数学的表現の構造的情報のトークン化手法による ProcessBERT の性能改善

張 純朴　加藤 祥太　加納 学
京都大学大学院情報学研究科
{shota,manabu}@human.sys.i.kyoto-u.ac.jp

概要

プロセス産業において重要な役割を果たす物理モデルの構築には，文献調査を含めた多大な労力を要する．その労力を削減するために，物理モデルを自動で構築するシステム (AutoPMoB) の実現を目指している．本研究の目的は，AutoPMoB の実現に必要な要素技術として，複数文献間の変数定義の同義性判定手法を開発することである．本研究では，まず，化学プロセス関連の論文約 80 万報からなるコーパスを作成した．そして，数学的表現の構造的情報を扱う前処理およびトークン化手法を提案し，化学プロセスに特化した言語モデル $\text{ProcessBERT}_{2022}$ を構築した．$\text{ProcessBERT}_{2022}$ は変数同義性判定において，既存モデルを上回り，F1 値 0.872 を達成した．

1　はじめに

化学や鉄鋼などのプロセス産業では，プロセスの設計や運転に物理モデルに基づくプロセスシミュレータが用いられている．物理モデルの構築にはプロセスに関する深い理解と専門知識だけでなく，精度向上のための試行錯誤的な取り組みが必要とされる．このため，物理モデルの構築には多大な労力がかかる．本研究の最終目的は，複数の文献から物理モデルを自動で構築するシステム (Automated physical model builder; AutoPMoB) を開発することである．AutoPMoB の実現には，物理モデル構築に必要な情報を文献から抽出し，抽出した情報の表記を統一する必要がある．本研究の目的は，表記を統一するために，異なる文献から抽出した変数の定義が同じかどうか (同義性) を正確に判定する手法を開発することである．本研究では，1) 約 80 万報の化学工学関連論文を収集し，2) BERT モデルが数学的表現の構造的情報を学習できるような前処理手法を提案し，BERT モデルをゼロから学習する．そして，提案手法と既存手法で構築した BERT モデルによる変数定義の同義性判定性能を比較する．

2　関連研究

自然言語処理の分野において，単語の意味をベクトル (埋込ベクトル) で表現することの有効性が多くの研究で検証されてきた．Word2Vec [1]，GloVe [2]，FastText [3] は，文脈に依存しない埋込ベクトルである．しかし，単語の意味は文脈によって変わる場合がある．単語に対して一意に決まる埋込ベクトルを学習する手法はそのような状況に対応できない．文脈に依存した埋込ベクトルを学習する言語モデルの一つが BERT (Bidirectional Encoder Representations from Transformers) [4] である．Peng ら [5] は，数式とその周辺の文脈から構成されるコーパスを用いて，数学分野の情報抽出，トピック分類および文生成のタスクに特化したモデル MathBERT を構築した．Dadure ら [6] は，数学分野の情報抽出と質問応答の研究を行う ARQMath [7] が作成したコーパスを用いて，数式の情報抽出タスクに特化した BERT モデルを作成した．これらの研究では，一般的な数学的情報を扱うコーパスを用いて追加で BERT モデルを訓練していた．しかし，化学プロセス分野における変数定義同義性判定のタスクに対応するために，その分野に特化したコーパスでモデルを構築する必要がある．金上ら [8] は，化学工学分野のコーパス (約 7 億語) を用いて BERT を追加で訓練し，ProcessBERT を構築した．彼らの前処理では，数学的表現の構造的な情報を無視したテキストを使用していた．図 1 に数式 $\frac{c_k}{c_l}$ の前処理およびトークン化後の文字列を示す．この前処理後の文字列が元々数学的表現の一部なのか，単語の一部なのかを区別することはできないため，数学的表現の構造的情報は学習されない．また，学習に使用したコーパスのサイズが小さいという課題があった．

