Master of Science

Development of a Construction Specialized Pretrained Language Model

February 2022

Department of Civil & Environmental Engineering
The Graduate School
Seoul National University

Geonwoo Kim

Development of a Construction Specialized Pretrained Language Model

지도교수 지석호

이 논문을 공학석사학위논문으로 제출함

2022년 2월

서울대학교 대학원

건설환경공학부

김건우

김건우의 석사학위논문을 인준함

2022년 2월

위원장 ____________________(인)

부위원장 ____________________(인)

위원 ____________________(인)

Abstract

Development of a Construction Specialized Pretrained Language Model

Geonwoo Kim

Department of Civil & Environmental Engineering

The Graduate School

Seoul National University

Due to the nature of construction fields, various irregular text data are generated, and natural language processing is being used in many studies to analyze these data. However, previous studies have limitations that individual models should be created for the study to utilize models that have not been pretrained and lots of labeled data to learn each model is required. On the other hand, there is a difference in the case of pretrained language model that pretraining using unlabeled data in the early days makes a basic model, and then various tasks can be performed only with simple finetuning without creating individual models.

In recent years, some studies have used the pretrained language model, but the pretrained language model used was taught based on general terms, not the term used mainly in the construction field, so there was a limitation in terms of accuracy in analyzing terms of construction.

In order to solve these limitations, this research collected text data used in the construction field and built a construction corpus, and developed and verified a construction specialized pretrained language model by pretraining it. This research consists of two main stages. First, develop a pretrained language model for construction specialization through data collection and comparison between pretrained language models according to corpus. Second, the superiority of the developed model was verified through experiments and comparisons in terms of accuracy, efficiency, and adaptability between the developed pretrained language model and the previously un-pretrained language model.

The results of these experiments show that the pretrained language model developed in this research is superior in terms of accuracy, efficiency, and adaptability compared to the language model that has not been pretrained, and the accuracy is higher than that of the language model that has been pretrained in general corpus. It is expected that the developed construction specialized pretrained language model can be used to perform various natural language processing tasks in the construction field.

Keywords: Pretrained Language Model, Natural Language Processing, Corpus, ELECTRA

Student Number: 2020-20589

Contents

List of Tables

List of Figures

List of Equations

Chapter 1. Introduction

1.1 Research Background

Most of the data in the construction field is text data, and various results can be obtained through analysis of these text data. However, the amount of these text data is too large and costly to handle manually, so natural language processing is essential for analyzing text data in the construction field (Jeong and Kim, 2012; Lee et al., 2016; Liu and El-Gohary 2017). In the meantime, various studies have been conducted to conduct construction-related tasks using natural language processing (M.-Y. Cheng et al., 2020; S. Chi & Han, 2013; Gibb et al., 2014; Rowlinson & Jia, 2015). However, most studies have used a model rather than a pretrained language model, so it is necessary to develop individual models to perform each task, and there is a limit to the need for a large amount of learning data to develop each model.

To solve these limitations, natural language processing techniques using pre-trained language models have emerged.(Devlin et al., 2018; Howard & Ruder, 2018; Peters et al., 2018) The pretrained language model is a technique that uses unsupervised learning from a large-scale corpus that is not labeled to learn the language model first, and performs additional supervised learning by using labeled data according to the task that is performed afterwards. Pretrained language model learns contextual meaning from large capacity corpus, so it shows high learning performance with little additional data. In addition, the task can be performed only by finetuning without the need to generate an individual model according to the task, thereby reducing time and costs required for building the individual model. In various fields such as economy, bio, and law build their own corpus from text data composed of terms mainly used in each specialized field and used it to make pretrained language model (Chalkidis et al., 2020; Clark et al., 2020; Lee et al., 2020). Previous studies have demonstrated that in specialized fields where general terms and mainly used terms are different, the performance of the pretrained language model constructed as a corpus suitable for the specialized field is higher when performing tasks in each field than in the pretrained language model learned in general terms.

In the construction field, recent research using the pretrained language model have been conducted. (Li et al., 2021; Xiong et al., 2021; Xue & Zhang, 2021). However, there is a limitation that the analysis accuracy of construction terminology is low because many technical terms that are very different from the commonly used terminology are used in the task mainly performed in the construction field (Kim & Chi, 2019; Zhong, He, et al., 2020)

1.2 Problem Statement

Previous studies on using natural language processing for construction fields mainly focused on un-pretrained model. Therefore, Each task had to be developed separately to suit the task, and in the process, various labeled data were required to suit the task. In order to obtain labeled data, it is expensive and time-consuming to mobilize personnel with expertise in the construction field, and its applicability is low because only one task can be performed through the completed labeled data. Although these problems have been solved in studies using some pretrained language models, there is a limitation in that the model's understanding of construction terms is poor due to the characteristics that construction terms are different from general terms. To overcome these limitations, it is necessary to develop a pretrained language model specialized in the construction field.

1.3 Research Objectives and Scope

This research aims to develop construction specialized pretrained language model for various kinds of construction related natural language processing tasks. In addition, the developed construction specialized pretrained language model is compared with the natural language processing model, which was mainly used in the construction field, in terms of (1) accuracy (2) efficiency (3) adaptability. And this research set the scope of the research as data related to the construction of Korean words.

A research methodology and the specific objectives to achieve the primary objective are as follows:

1) Objective 1: Text data in the construction field is collected and corpus in the construction field is constructed using the data. In this process, text data in general fields also enables comparative experiments between models learned with corpus in the construction field and models learned with general corpus through collection and corpus creation.
2) Objective 2: By training the ELECTRA model based on the corpus of each construction field and general field, and comparing each performance, proving that the pretrained language model based on the corpus in the construction field is more suitable for natural language treatment tasks.

3) Objective 3: The developed construction specialized pretrained language model is compared with the natural language processing model, which was mainly used in the construction field, to prove its accuracy performance.
4) Objective 4: The developed construction specialized pretrained language model is compared with the natural language processing model, which was mainly used in the construction field, to prove its efficiency performance.
5) Objective 5: The developed construction specialized pretrained language model is compared with the natural language processing model, which was mainly used in the construction field, to prove its adaptability performance.

1.4 Dissertation Outline

This dissertation is composed of five chapters and the details for each chapter are as below.

Chapter 1. Introduction: This chapter covers the backgrounds, problems of the research, objectives, and scope of the research.

Chapter 2. Theoretical Background and Related Works: This chapter provides a comprehensive review of using pretrained language model in natural language processing, studies in which natural language processing was used in the construction field, and studies in which a pretrained language model was used in the construction field.

Chapter 3. Construction Specialized Pretrained Language Model: This chapter presents a process of build construction specialized pretrained language model and describe the process of comparing and verifying the developed model with previous model.

Chapter 4. Results and Discussions: This chapter covers the result of comparison of the pretrained language model learned with the construction corpus and the pre-trained language model learned with the general language corpus and the results of comparison and verification in three aspects (accuracy, efficiency, adaptability) with the developed construction specialized pretrained language model and the previous model mainly used in the construction field are explained.

Chapter 5. Conclusions: This chapter summarize achievements, contributions, limitations of this research, and describe the contents of future study.

Chapter 2. Theoretical Background and Related Works

This chapter describes a comprehensive review of what is and why using pretrained language model in natural language processing. In addition, the research that developed and used a pretrained language model specialized in each domain area in domain areas where there is a large difference between general and mainly used languages is described. It is also described various studies using natural language processing models that were not pretrained in the construction field and studies using pretrained language models.

2.1 Pretrained Language Model

2.1.1 pretrained language model description

Pretrained language model is a technique that applies transfer learning, which is a method of performing supervised learning for labeled data suitable for the desired task after performing unsupervised learning for large amount of unlabeled data (Figure 2.1).

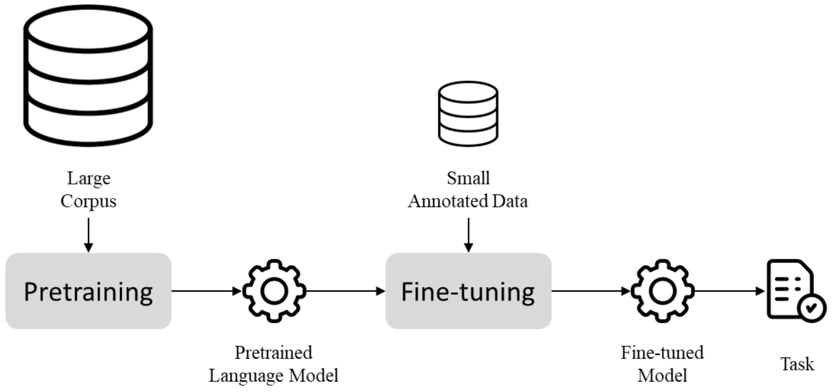


Figure 2.1 process of pretraining and fine-tuning

Before the emergence of the Pretrained language model, the previous models did not include information according to context, and the same word was expressed in the same embedding. Peters et al. (2018) developed 'ELMO (Embedding from Language Model)' and this pretrained language model implies the contextual meaning of the whole word sequence, not the individual word, and it showed excellent results in the processing of homonyms, which were the limitations of the existing model. After the development of ELMO, transfer learning has spread in the field of natural language processing. ULMFiT (Universal Language Model Fine-Tuning) showed that the pretrained model weighted value can be applied to new task, and the data with much less amount can show the same or superior performance as the existing algorithm (Howard & Ruder, 2018.). Vaswani et al. (2017) developed a Transformer structure. Transformer model introduced an attachment mechanism to overcome the limitations caused by the serial connection of recurrent models (RNN, LSTM, GRU), and it dramatically improved the performance of the existing pretrained language model by calculating word representation based on the whole sequence. Recently, various pretrained language models using Transformer's Encoder stack and Decoder stack have appeared in the field of natural language processing. BERT (Bidirectional Encoder Representation from Transformers) improved the inefficiency of learning caused by the left-to-right form of the existing pretrained language model by using MLM (Masked Language Model) in the encoder stack of Transformers (Devlin et al., 2018). Then, based on BERT, Clark et al. (2020) proposed a new pre-training task called Replaced Token Detection (RTD) to develop ELECTRA. In the case of ELECTRA model, it has been developed that are much more economical in computing resources and faster and more effective than existing models, such as Figure 2.2, compared to BERT and existing language models.

