', 1], ['sensing', 1], ['gps receiver', 3], ['laser', 1], ['sensing device', 2], ['integral display', 1], ['remote display', 3], ['phone', 1], ['smart phone)', 1], ['photosensor', 2], ['electronic angle sensor', 1], ['electronic distance sensor', 1], ['link', 1] **CS**: ['operator', 3], ['machine operator', 2]

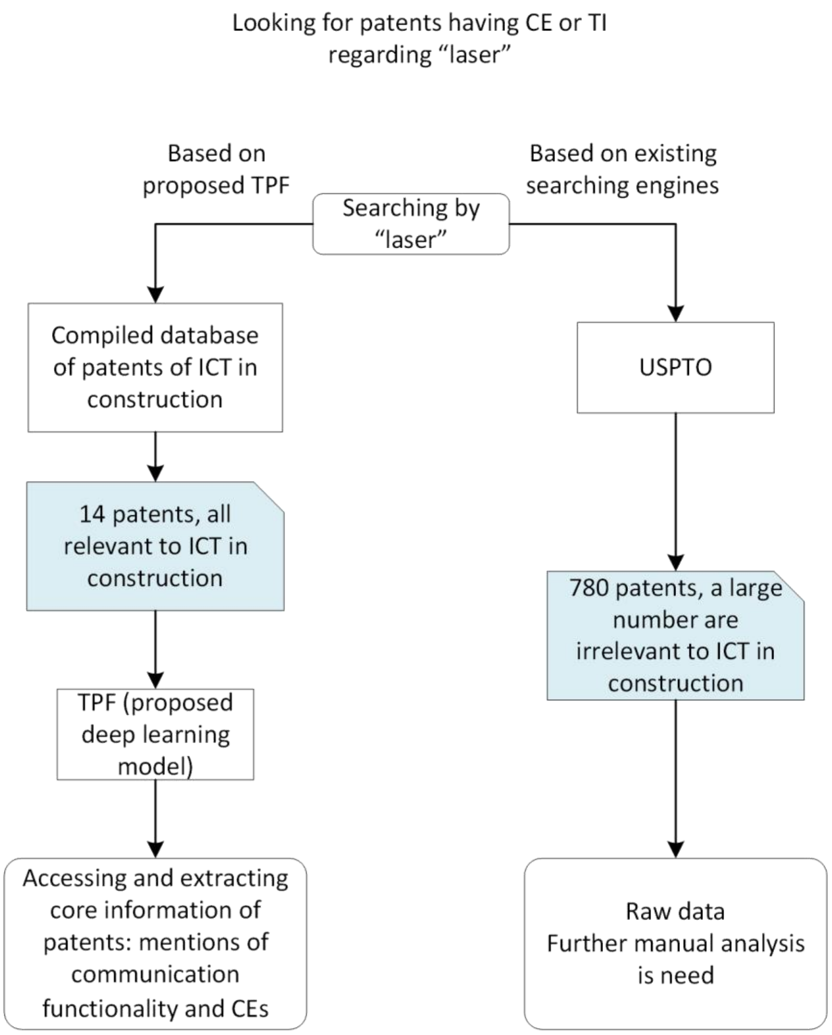
8.2.2.2 QA system for the technical documents of ICT in construction

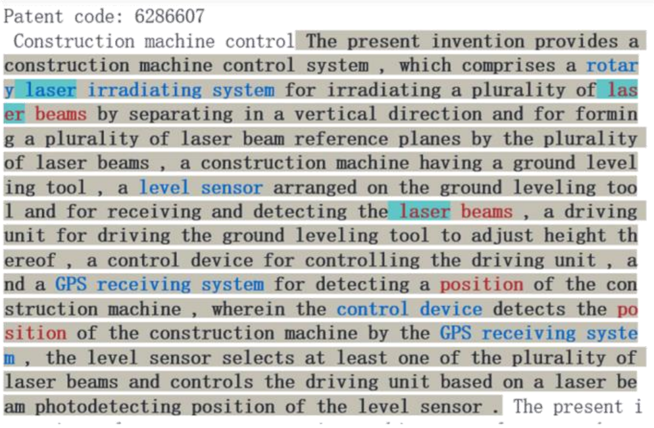
Question Answering (QA) systems aim to automatically create answers to questions proposed by people. The answers and questions are always in textual form. One of the most frequently employed QA system is built upon the so-called "facts", in which the questions are typically started with *What, When, Which, Where and Who*. These factoid questions expect short-terms as answers, phrases or sub-sentences. Therefore, in the field of ICT in construction, the CER may provide basic elements for communication relevant facts. This may satisfy the aforementioned question such as *what information is transferred to what models by who*.

8.3 Application in practice 1: Communication-oriented information retrieval

As mentioned in Section 1.2, it is a challenging work for the managers in construction projects to correctly adapt and properly implement ICT applications to solve problems in the communication process. Conventionally, shaping and enhancing desired communication was conducted largely through trial-and-error experiments or personal judgment, frequently resulting in the failure to make the best usage of potential ICTs. These traditional methods lacked effective analysis of the communication functionality of previous innovations and overlooked the existing knowledge embedded in the written language in the technical documents (especially the patents) of ICT in construction (Tan, 2007).

Based on the compiled database of patents of ICT in construction and the recognized CEs by the TPF model, this study forms an approach to retrieve, extract and abstract information indicating the communication functionality of ICT in construction. This approach forms an efficient method for the users to access desired patents of ICT in construction, identify the process and reveal the attributes of the communication functionality in the patents.


Looking for patents having CE or TI regarding “laser”
Based on proposed TPF
Searching by “laser”
Based on existing searching engines
Compiled database of patents of ICT in construction
14 patents, all relevant to ICT in construction
TPF (proposed deep learning model)
Accessing and extracting core information of patents: mentions of communication functionality and CEs
USPTO
780 patents, a large number are irrelevant to ICT in construction
Raw data Further manual analysis is need



Patent code: 6286607

Construction machine control The present invention provides a construction machine control system , which comprises a rotary laser irradiating system for irradiating a plurality of laser beams by separating in a vertical direction and for forming a plurality of laser beam reference planes by the plurality of laser beams , a construction machine having a ground leveling tool , a level sensor arranged on the ground leveling tool and for receiving and detecting the laser beams , a driving unit for driving the ground leveling tool to adjust height thereof , a control device for controlling the driving unit , and a GPS receiving system for detecting a position of the construction machine , wherein the control device detects the position of the construction machine by the GPS receiving system , the level sensor selects at least one of the plurality of laser beams and controls the driving unit based on a laser beam photodetecting position of the level sensor . The present i

...

Figure 8-2 Retrieval of communication-related information of ICT in construction：TPF vs Searching Engines

The proposed approach uses the compiled collection of patents of ICT in construction as the basic database. Moreover, the recognized CEs are indexed as the basic elements to enable the communication-oriented retrieval, allowing users to retrieve patents by using the CEs. Such a retrieval system is specifically for searching patents of ICT in construction and accessing the content that indicates the communication functionality.

Figure 8-2 presents an example querying "laser" in the proposed retrieval system and USPTO searching engine. According to the query, the retrieval system would return the patents of ICT in construction with "laser" as CEs, whereas USPTO searching engine would return all the patents in which the word "laser" appears. Moreover, the content that mentions the communication functionality in the patents would be extract and all the CEs can be recognized by the retrieval system. Compare with the traditional approach such as searching engines in the patent offices, this approach can specify the retrieval into the patents of ICT in construction, and provide a much more efficient method for the users to access the information indicating the communication functionality in the patents by displaying the content that describes the communication and the CEs. The traditional searching engine would return a lot of irrelevant patents that have mentioned the word "laser" throughout the patent. These resulting patents are no more than raw data, and would take substantial time and energy for users to screen patents of ICT in construction and access key information of the communication functionality.

8.4 Application in practice 2: Classification of Patents of ICT in Construction

8.4.1 Pre-defined Classes of Classification

ICT stands for the information technologies that emphasize the communication role. In the construction industry, the ICTs were particularly utilized to manage project data, advance the collaborations, reduce the risks by improving the information coordination, not only between the physical sites and communication models, but also between stakeholders of construction projects (Alsafouri and Ayer, 2018; Dainty et al., 2007; Rimmimgton et al., 2015). This study, by referring to the work of Alsafouri and Ayer (2018), proposes a classification scheme comprising of three communication modes. Figure 8-3 indicates the information flows in the three communication modes. (1) The first communication mode includes ICTs that enable *semi-automatic information coordination*, in which relevant information & data is obtained from the construction site or database remotely and the intervene of people is needed. In semi-automatic information coordination, the information is usually obtained either from the physical site, normally through monitor, survey or measure apparatus, or from databases in which relevant information or building model data was stored. Typically, that information is delivered to people (users, operators, etc.). For example, a field user operates a mobile computing device to capture structure images. The information in the mode of *semi-automatic information coordination* may be transferred unidirectionally or bidirectionally. (2) The ICTs in the second communication mode automatically transfers the information & data without manual manipulations (automatic information coordination). For example, a location tracking system to automatically identifying environmental information. (3) The last communication mode incorporates the ICTs transferring information & data from a stakeholder to another. For example, a construction payment management system transfers requests for payment to participants in a construction project.

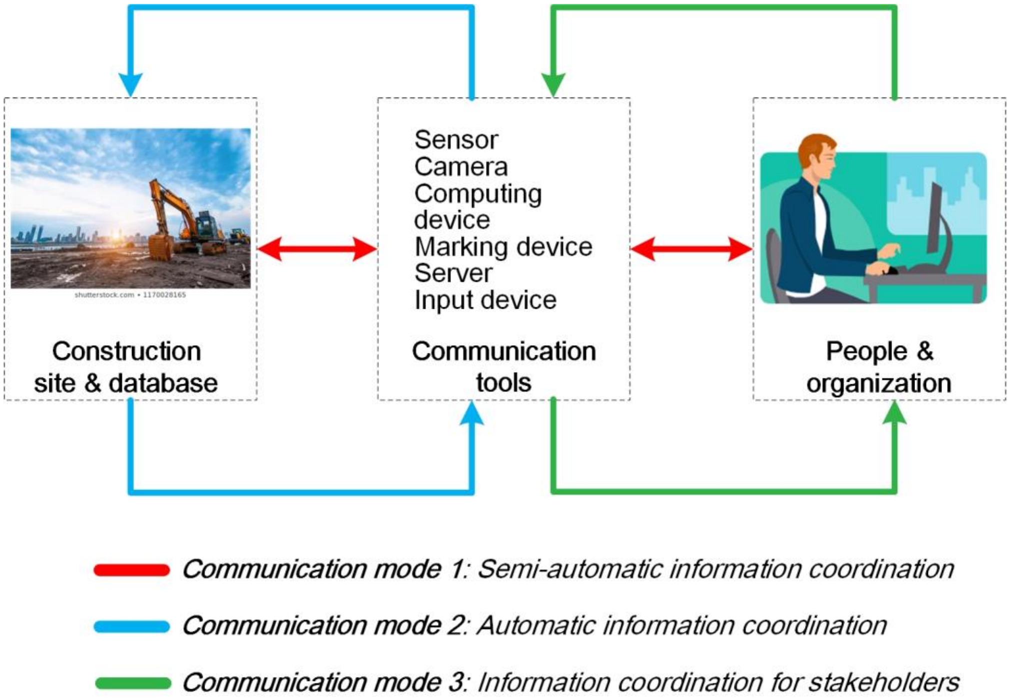

shutterstock.com • 1170028165
Construction site & database
Sensor
Camera
Computing device
Marking device
Server
Input device
Communication tools
People & organization
Communication mode 1: Semi-automatic information coordination
Communication mode 2: Automatic information coordination
Communication mode 3: Information coordination for stakeholders

Figure 8-3 The communication modes of ICT in construction

8.4.2 Methods for Classification

Intuitively, the communication-oriented entities, including the TIs, CMs, and CSs are more informative than common words and phrases in discerning the communication modes. For example, suppose that we are seeking to classify a patent that should be categorized into communication mode 3, the appearance of CSs, such as *requester* or *supplier*, and TIs such as *contract* or *command* may directly determine the classification. In this way, the common words and phrases may not provide sufficient information as CEs. Typically, it is a general wisdom that fewer informative features are sufficient for a classification task, and the other common features are noisy features that may harm the learning process for machine learning or deep learning models (Gabrilovich and Markovitch, 2004).