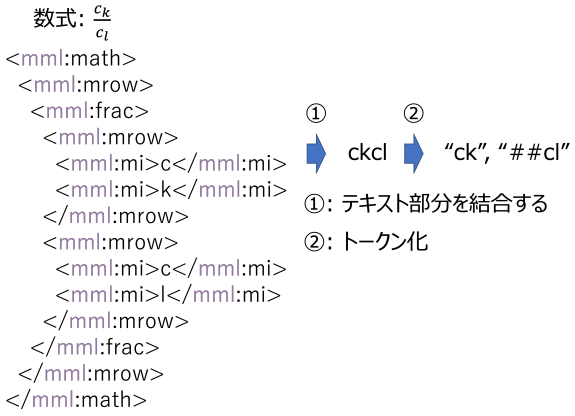

図 1　先行研究 [8] の前処理およびトークン化手法による数式の変換例.

3　提案モデル (ProcessBERT$_{2022}$)

3.1　コーパス

Elsevier 社が提供する Elsevier Research Product APIs [9] を用いて，化学工学分野の 130 ジャーナルから XML (Extensible Markup Language) 形式の論文ファイルを収集した．XML はタグを用いて文章の見た目と構造を記述するマークアップ言語の一種である．XML の数学的表現に対応するタグは 2004 年に ⟨formula⟩ タグから ⟨mml:math⟩ タグに変更された．本研究では，論文中の数学的表現を統一的に処理するために，数学的表現の記述に ⟨mml:math⟩ が用いられている 2005 年以降の論文約 80 万報を使用する．簡単のため，以降の文では，数学的表現の記述に用いられるタグを “mml” が省略された形で表記する．先行研究 [8] と同様に，論文の抄録と本文を用い，タイトル・著者情報・キーワード・参考文献・付録は用いない．

3.2　前処理とトークン化

前処理では，まず数学的表現の置換を行う．数学的表現は変数と数式の二種類に分けられる．本研究では，以下の 2 つの条件を全て満たす XML 構文を変数を表現するものとみなす．

1. ⟨mi⟩, ⟨msub⟩, ⟨msup⟩, ⟨msubsup⟩, ⟨mover⟩, ⟨munder⟩, ⟨moverunder⟩, ⟨math⟩, ⟨mrow⟩ の 9 種類のタグのみを使用する．
2. 一番外側の ⟨math⟩ 要素の直下にある子要素の数が 1 である．

表 1　7 種類の XML タグごとの変数変換ルール

タグ	変数例	処理後文字列
⟨mi⟩	V	[VAR] v
⟨msub⟩	V_i	[VAR] v [SUB] [VAR] i
⟨msup⟩	V^i	[VAR] v [SUP] [VAR] i
⟨msubsup⟩	V_i^j	[VAR] v [SUB] [VAR] i [SUP] [VAR] j
⟨mover⟩	$\overline{V}$	[VAR] v [OVER] [VAR]¯
⟨munder⟩	$\underline{V}$	[VAR] v [UNDER] [VAR] _
⟨moverunder⟩	$\dot{\underline{V}}$	[VAR] v [OVER] [VAR] . [UNDER] [VAR] _

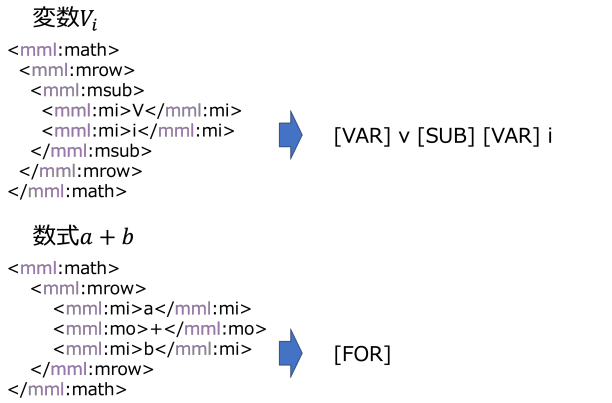

図 2　提案する前処理による変数と数式の変換例．XML タグと変換ルールに基づいてタグを含まない文字列に変換する．

XML で変数の表現に用いられる 7 種類のタグに注目し，表 1 に示す変換ルールを定義した．このルールに基づき，各変数を処理する．一つの変数に複数のタグが用いられる場合，外側から順に処理を行う．数式は変数と比較してタグの種類が多く，構造も多様である．また，数式より変数の方が変数定義の同義性判定を行う上で重要であると考えたため，数式は一律に “[FOR]” トークンで置換することとした．図 2 に変数と数式の変換処理の例を示す．