In this research, we collected text data in the construction field and decided to use ELECTRA, which can learn quickly with less computing resources and relatively little data, to derive the most efficient results based on the collected text data.

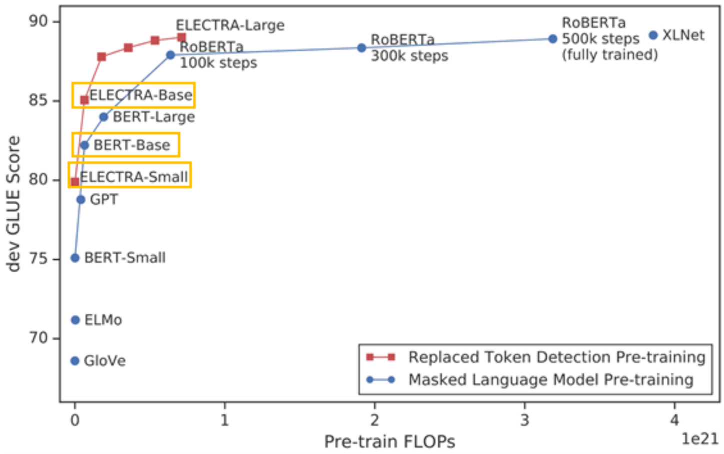


Figure 2.2 ELECTRA model performance

2.1.2 Domain specialized pretrained language model

With the emergence of various pretrained language models, natural language processing studies using the models have been conducted in various domain fields. In the field of using general language, the pretrained language model learned by the existing general text data was used to obtain excellent performance, but some domain term was relatively low in general term and other fields. Many studies have been made to solve these problems and it is very time and cost to build this corpus (Roziewski & Kozłowski, 2021).

In some studies, BERT was learned according to domain specific corpus, which resulted in higher performance when performing domain natural language tasks. Araci, (2019) developed FinBERT for financial sentiment analysis. The existing financial sector analysis has a problem of 'no specialized language and available data, and the general-purpose model is not effective cause of the specialized language used in domain', so to solve this problem, TRC2-financial, Financial Phrasebank, and FiQA Sentiment dataset was pretrained to BERT to develop FinBERT. The FinBERT was obtained from the financial domain task with SOTA (State-of-the-art) and proved its excellence.
Lee et al., (2020) developed BioBERT for biomedical natural language processing tasks. BioBERT was pretrained by biomedical domain corpus such as PubMed, and PMC. Developed BioBERT recorded SOTA in 12 biomedical natural language processing tasks and performed best average score in 15 tasks. Chalkidis et al., (2020) developed LEGAL-BERT and It has good performance in the legal domain field. In addition, the training domain corpus from scratch

in the process of LEGAL-BERT was demonstrated to achieve better performance during the adapt domain corpus from pretraining.

2.2 Natural Language Processing in Construction

2.2.1 Using Un-pretrained language model for Construction tasks

In the construction field, there were various studies using un-pretrained language models, not pretrained language models. Most previous studies using natural language processing tasks were about information extraction (e.g., Identify keywords, identify similar documentation), classification (document classification, construction risk classification), Named entity recognition (extract risk factor, predict classes of words), Question answering (extract relate regulation, chatbot).

Liu & El-Gohary, (2017) proposed ontology-based, semi-supervised conditional random fields (CRF) based information extraction methodology from bridge inspection reports. Tian et al., (2021) used convolutional neural networks (CNN) for text classification and used term frequency-inverse document frequency (TF-IDF) for extract construction knowledge. Ren & Zhang, (2021) proposed a semantic rule-based information extraction (IE) methodology to extract construction execution steps from construction procedural documents automatically. Goh & Ubeynarayana, (2017) used natural language processing with various kinds of models (e.g., Support Vector Machine (SVM), Linear Regression (LR), Random Forest (RF), etc.) for classify the accident causes and reasons. M. Y. Cheng et al., (2020) developed hybrid model incorporating Gated Recurrent Unit (GRU) and Symbiotic Organisms Search (SOS) and named Symbiotic Gated Recurrent Unit (SGRU).

In addition, various studies for classification were conducted (Ayhan et al., 2019; N. W. Chi et al., 2016; Fang et al., 2020; Mo et al., 2020; Zhang, 2019; Zhong, Pan, et al., 2020). Moon et al., (2020) used NER for extract main element from bridge inspection report and proposed active learning to reduce time and cost for labeling. H. Liu et al., (2021) used natural language processing for data preprocessing and encoded to tokenize item descriptions and link pay items across different catalogs.

However, for most studies that did not use the pretrained language model, a large amount of labeled data is essential for the task, and there is a limit that the developed model cannot be used for other models.

2.2.2 Using pretrained Language Model for Construction tasks

In the construction field, there were several studies using pretrained language models for natural language processing tasks.

Amer et al., (2021) proposed the first attempt to automate linking look-ahead planning tasks to master-schedule activities following an natural language processing-based multi-stage ranking formulation. This study used a distance-based matching for candidate generation and a transformer architecture for final matching. By presenting a list of the top five games with a 76.5% precision, it proved that the look-ahead planning task helps match master scheduling activities.

Xue & Zhang, (2021) proposed automated code compliance checking systems to enable an automated regulatory rule conversion. Accurate Part-of-Speech (POS) tagging of building code texts is crucial to this conversion. Therefore, this study used BERT_Cased_Base pretrained language model. This model outperformed the previous SOTA POS taggers.

Zhong, He, et al., (2020) develops a robust end-to-end methodology to improve the efficiency and effectiveness of retrieving queries pertaining to building regulations. The developed methodology integrates information search with a deep learning model of natural language processing to provide accurate and fast answers to user questions in the building regulation collection.

Li et al., (2021) proposed a new NER neural network model using pretrained BERT model based on vocabulary enhancement machine reading to identify plane and overlapped entities in Chinese bridge inspection texts.

In the case of these studies, the pretrained language model was used to obtain superior performance compared to the existing previous model. However, since the pretrained language model was not pretrained based on construction corpus, there is a limit of accuracy in construction task.

Therefore, this study is to develop a model optimized for natural language processing in construction field by creating a pretrained language model learned by corpus in construction field.

Chapter 3. Construction Specialized Pretrained Language Model

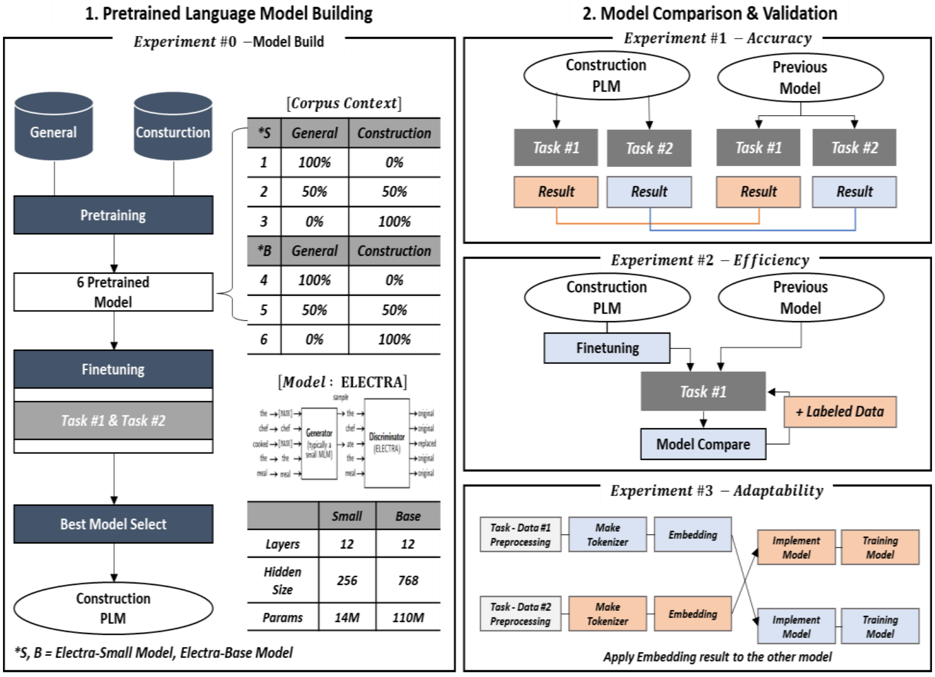
This chapter describes the data collection, development of a pretrained language model, and verification of the developed model. First, in the case of language model development, divided into two steps: (1) collect data and make corpus, (2) pretraining model.

Collect data and make corpus data is the process of collecting text data related to construction, and making construction corpus through preprocessing and integration of the collected data. In this process, a total of 6.6GB of construction-related text data was collected from various sources such as bridge precise safety inspection report, construction standard specification, construction information related laws and enforcement regulations, and construction related articles.

For the verification of the developed language model, divided into three steps: (1) Accuracy, (2) Efficiency (3) Adaptability Validation

To compare accuracy, we studied the model developed with the same epoch and the natural language processing model used mainly in the construction field, and then experimented to compare the F1 score of the two models. To verify the efficiency, the amount of data was gradually increased from 10% to 100% of the total data 10 times, and the F1 score was compared according to the amount of data. In order to verify the applicability, the accuracy of the existing model was calculated by F1 Score by cross-verification method of introducing the results from one task to the embedding stage to another task.

Overview of research methodology is like Figure 3.1.