8.4.2.1 Basic Methods

Text processing and machine learning methods in this chapter are similar to the methods used in chapter 6, including data processing by NLP techniques, optimization and validation, and prediction. Totally, 88 of the 392 patents of ICT in construction were manually annotated with corresponding communication modes. Three common machine learning methods, Gaussian Naive Bayes (GNB), SVM and Bernoulli Naive Bayes (BNB), were used to learn the relations between the features and classifications from the annotated data.

8.4.2.2 Vectorizing with Recognized CEs

Feature selection plays a central role in text classification tasks. The aim is to select a collection of features that are informative to discern the different classes. Using recognized as features for further classifying documents has been implemented in several studies (Anđelić et al., 2017; Gui et al., 2012; Montalvo et al., 2007). Some of the studies used recognized entities exclusively as features for classification, others used the concatenations of common features and recognized entities. These studies reported that the recognized entities and integration of entities and common words could lead to better performance in classifying the documents into the classes that are highly related to the entities. To validate the performance of the classification based on CEs, this study trained two classifiers based on the features of common words and integration of common words and CEs respectively. The baseline feature extraction method of common words is similar to the method in chapter 6, using statistic methods to extract top k features from the n-grams and adopting Tf-Idf algorithm to vectorize the documents.

With respect to the CEs, this study obtained two vectors from common words and CEs respectively, using unigram and Tf-Idf algorithm. A combined vector that concatenates the two vectors is used to represent the patent document. The number of common words is set as a hyperparameter which would be tuned during the training process. The number of CEs is set as the size of the vocabulary of CEs due to the small number.

8.4.3 Empirical Validation

Similar to the validations in previous chapters, this study adopts K-fold cross-validation (K is set as 5 in this case) to validate the performance for the proposed feature extraction based on recognized CEs. To statistically estimate the performance, the validation for each training round would be repeated 10 times. For each training round, the repeated training and testing share the same instances that were randomly obtained in the k-fold instance generation process. Each repeated training process with a specific hyperparameter has precision, recall, and F-score values, and the average values of the 10 times training processes would be computed.

Figure 8-4 illustrates the validation comparison of two types of features over the three machine learning methods. In most of the instances, the concatenations of vectors of common words and CEs lead to better performance than exclusive vectors of common words.

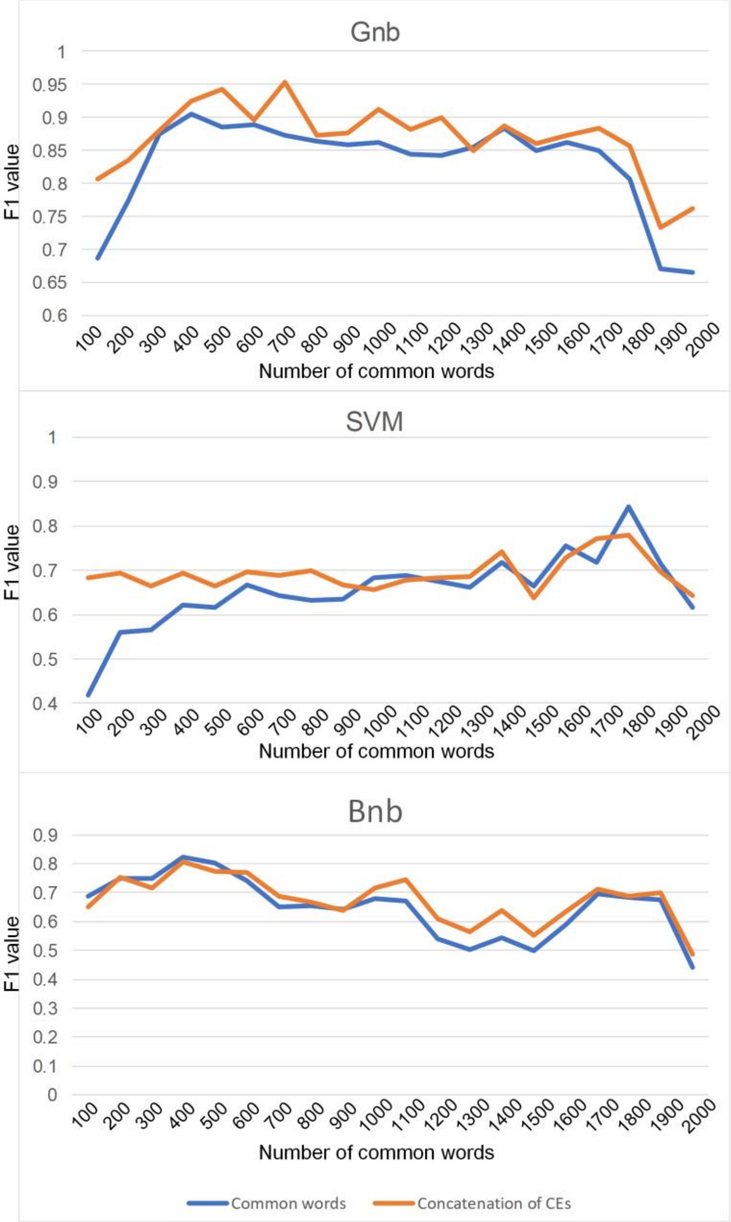


Figure 8-4 Performance comparison of two types of features over the three machine learning methods

Through the hyperparameter tuning process, the machine learning model and hyperparameters that lead to the best performance for each feature type is shown in Table 8.2. Using the proposed CEs as a part of the patent representing vectors outperformed the feature type of common words.

Table 8.2 Best machine learning model and hyperparameters for classification over common words and concatenation of CEs

Common words (Gnb+400 common words)			**Concatenation of CEs (Gnb+700 common words + 196 CEs)**		
Precision	Recall	F1	Precision	Recall	F1
0.9173	0.9058	0.905	0.9653	0.9529	0.9535

8.4.4 Classification Results and Discussion

Besides the annotated 88 patents, the classifications of the rest patents were predicted using the achieved best classifier. Therefore, the 392 patents of ICT in construction were categorized into the pre-defined communication modes. Table 8.3 indicates the number of patents classified into each communication mode.

Table 8.3 Total number of patents of ICT in construction for each communication mode

Communication name	Number of patents of ICT in construction
Mode 1: Semi-automatic information coordination	129
Mode 2: Automatic information coordination	164
Mode 3: Information coordination for stakeholders	99

Additionally, Table 8.4 illustrates the combinations of common TI and CM used in different communication modes. The selected common TIs comprise of (1) image data, including mentions of *'digital image', 'image information', 'photographic image', 'imagery', 'image or symbol' and 'image data'*; (2) geographic or location data, including mentions of *'geographic or location data', 'location information', 'component location', 'geographic location of the marking apparatus', 'location data point', 'geographic location', 'planning location', 'site location', 'asset location', 'position fixing coordinate location information',* *'geographical location', 'location of visual marker', 'location of data center device', 'location of bird', 'location identification information', 'latitude location', 'longitude location', 'location point', 'location antenna', 'location signal', 'dimensional location', 'geographical location of the', 'physical location', 'location of construction equipment', 'locational information', 'automatic location property', 'field location', 'geographic information', 'geographic location of the marking apparatus', 'geographic location', 'geographic coordinate information', 'geographical location', 'objective geographic information', 'geographic data', 'geographically referenced information', 'geographical data', 'magnetic field', 'magnetic field information', 'field data',* and *'field related data'*; (3) radio frequency data, including mentions of *'rf interrogation signal', 'rf signal', 'rf interference signal', 'rfid tag information', 'component', 'component location', 'component identifier', 'non structural building component', 'mep component', 'built component',* and *'designed room component'*; (4) payment, including mentions of *'payment',* and *'payment information'*; (5) project information, including mentions of *'project information', 'status of the project', 'construction project profile', 'project type information', 'construction project information', 'project design requirement', 'back projected image', 'project description', 'project status report', 'project task duration', 'projected imagery', 'construction project plan', 'construction project schedule', 'construction project budget', 'building project permit plan data', 'project drawing', 'project data file',* and *'project progress'*; (6) light or beam data, including mentions of *'light beam', 'laser beam', 'beam receiving position', 'beam', 'tracking laser beam', 'fan beam', 'shaped beam', 'light', 'light information', 'pulsed laser light', 'light frequency wavelength', 'light beam', 'flight plan',*

'ambient light level measurement', 'reflected laser light', 'laser plane of light', 'laser light energy', 'position of incoming laser light', 'structured light image', 'reflection light', 'laser light ray', 'light ray', 'laser light', 'visible light', and *'light signal'*; (7) position information, including mentions of *'dimensional position', 'position data', 'position information signal', 'clock and position information', 'vehicle position', 'scan position', 'real time positional information', 'position fixing coordinate location information', 'position information', 'shooting position', 'layout position', 'display position', 'target position', 'position of incoming laser light', 'global positioning system time signal', 'unit position', 'positioning signal', 'position marker', 'position and orientation',* and *'coordinate position'.*

The selected common CMs comprise of (1) camera device, including mentions of *'camera', 'digital camera', 'flight and camera controller', 'virtual camera', 'registration camera', 'format digital camera',* and *'video camera'* ; (2) display device, including mentions of *'display device', 'user interface', 'display', 'interface', 'graphical user interface', 'display screen', 'visual display', 'viewer', 'geographic information display control system', 'geographic data display controller', 'geographic display controller', 'graphic display', 'overview display',* and *'geographical overview display'*; (3) computing device, including mentions of *'computing device', 'computer system', 'computer implemented', 'computer processor', 'computing system',* and *'computer'*; (4) mobile device, including mentions of *'mobile device', 'mobile computing device', 'mobile evaluation vehicle', 'mobile electronic data processing apparatus', 'mobile', 'mobile client device', 'mobile client computing device', 'mobile interactive computer', 'mobile terminal',* and *'mobile hand held device'*; (5) sensor, including mentions of *'sensor', 'electromagnetic sensor',* and *'sensor unit'*; (6)

database, including mentions of *'database', 'environmental condition database', 'purchase order database', 'project database', 'contractor database', 'subcontractor database', 'bid database', 'asset information database', 'contract database',* and *'project schedule database'*; (7) laser device, including mentions of *'laser scanner', 'laser receiver', 'rotary laser irradiating system', 'laser light receiver', 'laser surveying instrument',* and *'semiconductor laser impinge'*; (8) server device, including mentions of *'server', 'construction project management server', 'enterprise server', 'enterprise based server', 'remote server system', 'web server', 'application server', 'remote computer server', 'bim server', 'enterprise resource planning erp server computer', 'management server', 'client server communication interface', 'task management server', 'server system', 'local server system', 'control server', 'server computer', 'remote management server', 'server application', 'server network device', 'server network', 'remote server',* and *'insurance provider remote server'*.

