ProcessBERT: Towards Equivalence Judgment of Variable Definitions among Multiple Engineering Documents

Anonymous ACL submission

Abstract

Physical models play an important role in the process industry. However, conventional physical model building requires a survey on a huge amount of literature and trial-and-error to improve the model performance. We aim to develop an automated physical model builder (AutoPMoB), which automatically collects documents about a target process from literature databases, extracts necessary information from them, and builds a desired physical model by reorganizing the information. In this study, we proposed a method of judging equivalence of variable definitions, which is one of the fundamental technologies to realize AutoPMoB. We built a large-scale corpus specialized in chemical engineering and developed ProcessBERT, which is a domain-specific language model pretrained on our corpus. We created datasets from papers related to chemical processes and evaluated the performance of ProcessBERT in the equivalence judgment task. We found that ProcessBERT outperformed the other language models in the similarity-based method.

1 Introduction

In the process industry, physical models play an important role in process design and operation. Conventional physical model building requires engineers to have a deep understanding of a target process and survey a vast amount of documents. In addition, they need to improve the model by trial-and-error until a desired model is obtained.

To free the engineers from the laborious tasks of physical model building, we aim to develop an automated physical model builder (AutoPMoB). AutoPMoB automatically collects relevant documents from literature databases, extracts necessary information (formulas, variables, and experiment data) from them, and builds a desired physical model by combining the information. In order to realize AutoPMoB, several fundamental technologies need to be developed. In this study, we proposed a method for judging the equivalence of variable definitions: whether two noun phrases represent the same variable or not.

BERT (Devlin et al., 2019) achieved state-of-the-art results on various natural language processing (NLP) tasks at the time. Previous studies have also shown that the pre-trained language models using in-domain corpora perform better than those using general-domain corpora when solving NLP tasks in a specialized domain (Lee et al., 2019; Beltagy et al., 2019; Alsentzer et al., 2019; Peng et al., 2019; Gu et al., 2021). With this in mind, we expect that a pre-trained model with a corpus related to chemical engineering will benefit the equivalence judgment of variable definitions.

In this paper, we constructed a corpus specific to chemical engineering and pre-trained ProcessBERT using it. We evaluated the model by judging the equivalence between variable definitions in papers on chemical processes. We finally compared the model's performance with original BERT and SciBERT (Beltagy et al., 2019).

2 Methods

2.1 Corpus

We collected papers related to chemical engineering using Elsevier Research Product APIs[1] from 17 journals as shown in Table 1. We first obtained a list of DOIs and then downloaded documents. We next removed some of the documents that were not journal articles and finally obtained 133,319 papers. The numbers of DOIs and obtained papers are summarized in Table 1. Then, we extracted the abstracts and full texts (excluding figures and tables) from the papers. Then, we split the sentences in the papers using ScispaCy (Neumann et al., 2019), a Python library for practical biomedical/scientific text processing. We finally constructed a chemical engineering corpus (ChemECorpus) with a total

[1] https://dev.elsevier.com/

Journal	DOIs	Papers in ChemECorpus
Applied Catalysis B Environmental	11,369	10,727
Carbohydrate Polymers	17,280	16,361
Chemical Engineering and Processing - Process Intensification	4,200	3,935
Chemical Engineering Journal	27,818	27,222
Chemical Engineering Research and Design	6,000	5,375
Chemical Engineering Science	14,572	13,527
Chinese Journal of Catalysis	2,731	2,709
Computers & Chemical Engineering	13,823	6,584
Current Opinion in Chemical Biology	2,605	2,201
Journal of Catalysis	10,849	10,248
Journal of Cleaner Production	27,814	26,994
Journal of Energy Chemistry	2,251	2,236
Journal of Process Control	3,048	2,744
Progress in Crystal Growth and Characterization of Materials	332	256
Progress in Polymer Science	1,252	1,017
South African Journal of Chemical Engineering	242	233
Thermal Science and Engineering Progress	986	950
Total	147,172	133,319

Table 1: Journal and the number of DOIs and the obtained papers.

word count of approximately 0.68 billion (4.0GB).

2.2 Pre-training

We performed additional pre-training from $\text{BERT}_{\text{BASE}}$ using ChemECorpus. ProcessBERT was first trained using a maximum sequence length of 128 for 900,000 steps on the two pre-training tasks (masked language model and next sentence prediction), with a batch size of 64. Next, the model was trained on longer sequences of maximum length 512 for additional 100,000 steps with a batch size of 8.