*S	General	Construction
1	100%	0%
2	50%	50%
3	0%	100%
*B	General	Construction
4	100%	0%
5	50%	50%
6	0%	100%

	Small	Base
Layers	12	12
Hidden Size	256	768
Params	14M	110M

*S, B = Electra-Small Model, Electra-Base Model

Figure 3.1 Overview of research methodology

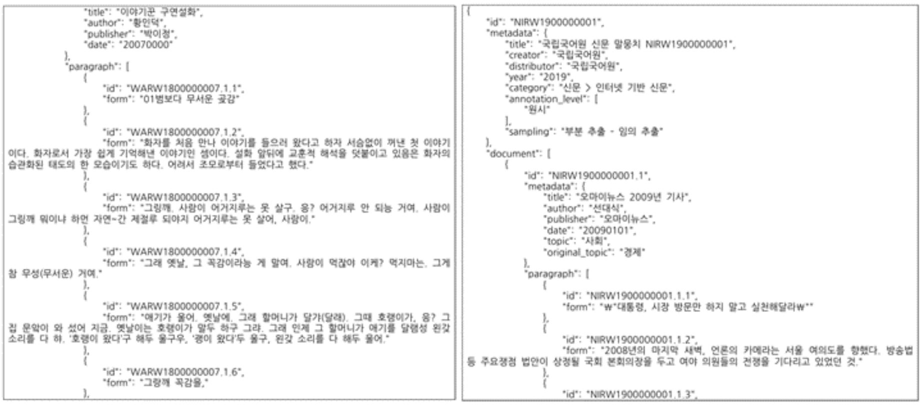
3.1 Pretrained Language Model Building

3.1.1 Collect data and make corpus

It is a step of constructing corpus based on data collected and collected to learn language model in advance. The entire step for Collect data and make corpus consists of (1) collecting text data for each field (2) preprocessing and data extraction (3) constructing three different corpus.

(1) Collecting text data for each field.

Text data consisting of general language are NSMC(Naver Sentiment Movie Corpus), Korean corpus dataset(Korean spoken language, octopus, newspaper, hot speech, etc), ‘CheongWaDae’ national petition comment data, chatbot question and answer fair, KcBERT learning comment. All of the data are open source and data can be provided through a simple data utilization plan submission. An example of text data in a general language is the same as Figure 3.2.



```
                "title": "이야기꾼 구연설화",
                "author": "황인덕",
                "publisher": "박이정",
                "date": "20070000"
            },
            "paragraph": [
                {
                    "id": "WARW1800000007.1.1",
                    "form": "01범보다 무서운 곶감"
                },
                {
                    "id": "WARW1800000007.1.2",
                    "form": "화자를 처음 만나 이야기를 들으러 왔다고 하자 서슴없이 꺼낸 첫 이야기
이다. 화자로서 가장 쉽게 기억해낸 이야기인 셈이다. 설화 앞뒤에 교훈적 해석을 덧붙이고 있음은 화자의
습관화된 태도의 한 모습이기도 하다. 어려서 조모로부터 들었다고 했다."
                },
                {
                    "id": "WARW1800000007.1.3",
                    "form": "그링께, 사람이 어거지루는 못 살구. 응? 어거지루 안 되능 거여. 사람이
그링께 뭐이냐 하먼 자연~간 제절루 되야지 어거지루는 못 살어, 사람이."
                },
                {
                    "id": "WARW1800000007.1.4",
                    "form": "그래 옛날, 그 곶감이라능 게 말여. 사람이 먹잖아 이케? 먹지마는. 그게
참 무성(무서운) 거여."
                },
                {
                    "id": "WARW1800000007.1.5",
                    "form": "애기가 울어. 옛날에. 그래 할머니가 달갸(달래). 그때 호랭이가, 응? 그
집 문앞이 와 섰어 지금. 옛날이는 호랭이가 말두 하구 그랴. 그래 인제 그 할머니가 애기를 달랭성 왼갖
소리를 다 하. '호랭이 왔다'구 해두 울구우, '괭이 왔다'두 울구, 왼갖 소리를 다 해두 울어."
                },
                {
                    "id": "WARW1800000007.1.6",
                    "form": "그랑께 곶감을."
                },
```

```
{
    "id": "NIRW1900000001",
    "metadata": {
        "title": "국립국어원 신문 말뭉치 NIRW1900000001",
        "creator": "국립국어원",
        "distributor": "국립국어원",
        "year": "2019",
        "category": "신문 > 인터넷 기반 신문",
        "annotation_level": [
            "원시"
        ],
        "sampling": "부분 추출 - 임의 추출"
    },
    "document": [
        {
            "id": "NIRW1900000001.1",
            "metadata": {
                "title": "오마이뉴스 2009년 기사",
                "author": "선대식",
                "publisher": "오마이뉴스",
                "date": "20090101",
                "topic": "사회",
                "original_topic": "경제"
            },
            "paragraph": [
                {
                    "id": "NIRW1900000001.1.1",
                    "form": "₩"대통령, 시장 방문만 하지 말고 실천해달라₩""
                },
                {
                    "id": "NIRW1900000001.1.2",
                    "form": "2008년의 마지막 새벽, 언론의 카메라는 서울 여의도를 향했다. 방송법
등 주요쟁점 법안이 상정될 국회 본회의장을 두고 여야 의원들의 전쟁을 기다리고 있었던 것."
                },
                {
                    "id": "NIRW1900000001.1.3",
```

Figure 3.2 General text example

The collected construction text data consists of safety diagnosis report, construction standard specification, construction information related laws and enforcement regulations, safety newspaper, construction related articles, construction field papers, construction term dictionary. Among them, data were collected through web crawling in the case of construction information related laws and enforcement regulations, safety newspapers, and construction related articles, and data were collected through cooperation of related organizations in the remaining data. In the case of web crawling, we used the ‘BeautifulSoup’ and ‘Selenium’ of Python Library. We used the news search query of Naver portal, which is the most used search site in Korea. Based on the 'construction' keyword, we collected news data for 15 years from 2005 to 2020. The construction text data collected example is equal to Figure 3.3.

Figure 3.3 Construction text example

(2) Preprocessing and data extraction

For preprocessing text data, 'soynlp' and 'kss (Korean Sentence splitter)', which are Python libraries, were used. Preprocessing carried out the removal of Chinese characters, special characters and open characters, the removal of URL patterns, and the removal of repeated letters.

In order to reduce the variables according to the amount of data, the total amount of general text data and construction text data was unified to 6.6GB through Random sampling. The summary of the collected general text data and construction text data is equal to Table 3.1

Table 3.1 Summary of collected text data

일반 데이터(General)		건설 데이터(Construction)	
Naver 영화감성분석(NSMC)	0.1GB	안전진단보고서	0.5GB
모두의 말뭉치 한국어 데이터셋	4.7GB	건설공사기준 시방서	0.9GB
청와대 국민청원 댓글 데이터	0.5GB	건설정보 관련 법령 및 시행규칙	0.1GB
챗봇 문답페어	0.4GB	안전신문, 건설관련 기사	3.3GB
기타(KcBERT 학습용 댓글 등)	0.9GB	기타(건설 논문, 건설용어사전 등)	1.8GB
총합	6.6GB	총합	6.6GB

(3) Corpus creation

In order to derive the optimal model according to the change in the corpus composition ratio, three corpus were constructed by extracting 0%, 50%, and 100% of plain text data and construction text data. In the case of constructing a corpus that combines 50% general data and 50% construction data, random sampling was extracted from each data.

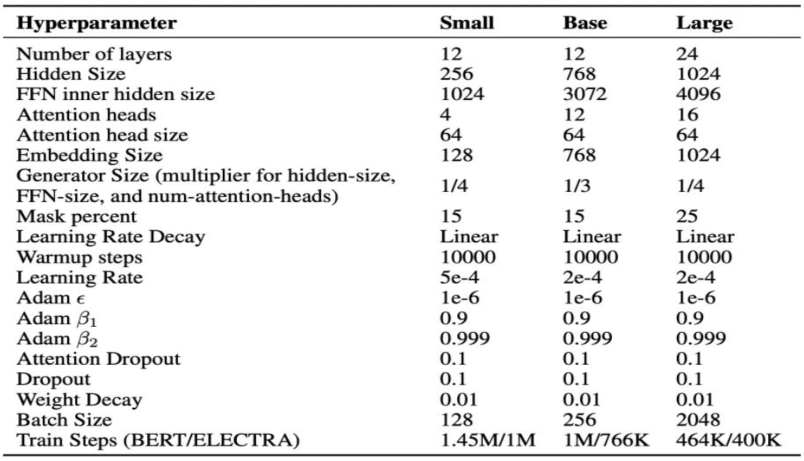
3.1.2 Pretraining model

This step is a process of pretraining based on the completed corpus. Pre-training was conducted for a week using four Google TPU V3-8 from TFRC (TensorFlow Research Cloud), and the basic model performed pretraining by learning three corpus built ahead of the two types of ELECTRA models developed by the Google Research Team (small, base). The entire step for the pretraining model consists of (1) selection of a basic model (2) generation of Vocab (3) model learning.

(1) Two types of ELECTRA model description

The small model consists of a hidden size 256, an embedding size 128, and a batch size 128. In the case of the base model, it consists of a hidden size 768, an embedding size 768, and a batch size 256, etc. The parameters of the small model and the base model are shown in Table 3.2.

Table 3.2 Hyperparameter of the ELECTRA model



Hyperparameter	Small	Base	Large
Number of layers	12	12	24
Hidden Size	256	768	1024
FFN inner hidden size	1024	3072	4096
Attention heads	4	12	16
Attention head size	64	64	64
Embedding Size	128	768	1024
Generator Size (multiplier for hidden-size, FFN-size, and num-attention-heads)	1/4	1/3	1/4
Mask percent	15	15	25
Learning Rate Decay	Linear	Linear	Linear
Warmup steps	10000	10000	10000
Learning Rate	5e-4	2e-4	2e-4
Adam ϵ	1e-6	1e-6	1e-6
Adam β_1	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.999
Attention Dropout	0.1	0.1	0.1
Dropout	0.1	0.1	0.1
Weight Decay	0.01	0.01	0.01
Batch Size	128	256	2048
Train Steps (BERT/ELECTRA)	1.45M/1M	1M/766K	464K/400K

(2) Vocab creation

To learn a language model based on the corpus, an embedding process that converts the corpus into a vector is needed. Vocab was created for this embedding process.

Vocab for each corpus was established to learn the configured corpus. Vocab consists of 32000 words each, and Wordpiece, a basic compatible tokenizer method of the ELECTRA model, was used. Wordpiece is a representative method used in subword tokenizer and is an algorithm that performed Byte Pair Encoding (BPE) based on likelihood of the corpus. An example of how BPE obtains vocab given a sequence 'abcabc' is as shown in Table 3.3.