The most common data to be transferred by the ICTs in semi-automatic information coordination mode (mode 1) are image data and geographic or location data. In this communication mode, ICTs utilize information flows, such as “image data ↔ camera devices”, “image data ↔ display devices”, “image data ↔ mobile devices”, “geographic or location data ↔ display devices”, and “geographic or location data ↔ computing devices” to enable information coordination between construction sites and users. From Table 8.4, it can be seen that not all of the above information flows were exclusively used to enable semi-automatic communication. On one hand, information flows of “image data ↔ display devices” and “geographic or location data ↔ computing devices” both appear in mode 1 and mode 2, indicating that information flows may enable automatic communication from construction or semi-automatic. On the other hand, in transferring image data, the communication models "camera" and "mobile" only enable semi-automatic communication mode. The image data captured from the construction site always need confirmation of managers or engineers on the field (Ju et al., 2012).

The communication mode 2 incorporates the largest number of patents of ICT in construction, enabling automatic information flows, incorporating various communication patterns, such as *image or geographic or location data* displayed by *display devices*, *light or beam data* transferred by *laser devices*, *radio frequency data* transmitted through *databases*, and *position information* transferred by *sensors*. Some of the TIs are exclusively transferred in the automatic information coordination mode, such as beam data and magnetic field information. Some of the communication models are common apparatus that were utilized in different communication modes, such as computing devices and databases, while some tools are exclusively used to enable automatic information coordination, such as laser devices.

Prior studies indicate that the construction industry needs more ICTs that enable automatic information transmission (Alsafouri and Ayer, 2018). The results show the main information flows used in ICTs in construction enable communication mode 2. Prior research regards the laser devices, especially 3D laser devices need people to acquire data (Alsafouri and Ayer, 2018). But in data transmission, the results in this study reveals that ICTs follows an automatic manner when using the laser devices to transmit the light or beam data. Figure 8-5 plots the distribution of TI, CM, and CS in different communication modes. It can be seen that most of the ICTs in construction transfer the light or beam data automatically.

Table 8.4 Combinations of common TI and CM used in different communication modes

Transferred data or information (TI)							Communication models (CM)								Mentioned communication subjects (CS)	Number of patents		
image data	geographic or location data	radio frequency data	payment	project information	light or beam data	position information	camera devices	display device	computing device	mobile device	sensor	database	laser device	server device		Mode 1	Mode 2	Mode 3
√							√								system administrator	11		
√								√							user	7		
	√							√							user	5		
	√								√						client	4		
√										√					person	4		
					√								√				9	
√								√							user		7	
						√					√						6	
	√							√									6	
√											√						6	
						√							√				5	
		√										√					5	
			√											√	participant, user, payee, etc.			8
				√				√							user, bidder, employee, etc.			6
			√						√						developer, construction design office, etc.			4
				√										√	participant, user, payee, etc.			4

The communication mode 3, information coordination between stakeholders, have unique transferred information types, such as payment and project information. The communication models used in this communication mode are common tools, including server, computing and display devices. However, this communication mode involves various communication subjects (CS), such as participant, user, payee, user, bidder, employee, developer, and construction design office.

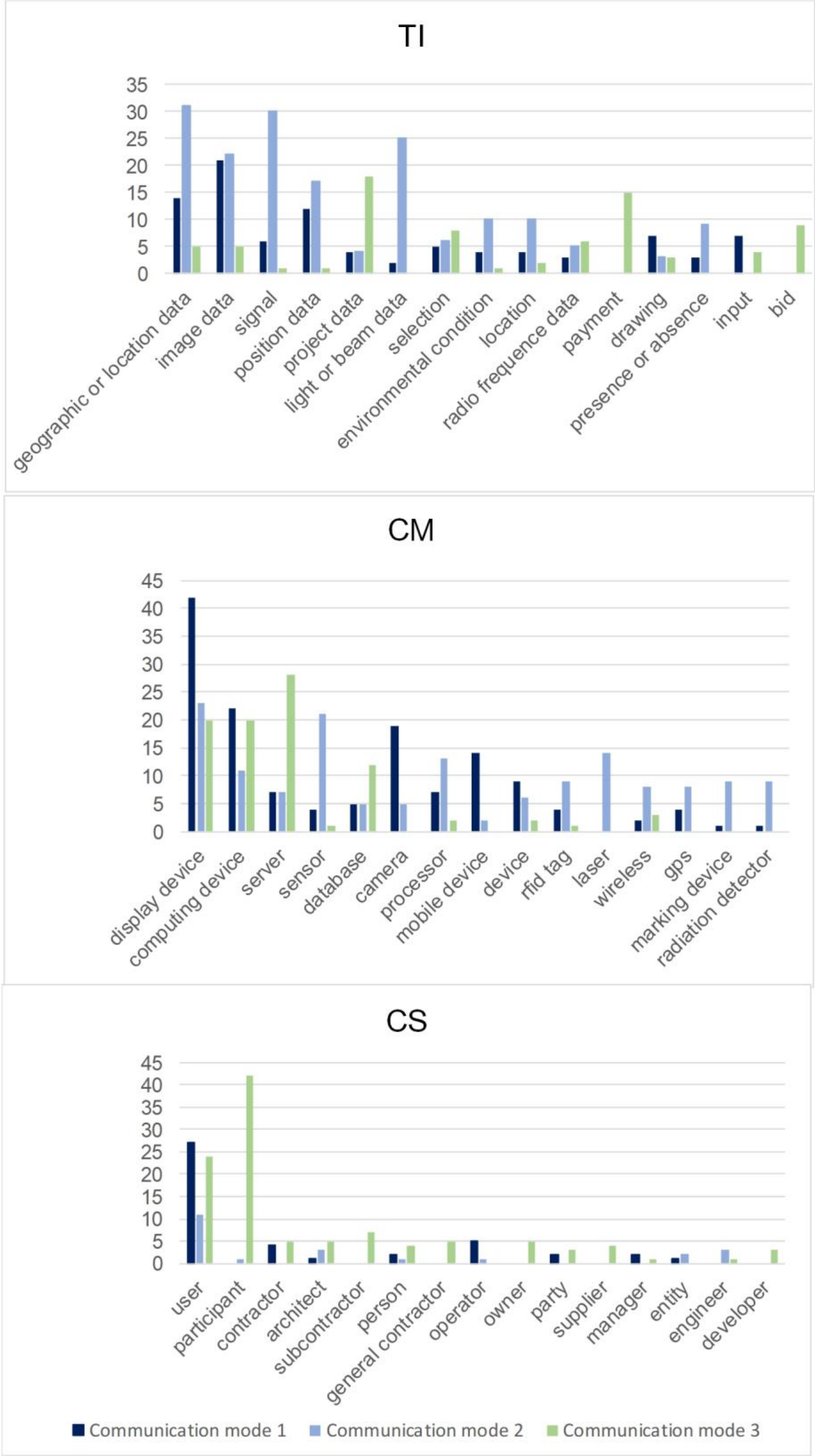


Figure 8-5 Distribution of TI, CM, and CS in different communication modes

8.5 Summary of this Chapter

This chapter firstly describes the potential applications in two perspectives: direct and indirect applications. Secondly, the chapter presents a specific application classifying the patents of ICT in construction into communication modes by using the recognized CEs. In this application, the recognized CEs were used as additional features in generating patent vectors, resulting in better performance in the classification task. The classified results indicate that the same data type and communication models generating similar information flows may enable different communication modes for ICT. In addition, in the construction industry, although most of the data types may be conveyed with or without human intervene, some data types such as light or beam data and presence or absence information were mostly transferred in automatic mode. The results also show that the ICTs in the mode of information coordination for stakeholders incorporate specific transferred information and communication models.

Chapter 9 Conclusion

9.1 Introduction

This chapter summarizes and brings together the major findings, as well as the limitations and recommendations for future studies. The research aim and objectives are first examined whether they were achieved. The main contributions to the body of knowledge are presented. Finally, the limitations associated with this study are discussed, and future research directions are proposed.

9.2 Review of Research Objectives

In the web 2.0 era, textual data is much easier to be stored in digital forms, leaving a considerable volume of unstructured data. The ICT in construction is not an exception, embracing a wide range of technical documents in which relevant knowledge was embodied. However, without effective diffusion, knowledge is no more than raw data. It is a general wisdom that employing deep learning models, as well as NLP techniques can make important assistance in converting these raw data into useful information. Therefore, the motivation of this study is to analyze these textual data in the technical documents of ICT in construction based on the deep learning model.

The primary aim of this study is to develop a deep learning model to recognize the CEs from the technical documents of ICT in construction. In this investigation, the specific objectives are as follows:

(1) To develop a deep learning model for entity recognition which can automatically

recognize CEs from raw text through utilizing the contextual information.

(2) To compile a patent database of ICT in construction and to acquire annotated data as training and testing instances for the deep learning model of communication-oriented entity recognition (CER).

(3) To train and validate the deep learning model and to make it tailored for CER, which achieves intelligence to recognize the CEs from technical documents of ICT in construction like human beings to understand the contextual meanings.

Chapter 2 and 3 review the ICT in construction and entity recognition to highlight the research gaps. To achieve objective 1, chapter 5 clarified basic definitions of CER of ICT in construction and highlighted the technical problems. A deep learning model based on the transformer was developed for entity recognition that is tailored to the recognition of CEs from technical documents of ICT in construction. To achieve objective 2, chapter 6 used an MLP model and NLP techniques to compile a patent database of ICT in construction. Based on the database, the annotated data for training and testing were achieved by manually labeling. To achieve objective 3, the deep learning model was validated in two perspectives: chapter 7 validated the accuracy of proposed TPF in recognizing the CEs, compared to the accuracy of the traditional RNN-based deep learning model; chapter 8 described the potential applications of the deep learning model for CER, and used one case of the applications to validate the practical value of recognized CEs by using the CEs as features to achieve a classification scheme that categorized the patents of ICT in construction into different communication modes.

9.3 Summary of Research Findings

The key findings are highlighted as follows:

(1) An automatic entity recognition approach was proposed based on a deep learning model. The developed model can recognize entities merely based on the input text, rather than pre-defined vocabularies (used to match entities in the raw text) and external lexical databases (used to discern ambiguous entities). The validation results indicate that the deep learning model can recognize unknown entities that cannot be recognized by previous entity recognition approaches.

(2) This study has compiled a patent database of ICT in construction. The current classification systems of patent offices do not provide a specific patent class for ICT in construction. Such a database not only provides training and testing data for the proposed model in this study but also supplements a collection of technical documents of ICT in construction for further studies.

(3) The deep learning model was developed based on the transformer, with various neural networks and NLP techniques and algorithms, yielding better performance than the BLC model which was reported as one of the most outperformed deep learning models for NLP tasks. The validation results show that the deep learning model outperforms in each training round, and the performance in terms of F-score, precision, and recall are at least 15% higher than the traditional model (RNN-based model).

(4) The validation over the ambiguous CEs (the same spellings may have different

meanings in different contexts) revealed that the deep learning model is better in addressing contextual information to predict CEs. The validation also indicated that the deep learning model has similar performance toward ambiguous and overall CEs, which exceeded the traditional model by at least 10%. The validation results also suggested that when a small percentage of a phrase's appearance as a CE, both models' performance decrease due to fewer opportunities to learn from true labels, but the proposed model remains the superior one, and the gap between the two models even increases.

(5) The classification results indicated that the recognized CEs for each patent of ICT in construction is more informative than common words when they were used as features for classifying the patents into different communication modes. The classification results revealed how ICT in construction uses different information flows to enable different communication modes.

9.4 Contributions of the Research

This study made original contributions to the body of knowledge in the following three aspects.

(1) The proposed deep learning model contributes an automatic entity recognition approach that is more effective than those in previous CEM studies. Entity recognition (sometimes called concept extraction, term identification or event extraction) has become an important approach in recent informatics studies of the CEM domain, aiming to automatically extract all the entities (sometimes called concepts, textual elements, textual items, or terminologies) from raw text that are relevant to a specific domain. Recently, a small number of studies

utilized rule-based and pre-defined vocabularies to enable automatic entity recognition to create digital dictionaries of the CEM domain. Those approaches, despite their convenience, suffered from two main drawbacks. First, these approaches require expertise knowledge and enormous efforts for the establishment of rules and vocabularies, which are extremely time-consuming and labor-intensive. In addition, pre-defined vocabularies exclude the unknown entities (the entities exist but are not known by the developer), resulting in the missing of numerous related entities. Second, those methods always lead to lower accuracy in discerning ambiguous entities (the spellings can appear as an entity at one position and common noun at another position, or appear as different entity types).