次に，前処理後の文字列をトークン化する．トークン化には BERT と同じ WordPiece [10] アルゴリズムで作成したトークナイザを用いた．トークナイザの作成時には 6 種類の特殊トークン (“[FOR]”, “[VAR]”, “[SUB]”, “[SUP]”, “[OVER]”, “[UNDER]”) を追加した．

3.3　事前学習

以下の 3 つのステップを経て，XML ファイルを一文一行 (one sentence per line) のテキストファイルに変換する．

1. 数学的表現を提案した前処理で変換する．

2. 抄録と本文のテキスト部分を結合する．
3. 文分割のツール Spacy [11] を用いて，結合されたテキストを文に分割し，一行が一文のテキストファイルに変換する．

以上の処理で事前学習データ (36 億語) を得た．これを用いて BERT モデルをゼロから訓練し，化学プロセスに特化した言語モデル $\text{ProcessBERT}_{2022}$ を構築した．$\text{ProcessBERT}_{2022}$ の学習では，バッチサイズとシーケンスの最大長をそれぞれ 256 と 128 とし，学習ステップを 1,000,000 とした．それ以外のハイパーパラメータは $\text{BERT}_{\text{BASE}}$ [12] と同じとした．事前学習タスクは Masked Language Model と Next Sentence Prediction とした．計算には Google Cloud Platform [13] で 8 コアの TPU v3 を使用した．学習には約 32 時間要した．事前学習の際に指定する語彙ファイルはトークナイザの作成時と同じものを使用した．

4 実験

4.1 データセット

先行研究 [8] で使用したデータセットに新たに 2 プロセスを追加した合計 5 つの化学プロセスに関する論文 45 報からなるデータセットを用いる．5 つのプロセスは，バイオディーゼルプロセス (Biodiesel; BD)，晶析プロセス (Crystallization; CRYST)，連続槽型反応器 (Continuous Stirred Tank Reactor; CSTR)，チョクラルスキープロセス (Czochralski; CZ)，多管式熱交換器 (Shell and Tube Heat Exchanger; STHE) である．同一プロセスの異なる 2 つの論文に含まれるすべての変数定義のペアについて，同義 (1) もしくは非同義 (0) のラベルが付与されている．プロセスごとの同義と非同義の変数定義ペアの数を表 2 に示す．このデータセットは，同義ペアの数が非同義変数定義ペアの数よりもかなり少ない不均衡データセットであるため，以下のように各プロセスについて，訓練用およびテスト用データを作成した．

訓練用 同義変数定義ペアの半分をランダムにサンプリングし，どのプロセスでもデータの総数が 1,500 になるように非同義の変数定義ペアをランダムにサンプリングした．

テスト用 訓練用データ以外の同義変数定義ペアをテスト用とした．また，テスト用データの数が全体のサンプル数の 10%になるように非同義の変数定義ペアをランダムにサンプリングした．

表 2 各プロセスの論文数と同義および非同義のペア数

プロセス	論文数	同義	非同義
BD	9	41	3,770
CRYST	11	165	22,186
CSTR	10	202	7,391
CZ	8	329	22,144
STHE	7	69	16,995