Table 3.3 Byte Pair Encoding (BPE) example

Iteration	Sequence	Vocabulary
0	a b a b c a b c	{a,b,c}
1	ab ab c ab c	{a,b,c,ab}
2	ab abc abc	{a,b,c,ab,abc}
3	ababc abc	{a,b,c,ab,abc,ababc}
4	ababcabc	{a,b,c,ab,abc,ababc,ababcabc}

(3) Model training

After constructing vocab by corpus, TPU V3-8 and N1-standard-1 VM (Virtual Machine) were used to study. The Python Library used in the learning process is the same as Table 3.4.

Hyperparameter for model learning is the same as Vocab_Size: 32000, Num_train_steps: 700,000, Train_batch_size: 256 Learning_rate: 2e-4, Max_seq_length: 512, No_lower_case. When 100% corpus of construction text data was trained as the ELECTRA-Base model, it was learned by converging to 10.5331 loss like Figure 3.4.

Table 3.4 Python Library list

Number	Library version	Number	Library version
1	huggingface-hub==0.0.12	6	soynlp==0.0.493
2	Korpora==0.2.0	7	tensorflow==1.15.0
3	kss==2.5.0	8	tokenizers==0.9.3
4	regex==2020.11.13	9	torch==1.6.0
5	sentencepiece==0.1.91	10	transformers==3.5.1

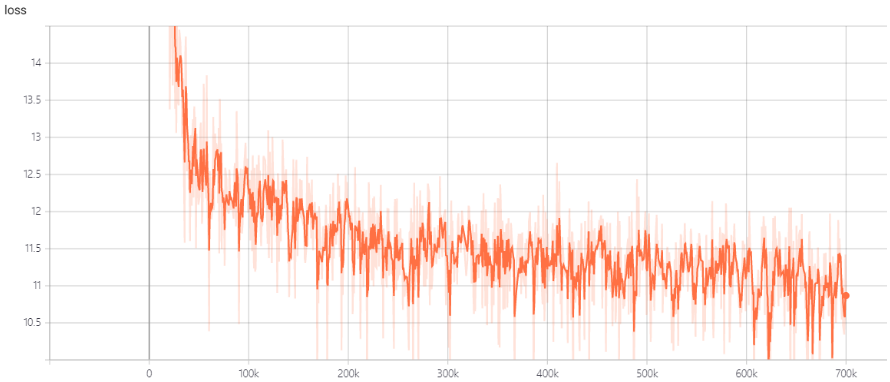


Figure 3.4 Model loss graph

3.2 Model Comparison and Validation

3.2.1 Task Selection and Data Collection for Model Verification

The step of collecting the task selection and related data for verifying the developed pretrained language model is. In this study, Classification and Named Entity Recognition (NER) were selected among natural language processing tasks mainly performed in construction field.

(1) Classification

Text classification is the process of classifying text into word groups. Using NLP, text classification can automatically analyze text and then assign a predefined set of labels or categories according to the context.

In this experiment, Classification performed a task to classify accident types based on construction accident case data. The construction accident case data was provided from Construction Safety Management Integrated Information (CSI) of the National Land Safety Management Agency under the Ministry of Land, Infrastructure and Transport. The data is accident case data for one year from July 2019 to July 2020, and consists of a total of 3,719 cases, excluding data without accident details, which is the main information. Among the total data, a column on how to conduct text analysis and a column on the type of personal accident that will provide label information were extracted, and the extracted data is shown in Table 3.5. There are six types of labels: 'hit object', 'fall down', 'fall', ‘get stuck’, 'cut', and 'others'.

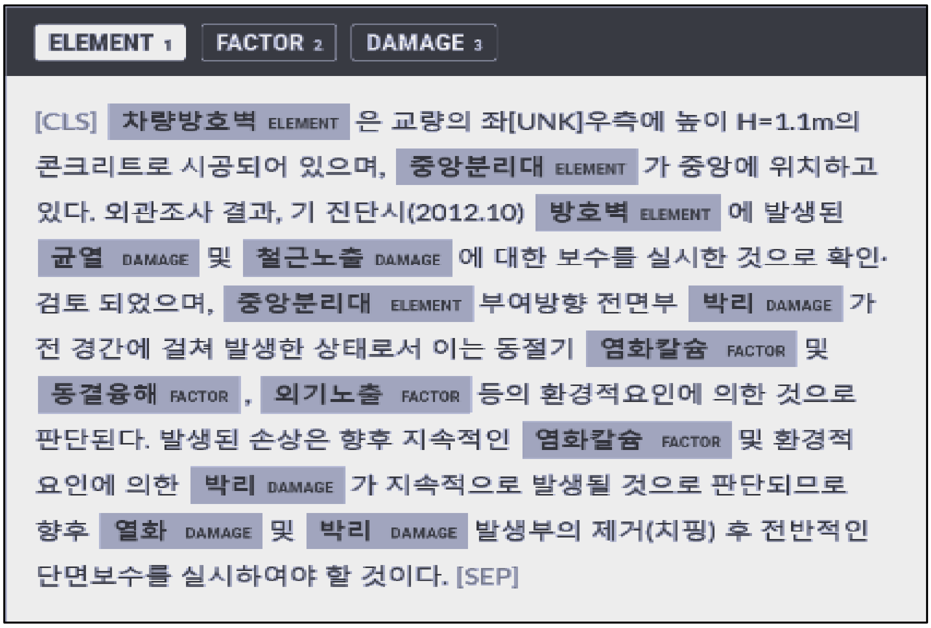
Table 3.5 Accident case data example

Content	Label text	Label
신당교 A2 시스템비계 하부 베이스 플레이트 해체 작업을 위한 비계 인상 작업 중에 작업발판과 작업자와의 부딪힘으로 인한 찰과상	물체에 맞음	4
이동식 고소작업대 이동 중 작업대 상부가 지붕에 걸려 이동이 불가하였으나 이를 인지하지 못하고 더 세게 작업대를 밀다 작업대가 넘어져 그 밑에 깔린 사고임	물체에 맞음	4
일반공 안왕근씨가 시멘트를 어깨에 메고 가설계단을 이동중 뒤로 넘어지며 허리와 늑골을 가설계단에 부딪히며 넘어짐	넘어짐	2
현장식당 앞 이동통로에 재해자가 의식을 잃고 앉아있는 상태에서 입에서 거품이 나오는 것을 보건관리자가 목격하여 119 신고조치	기타	0
펌프카차량의 아웃트리거 설치를 위해 지반상태 및 여유폭 확인 중 도로에서 미끄러져 손목 골절	넘어짐	2
상수도 공사 사진 촬영 중 실족으로 인한 낙상사고	떨어짐	3
굴삭기 인양 중인 복공판이 흔들리며 장비 몸체와 복공판 사이 손가락 끼임(왼손 중지)	끼임	1
…	…	…

(2) Named entity recognition (NER)

Named entity recognition (NER) is a sub-task of information extraction (IE) that finds and classifies a particular entity in the body or body. NER is also known simply as entity identification, entity chunking, and entity extraction. NER is used in many areas of artificial intelligence (AI), including natural language processing (NLP) and machine learning. Information extraction is based on NER and uses a model that operates based on grammar or statistical model to find target information. The NER first recognizes the entity as one of several categories: people, location, organization, expression, percentage, and monetary value. The category is abbreviated to location (LOC), person (PER), and organization (ORG). When the information category is recognized, the information extraction utility extracts the relevant information of the named entity and constructs a document that the machine can read from the information so that other tools can process further to extract meaning.

In NER, we performed task to derive each element based on the bridge safety inspection report. The bridge safety inspection diagnosis report was provided by the Bridge Management System (BMS) of the Korea Institute of Construction Management and Technology under the Ministry of Land, Transport and Maritime Affairs. Data extracted a total of 1,650 paragraphs from 10 of the bridge safety inspection reports conducted in 2014. The labels to perform NER were divided into four categories: ELEMENT, FACTOR, DAMAGE, and NONE. Labeling software Prodigy was used to label NER data, and the example labeled on the actual UI is the same as Figure 3.5.



[CLS] 차량방호벽 ELEMENT 은 교량의 좌[UNK]우측에 높이 H=1.1m의
콘크리트로 시공되어 있으며, 중앙분리대 ELEMENT 가 중앙에 위치하고
있다. 외관조사 결과, 기 진단시(2012.10) 방호벽 ELEMENT 에 발생된
균열 DAMAGE 및 철근노출 DAMAGE 에 대한 보수를 실시한 것으로 확인·
검토 되었으며, 중앙분리대 ELEMENT 부여방향 전면부 박리 DAMAGE 가
전 경간에 걸쳐 발생한 상태로서 이는 동절기 염화칼슘 FACTOR 및
동결융해 FACTOR , 외기노출 FACTOR 등의 환경적요인에 의한 것으로
판단된다. 발생된 손상은 향후 지속적인 염화칼슘 FACTOR 및 환경적
요인에 의한 박리 DAMAGE 가 지속적으로 발생될 것으로 판단되므로
향후 열화 DAMAGE 및 박리 DAMAGE 발생부의 제거(치핑) 후 전반적인
단면보수를 실시하여야 할 것이다. [SEP]

Figure 3.5 NER labeling example

3.2.2 Set up a previous model for comparative verification

To compare with the developed pretrained language model, the model used mainly in the natural language processing research in the construction field was selected. The basic framework of the selected model consists of Bi-LSTM + CRF structure, and word embedding uses word2vec. The detailed structure of the model is like Figure 3.6.