To remedy these problems, the proposed approach resorts to the techniques from the realm of deep learning that can utilize the contextual information within raw texts. In recent years, deep learning techniques have been recognized as a powerful tool to aid human beings in solving complex tasks to explore and utilize the unstructured text data in a robust way, in which the embedded information can be extracted in a readable manner. The key merit of deep learning is its capacity to address contextual information and generate contextual representations for each input token, rather than the external linguistic resources that are required in previous studies.

(2) This is the first time that a deep learning model has been developed with NLP techniques for representing information embedded in the raw text in the CEM domain. Deep learning techniques have been widely used in NLP tasks, fast becoming necessary tools to explore and utilize the unstructured text data in a robust way, in which the information in the unstructured data could be extracted and represented in a readable way.

In the CEM domain, deep learning models have been majorly developed to process images and video data. However, with respect to the textual data, deep learning models have rarely been developed regardless of the outstanding performance that has been achieved in NLP tasks.

(3) Practically, this study contributes an efficient and smart approach for practitioners in construction projects to perceive the communication functionality about how the up-to-date patents of ICT in construction employ and utilize devices to arrange information flows. Due to the fragment and knowledge-intensive nature of the construction industry, it is a challenging work for managers in construction projects to correctly adapt and properly implement ICT applications to enable and enhance communication. Traditionally, adoption and innovation of ICT highly depend on trial-and-error experiments or personal judgment to raise ideas for shaping the communication and conceive solutions for the problems in the communication process of construction projects. This overlooks the existing knowledge embedded in the written language in the technical documents (especially the patents) of ICT in construction (Tan, 2007).

Although a large amount of efforts have been devoted to developing advanced tools for patent analysis, these methods are not tailored for accessing and extracting key information of communication functionality of ICT in construction. Therefore, there is still a lack of a smart NLP approach to help practitioners link potential appropriate inventions of ICT in construction to their practical needs and to motivate creative thinking for innovative solutions.

The existing searching engines of the patent offices would retrieve a lot of irrelevant cases by simple queries and misses important cases by complex queries. Moreover, the retrieved patents are raw data that require a large amount of time and energy for users to understand the communication functionality . A few number approaches for patent analysis approaches (such as TRIZ) could process text in the patents and identify the problems and solutions. However, these approaches are developed for general analysis, without the focus on the communication functionality of ICT in construction.

Based on the compiled database of patents of ICT in construction and the recognized CEs, a communication-oriented retrieval system was formed to allow the users to access the patents of ICT in construction by querying CEs, and to extract the information indicating the communication functionality and recognize the basic entities. This retrieval system provides an efficient method to access key information of communication functionality embedded in the raw text. The retrieval system can help the practitioners efficiently connect the potential inventions of ICT in construction to the communication problems to be solved.

9.5 Limitations

This study is not without limitations.

First, the deep learning model could automatically identify and classify the CEs into pre-defined classes but cannot extract the relations between the recognized CEs. These relations might provide more knowledge to indicating the communication patterns embodied in the texts of ICT in construction.

Second, a pre-training process based on materials of ICT and Construction Engineering and Management might be taken into consideration for developing the transformer-based deep learning models. This study employed the pre-training parameters based on Wikipedia materials. These pre-trained parameters draw the contextual representations among a widely covered corpus, but the specific context of ICT in construction may be overlooked.

Third, more annotated data might be considered to verify if they can improve the CER performance of the deep learning model. The performance of the deep learning model is superior to BLC, but it seems that more annotated data might improve the overall performance.

Finally, more clarified annotated rules for tagging the raw data might improve the performance. Since the CER highly depends on the contextual information, it may confuse the annotators regardless of the specific annotate rules. Taking linguistic and syntax rules into consideration might reduce the confusion.

9.6 Future Research Recommendations

This study recommends future research in two directions. First is the application of NLP and deep learning techniques in other specific sub-fields of Construction Engineering and Management. The construction industry is characterized by its complex and fragmented nature. This needs advanced computer-aided tools to help practitioners to extract valuable knowledge from unstructured data. These sub-fields include but not exhaust safety engineering, lean construction, mega construction project, and prefabrication.

Second, with respect to ICT in construction, more advanced deep learning and NLP techniques may be adopted for other specific tasks, such as question answering, information retrieval, text summarization, sentiment analysis, and relation extraction. Furthermore, beyond contextual representations, multimodal representations are recommended because they can learn the dependencies between different types of data, such as between texts and images. This might provide complex knowledge by tagging text into images for ICT in construction, showing the communication patterns textually and visibly.

Appendix I: A Python Program for Crawling the Textual Data of Patents of ICT in Construction from USPTO

```
import re
from bs4 import BeautifulSoup
import requests
import os
import xlwt
import pandas as pd
global ori_dir, ori_dir1
ori_dir1 = 'D:/IN HONG KONG/patents/self_attention/'

ori_dir = 'D:/IN HONG KONG/patents/self_attention/data_1818_1/'

if not os.path.exists(ori_dir):
    os.makedirs(ori_dir)


def store_data(dictionary, Num):

    dictionary['patent_name'] = dictionary['patent_name'].replace('\n', ' ').replace('/', ' ').replace('"',
' ')[:100]

    # path of fields
    folder_name = ori_dir + dictionary['patent_name'] + '(' + dictionary['patent_code'] + ')'

    filename = folder_name + '.txt'
    data_list = ['patent_code', 'patent_name', 'year', 'inventor_and_country_data', 'patent_abstract',
'patent_claim', 'CPC_class', 'des_titlewith']
    with open(filename, 'w') as f:
        for data in data_list:
            f.write(data + ': \n' + dictionary[data] + '\n')

    f.close()
    print('No.' + str(Num) + ': ' + dictionary['patent_name'] + ' ' + dictionary['patent_code'] + ' has

been recorded' + '\n')
```

```
def fetch_data(url, Num):
    tmp_s = requests.session()

    headers = {
        'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8',
        'Accept-Encoding': 'gzip, deflate',
        'Accept-Language': 'zh-CN,zh;q=0.9',
        'Connection': 'keep-alive',
        'Host': 'patft.uspto.gov',
        'Upgrade-Insecure-Requests': '1',
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/69.0.3497.100 Safari/537.36',
    }

    try:

        r2 = tmp_s.get(url, headers=headers)
    except:
        print('There is something wrong with this patent')
    text2 = r2.text
    # print(text2)
    tmp_soup = BeautifulSoup(text2, "html.parser")
    patent_data = dict()
    # print(text2)
    # fetch patent_code
    patent_data['patent_code'] = tmp_soup.find('title').next[22:]
    # fetch patent_name
    patent_data['patent_name'] = tmp_soup.find('font', size="+1").text[:-1]

    # patent_data['inventor_and_country_data'] = tmp_soup.find_all('table',
width="100%")[2].contents

    # fetch patent_abstract
    tmp4 = text2[re.search('Abstract', text2).span()[1]:]
    patent_data['patent_abstract'] = tmp4[re.search('<p>', tmp4).span()[1]:(re.search('</p>',
tmp4).span()[0] - 1)].\
        replace('<BR><BR> ', ").replace("<BR><BR>", ").replace("<B><I>",
").replace("</I></B>", ").replace("\n      ", ' ')

    # fetch patent_claim
    tmp7 = text2[re.search('<CENTER><b><i>Claims</b></i></CENTER>', text2).span()[1]:]
    tmp8 = tmp7[re.search('1.', tmp7).span()[0]:]
    patent_data['patent_claim'] = tmp8[0:(re.search('<HR>', tmp8).span()[0] - 1)].\
        replace('<BR><BR> ', '').replace("<BR><BR>", '').replace("<B><I>", 
'').replace("</I></B>", '').replace("\n", ' ')
    # fetch year and inventor
    tmp1 = text2[re.search('BUF7=', text2).span()[1]:]
    patent_data['year'] = tmp1[:re.search('\n', tmp1).span()[0]]
    patent_data['inventor_and_country_data'] = tmp_soup.find_all('table', 
width="100%")[2].contents[1].text

    # fetch CPC class
    tmp3 = text2[re.search('Current CPC Class: ', text2).span()[1]:]
    patent_data['CPC_class'] = tmp3[re.search('width="70%"', 
tmp3).span()[1]:re.search('</TD></TR>', tmp3).span()[0]].\
        replace('&nbsp', ' ')
    # fetch other references
    patent_reference = tmp_soup.find('table', width="90%")
    if patent_reference != None:  # 'other reference' item may not exist
        patent_data['other_reference'] = patent_reference.text
    # fetch description

    tmp9 = tmp8[re.search('<CENTER><b><i>Description</b></i>', tmp8).span()[1]:]
    end_index = tmp9.index('<BR><BR><CENTER><b>* * * * *</b></CENTER>')
    tmp10 = tmp9[:end_index]

    sentences = tmp10.split('<BR><BR>')[1:]

    token_pattern = re.compile(r"(?u)\b\w\w+\b")

    a = 0
    des_titles = []
    title_indexs = []
    for line in sentences:

        a = a + 1
        if len(token_pattern.findall(line)) <= 20 and str.isupper(line) == True and 
str.isalpha(line.replace(' ', '').replace('\n', '')) == True:
            des_titles.append(line)
            title_indexs.append(sentences.index(line))
    print('\n\ndes_titles\n\n')
    print(des_titles)

    if des_titles != []:
        for a in range(len(des_titles)):
            if a != len(des_titles) - 1:

                for line in sentences[title_indexs[a] + 1:title_indexs[a + 1]]:
                    sentences[sentences.index(line)] = des_titles[a][:-1] + '<ITEM>' + line
            else:
                for line in sentences[title_indexs[a] + 1:]:
                    sentences[sentences.index(line)] = des_titles[a][:-1] + '<ITEM>' + line
    print('sentences\n\n')
    print(sentences)
    des_list = []
    for line in sentences:

        if line not in des_titles:
            des_list.append(line)
    print('\n\ndes_list\n\n')
    print(des_list)
    patent_data['des_titlewith'] = ''.join(des_list)

    store_data(patent_data, Num)


def main():
    url1 = 'http://patft.uspto.gov/netacgi/nph-
Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum
.htm&r=1&f=G&l=50&s1=7685013.PN.&OS=PN/7685013&RS=PN/7685013'
    fetch_data(url1, 2)

    code_file = ori_dir1 + 'NER_train_codes.csv'
    df = pd.read_csv(code_file, index_col=0)
    codelist1 = df['code'].values.tolist()
    df_1818 = pd.read_csv(ori_dir1 + '1818.csv', index_col=0)
    codelist2 = df_1818['codes'].values.tolist()

    print('main')
    for code in codelist2[423:]:
        if code in codelist1:
            pass
        else:

            patent_url                    =                    'http://patft.uspto.gov/netacgi/nph-

Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum

.htm&r=1&f=G&l=50&s1=' + str(code) + '.PN.&OS=PN/' + str(code) + '&RS=PN/' + str(code)
```

```
            print(patent_url)
            patent_dict = fetch_data(patent_url, codelist2.index(code))
            print(codelist2.index(code))
    df.loc[df['codes'] == int(patent_dict['patent_code']), 'patent_name'] =
patent_dict['patent_name']
    df.loc[df['codes'] == int(patent_dict['patent_code']), 'url3'] = patent_url
    df.loc[df['codes'] == int(patent_dict['patent_code']), 'year'] = patent_dict['year']
    df.loc[df['codes'] == int(patent_dict['patent_code']), 'CPC_class'] = patent_dict['CPC_class']
    df.loc[df['codes'] == int(patent_dict['patent_code']), 'inventor_and_country_data'] =
patent_dict['inventor_and_country_data']


if __name__ == '__main__':
    main()
data_list = ['patent_code', 'patent_name', 'year', 'inventor_and_country_data', 'patent_abstract',

'patent_claim', 'CPC_class']
```