4.2 同義性判定手法

先行研究 [8] と同様に，「変数定義の類似度に基づく手法」と「ファインチューニング済モデルに基づく手法」を用いる．

4.2.1 変数定義の類似度に基づく手法

2 つの変数定義をそれぞれ BERT モデルに入力して変数定義ベクトルを算出し，それらのコサイン類似度が閾値より大きければ同義と判定する．変数定義ベクトルとして，先行研究 [8] では 12 層の Transformer encoder の出力ベクトルの平均を使用していたが，本研究では，最終層の出力ベクトルのみを使用する．変数定義ベクトルの計算には Devlin らが Github 上で公開しているプログラム [12] (extract_features.py) を使用した．閾値には Youden's Index [14] を採用した．

4.2.2 ファインチューニング済モデルに基づく手法

各プロセスについて，$\text{ProcessBERT}_{2022}$ の事前学習済モデルを訓練用データでファインチューニングし，そのモデルで同義性を判定する．ファインチューニングの際の下流タスクとして，2 つの名詞句が言い換えであるかどうかを判定する Microsoft Research Paraphrase Corpus (MRPC) [15] を用いたタスクを使用し，訓練には Devlin らが Github 上で公開しているプログラム [12] (run_classifier.py) を使用した．

5 結果と考察

5.1 結果

類似度による同義性判定の結果を表 3 に示す．本研究で構築した $\text{ProcessBERT}_{2022}$ に加えて，ProcessBERT [8]，$\text{BERT}_{\text{BASE}}$ [4]，SciBERT [16] の結果も示した．$\text{ProcessBERT}_{2022}$ は CRYST と CZ の 2 つのプロセスにおいて最も高い F1 値を達成したのに

表 3　類似度に基づく変数定義同義性判定結果 (F1 値)

モデル	BD	CRYST	CSTR	CZ	STHE	All
ProcessBERT	0.760	0.789	**0.757**	0.591	0.757	0.622
ProcessBERT$_{2022}$	0.655	**0.847**	0.730	**0.627**	0.750	0.720
BERT$_{BASE}$	0.731	0.817	0.632	0.617	0.750	0.672
SciBERT	**0.776**	0.811	0.719	0.558	**0.792**	**0.725**

対して，SciBERT は BD，STHE および全プロセスをまとめたデータセット (All)，ProcessBERT は CSTR においてそれぞれ最高値を達成した．

ProcessBERT$_{2022}$ と ProcessBERT のファイチューニング済モデルによる同義性判定結果 (正解率，適合率，再現率，F 値) をそれぞれ表 4 と表 5 に示す．いずれの場合も類似度を用いた手法と比較して，BD と STHE を除いたデータセットで性能が向上した．BD と CSTR 以外のデータセットにおいて，ProcessBERT$_{2022}$ の性能は ProcessBERT よりも高かった．

5.2　考察

本研究で構築したコーパス (36 億語) は，他のドメイン特化 BERT モデルのそれと同等のサイズである (SciBERT [16]: 32 億語，BioBERT [17]: 45 億語，PubMedBERT [18]: 31 億語)．そして，先行研究 [8] では事前学習済の BERT$_{BASE}$ のモデルに追加で学習を行ったのに対して，本研究では，構築したコーパスのみを用いてゼロから学習を行った．このため，先行研究で報告された，コーパスのサイズが小さく，モデルが十分に化学プロセス分野の専門的知識を十分に学習できない問題は改善された．

ファインチューニング済モデルによる同義性判定の結果において，ProcessBERT$_{2022}$ は BD のデータセットに対する再現率と F 値が，他のプロセスの場合よりも低かった．これは，BD の同義変数定義ペアの数が少なく，モデルが十分に正例を学習できなかったことが原因として考えられる．高い性能を達成するには，他のプロセスと同等の数の正例データを確保する必要がある．

また，本研究では変数定義の類似度に基づく手法とファインチューニング済モデルに基づく手法の両方において，変数定義を入力とした．これは，変数定義を含む文を入力とし，類似度に基づく手法で同義性判定を行ったところ，モデルの性能が著しく低下したためである．単語は周辺の文脈から意味が決まるという分布仮説や BERT モデルが事前学習時に