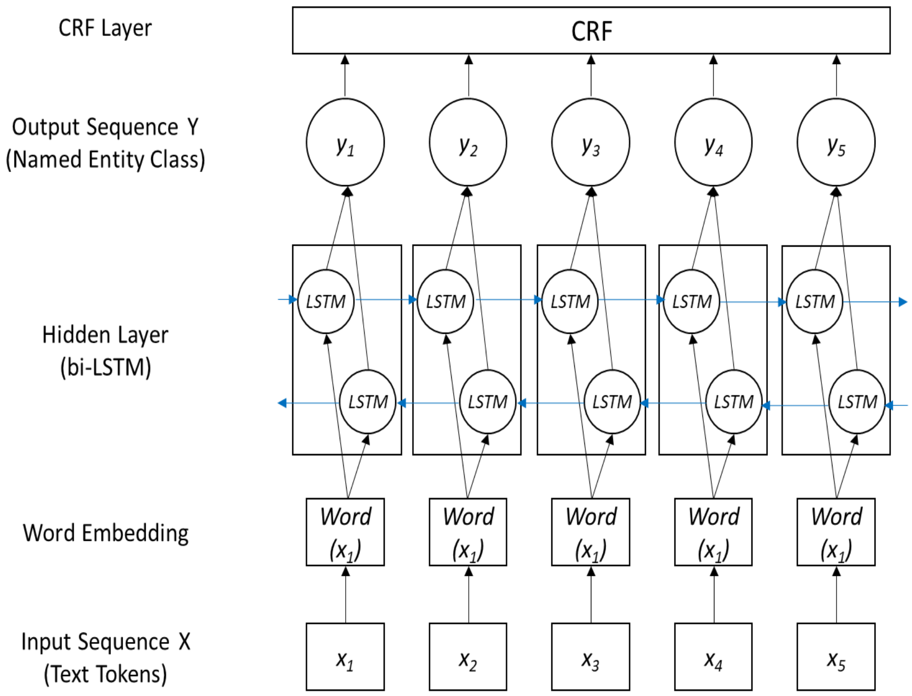


Figure 3.6 Pervious model structure

Text tokens that enter the input are embed using word2vec. The word2vec algorithm is based on the distribution hypothesis that word sets with similar relationships exist in similar vector spaces and that relationships also have constant vector values. For example Since the relationship between 'MAN' and 'WOMAN' is similar to that between 'UNCLE' and 'AUNT', and 'KING' and 'QUEEN', it can be seen that they are distributed with the same vector difference (Figure 3.7). The word2vec algorithm uses a simple two-tier neural network to insert words into a numeric vector. Neural networks are trained to predict specific words when other adjacent words for target words are provided to the input layer of the network. In particular, the input layer provides a one-hot vector of context words for the target word; the value of the input layer is passed to a hidden layer as much as the size specified by the user. When the value of the hidden layer is finally provided to the output layer and the context of the target word is provided to the network, the weight of the neural network is adjusted so that the network can successfully predict the target word. After network training, certain words are mapped to the values of the hidden layer, which is ultimately defined as the word vector for certain words.

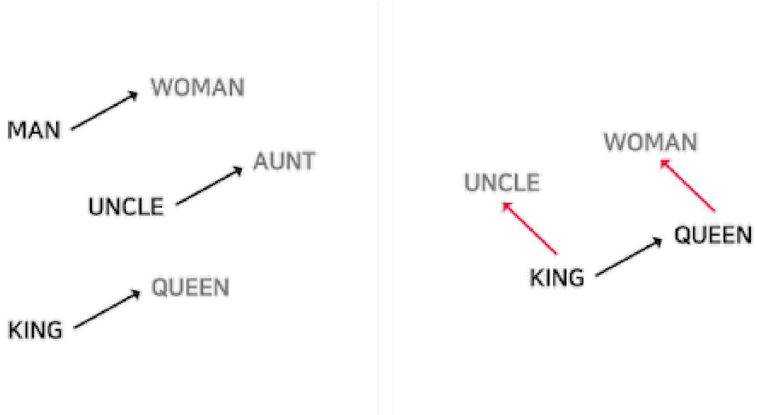


Figure 3.7 Wod2vec example

The two hidden layers, the forward layer, and the other reverse layer, have the same number of LSTM cells as the length of the input sequence or the number of words in the input sentence. Each cell of one hidden layer is connected in one direction so that the model can learn language patterns to express causality. The output of the hidden layer is a vector sequence of size 4. Each element of a vector corresponds to each class in which a word is classified by a model. By interpreting the vector of the 3-output layer as the logit of prediction for each word, the model classifies each word in the input sentence as the logit largest class among the four classes. The model's loss function uses a softmax cross entropy function normalized to the length of the input statement. After classification, go through the Conditional Random Field (CRF) layer. Adding a CRF layer allows the model to consider the dependence between predictive object names, that is, labels. The output value past the activation function for all words is the input of the CRF layer, and the CRF layer predicts the sequence with the highest score for the label sequence. This reflects the two-way context of the output label.

3.2.3 Verification of Accuracy

The developed pretrained language model and the previous model are compared and verified in terms of accuracy in this part. In this step, two tasks are performed for the pretrained language model and the previous model, and the results are compared. The process for the accuracy comparison experiment is the same as Figure 3.9. In this experiment, four types of evaluation metrics were used to evaluate the performance of the pretrained language model and the previous model: accuracy, precision, recall, F1 score.

Accuracy is the most intuitive performance measure, and it is simply a ratio of correctly predicted observation to the total observations. (Eq. 1) Precision is the ratio of the correctly predicted instances among the retrieved instances (Eq. 2). Recall is the ratio of the total amount of relevant instances among the instances actually retrieved (Eq. 3). Lastly, F1 score is a harmonic mean of precision and recall, and this metric is used due to the trade-off that exists between precision and recall (Eq. 4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad Eq. 1$$

$$Precision = \frac{TP}{TP + FP} \qquad Eq. 2$$

$$Recall = \frac{TP}{TP + FN} \qquad Eq. 3$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad Eq. 4$$

$$Micro\ F1-score\ = \frac{2*Micro-Precision*Micro-Recall}{Micro-Precision+Micro-Recall} \qquad Eq.5$$

		Actual Value	
		TRUE	FALSE
Predicted Value	TRUE	TP	FP
	FASLE	FN	TN

Figure 3.8 Confusion matrix

In these equations (Eq.1~Eq.5, Figure 3.8), True Positive (TP) refers to the cases where the model get the answer right. False Positive (FP) refers to the cases where the model incorrectly get the answer, i.e. model predicted the answer is 'Fall off' but the real answer is not 'Fall off'. False Negative (FN) refers to the cases model predicted the answer except 'Fall off' but the real answer is 'Fall off'. Cause these experiments have various label class and our labels are imbalanced, we use micro F1 score for measurement.

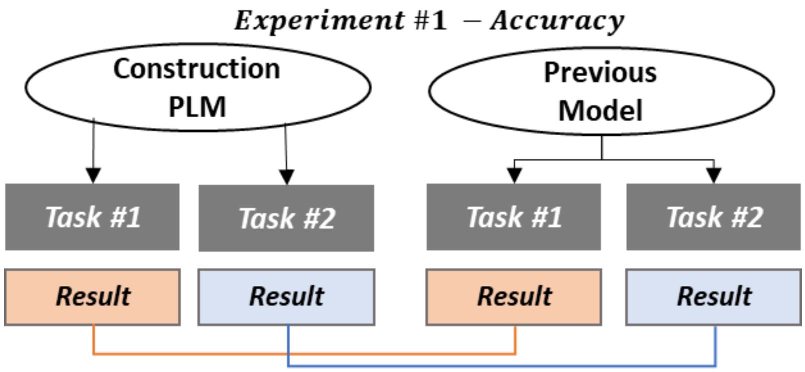


Figure 3.9 Experiment #1 process

3.2.4 Verification of Efficiency

The developed pretrained language model is a step to compare and verify the developed pretrained language model in terms of previous model and efficiency. In this step, we increase the amount of learning data from 10% to 100%, and perform two tasks for the pretrained language model and the previous model developed in 10 experiments. By the F1 score according to amount of learning data being recorded respectively and comparing the comparison about the efficiency part of the pretrained language model and previous model are proceed and the result is compared. The process for the efficiency comparison experiment is the same as Figure 3.10.

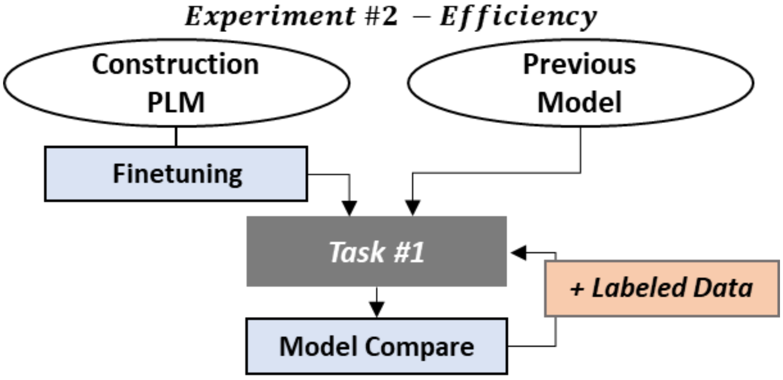


Figure 3.10 Experiment #2 process

3.2.5 Verification of Adaptability

The developed pretrained language model is a step to compare and verify the existing natural language processing model and efficiency. In order to compare the application of the pretrained language model to the existing natural language processing techniques to several Tasks, the results of the text preprocessing of one Task, the creation of a tokenizer, and the embedding learning are applied to the model learning of the other Tasks (4) model application (5). The application of F1 score is compared by cross-application to two types of tasks. The process of the application comparison experiment is equal to Figure 3.11.

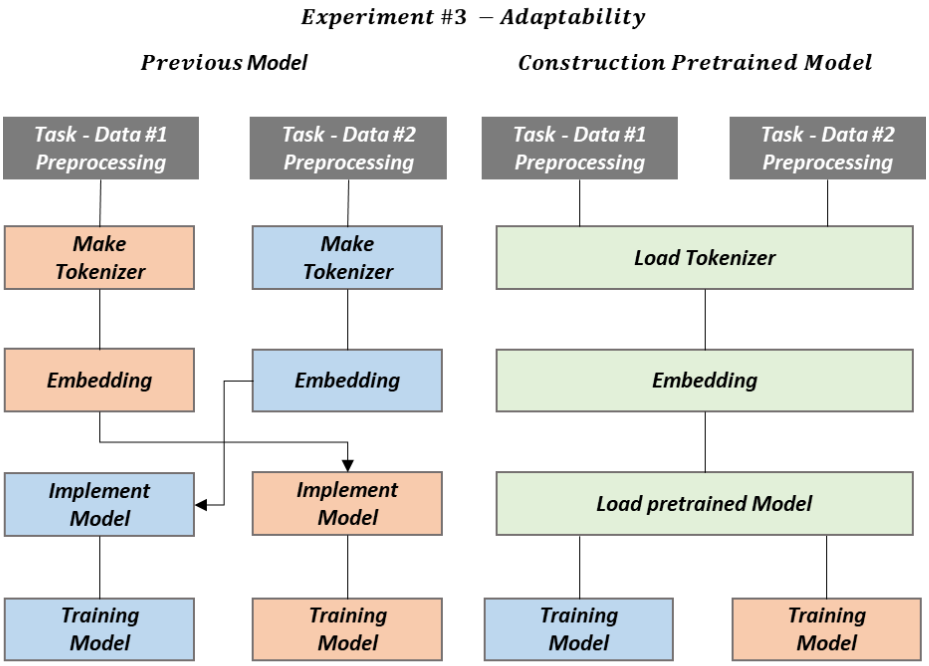


Figure 3.11 Experiment #3 process

3.3 Summary of the methodology

In summary, for developing construction specialized pretrained language model, we collected text data consisting of general language and text data consisting of construction language and built three different corpus based on this. Based on the three corpus, we learned six models for ELECTRA-Small and ELECTRA-Base models, respectively.