Appendix II: Python Programs for Screening Patents of ICT in Construction

II-1 **Vectorizing the patents**

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.model_selection import train_test_split
from sklearn.externals import joblib
from sklearn.model_selection import train_test_split, KFold

import pandas as pd

NGRAM_RANGE = (1, 2)

TOKEN_MODE = 'word'

MIN_DOCUMENT_FREQUENCY = 2

common_words = ['element', 'wherein', 'model', 'give', 'item', 'key', 'data', 'device', 'function']
```

```
def ngram_vectorize_K(All_texts, All_label, Train_textlist, Test_textlist, TOP_K, DIR):
    TOP_K = TOP_K
    kwargs = {
        'input': 'filename',
        'ngram_range': NGRAM_RANGE,
        'dtype': 'int32',
        'strip_accents': 'unicode',
        'decode_error': 'replace',
        'analyzer': TOKEN_MODE,
        'min_df': MIN_DOCUMENT_FREQUENCY,
        'stop_words': common_words,
    }
    vectorizer = TfidfVectorizer(**kwargs)

    vectorizer.fit(All_texts)
    All_tf = vectorizer.transform(All_texts)
    joblib.dump(vectorizer, DIR + str(TOP_K) + "TF_IDF.m")
    selector = SelectKBest(f_classif, k=min(TOP_K, All_tf.shape[1]))
    selector.fit(All_tf, All_label)
    joblib.dump(selector, DIR + str(TOP_K) + "selector.m")
    All_sel = selector.transform(All_tf)
    x_train_veclist = []
    x_test_veclist = []
    for a in range(5):
        train_texts = Train_textlist[a]
        test_texts = Test_textlist[a]
        x_train = vectorizer.transform(train_texts)
        x_train = selector.transform(x_train).astype('float32')
        x_train_veclist.append(x_train)
        x_test = vectorizer.transform(test_texts)
        x_test = selector.transform(x_test).astype('float32')
        x_test_veclist.append(x_test)

    return x_train_veclist, x_test_veclist, All_sel


def test_vectorize_K(All_texts, TOP_K):
    TOP_K = TOP_K
    kwargs = {
        'input': 'filename',
        'ngram_range': NGRAM_RANGE,  # Use 1-grams + 2-grams.
        'dtype': 'int32',
        'strip_accents': 'unicode',
        'decode_error': 'replace',
        'analyzer': TOKEN_MODE,  # Split text into word tokens
        'min_df': MIN_DOCUMENT_FREQUENCY,
        'stop_words': common_words,
    }
    vectorizer = TfidfVectorizer(**kwargs)
    vectorizer.fit(All_texts)
    All_tf = vectorizer.transform(All_texts)
    selector = SelectKBest(f_classif, k=min(TOP_K, All_tf.shape[1]))
    selector.fit(All_tf, All_label)
    x_train_veclist = []
    x_test_veclist = []
    for a in range(5):
        train_texts = Train_textlist[a]
        test_texts = Test_textlist[a]
        x_train = vectorizer.transform(train_texts)
        x_train = selector.transform(x_train).astype('float32')
        x_train_veclist.append(x_train)
        x_test = vectorizer.transform(test_texts)
        x_test = selector.transform(x_test).astype('float32')
        x_test_veclist.append(x_test)

    return x_train_veclist, x_test_veclist


def main():

    data_dir = 'D:/IN HONG KONG/patents/LML/ICT/12.1.2018/'

    label_dir = data_dir + 'label_filelist.csv'
    df = pd.read_csv(label_dir, index_col=0)
    remain_label = 'remain' + str(2) + 'list'
    all_texts = df[remain_label].values
    all_labels = df['Class'].values
    X_trainlist = []
    x_testlist = []
    y_trainlist = []
    y_testlist = []
    kf = KFold(5, shuffle=True)
    a = 0
    for train_index, test_index in kf.split(all_texts):
        a = a + 1
        x_traintext, x_testtext, y_train, y_test = all_texts[train_index], all_texts[test_index],
all_labels[train_index], all_labels[test_index]
        X_trainlist.append(x_traintext)
        x_testlist.append(x_testtext)
        y_trainlist.append(y_train)
        y_testlist.append(y_test)
        ngram_vectorize_K(all_texts, all_labels, X_trainlist, x_testlist, 5000)


if __name__ == '__main__':
    main()
```

II-2 **Main program of MLP training**

```
import pandas as pd
import argparse
import time
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB
from sklearn.preprocessing import FunctionTransformer
import tensorflow as tf
import numpy as np
import B01_explore_data
import B02_vectors
import B03_build_model
import re
from sklearn.externals import joblib

import logging
from time import gmtime, strftime

FLAGS = None



def Vectorize_split(DF,f_num,DIR):
    df = DF
    remain_label = 'dir'
    all_texts = df[remain_label].values.tolist()
    all_labels = df['Class']
    x_traintext, x_testtext, y_train, y_test = train_test_split(all_texts, all_labels, test_size=0.2)
```

```
y_train_ori = y_train
    y_test_ori = y_test

    num_classes = B01_explore_data.get_num_classes(all_labels)
    print(num_classes)
    x_trainvec, x_testvec,All_tf = B02_vectors.ngram_vectorize_split(all_texts, all_labels,
x_traintext, x_testtext, f_num,DIR)
    return y_train, y_test, x_trainvec, x_testvec, All_tf, all_labels

def train_allmodel_split( X_train,
                          X_test,
                          Y_train,
                          Y_test,
                          units,
                          f_num,
                          Save_dir,
                          All_sel,
                          all_labels,
                          learning_rate=1e-3,
                          epochs=1000,
                          batch_size=128,
                          layers=2,
                          dropout_rate=0.2):
    num_classes =2
    x_trainvec = X_train
    x_testvec = X_test
    y_train = Y_train
    y_test = Y_test
    gnb = GaussianNB(var_smoothing=1e-3)
    svclassifier = SVC(kernel='linear')
    Bnb = BernoulliNB()


    c = 0
    precision = []

    model = B03_build_model.mlp_model_vunits(layers=layers,
                                             units=units,
                                             dropout_rate=dropout_rate,
                                             input_shape=x_trainvec.shape[1:],
                                             num_classes=2)
```

```
y_predall_gnb = y_pred_gnb
y_predall_SVM = y_pred_SVM
y_predall_Bnb = y_pred_Bnb

MLP_name = str(f_num)+ str(units[0]) + str(units[1])

model.save(Save_dir + MLP_name)


df_Y = pd.DataFrame(y_testall, columns=['Class'])
df_Y.reset_index(drop=True, inplace=True)
df_MLP = pd.DataFrame(y_predall_DP, columns=['P_MLP'])
df_gnb = pd.DataFrame(y_predall_gnb, columns=['P_gnb'])
df_SVM = pd.DataFrame(y_predall_SVM, columns=['P_SVM'])
df_Bnb = pd.DataFrame(y_predall_Bnb, columns=['P_Bnb'])

result = pd.concat([df_Y,df_MLP,df_gnb,df_SVM,df_Bnb], axis=1)
return result
```

Appendix III: A Python Program for Training the Deep Learning Model

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import collections
import os
import modeling
import transformer_optm
import transformer_tokenization
import tensorflow as tf
from sklearn.metrics import f1_score,precision_score,recall_score
from tensorflow.python.ops import math_ops
import tf_metrics
import pickle
from params import params1

def main(_):
    tf.logging.set_verbosity(tf.logging.INFO)
    processors = {
        "ner": Process_CER
    }
    if not params1.do_train and not params1.do_eval:
        raise ValueError("Some thing wrong")

    CER_initial = modeling.BertConfig.from_json_file(params1.bert_config_file)

    task_name = params1.task_name.lower()

    processor = processors[task_name]()

    label_list = processor.Obtain_lbs()

    tokenizer = transformer_tokenization.FullTokenizer(
        vocab_file=params1.vocab_file, do_lower_case=params1.do_lower_case)
    tpu_cluster_resolver = None
```

```
if params1.use_tpu and params1.tpu_name:
    tpu_cluster_resolver = tf.contrib.cluster_resolver.TPUClusterResolver(
      params1.tpu_name, zone=params1.tpu_zone, project=params1.gcp_project)

  is_per_host = tf.contrib.tpu.InputPipelineConfig.PER_HOST_V2

  run_config = tf.contrib.tpu.RunConfig(
    cluster=tpu_cluster_resolver,
    master=params1.master,
    model_dir=params1.output_dir,
    save_checkpoints_steps=params1.save_checkpoints_steps,
    tpu_config=tf.contrib.tpu.TPUConfig(
      iterations_per_loop=params1.iterations_per_loop,
      num_shards=params1.num_tpu_cores,
      per_host_input_for_training=is_per_host))

  train_examples = None
  num_train_steps = None
  num_warmup_steps = None

  if params1.do_train:
    train_examples = processor.Obtain_instances_for_training(params1.data_dir)
    num_train_steps = int(
      len(train_examples) / params1.train_batch_size * params1.num_train_epochs)
    num_warmup_steps = int(num_train_steps * params1.warmup_proportion)

  model_fn = model_fn_builder(
    CER_initial=CER_initial,
    num_labels=len(label_list)+1,
    init_checkpoint=params1.init_checkpoint,
    learning_rate=params1.learning_rate,
    num_train_steps=num_train_steps,
    num_warmup_steps=num_warmup_steps,
    use_tpu=params1.use_tpu,
    use_one_hot_embeddings=params1.use_tpu)

  estimator = tf.contrib.tpu.TPUEstimator(
    use_tpu=params1.use_tpu,
    model_fn=model_fn,
    config=run_config,
    train_batch_size=params1.train_batch_size,
    eval_batch_size=params1.eval_batch_size,
    predict_batch_size=params1.predict_batch_size)
```

```
if params1.do_train:
    train_file = os.path.join(params1.output_dir, "train.tf_record")
    filed_based_convert_examples_to_features(
      train_examples, label_list, params1.max_seq_length, tokenizer, train_file)
    tf.logging.info("***** Running training *****")
    tf.logging.info("  Num examples = %d", len(train_examples))
    tf.logging.info("  Batch size = %d", params1.train_batch_size)
    tf.logging.info("  Num steps = %d", num_train_steps)
    train_input_fn = file_based_input_fn_builder(
      input_file=train_file,
      seq_length=params1.max_seq_length,
      is_training=True,
      drop_remainder=True)
    estimator.train(input_fn=train_input_fn, max_steps=num_train_steps)
  if params1.do_eval:
    eval_examples = processor.Obtain_instances_for_example(params1.data_dir)
    eval_file = os.path.join(params1.output_dir, "eval.tf_record")
    filed_based_convert_examples_to_features(
      eval_examples, label_list, params1.max_seq_length, tokenizer, eval_file)

    tf.logging.info("***** Running evaluation *****")
    tf.logging.info("  Num examples = %d", len(eval_examples))
    tf.logging.info("  Batch size = %d", params1.eval_batch_size)
    eval_steps = None
    if params1.use_tpu:
      eval_steps = int(len(eval_examples) / params1.eval_batch_size)
    eval_drop_remainder = True if params1.use_tpu else False
    eval_input_fn = file_based_input_fn_builder(
      input_file=eval_file,
      seq_length=params1.max_seq_length,
      is_training=False,
      drop_remainder=eval_drop_remainder)
    result = estimator.evaluate(input_fn=eval_input_fn, steps=eval_steps)
    output_eval_file = os.path.join(params1.output_dir, "eval_results.txt")
    with open(output_eval_file, "w") as writer:
      tf.logging.info("***** Eval results *****")
      for key in sorted(result.keys()):
        tf.logging.info("  %s = %s", key, str(result[key]))
        writer.write("%s = %s\n" % (key, str(result[key])))
  if params1.do_predict:
    token_path = os.path.join(params1.output_dir, "token_test.txt")
    with open('./output/label2id.pkl','rb') as rf:
      label2id = pickle.load(rf)
```

```
            id2label = {value:key for key,value in label2id.items()}
        if os.path.exists(token_path):
            os.remove(token_path)
        predict_examples = processor.get_test_examples(params1.data_dir)

        predict_file = os.path.join(params1.output_dir, "predict.tf_record")
        filed_based_convert_examples_to_features(predict_examples, label_list,
                                                 params1.max_seq_length, tokenizer,
                                                 predict_file,mode="test")

        tf.logging.info("***** Running prediction*****")
        tf.logging.info("  Num examples = %d", len(predict_examples))
        tf.logging.info("  Batch size = %d", params1.predict_batch_size)

        predict_drop_remainder = True if params1.use_tpu else False
        predict_input_fn = file_based_input_fn_builder(
            input_file=predict_file,
            seq_length=params1.max_seq_length,
            is_training=False,
            drop_remainder=predict_drop_remainder)

        result = estimator.predict(input_fn=predict_input_fn)
        output_predict_file = os.path.join(params1.output_dir, "label_test.txt")
        with open(output_predict_file,'w') as writer:
            for prediction in result:
                output_line = "\n".join(id2label[id] for id in prediction if id!=0) + "\n"
                writer.write(output_line)

def update_ner(p):
    p.data_dir = 'D:/IN HONG KONG/patents/self_data/NERdata/'
    p.task_name ="NER"
    return p

if __name__ == "__main__":
    Bert_dir = 'D:/IN HONG KONG/patents/self_attention/uncased_L-12_H-768_A-12'
    params1 = params1()
    params1.output_dir = 'D:/IN HONG KONG/patents/self_attention/output4/'
    params1.update_Bert(Bert_dir)
    params1 = update_ner(params1)
    print('\n'.join(['%s:%s' % item for item in params1.__dict__.items()]))

    tf.app.run()
```