In order to verify the model performance difference due to the number of training steps, we also constructed a model with double the number of pre-training steps ($\text{ProcessBERT}_{\text{double}}$). Training of ProcessBERT was performed on a single TPU v3 with 8 cores[2] and this pre-training took about 13 hours to complete.

For pre-training, we used the original BERT code[3]. The vocabulary and hyper-parameters used in the pre-training were the same as those used in $\text{BERT}_{\text{BASE}}$ pre-training.

[2] https://cloud.google.com/

[3] https://github.com/google-research/bert/blob/master/run_pretraining.py (Apache License, Version 2.0)

3 Experiment

3.1 Datasets

First, we collected 11, 10, and 7 papers respectively on Crystallization (CRYST), Continuous Stirred Tank Reactor (CSTR), and Shell and Tube Heat Exchanger (STHE). Next, we extracted the noun phrases corresponding to variable definitions from their full text. We then created all combinations of variable definitions in two different papers of the same process and manually assigned a label: "equivalent" (1) or "non-equivalent" (0). The number of equivalent and non-equivalent pairs for each process is shown in Table 2. Because all the datasets were imbalanced with tiny proportions of equivalent pairs, we created training and test data as follows.

Training Data To keep constant the number of training steps for the experiment in section 3.2.2, we randomly sampled non-equivalent pairs so that the total number of data was 2,500.

Test Data We randomly sampled non-equivalent pairs so that the number of equivalent pairs was 10% of the total.

	Equivalent	Non-equivalent
CRYST	54	12,200
CSTR	122	4,693
STHE	61	13,720

Table 2: The number of equivalent or non-equivalent pairs for three processes.

3.2 Methods of Equivalence Judgment

To evaluate the performance of ProcessBERT and $\text{ProcessBERT}_{\text{double}}$, we conducted experiments by the following two methods comparing with $\text{BERT}_{\text{BASE}}$ and SciBERT (Beltagy et al., 2019).

3.2.1 Similarity between Variable Definitions

As shown in Figure 1, we judge the equivalence of variable definitions by the similarity between their embedding vectors calculated by each language model.

First, we obtain the vector representing the variable definition by the following steps.

1. Input the noun phrase corresponding to the variable definition into the model and extract the embedding vectors of the words except stopwords (e.g. articles, prepositions, and conjunctions) from the twelve layers of Transformer Encoder.

2. Calculate the vector representing the variable definition ($\boldsymbol{d}$) according to Eq. (1):

$$\boldsymbol{d} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{12}\sum_{j=1}^{12}\boldsymbol{v}_{i,j}\right), \qquad (1)$$

   where n is the number of the extracted vectors and $\boldsymbol{v}_{i,j}$ is the ith word's embedding vector from the jth Transformer Encoder ($1 \leq i \leq n$, $1 \leq j \leq 12$).

Next, we calculate the cosine-similarity of the two vectors representing the variable definitions. If the similarity exceeds a threshold, we judge the two definitions as equivalent.

3.2.2 Fine-tuned BERT Model

We first fine-tune a model using the training data of two processes in section 3.1. Next, we evaluate the performance of the fine-tuned model using the test data of the remaining one process. We perform the above steps three times while changing the test data of one process.

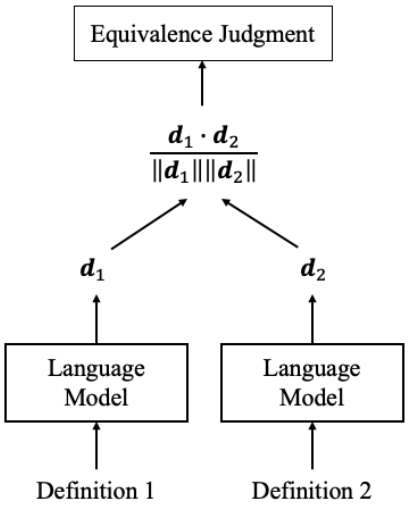

Figure 1: Equivalence judgment by similarity between variable definitions.

For fine-tuning, we used the original BERT code [4]. The task of classifying whether two noun phrases are equivalent or not is similar to the task using Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005). For this reason, we assign "MRPC" to the argument (TASK_NAME) when running run_classifier.py. For the other hyper-parameters in fine-tuning, we follow the recommended settings by Devlin et al. (2019).

Fine-tuning procedure of BERT on MRPC task is shown in Figure 2. First, a word sequence is input into a model, consisting of two variable definitions connected with [SEP] token and prefixed with [CLS] token. Next, the predicted values of the two classes are computed from the embedding vector from the final layer corresponding to [CLS] token ($C \in \mathbb{R}^{768}$) and the weight matrix ($W \in \mathbb{R}^{2\times768}$). If the predicted value of the class of "equivalent" is greater than that of the class of "non-equivalent", the two definitions are judged as equivalent.