To verify the model, two NLP tasks were set up to find out each element in the classification and the bridge precise safety inspection report that fit the accident type based on the construction accident case, and data were collected and labeled according to each task.

In addition, the model used mainly in the natural language processing of the existing construction field was selected through word2vec-based embedding and Bi-LSTM + CRF was selected as the basic framework. The model selected is set up as the control group. The comparison with the developed pretrained language model is performed. The comparison and verification of the pretrained language model with the existing model are performed by experimenting with three things: accuracy, efficiency, and applicability.

Chapter 4. Results and Discussions

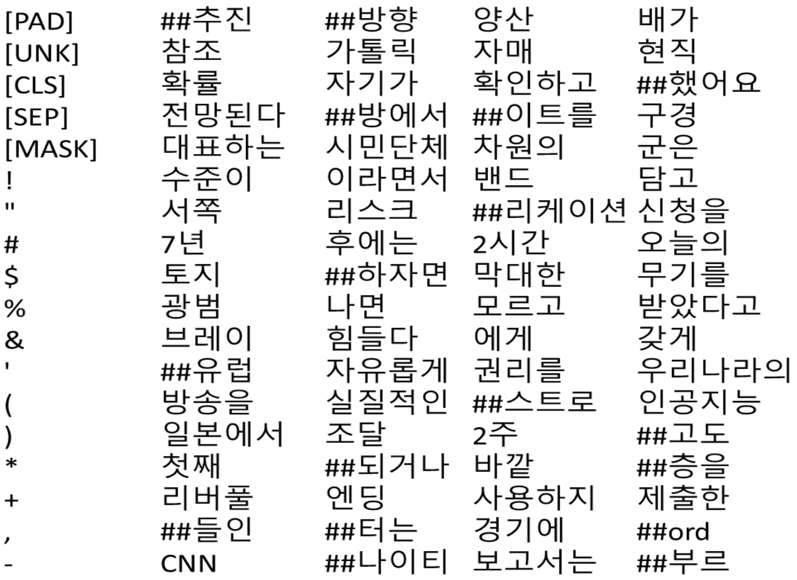
This chapter covers the results of building construction specialized pretrained language model and comparing and validation the developed model with the previous model. The data used in all the processes are the above-mentioned construction accident case classification and bridge safety inspection report NER, and the effectiveness of the construction specialized pretrained language model is proved based on the F1 score obtained from the experiments.

4.1 Construction specialized pretrained language model

4.1.1 Embedding according to learning by corpus

In order to derive the optimal model according to the change in the corpus composition ratio, three corpus were constructed by extracting 0%, 50%, and 100% of plain text data and construction text data.

Based on the three corpus, vocab with 32,000 words each was created using Wordpiece tokenizer. An example of Vocab consisting of 100% general text data is shown in Figure 4.1. Since it is based on Wordpiece, Token, which has acquired a likelihood of less than threshold within the corpus, is connected to the '##' mark.



[PAD]
[UNK]
[CLS]
[SEP]
[MASK]
!
"
#
$
%
&
'
(
)
*
+
,
-

##추진
참조
확률
전망된다
대표하는
수준이
서쪽
7년
토지
광범
브레이
##유럽
방송을
일본에서
첫째
리버풀
##들인
CNN

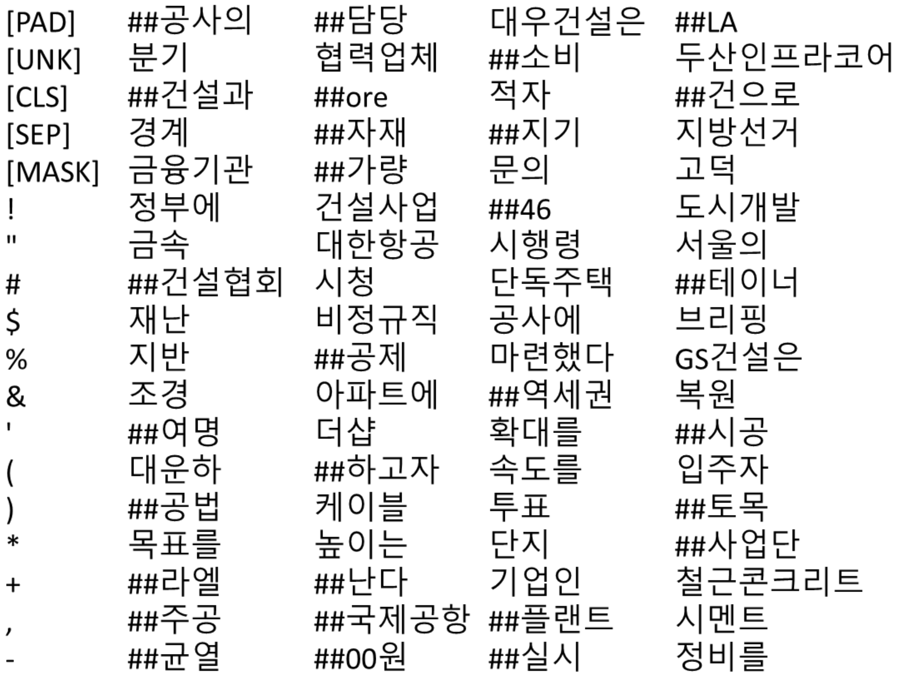
##방향
가톨릭
자기가
##방에서
시민단체
이라면서
리스크
후에는
##하자면
나면
힘들다
자유롭게
실질적인
조달
##되거나
엔딩
##터는
##나이티

양산
자매
확인하고
##이트를
차원의
밴드
##리케이션
2시간
막대한
모르고
에게
권리를
##스트로
2주
바깥
사용하지
경기에
보고서는

배가
현직
##했어요
구경
군은
담고
신청을
오늘의
무기를
받았다고
갖게
우리나라의
인공지능
##고도
##층을
제출한
##ord
##부르

Figure 4.1 General corpus vocab

The vocab of the corpus consisting of 100% of construction-related text data is the same as Figure 4.2.



```
[PAD]    ##공사의      ##담당       대우건설은    ##LA
[UNK]    분기          협력업체     ##소비        두산인프라코어
[CLS]    ##건설과      ##ore        적자          ##건으로
[SEP]    경계          ##자재       ##지기        지방선거
[MASK]   금융기관      ##가량       문의          고덕
!        정부에        건설사업     ##46          도시개발
"        금속          대한항공     시행령        서울의
#        ##건설협회    시청         단독주택      ##테이너
$        재난          비정규직     공사에        브리핑
%        지반          ##공제       마련했다      GS건설은
&        조경          아파트에     ##역세권      복원
'        ##여명        더샵         확대를        ##시공
(        대운하        ##하고자     속도를        입주자
)        ##공법        케이블       투표          ##토목
*        목표를        높이는       단지          ##사업단
+        ##라엘        ##난다       기업인        철근콘크리트
,        ##주공        ##국제공항   ##플랜트      시멘트
-        ##균열        ##00원       ##실시        정비를
```

Figure 4.2 Construction corpus vocab

The corpus consisting of construction-related text data shows that the words used mainly in the construction field such as ‘##플랜트’, ‘##시공’, ‘철근콘크리트’, ‘##공법’, ‘##균열’ have been learned.

Based on the trained tokenizer, the result of embedding when you put an example sentence "소성수축균열은 경화하는 과정에서 외부에 수분을

빼앗기면서 발생하는 초기재령균열을 의미하는데, 비표면적이 큰 방호벽이나 슬래브에서 크게 발생한다." is like Table 4.1.

Table 4.1 Embedding example

	Embedding example
Text Example	"소성수축균열은 경화하는 과정에서 외부에 수분을 빼앗기면서 발생하는 초기재령균열을 의미하는데, 비표면적이 큰 방호벽이나 슬래브에서 크게 발생한다."
General Corpus	['[CLS]', '소', '##성', '##수', '##축', '##균', '##열', '##은', '경', '##화하는', '과정에서', '외부에', '수분', '##을', '빼앗', '##기', '##면서', '발생하는', '초기', '##재', '##령', '##균', '##열을', '의미하는', '##데', ',', '비', '##표', '##면적', '##이', '큰', '방', '##호', '##벽', '##이나', '슬', '##래', '##브', '##에서', '크게', '발생한다', '[SEP]']
Construction Corpus	['[CLS]', '소성', '##수축', '##균열', '##은', '경화', '##하는', '과정에서', '외부에', '수', '##분을', '빼앗', '##기', '##면서', '발생하는', '초기', '##재', '##령', '##균열', '##을', '의미하는', '##데', ',', '비', '##표', '##면적이', '큰', '방호벽', '##이나', '슬래브', '##에서', '크게', '발생한다', '[SEP]']

According to the example, words such as '소성', '수축', '균열', '경화', '방호벽', '슬래브' are trained when the corpus was trained from construction text data, but they are not trained when they are based on corpus trained from general text data.

4.1.2 The Results of Task Performance and Optimal Model Selection according to the Type of Corpus

As a result of experiment on Task#1 – classification and Task#2-NER according to corpus, it was confirmed that F1 score increased as corpus with high proportion of construction data such as Table 4.2 and Table 4.3, and F1 score was higher than ELECTRA-Base model when ELECTRA-Base model was taught. I confirmed the appearance.