References

Aggarwal, C.C., Reddy, C.K., 2013. Data clustering: algorithms and applications. *CRC press*.

Agrawal, A., Henderson, R., 2002. Putting patents in context: Exploring knowledge transfer from MIT. *Management Science*, 48(1), 44-60.

Ahuja, V., Yang, J., Shankar, R., 2009. Study of ICT adoption for building project management in the Indian construction industry. *Automation in Construction*, 18(4), 415-423.

Akarowhe, K., 2017. Information Communication Technology (Ict) in the Educational System of the Third World Countries as a Pivotal to Meet Global Best Practice in Teaching and Development. *Am J Compt Sci Inform Technol*, 5(2).

Al Qady, M., Kandil, A., 2014. Automatic clustering of construction project documents based on textual similarity. *Automation in Construction*, 42, 36-49.

Al Rahhal, M.M., Bazi, Y., Al Zuair, M., Othman, E., BenJdira, B., 2018. Convolutional Neural Networks for Electrocardiogram Classification. *Journal of Medical and Biological Engineering*, 38(6), 1014-1025.

Alsafouri, S., Ayer, S.K., 2018. Review of ICT Implementations for Facilitating Information Flow between Virtual Models and Construction Project Sites. *Automation in Construction*, 86(August 2016), 176-189.

Anagnostou, K., Vlamos, P., 2011. Square AR: Using Augmented Reality for Urban Planning. *Third International Conference on Games & Virtual Worlds for Serious Applications*.

Anđelić, S., Kondić, M., Perić, I., Jocić, M., Kovačević, A., 2017. Text Classification Based on Named Entities. *International Conference on Information Society and Technology*.

Aouad, G., Wu, S., Lee, A., 2009. nD Modelling, Present and Future.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J.F., Iglesias, C.A., 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77, 236-246.

Aronson, A.R., 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings Annual Symposium*, 2001(1), 17-21.

Baek, F., Ha, I., Kim, H., 2019. Augmented reality system for facility management using image-based indoor localization. *Automation in Construction*, 99, 18-26.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bekoulis, G., Deleu, J., Demeester, T., Develder, C., 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114, 34-45.

Bell, G., Hey, T., Szalay, A., 2009. Beyond the data deluge. *Science*, 323(5919), 1297-1298.

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb), 1137-1155.

Benson, C.L., Magee, C.L., 2013. A hybrid keyword and patent class methodology for selecting relevant sets of patents for a technological field. *Scientometrics*, 96(1), 69-82.

Bernardini, F., Rushmeier, H., 2002. The 3D model acquisition pipeline. *Computer graphics forum*, 21(2), 149-172.

Bertram, J., Mandl, T., 2017. Ambiguity in patent vocabulary: Experiments with clarity scores for claims and descriptions. *International Conference on Knowledge & Smart Technology*.

Bick, E., 2004. A Named Entity Recognizer for Danish. *Conference on Language Resources and Evaluation*, 305-308.

Bird, S., Loper, E., 2004. NLTK: the natural language toolkit. *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 63-70.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993-1022.

Bosché, F., Abdel-Wahab, M., Carozza, L., 2016. Towards a Mixed Reality System for Construction Trade Training. *Journal of Computing in Civil Engineering*, 30(2), 04015016.

Bosché, F., Guenet, E., 2014. Automating surface flatness control using terrestrial laser scanning and building information models. *Automation in Construction*, 44, 212-226.

Bouadjenek, M.R., Sanner, S., Ferraro, G., 2015. A Study of Query Reformulation for Patent Prior Art Search with Partial Patent Applications. *International Conference on Artificial Intelligence & Law*.

Brown, I., Mues, C., 2012. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3), 3446-3453.

Cakir, L., Yilmaz, N., 2014. Polynomials, radial basis functions and multilayer perceptron neural network methods in local geoid determination with GPS/levelling. *Measurement*, 57, 148-153.

Camacho-Collados, J., Pilehvar, M.T., Navigli, R., 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240, 36-64.

Cambria, E., White, B., 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2), 48-57.

Cassetta, E., Marra, A., Pozzi, C., Antonelli, P., 2017. Emerging technological trajectories and new mobility solutions. A large-scale investigation on transport-related innovative start-ups and implications for policy. *Transportation Research Part A: Policy and Practice*, 106(March), 1-11.

Chakrabarti, S., Dom, B., Indyk, P., 1998. Enhanced hypertext categorization using hyperlinks. *SIGMOD Rec.*, 27(2), 307-318.

Chan, A.P.C., Chan, D.W.M., Chiang, Y.H., Tang, B.S., Chan, E.H.W., Ho, K.S.K., 2004. Exploring critical success factors for partnering in construction projects. *Journal of*

Construction Engineering and Management-Asce, 130(2), 188-198.

Chen, Q., Zhuo, Z., Wang, W., 2019. BERT for Joint Intent Classification and Slot Filling. *arXiv preprint arXiv:1902.10909*.

Cheng Eddie, W.L., Li, H., Love Peter, E.D., Irani, Z., 2001. Network communication in the construction industry. *Corporate Communications: An International Journal*, 6(2), 61-70.

Cho, C.-Y., Kwon, S., Shin, T.-H., Chin, S., Kim, Y.-S., 2011. A development of next generation intelligent construction liftcar toolkit for vertical material movement management. *Automation in Construction*, 20(1), 14-27.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, S., Kang, D., Lim, J., Kim, K., 2012. A fact-oriented ontological approach to SAO-based function modeling of patents for implementing Function-based Technology Database. *Expert Systems with Applications*, 39(10), 9129-9140.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug), 2493-2537.

Costin, A., Pradhananga, N., Teizer, J., 2012. Leveraging passive RFID technology for construction resource field mobility and status monitoring in a high-rise renovation project. *Automation in Construction*, 24, 1-15.

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., Leite, F., 2018. 6D DBSCAN-based segmentation of building point clouds for planar object classification. *Automation in Construction*, 88, 44-58.

Dai, A.M., Le, Q.V., 2015. Semi-supervised sequence learning. *Proceedings of the 29th International Conference on Neural Information Processing Systems*, 3079-3087.

Dainty, A., Moore, D., Murray, M., 2007. Communication in construction: Theory and practice. *Routledge*.

Deng, Y., Cheng, J.C.P., Anumba, C., 2016. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67, 1-21.

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

Dhingra, B., Liu, H., Salakhutdinov, R., Cohen, W.W., 2017. A comparative study of word embeddings for reading comprehension. *arXiv preprint arXiv:1703.00993*.

Dong, L.Y., Ji, S.J., Zhang, C.J., Zhang, Q., Chiu, D.W., Qiu, L.Q., Li, D., 2018. An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Systems with Applications*, 114, 210-223.

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul), 2121-2159.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N.A., 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint* *arXiv:1505.08075*.

Eftimov, T., Seljak, B.K., Korosec, P., 2017. A rule-based named-entity recognition method for knowledge extraction of evidence based dietary recommendations. *PLoS ONE*, 12(6).

El-Diraby, T., Lima, C., Feis, B., 2005. Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge. *Journal of Computing in Civil Engineering*, 19(4), 394-406.

El-Diraby, T.E., Kashif, K.F., 2005. Distributed ontology architecture for knowledge management in highway construction. *Journal of Construction Engineering and Management*, 131(5), 591-603.

El-Ghandour, W., Al-Hussein, M., 2004. Survey of information technology applications in construction. *Construction innovation*, 4(2), 83-98.

El Ghazali, Y., Lefebvre, É., Lefebvre, L.A., 2012. The potential of RFID as an enabler of knowledge management and collaboration for the procurement cycle in the construction industry. *Journal of technology management & innovation*, 7(4), 81-102.

Ergen, E., Akinci, B., Sacks, R., 2007. Tracking and locating components in a precast storage yard utilizing radio frequency identification technology and GPS. *Automation in Construction*, 16(3), 354-367.

Fan, H.Q., Xue, F., Li, H., 2015a. Project-Based As-Needed Information Retrieval from Unstructured AEC Documents. *Journal of Management in Engineering*, 31(1).

Fan, X., Li, S., Tian, L., 2015b. Chaotic characteristic identification for carbon price and an multi-layer perceptron network prediction model. *Expert Systems with Applications*, 42(8), 3945-3952.

Finch, E., 2000. Net gain in construction: Using the Internet in the construction industry. *Butterworth-Heinemann*.

Flyvbjerg, B., 2014. What You Should Know About Megaprojects and Why: An Overview. *Project Management Journal*, 45(2), 6-19.

Forman, G., 2002. Choose Your Words Carefully: An Empirical Study of Feature Selection Metrics for Text Classification. 150-162.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. *Springer series in statistics New York*, New York, USA.

Fu, X., Liu, W., Xu, Y., Cui, L., 2017. Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis. *Neurocomputing*, 241, 18-27.

Gabrilovich, E., Markovitch, S., 2004. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5. *Proceedings of the twenty-first international conference on Machine learning*, 41.

Gaizauskas, R., Humphreys, K., Cunningham, H., Wilks, Y., 1995. University of Sheffield: description of the LaSIE system as used for MUC-6. *Conference on Message Understanding*, 207-220.

Gao, G., Liu, Y.S., Wang, M., Gu, M., Yong, J.H., 2015. A query expansion method for retrieving online BIM resources based on Industry Foundation Classes. *Automation in Construction*, 56, 14-25.