表 4　ProcessBERT$_{2022}$ のファインチューニング済モデルに基づく変数定義同義性判定結果

データセット	正解率	適合率	再現率	F 値
BD	0.928	0	0	0
CRYST	0.990	0.980	0.893	0.935
CSTR	0.980	0.932	0.831	0.879
CZ	0.983	0.868	0.941	0.903
STHE	0.972	0.926	0.735	0.820
All	0.979	0.911	0.837	0.872

表 5　ProcessBERT のファインチューニング済モデルに基づく変数定義同義性判定結果

データセット	正解率	適合率	再現率	F 値
BD	0.942	0	0	0
CRYST	0.978	0.885	0.821	0.852
CSTR	0.979	0.910	0.843	0.875
CZ	0.955	0.701	0.814	0.753
STHE	0.921	0	0	0
All	0.961	0.797	0.709	0.750

文を入力とする事実を踏まえると，本来は文を入力することで性能向上を期待できるはずである．さらに，定義を含む文を入力とする方法はファイチューニング済モデルに基づく手法に適用できるが，その際に適切な下流タスクを考案する必要がある．このような文を入力として用いる方法の開発は今後の課題である．

6　おわりに

本研究では先行研究 [8] に引き続き，物理モデル自動構築システム (AutoPMoB) の要素技術である複数文献間における変数の同義性判定手法の開発に取り組み，XML 形式のコーパスに含まれる数学的表現の構造的情報を扱う前処理およびトークン化手法を提案した．提案手法で構築した言語モデル ProcessBERT$_{2022}$ は先行研究で構築された ProcessBERT [8] よりも高い同義性判定性能を達成した．今後は，5.2 節で議論した課題に対応し，さらなる性能向上を目指す．

謝辞

本研究は JSPS 科研費 JP21K18849 および Google Cloud Research Credits プログラムの助成 (GCP19980904) を受けたものです.

参考文献

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

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

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, Vol. 5, pp. 135–146, 2017.

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

[5] S. Peng, K. Yuan, L. Gao, and Z. Tang. Mathbert: A pre-trained model for mathematical formula understanding. **arXiv preprint arXiv:2105.00377**, 2021.

[6] P. Dadure, P. Pakray, and S. Bandyopadhyay. Bert-based embedding model for formula retrieval. In **CLEF (Working Notes)**, pp. 36–46, 2021.

[7] R. Zanibbi, D. W Oard, A. Agarwal, and B. Mansouri. Overview of arqmath 2020: Clef lab on answer retrieval for questions on math. In **International Conference of the Cross-Language Evaluation Forum for European Languages**, pp. 169–193. Springer, 2020.

[8] 金上和毅, 加藤祥太, 加納学. 複数文献間の変数の同義性判定に向けた ProcessBERT の構築. 言語処理学会 第 28 回年次大会 発表論文集, 2022.

[9] Elsevier developer protal. https://dev.elsevier.com/., Accessed on 2022/01/16.

[10] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou. Fast wordpiece tokenization. **arXiv preprint arXiv:2012.15524**, 2020.

[11] M. Neumann, D. King, I. Beltagy, and W. Ammar. Scispacy: fast and robust models for biomedical natural language processing. **arXiv preprint arXiv:1902.07669**, 2019.

[12] Original bert codes. https://github.com/google-research/bert, Accessed on 2022/07/13.

[13] Cloud computing services—google cloud. https://cloud.google.com/., Accessed on 2022/12/29.

[14] W. J. Youden. Index for rating diagnostic tests. **Cancer**, pp. 32–35, 1950.

[15] B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In **Third International Workshop on Paraphrasing (IWP2005)**, 2005.

[16] I. Beltagy, K. Lo, and A. Cohan. Scibert: A pre-trained language model for scientific text. **arXiv preprint arXiv:1903.10676**, 2019.

[17] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, Vol. 36, No. 4, pp. 1234–1240, 2020.

[18] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. **ACM Transactions on Computing for Healthcare (HEALTH)**, Vol. 3, No. 1, pp. 1–23, 2021.