Table 4.2 Task #1 results by Corpus

Corpus Model	General 100%	General 50% + Construction 50%	Construction 100%
Small	73.34	74.16	75.31
Base	77.53	78.18	78.92

Table 4.3 Task #2 results by Corpus

Corpus Model	General 100%	General 50% + Construction 50%	Construction 100%
Small	90.04	91.16	91.34
Base	91.33	91.42	92.24

The results of the experiment on two types of tasks showed that the performance of the model, which was studied with 100% Construction Corpus in ELECTRA-Base, was the best with the F1 score of 78.92 in Task#1 and 92.24

in Task#2, respectively. Therefore, this model was selected as a construction specialized pretrained language model.

4.2 Comparison and validation results

4.2.1 Set up a previous model

The description of the previous model set for comparison is as follows. First, the hyperparameters of the Word2vec model applied for word embedding are the same as Table 4.4.

Table 4.4 Hyperparameters of the Word2vec model

Hyperparameter	Value	Description
Vector Size	200	The size of word vector
Window Size	10	The number of neighboring words used to train the word distribution
Minimum Count	30	The minimum threshold for each word to train
Epochs	100	The number of iterations

Both task models were used in the same structure, and only the learning of input-related embedding and tokenizer was different according to each task. The basic skeleton Bi-LSTM consists of 128 units, and Dropout is set at 0.2. In addition, the Softmax function was used for activation function. Optimzier used adam, and loss function used categorical_crossentropy. The overall model consisted of 128 for batch_size and 30 for epochs. The idealization was stopped when the optimal model was obtained within the epochs by applying early stopping.

4.2.2 Verification of Accuracy result

To compare the construction specificized pretrained language model with the Accuracy of the previous model, the experiment for Task #1-classification, Task #2-NER was performed. For the experiment, the data for each task were divided into 8:1:1 for the training: test: Validation, which is the ratio used mainly in the natural language processing field. For task #1-classification, 2,976 data out of a total of 3,719 data were divided into a train set, 371 data into a test set, and 372 data into a validation set. For task #2-classification, 1,320 data out of 1,650 data were divided into train set, 165 data were divided into test set, and 165 data were divided into validation set. The test results of the pretrained language model are the same as Table 4.5 and Table 4.6.

Table 4.5 Accuracy experiment Task #1 result

Task#1	Precision	Recall	F1 score	Support
Others	0.6786	0.5938	0.6333	64
Get stuck	0.8059	0.8059	0.8059	36
Fall down	0.7593	0.8951	0.8217	95
Fall	0.8939	0.831	0.8613	71
Hit object	0.7691	0.7897	0.7788	71
Cut	0.8624	0.7147	0.7817	35
accuracy			0.7892	372
macro avg	0.7964	0.7743	0.7841	372
weighted avg	0.7887	0.7859	0.7892	372

Table 4.6 Accuracy experiment Task #2 result

Task#2	Precision	Recall	F1 score	Support
Damage	0.94	0.94	0.94	745
Element	0.93	0.94	0.93	425
Factor	0.84	0.85	0.84	235
micro avg	0.92	0.93	0.92	1405
macro avg	0.90	0.91	0.91	1405
weighted avg	0.92	0.93	0.92	1405

The comparison result of the Pretrained language model and the Previous model is the same as the Table 4.7.

Table 4.7 Accuracy experiment result comparison

	Pretrained language model	Previous model
Task#1 Classification	78.92	67.56
Task#2 NER	92.24	76.23

In task #1-Classification, the pretrained language model was 11.36 points higher than the previous model, and the task #2-NER was 16.01 points higher. In the case of actual classification example, it was confirmed that the previous model was mispredicted as 'Get stick' in case of 'the moment the connection part is broken when removing the bottom connection pin for the crawler crane boom extension in the field', while the previous model was well judged as 'Hit object'. In the case of the example, there are a lot of construction-related terms, and the actual example sentence itself does not have words related to 'Hit object', so it was difficult to predict the correct answer in the previous model. However, in the case of the pretrained language model, contextual information was grasped and correct answers were predicted.

4.2.3 Verification of Efficiency result

To compare the efficiency of the construction specialized pretrained language model with the previous model, the experiment was performed on task #1-classification, task #2-NER. In this experiment, the total data was divided into 10 parts and experimented with increasing the amount of data used for learning from 10% to 100% 10 times. The experimental results according to the change of data amount by each task are the same as Table 4.8 and Table 4.9.

Previous model showed that the accuracy of data increased as the amount of data increased in both task #1-classification and task #2-NER. Especially, it was found that the amount of data was very low in 10% and 20%, and the result of learning with 100% data was very low compared to the result of learning, which means that there is a lot of labeled data needed for learning of the previous model. On the other hand, the Pretrained language model showed good performance in both Tasks despite the fact that the amount of data was 10%. In case of Task#2, the increase of F1 score as the data increases is the same as the previous model, but the width is small, and it is possible to learn enough through a small amount of labeled data. The results of this study showed that the effectiveness of the pretrained language model was higher than that of the previous model

Table 4.8 Efficiency experiment Task #1 result

Model Data size	Pretrained language model	Previous model
10%	78.56	52.34
20%	78.52	53.62
30%	78.44	54.11
40%	78.20	58.00
50%	77.97	60.23
60%	78.35	62.41
70%	78.88	66.79
80%	77.92	67.51
90%	78.16	67.69
100%	78.92	67.56

Table 4.9 Efficiency experiment Task #2 result

Model Data size	Pretrained language model	Previous model
10%	90.83	60.69
20%	88.88	61.21
30%	88.99	68.93
40%	92.13	73.84
50%	92.06	70.65
60%	91.52	75.44
70%	90.23	76.17
80%	89.83	75.24
90%	91.27	76.84
100%	92.24	76.23

.

4.2.2 Verification of Adaptability result

To compare the adaptability of the construction specialized pretrained language model with the previous model, the experiment was performed on task #1-classification, task #2-NER.

In this experiment, the results of (1) text preprocessing, (2) tokenizer generation, and (3) embedding learning of one task are applied to (4) model application (5) model learning of another task. Cross-applied to two tasks and compared adaptability with F1 score. In the case of the pretrained language model, the results are the same as the results of the accuracy experiment performed earlier because new embedding is not trained and applied as the task changes. On the other hand, in the case of the previous model, embedding training according to task is performed separately, and as a result, F1 score fell significantly compared to the existing cross-applied objective result (Table 4.10, Table 4.11).

Table 4.10 Adaptability experiment Task #1 result

Cross-apply Model	Task#2 → Task#1	Adaptability experiment result
Pretrained language model	78.92	78.92
Previous model	61.45	67.56

Table 4.11 Adaptability experiment Task #2 result

Cross-apply Model	Task#1 → Task#2	Adaptability experiment result
Pretrained language model	92.24	92.24
Previous model	47.63	76.23

4.3 Discussion

First, it was proved through experiments that the accuracy of the NLP task in the domain-knowledge part of the pretrained language model was changed according to the corpus pretraining. The higher the proportion of construction text data, the better the learning of the language used in the construction field, and the embedding result according to the example sentence was confirmed. The results of this study also led to the comparison of the accuracy of two actual tasks. This study proved the necessity of pretrained language model based on corpus composed of text data in the field and built construction specialized pretrained language model.

The developed construction specialized pretrained language model and the previous three experiments on the previous model, which was mainly used in the construction field natural language treatment, proved that the pretrained language model is superior in terms of accuracy, efficiency and adaptability compared to the previous model. In addition, the pretrained language model was more effective in grasping contextual information through example data.

Chapter 5. Conclusion

5.1 Summary and Contributions

In the construction industry, a vast amount of text data such as safety accident cases, specifications, construction laws, and construction contract documents are accumulated, and the processing and analysis of these data consumes a lot of time and money. To solve this problem, various natural language processing techniques have been used in many previous studies. However, in the case of natural language processing techniques used in the construction field, there is a limit that data analysis is still inefficient because a large amount of labeled learning data is needed and individual models must be created according to the task. Although there have been studies on construction using pretrained language model to overcome the limitations of existing natural language processing techniques, there are limitations that the accuracy of construction term analysis is not high because it was pretrained based on corpus composed from text data of general terms, not the term used mainly in construction. To overcome these problems, this research develops a construction specialized pretrained language model. In addition, the developed model was compared with the existing natural language processing techniques in terms of accuracy, efficiency, and applicability.

This research has the following contributions. First, based on the data related to Korean construction, we have built a Korean construction corpus, and confirmed that the corpus is suitable for the natural language processing task in the construction field compared to the corpus composed of other general languages. Second, it proposed a method applicable to natural language processing in construction fields that solved the limitations of previous studies (e.g., The need for individual models for each task, the need for large amounts of labeled data, and the low accuracy in analyzing construction terminology because they are not learned by corpus in construction field due to the use of language models that have not been pretrained.). Finally, the developed pretrained language model can be used in various studies in the future because it can be used directly through fine adjustment in industrial part.

5.2 Limitation and Future Study

In this research, two different natural language processing tasks related with construction were experimented, but in addition to the two tasks that have been tested by but, various natural language processing tasks are being used in the construction field. In some cases, question answering (QA) related to construction regulations is created and used, and similar documents are automatically found and compared and contrasted. There is a limit that the test for these various tasks has not been applied. However, given previous studies in other research fields, it is found that if the performance of both the basic classification and the NER is excellent, the overall high index is obtained in other tasks, so it is possible to solve it if only the supplementary experiment is performed.

As a future study, various kinds of NLP tasks such as QA, Text summarization, chatbot, etc. will be proved that the developed pretrained language model can be applied to other tasks in addition to the classification and NER performed earlier and is superior to previous NLP models. And in addition to Korean construction data, it is also intended to apply to construction text data in various languages by learning based on English or other language-based data. Also, In the construction field, such as GLUE test, which is generally performed in computer science, and KLUE, which is an evaluation index of Korean natural language processing model, It could be possible to make test datasets for performance verification of natural language processing model.