García Adeva, J.J., Pikatza Atxa, J.M., Ubeda Carrillo, M., Ansuategi Zengotitabengoa, E.,

2014. Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, 41(4), 1498-1508.

Garcia-Laencina, P.J., Sancho-Gomez, J.L., Figueiras-Vidal, A.R., 2013. Classifying patterns with missing values using Multi-Task Learning perceptrons. *Expert Systems with Applications*, 40(4), 1333-1341.

Gerken, J.M., 2012. A new instrument for technology monitoring: novelty in patents measured by semantic patent analysis. *Scientometrics*, 91(3), 645-670.

Giachanou, A., Salampasis, M., Paltoglou, G., 2015. Multilayer source selection as a tool for supporting patent search and classification. *Information Retrieval Journal*, 18(6), 559-585.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A., 2011. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. *Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers*, 42-47.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. *MIT press Cambridge*.

Goyal, A., Gupta, V., Kumar, M., 2018. Recent Named Entity Recognition and Classification techniques: A systematic review. *Computer Science Review*, 29, 21-43.

Graves, A., 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gredel, D., Kramer, M., Bend, B., 2012. Patent-based investment funds as innovation intermediaries for SMEs: In-depth analysis of reciprocal interactions, motives and fallacies. *Technovation*, 32(9), 536-549.

Greiman, V.A., 2013. Megaproject management: Lessons on risk and project management from the Big Dig. *John Wiley & Sons*.

Grishman, R., Sundheim, B., 1996a. Message understanding conference-6: A brief history. *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1.

Grishman, R., Sundheim, B., 1996b. Message Understanding Conference 6: A Brief History. *Conference on Computational Linguistics*, 466-471.

Guan, Y., Wei, Q., Chen, G.Q., 2019. Deep learning based personalized recommendation with multi-view information integration. *Decision Support Systems*, 118, 58-69.

Gui, Y., Gao, Z., Li, R., Yang, X., 2012. Hierarchical text classification for news articles based-on named entities. *International Conference on Advanced Data Mining and Applications*, 318-329.

Gwak, J.H., Sohn, S.Y., 2018. A novel approach to explore patent development paths for subfield technologies. *Journal of the Association for Information Science and Technology*, 69(3), 410-419.

Habash, N., Rambow, O., Roth, R., 2009. MADA+ TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR)*, 41, 62.

Haddi, E., Liu, X., Shi, Y., 2013. The role of text pre-processing in sentiment analysis.

Procedia Computer Science, 17, 26-32.

Haykin, S., 1999. Neural networks a comprehensive introduction. Prentice Hall, New Jersey.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

Heinzerling, B., Strube, M., 2018. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2989-2993.

Hemati, W., Mehler, A., 2019. LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools. *Journal of Cheminformatics*, 11.

Hillier, S.P., 2007. Microsoft SharePoint: Building Office 2007 Solutions in VB 2005. *Apress*.

Hindle, D., 1989. Acquiring disambiguation rules from text. *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, 118-125.

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

Hofer, M., Kormilitzin, A., Goldberg, P., Nevado-Holgado, A., 2018. Few-shot Learning for Named Entity Recognition in Medical Text. *arXiv preprint arXiv:1811.05468*.

Hosseini, R., Chileshe, N., Zou, J., Baroudi, B., 2013. Approaches of implementing ICT technologies within the construction industry. *Australasian Journal of Construction Economics and Building-Conference Series*, 1(2), 1-12.

Hou, L., Chi, H.-L., Tarng, W., Chai, J., Panuwatwanich, K., Wang, X., 2017. A framework of innovative learning for skill development in complex operational tasks. *Automation in Construction*, 83, 29-40.

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

Irizarry, J., Karan, E.P., Jalaei, F., 2013. Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction*, 31, 241-254.

Ittoo, A., Nguyen, L.M., van den Bosch, A., 2016. Text analytics in industry: Challenges, desiderata and trends. *Computers in Industry*, 78, 96-107.

Jain, A., Kulkarni, G., Shah, V., 2018. Natural Language Processing. *International Journal of Computer Sciences and Engineering*, 6(1), 161-167.

Ju, Y., Kim, C., Kim, H., 2012. RFID and CCTV-Based Material Delivery Monitoring for Cable-Stayed Bridge Construction. *Journal of Computing in Civil Engineering*, 26(2), 183-190.

Kalchbrenner, N., Grefenstette, E., Blunsom, P., 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kaplan, S., Vakili, K., 2013. Novelty vs. usefulness in innovative breakthroughs: A test using topic modeling of nanotechnology patents, Technical Report, Working Paper.

Karan, E.P., Irizarry, J., 2015. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Automation in Construction*, 53, 1-12.

Karatzoglou, A., Feinerer, I., 2010. Kernel-based machine learning for fast text mining in R. *Computational Statistics & Data Analysis*, 54(2), 290-297.

Kim, M.-K., Wang, Q., Park, J.-W., Cheng, J.C.P., Sohn, H., Chang, C.-C., 2016a. Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and BIM. *Automation in Construction*, 72, 102-114.

Kim, M., Park, Y., Yoon, J., 2016b. Generating patent development maps for technology monitoring using semantic patent-topic analysis. *Computers and Industrial Engineering*, 98, 289-299.

Kim, M.K., Cheng, J.C.P., Sohn, H., Chang, C.C., 2015. A framework for dimensional and surface quality assessment of precast concrete elements using BIM and 3D laser scanning. *Automation in Construction*, 49, 225-238.

Kim, S.A., Shin, D., Choe, Y., Seibert, T., Walz, S.P., 2012. Integrated energy monitoring and visualization system for Smart Green City development: Designing a spatial information integrated energy monitoring model in the context of massive data management on a web based platform. *Automation in Construction*, 22, 51-59.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. *International Conference for Learning Representations*.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai*, 14(2), 1137-1145.

Korvigo, I., Holmatov, M., Zaikovskii, A., Skoblov, M., 2018. Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules. *Journal of Cheminformatics*, 10(1), 28.

Kratzwald, B., Ilic, S., Kraus, M., Feuerriegel, S., Prendinger, H., 2018. Deep learning for affective computing: Text-based emotion recognition in decision support. *Decision Support Systems*, 115, 24-35.

Kumar, N.S., Muruganantham, D., 2016. Disambiguating the Twitter Stream Entities and Enhancing the Search Operation Using DBpedia Ontology: Named Entity Disambiguation for Twitter Streams. *International Journal of Information Technology and Web Engineering*, 11(2), 51-62.

Kurdi, M.Z., 2017. Natural language processing and computational linguistics 2: semantics, discourse and applications. *John Wiley & Sons*.

Lafferty, J., McCallum, A., Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lam Patrick, T.I., Wong Franky, W.H., Tse Kenny, T.C., 2010. Effectiveness of ICT for Construction Information Exchange among Multidisciplinary Project Teams. *Journal of Computing in Civil Engineering*, 24(4), 365-376.

Lam, P.T., Wong, F.W., Tse, K.T., 2009. Effectiveness of ICT for construction information exchange among multidisciplinary project teams. *Journal of Computing in Civil Engineering*, 24(4), 365-376.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Le, T.Y., Jeong, H.D., 2017. NLP-Based Approach to Semantic Classification of Heterogeneous Transportation Asset Data Terminology. *Journal of Computing in Civil Engineering*, 31(6).

Lee, C., Song, B., Park, Y., 2013. How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships. *Technology Analysis and Strategic Management*, 25(1), 23-38.

Levitt, R.E., 2007. CEM research for the next 50 years: Maximizing economic, environmental, and societal value of the built environment. *Journal of Construction Engineering and Management-Asce*, 133(9), 619-628.

Levy, O., Goldberg, Y., Dagan, I., 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211-225.

Li, H., Chan, G., Wong, J.K.W., Skitmore, M., 2016. Real-time locating systems applications in construction. *Automation in Construction*, 63, 37-47.

Li, N., Wu, D.D., 2010. Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems*, 48(2), 354-368.

Li, S., Hu, J., Cui, Y., Hu, J., 2018. DeepPatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2), 721-744.

Li, Z., Shen, G.Q., Xue, X., 2014. Critical review of the research on the management of prefabricated construction. *Habitat International*, 43, 240-249.

Li, Z., Tate, D., Lane, C., Adams, C., 2012. A framework for automatic TRIZ level of invention estimation of patents using natural language processing, knowledge-transfer and patent citation metrics. *Computer-Aided Design*, 44(10), 987-1010.

Lin, G.B., Shen, Q.P., 2007. Measuring the performance of value management studies in construction: Critical review. *Journal of Management in Engineering*, 23(1), 2-9.

Lin, K.-Y., Soibelman, L., 2006. Promoting transactions for A/E/C product information. *Automation in Construction*, 15(6), 746-757.

Liu, J., Zhang, Q., Wu, J., Zhao, Y., 2018. Dimensional accuracy and structural performance assessment of spatial structure components using 3D laser scanning. *Automation in Construction*, 96, 324-336.

Liu, K., El-Gohary, N., 2016. Ontology-based sequence labelling for automated information extraction for supporting bridge data analytics. *Procedia Engineering*, 145, 504-510.

Liu, K., El-Gohary, N., 2017. Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction*, 81, 313-327.

Liu, S.-H., Liao, H.-L., Pi, S.-M., Hu, J.-W., 2011. Development of a Patent Retrieval and Analysis Platform – A hybrid approach. *Expert Systems with Applications*, 38(6), 7864-7868.

Lu, Y., Li, Y., Skibniewski, M.J., Wu, Z., Wang, R., Le, Y., 2014. Information and Communication Technology Applications in Architecture , Engineering , and Construction Organizations : A 15-Year Review. *Journal of Management in Engineering*, 31(1), 1-19.

Lu, Y.J., Li, Y.K., Skibniewski, M., Wu, Z.L., Wang, R.S., Le, Y., 2015. Information and Communication Technology Applications in Architecture, Engineering, and Construction Organizations: A 15-Year Review. *Journal of Management in Engineering*, 31(1).

Lubowiecka, I., Armesto, J., Arias, P., Lorenzo, H., 2009. Historic bridge modelling using laser scanning, ground penetrating radar and finite element methods in the context of structural dynamics. *Engineering Structures*, 31(11), 2667-2676.

Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Wang, J., Lin, H., 2018. A neural network approach to chemical and gene/protein entity recognition in patents. *Journal of Cheminformatics*, 10(1), 65.

Ma, X., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.

Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O., 2016. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*.

Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2605), 2579-2605.

Mahdabi, P., Crestani, F., 2014. The effect of citation analysis on query expansion for patent retrieval. *Information Retrieval*, 17(5-6), 412-429.

Mahdabi, P., Keikha, M., Gerani, S., Landoni, M., Crestani, F., 2011. Building Queries for Prior-Art Search. *Information Retrieval Facility Conference*.

Mahmoudi, N., Docherty, P., Moscato, P., 2018. Deep neural networks understand investors better. *Decision Support Systems*, 112, 23-34.

Majumder, M., Barman, U., Prasad, R., Saurabh, K., Saha, S.K., 2012. A Novel Technique for Name Identification from Homeopathy Diagnosis Discussion Forum. *Procedia Technology*, 6, 379-386.

Manning, C.D., 1999. Foundations of statistical natural language processing. *Cambridge, Mass. : MIT Press*, Cambridge, Mass.

Martínez-Rojas, M., Marín, N., Vila, M.A., 2015. The role of information technologies to address data handling in construction project management. *Journal of Computing in Civil Engineering*, 30(4), 04015064.

Mathur, P., 2017. Technological Forms and Ecological Communication: A Theoretical Heuristic. *Lexington Books*.

Meho, L.I., Yang, K., 2007. Impact of data sources on citation counts and rankings of LIS faculty: Web of science versus scopus and google scholar. *Journal of the American Society for Information Science and Technology*, 58(13), 2105-2125.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient Estimation of Word Representations in Vector Space. 1-12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111-3119.