4 Results and Discussion

4.1 Results

Table 3 summarizes the equivalence judgment results of the similarity-based method and fine-tuned model. We used Youden's index (Youden, 1950) as the threshold in the similarity-based method. When using the similarity-based method, SciBERT achieved the best score for the CRYST dataset and

[4] https://github.com/google-research/bert/blob/master/run_classifier.py (Apache License, Version 2.0)

Model	Similarity-based method				Fine-tuned model			
	CRYST	CSTR	STHE	All	CRYST	CSTR	STHE	All
ProcessBERT	0.752	**0.642**	**0.726**	**0.653**	0.660	0.270	0.790	0.552
ProcessBERT$_{\text{double}}$	0.658	0.569	0.667	0.590	0.725	0.137	0.842	0.557
BERT$_{\text{BASE}}$	0.730	0.537	0.671	0.567	0.652	**0.336**	0.825	**0.583**
SciBERT	**0.766**	0.631	0.699	0.622	**0.827**	0.094	**0.855**	0.579

Table 3: F1 scores of four models in equivalence judgment test.

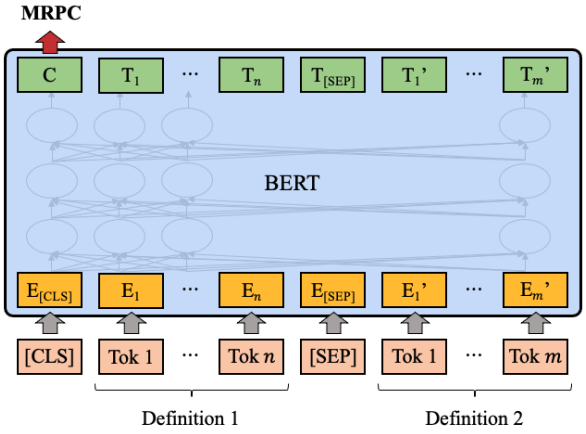

Figure 2: Fine-tuning procedure of BERT on MRPC task(Devlin et al., 2019)

ProcessBERT achieved the best score for CSTR, STHE, and the dataset made by combining the three datasets. When using the fine-tuned model, both ProcessBERT and ProcessBERT$_{\text{double}}$ underperformed the other models for all datasets.

4.2 Discussion

The size of ChemECorpus (0.7B words) is smaller than that of the corpora used to pre-train most of the previous domain-specific BERT models (SciBERT: 3.2B words, BioBERT: 4.5B words, PubMedBERT: 3.1B words). The limited size of ChemECorpus hinders ProcessBERT from learning enough specialized knowledge in the chemical engineering domain. In addition, previous work (Gu et al., 2021) has shown that a domain-specific language model pre-trained from scratch can outperform the one pre-trained from a general-domain language model like BERT$_{\text{BASE}}$. It can be possible to construct a higher-performance language model by constructing a corpus of sufficient size and pre-training from scratch.

In the experiment of equivalence judgment by the fine-tuned model, F1 scores for the CSTR dataset are clearly smaller than those for the other two processes. We found that this method judged as equivalent only the pairs in which the words constituting each noun phrase were almost the same. The fact that the CSTR dataset have less such equivalent pairs than the other datasets can lead to the poor performance for the CSTR dataset. This problem can be solved by splitting the dataset of each process into training and test data and having the model learn variations of the same variable definition. In order to conduct this experiment, we need to increase the number of positive examples in the datasets in the future.

The results of ProcessBERT and ProcessBERT$_{\text{double}}$ show that the performance of ProcessBERT does not improve with increasing the number of pre-training steps. This is in line with the previous study (Alsentzer et al., 2019).

5 Conclusion

To judge the equivalence between variable definitions among multiple documents, we constructed ChemECorpus from 133,319 papers related to chemical engineering and developed ProcessBERT pre-trained using ChemECorpus. We evaluated the performance of ProcessBERT with original BERT and SciBERT by two methods: one using the similarity of variable definitions and the other using the fine-tuned model. As a result, we found that the similarity-based method with ProcessBERT achieved the best performance.

For future work, we will increase the number of positive examples in the test dataset. We will also extend ChemECorpus and pre-train ProcessBERT from scratch to improve its performance.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 58–65. Association for Computational Linguistics.

W. J. Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3(1):32–35.