Bibliography

Amer, F., Jung, Y., & Golparvar-Fard, M. (2021). Transformer machine learning language model for auto-alignment of long-term and short-term plans in construction. Automation in Construction, 132. https://doi.org/10.1016/j.autcon.2021.103929

Araci, D. T. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. https://arxiv.org/abs/1908.10063v1

Ayhan, B. U., Tokdemir, O. B., & Asce, M. (2019). Accident Analysis for Construction Safety Using Latent Class Clustering and Artificial Neural Networks. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001762

Baker, H., Hallowell, M. R., & Tixier, A. J. P. (2020). AI-based prediction of independent construction safety outcomes from universal attributes. Automation in Construction, 118. https://doi.org/10.1016/J.AUTCON.2020.103146

Baker, H., Hallowell, M. R., & Tixier, A. J. P. (2020). Automatically learning construction injury precursors from text. Automation in Construction, 118. https://doi.org/10.1016/j.autcon.2020.103145

Bilge, E. Ç., & Yaman, H. (2021). Research trends analysis using text mining in construction management: 2000–2020. Engineering, Construction and

Architectural Management. https://doi.org/10.1108/ECAM-02-2021-0107

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

Cheng, M. Y., Kusoemo, D., & Gosno, R. A. (2020). Text mining-based construction site accident classification using hybrid supervised machine learning. Automation in Construction, 118, 103265. https://doi.org/10.1016/j.autcon.2020.103265

Chi, N. W., Lin, K. Y., El-Gohary, N., & Hsieh, S. H. (2016). Evaluating the strength of text classification categories for supporting construction field inspection. Automation in Construction, 64, 78–88. https://doi.org/10.1016/j.autcon.2016.01.001

Chi, S., & Han, S. (2013). Analyses of systems theory for construction accident prevention with specific reference to OSHA accident reports. International Journal of Project Management, 31(7), 1027–1041. https://doi.org/10.1016/j.ijproman.2012.12.004

Chou, C.-L., Chang, C.-H., Lin, Y.-H., Chien, K.-C., Chang, -H, Lin, Y.-H., & Chien, K.-C. (2020). On the Construction of Web NER Model Training Tool based on Distant Supervision. ACM Trans. Asian Low-Resour. Lang. Inf. Process, 19(6), 87. https://doi.org/10.1145/3422817

Clark, K., Luong, M.-T., Brain, G., Le Google Brain, Q. V, & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. https://arxiv.org/abs/2003.10555v1

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 4171–4186. https://arxiv.org/abs/1810.04805v2

Elgibreen, H., Faisal, M., Sulaiman, M. Al, Abdou, S., Mekhtiche, M. A., Moussa, A. M., Alohali, Y. A., Abdul, W., Muhammad, G., Rashwan, M., & Algabri, M. (2021). An Incremental Approach to Corpus Design and Construction: Application to a Large Contemporary Saudi Corpus. IEEE Access, 9, 88405–88428. https://doi.org/10.1109/ACCESS.2021.3089924

Fang, W., Luo, H., Xu, S., Love, P. E. D., Lu, Z., & Ye, C. (2020). Automated text classification of near-misses from safety reports: An improved deep learning approach. Advanced Engineering Informatics, 44, 101060. https://doi.org/10.1016/J.AEI.2020.101060

Gibb, A., Lingard, H., Behm, M., & Cooke, T. (2014). Construction accident causality: Learning from different countries and differing consequences. Construction Management and Economics, 32(5), 446–459. https://doi.org/10.1080/01446193.2014.907498

Goh, Y. M., & Ubeynarayana, C. U. (2017). Construction accident narrative classification: An evaluation of text mining techniques. Accident Analysis and Prevention, 108, 122–130. https://doi.org/10.1016/j.aap.2017.08.026

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) (Vol. 1). https://doi.org/10.18653/v1/p18-1031

Kim, T., & Chi, S. (2019). Accident Case Retrieval and Analyses: Using Natural Language Processing in the Construction Industry. Journal of Construction Engineering and Management, 145(3), 04019004. https://doi.org/10.1061/(asce)co.1943-7862.0001625

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234–1240. https://doi.org/10.1093/BIOINFORMATICS/BTZ682

Li, R., Mo, T., Yang, J., Li, D., Jiang, S., & Wang, D. (2021). Bridge inspection named entity recognition via BERT and lexicon augmented machine reading comprehension neural model. Advanced Engineering Informatics, 50. https://doi.org/10.1016/j.aei.2021.101416

Liu, H., Kwigizile, V., & Huang, W.-C. (2021). Holistic Framework for Highway Construction Cost Index Development Based on Inconsistent Pay Items. Journal of Construction Engineering and Management, 147(7). https://doi.org/10.1061/(ASCE)CO.1943-7862.0002080

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. https://doi.org/10.1016/j.autcon.2017.02.003

Mo, Y., Zhao, D., Du, J., Syal, M., Aziz, A., & Li, H. (2020). Automated staff assignment for building maintenance using natural language processing. Automation in Construction, 113. https://doi.org/10.1016/J.AUTCON.2020.103150

Mo, Y., Zhao, D., Du, J., Syal, M., Aziz, A., & Li, H. (2020). Automated staff assignment for building maintenance using natural language processing. Automation in Construction, 113. https://doi.org/10.1016/J.AUTCON.2020.103150

Moon, S., Chung, S., & Chi, S. (2020). Bridge Damage Recognition from Inspection Reports Using NER Based on Recurrent Neural Network with Active Learning. Journal of Performance of Constructed Facilities, 34(6), 04020119. https://doi.org/10.1061/(asce)cf.1943-5509.0001530

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 2227–2237. https://doi.org/10.18653/v1/n18-1202

Ren, R., & Zhang, J. (2021). Semantic Rule-Based Construction Procedural Information Extraction to Guide Jobsite Sensing and Monitoring. Journal of Computing in Civil Engineering, 35(6), 04021026. https://doi.org/10.1061/(asce)cp.1943-5487.0000971

Rowlinson, S., & Jia, Y. A. (2015). Construction accident causality: An institutional analysis of heat illness incidents on site. Safety Science, 78, 179–189. https://doi.org/10.1016/j.ssci.2015.04.021

Roziewski, S., & Kozłowski, M. (2021). LanguageCrawl: a generic tool for building language models upon common Crawl. Language Resources and Evaluation, 55(4), 1047–1075. https://doi.org/10.1007/s10579-021-09551-7

Suh, Y. (2021). Sectoral patterns of accident process for occupational safety using narrative texts of OSHA database. Safety Science, 142. https://doi.org/10.1016/j.ssci.2021.105363

Tian, D., Li, M., Shi, J., Shen, Y., & Han, S. (2021). On-site text classification and knowledge mining for large-scale projects construction by integrated intelligent approach. Advanced Engineering Informatics, 49. https://doi.org/10.1016/j.aei.2021.101355

Toosi, H., Ghaaderi, M. A., & Shokrani, Z. (2021). Comparative study of academic research on project management in Iran and the World with text mining approach and TF–IDF method. Engineering, Construction and Architectural Management. https://doi.org/10.1108/ECAM-05-2020-0325

Wu, H., Shen, G., Lin, X., Li, M., Zhang, B., & Li, C. Z. (2020). Screening patents of ICT in construction using deep learning and NLP techniques. Engineering, Construction and Architectural Management, 27(8), 1891–1912. https://doi.org/10.1108/ECAM-09-2019-0480

Xue, X., & Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. Advanced Engineering Informatics, 47. https://doi.org/10.1016/j.aei.2020.101235

Zhang, F. (2019). A hybrid structured deep neural network with Word2Vec for construction accident causes classification. International Journal of Construction Management, 0(0), 1–21. https://doi.org/10.1080/15623599.2019.1683692

Zhong, B., He, W., Huang, Z., Love, P. E. D., Tang, J., & Luo, H. (2020). A building regulation question answering system: A deep learning methodology. Advanced Engineering Informatics, 46(October), 101195. https://doi.org/10.1016/j.aei.2020.101195

Zhong, B., Pan, X., Love, P. E. D., Ding, L., & Fang, W. (2020). Deep learning and network analysis: Classifying and visualizing accident narratives in construction. Automation in Construction, 113. https://doi.org/10.1016/J.AUTCON.2020.103089

초 록

건설분야 특성상 다양한 비정형 텍스트 데이터가 발생하고 있으며, 이러한 데이터를 분석하기 위해 자연어처리가 많은 연구에서 활용되고 있다. 그러나 이전의 연구들은 주로 사전학습 되지 않은 언어모델을 활용하기에 연구 수행을 위해 개별 모델을 만들어야 하고, 각 모델을 학습시키기 위한 라벨링된 데이터를 많이 필요로 한다는 한계점이 있었다. 반면에 사전학습 언어모델의 경우 초기에 라벨링 되지 않은 데이터를 이용해서 사전학습시켜 기본 모델을 만들고, 이후 개별 모델을 만들 필요없이 간단한 미세조정 만으로 다양한 과업을 수행할 수 있다는 차이점이 있다.

최근에는 일부 연구에서 사전학습 언어모델을 활용한 사례도 있었으나 사용한 사전학습 언어모델이 건설분야에서 주로 사용하는 용어가 아닌 일반적인 용어를 기준으로 학습되었기에 건설분야의 용어를 분석하는데 정확도 측면에서 한계가 있었다.

본 연구는 이러한 한계를 해결하기 위해 건설분야에서 사용되는 텍스트 데이터를 수집하여 건설분야 코퍼스를 구축하고, 이를 사전학습 시켜서 건설특화 사전학습 언어모델을 개발 및 검증했다. 연구는 크게 두 가지 단계로 구성되어 있다. 첫째로, 데이터 수집 및 코퍼스에 따른 사전학습 언어모델간 비교를 통한 건설특화 사전학습 언어모델을 개발하였다. 둘째로, 개발된 건설특화 사전학습 언어모델과 기존에 주로 사용하던 사전학습 되지 않은 언어모델과의 정확성, 효율성, 적용성 측면에서의 실험 및 비교를 통해 개발된 모델의 우월성을 검증하였다.

그 결과 본 연구에서 개발한 사전학습 언어모델이 사전학습되지 않은 언어모델에 비해 정확성, 효율성, 적용성 측면에서 모두 우수함을 보였으며, 일반적인 언어로 사전학습된 언어모델에 비해서도 정확도가 더 높음을 확인하였다. 개발된 건설특화 사전학습 언어모델을 활용하여 건설분야 다양한 자연어처리 과업 수행에 활용할 수 있으리라 기대된다.

주요어: 사전학습 언어모델, 자연어처리, 코퍼스, ELECTRA

학 번: 2020-20589