Miller, R.A., Gieszczykiewicz, F.M., Vries, J.K., Cooper, G.F., 1992. CHARTLINE: providing bibliographic references relevant to patient charts using the UMLS Metathesaurus Knowledge Sources. *Proceedings of Symposium on Computer Applications in Medical Care*, 86-90.

Mirończuk, M.M., Protasiewicz, J., 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, 106, 36-54.

Mok, K.Y., Shen, G.Q., Yang, J., 2015. Stakeholder management studies in mega construction projects: A review and future directions. *International Journal of Project Management*, 33(2), 446-457.

Montalvo, S., Martínez, R., Casillas, A., Fresno, V., 2007. Bilingual news clustering using named entities and fuzzy similarity. *International Conference on Text, Speech and Dialogue*, 107-114.

Moraes, R., Valiati, J.F., Neto, W.P.G., 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, 40(2), 621-633.

Motamedi, A., Hammad, A., 2009. RFID-assisted lifecycle management of building components using BIM data. *Proceedings of the 26th international symposium on automation and robotics in construction*, 109-116.

Munková, D., Munk, M., Vozár, M., 2013. Data pre-processing evaluation for text mining: transaction/sequence model. *Procedia Computer Science*, 18, 1198-1207.

Nadeau, D., Sekine, S., 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1), 3-26.

Nave, Z., 2010. Patent Information Gathering and Intellectual Property, in: Globes. http://www.cgi.co.il/.

Niemann, H., Moehrle, M.G., Frischkorn, J., 2017. Use of a new patent text-mining and visualization method for identifying patenting patterns over time: Concept, method and test application. *Technological Forecasting and Social Change*, 115, 210-220.

Nitithamyong, P., Skibniewski, M.J., 2004. Web-based construction project management systems: how to make them successful? *Automation in Construction*, 13(4), 491-506.

Onan, A., Korukoğlu, S., Bulut, H., 2016. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57, 232-247.

Panakkat, A., Adeli, H., 2009. Recurrent Neural Network for Approximate Earthquake Time and Location Prediction Using Multiple Seismicity Indicators. *Computer-Aided Civil and Infrastructure Engineering*, 24(4), 280-292.

Paulus, R., Xiong, C., Socher, R., 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Pavlinek, M., Podgorelec, V., 2017. Text classification method based on self-training and LDA topic models. *Expert Systems with Applications*, 80, 83-93.

Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Poli, R., Healy, M., Kameas, A., 2010. Theory and applications of ontology: Computer applications. *Springer*.

Qiu, Q.J., Xie, Z., Wu, L., Li, W.J., 2019. Geoscience keyphrase extraction algorithm using enhanced word embedding. *Expert Systems with Applications*, 125, 157-169.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

Ramshaw, L.A., Marcus, M.P., 1999. Text chunking using transformation-based learning, Natural language processing using very large corpora. *Springer*, 157-176.

Ravin, Y., Wacholder, N., 1997. Extracting names from natural-language text. Citeseer.

Ren, Y., Wang, R., Ji, D., 2016. A topic-enhanced word embedding for Twitter sentiment classification. *Information Sciences*, 369, 188-198.

Rezaeian, M., Montazeri, H., Loonen, R.C.G.M., 2017. Science foresight using life-cycle analysis, text mining and clustering: A case study on natural ventilation. *Technological Forecasting and Social Change*, 118, 270-280.

Rezgui, Y., 2006. Ontology-centered knowledge management using information retrieval techniques. *Journal of Computing in Civil Engineering*, 20(4), 261-270.

Riedmiller, M., 1994. Advanced supervised learning in multi-layer perceptrons-from backpropagation to adaptive learning algorithms. *Computer standards and interfaces*, 16(3), 265-278.

Rimmimgton, A., Dickens, G., Pasqire, C., 2015. Impact of Information and Communication Technology (ICT) on construction projects. *Organization, technology & management in construction: an international journal*, 7(3), 1367-1382.

Robbins, H., Monro, S., 1985. A stochastic approximation method, Herbert Robbins Selected Papers. *Springer*, 102-109.

Rojas, E.M., Songer, A.D., 1999. Web-centric systems: a new paradigm for collaborative engineering. *Journal of Management in Engineering*, 15(1), 39-45.

Ruddock, L., 2006. ICT in the construction sector: Computing the economic benefits. *International Journal of Strategic Property Management*, 10(1), 39-50.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Sacks, R., Whyte, J., Swissa, D., Raviv, G., Zhou, W., Shapira, A., 2015. Safety by design: dialogues between designers and builders using virtual reality. *Construction Management and Economics*, 33(1), 55-72.

Saha, S.K., Narayan, S., Sarkar, S., Mitra, P., 2010. A composite kernel for named entity recognition. *Pattern Recognition Letters*, 31(12), 1591-1597.

Saha, S.K., Sarkar, S., Mitra, P., 2009. Feature selection techniques for maximum entropy based biomedical named entity recognition. *Journal of Biomedical Informatics*, 42(5), 905-911.

Sang, E.F.T.K., Meulder, F.D., 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Computer Science*.

Sardroud, J.M., 2015. Perceptions of automated data collection technology use in the construction industry. *Journal of Civil Engineering and Management*, 21(1), 54-66.

Schnabel, T., Labutov, I., Mimno, D., Joachims, T., 2015. Evaluation methods for unsupervised word embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 298-307.

Seedah, D.P.K., Choubassi, C., Leite, F., 2016. Ontology for Querying Heterogeneous Data Sources in Freight Transportation. *Journal of Computing in Civil Engineering*, 30(4).

Shannon, C.E., 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3), 379-423.

Shekarpour, S., Marx, E., Ngonga Ngomo, A.C., Auer, S., 2015. SINA: Semantic interpretation of user queries for question answering on interlinked data. *Journal of*

Web Semantics, 30, 39-51.

Shen, D., Zhang, J., Zhou, G., Su, J., Tan, C.-L., 2003. Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, 49-56.

Shin, D.H., Dunston, P.S., 2010. Technology development needs for advancing Augmented Reality-based inspection. *Automation in Construction*, 19(2), 169-182.

Silva, F.N., Amancio, D.R., Bardosova, M., Costa, L.D., Oliveira, O.N., 2016. Using network science and text analytics to produce surveys in a scientific topic. *Journal of Informetrics*, 10(2), 487-502.

Smith, H., 2002. Automation of patent classification. *World Patent Information*, 24(4), 269-271.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.

Souili, A., Cavallucci, D., Rousselot, F., 2015. Natural Language Processing (NLP) - A solution for knowledge extraction from patent unstructured data. *Procedia Engineering*, 131, 635-643.

Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), 11-21.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

Staub-French, S., Fischer, M., Kunz, J., Paulson, B., 2003. An ontology for relating features with activities to calculate costs. *Journal of Computing in Civil Engineering*, 17(4), 243-254.

Tan, R., 2007. Process of two stages analogy-based design employing TRIZ. *International Journal of Product Development*, 4(1-2), 109-121.

Tannebaum, W., Rauber, A., 2014. Using query logs of USPTO patent examiners for automatic query expansion in patent searching. *Information Retrieval*, 17(5-6), 452-470.

Teizer, J., Cheng, T., Fang, Y., 2013. Location tracking and data visualization technology to advance construction ironworkers' education and training in safety and productivity. *Automation in Construction*, 35, 53-68.

Thenmalar, S., Balaji, J., Geetha, T., 2015. Semi-supervised bootstrapping approach for named entity recognition. *arXiv preprint arXiv:1511.06833*.

Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 26-31.

Tixier, A.J.P., Hallowell, M.R., Rajagopalan, B., Bowman, D., 2016. Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports. *Automation in Construction*, 62, 45-56.

USPTO, 2007. Manual of patent examining procedure (8th ed.), Alexandria.

Vähä, P., Heikkilä, T., Kilpeläinen, P., Järviluoma, M., Gambao, E., 2013. Extending automation of building construction — Survey on potential sensor technologies and robotic applications. *Automation in Construction*, 36, 168-178.

Van Slyke, C., Belanger, F., 2003. E-business technologies. *New York.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 5998-6008.

Venugopalan, S., Rai, V., 2015. Topic based classification and pattern identification in patents. *Technological Forecasting and Social Change*, 94, 236-250.

Vlachidis, A., Tudhope, D., 2016. A knowledge-based approach to I nformation E xtraction for semantic interoperability in the archaeology domain. *Journal of the Association for Information Science and Technology*, 67(5), 1138-1152.

Voordijk, H., Van Leuven, A., Laan, A., 2003. Enterprise resource planning in a large construction firm: implementation analysis. *Construction Management and Economics*, 21(5), 511-521.

Wang, J., 2018. Innovation and government intervention: A comparison of Singapore and Hong Kong. *Research Policy*, 47(2), 399-412.

Wang, X., Jiang, W., Luo, Z., 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, 2428-2437.

Wang, Y., Yu, Z., Chen, L., Chen, Y., Liu, Y., Hu, X., Jiang, Y., 2014. Supervised methods for symptom name recognition in free-text clinical records of traditional Chinese medicine: An empirical study. *Journal of Biomedical Informatics*, 47, 91-104.

Weston, J., Chopra, S., Bordes, A., 2014. Memory networks. *arXiv preprint arXiv:1410.3916.*

Wetherill, M., Rezgui, Y., Lima, C., Zarli, A., 2002. Knowledge management for the construction industry: The e-COGNOS project. *Electronic journal of information technology in construction*, 7.

Wong, C.H., 2007. ICT implementation and evolution: Case studies of intranets and extranets in UK construction enterprises. *Construction innovation*, 7(3), 254-273.

Wu, C.H., Yun, K., Huang, T., 2010. Patent classification system using a new hybrid genetic algorithm support vector machine. *Applied Soft Computing Journal*, 10(4), 1164-1177.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yoon, S., Wang, Q., Sohn, H., 2018. Optimal placement of precast bridge deck slabs with respect to precast girders using 3D laser scanning. *Automation in Construction*, 86, 81-98.

Yu, S., Bai, S., Wu, P., 1998. Description of the Kent Ridge Digital Labs system used for MUC-7. *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998.*

Zhang, J.S., El-Gohary, N.M., 2016. Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques.

Journal of Computing in Civil Engineering, 30(5).

Zhao, W., Ma, H., Shi, Z., 2012. Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge & Information Systems*, 30(3), 569-587.

Zhao, Z., Xu, S., Kang, B.H., Kabir, M.M.J., Liu, Y., Wasinger, R., 2015. Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, 42(7), 3508-3516.

Zhou, P., El-Gohary, N., 2016. Ontology-Based Multilabel Text Classification of Construction Regulatory Documents. *Journal of Computing in Civil Engineering*, 30(4), 04015058.

Zhou, Z., Goh Yang, M., Shen, L., 2016. Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry. *Journal of Computing in Civil Engineering*, 30(6), 04016026.

Zhu, G., Iglesias, C.A., 2018. Exploiting semantic similarity for named entity disambiguation in knowledge graphs. *Expert Systems with Applications*, 101, 8-24.

Zhu, J., Uren, V., Motta, E., 2005. ESpotter: Adaptive Named Entity Recognition for Web Browsing. *Lecture Notes in Computer Science*, 3782, 518--529.

Zidane, Y.J.T., Johansen, A., Ekambaram, A., 2013. Megaprojects - Challenges and Lessons Learned, in: Pantouvakis, J.P. (Ed.), Selected Papers from the 26th Ipma, 349-357.

Zou, Y., Kiviniemi, A., Jones, S.W., 2017. Retrieving similar cases for construction project risk management using Natural Language Processing techniques. *Automation in Construction*, 80, 66-76.