BERT の教師なし分野適応による数学問題テキスト構文解析の精度向上要因の分析

吉田琉夏　松崎拓也
東京理科大学　理学部第一部　応用数学科
1419099@ed.tus.ac.jp　matuzaki@rs.tus.ac.jp

概要

本研究では，BERT の分野適応によって数学テキストの構文解析精度が向上する要因およびその限界を明らかにする．また，BERT の分野適応の効果は分野固有のサブ言語である数式を表現する方式に依存することを示し，適切な方式を明らかにする．

実験の結果，BERT の分野適用を行う場合，数式全体を専用の特殊トークンで置き換える方式が最も構文解析精度が高く，分野適応を行わない場合に比べて約 4 ポイント向上した．また，このとき数式を含む構造に関する誤りが多く改善されていた．一方で，fine-tuning に用いた新聞係り受けコーパスでの出現頻度が低い構造に関する誤りは，BERT の分野適応のみでは改善されにくいことがわかった．

1　はじめに

数学テキスト・法律・技術文書など，読解に一定の専門知識が必要になるテキストの構文アノテーションは，高コストである．一方で，近年 BERT の使用により高精度な係り受け解析が実現できることが示され [1]，さらに，BERT の事前訓練に構文解析の対象とする分野のテキストを用いることの有効性が示されている [2]．そこで本研究では，構文注釈付きデータが不要な BERT の分野適応のみによって，数学テキストの解析精度がどこまで向上するかを分析する．具体的には，数学テキストの文節単位の係り受け解析において，Wikipedia のみで事前訓練した BERT を用いる場合と数学テキストでの事前学習を追加で行う場合を比較した．

また，数学テキストにはサブ言語である数式が頻出する．数式はメイン言語，例えば日本語とは語彙も文法も全く異なるため，メイン言語のトークンと同列に扱って BERT に入力するのが適切であるか否かは不明である．同様なことは化学式など分野固有のサブ言語を含む種々のテキストでも起きる．そこで本研究では，BERT を数学テキストに適用する際の数式の表現として，数式全体を 1 つの未知語トークンで表す場合，新たに追加した特殊トークンで表す場合，数式に現れる記号を BERT の語彙に加える場合，数式を数式専用 BERT に入力したときの [CLS] ベクトルで表す場合の 4 つを比較した．

実験の結果，BERT の分野適用を行う場合，数式を専用の特殊トークンで表す方式が最も係り受け解析精度が高く，分野適応を行わない場合に比べて約 4 ポイント向上した．しかし，さらに係り受け注釈付き数学テキストを訓練に使用した場合と比べるとなお約 3 ポイントの差がある．

そこで，①BERT の分野適応によって改善する誤りと②注釈付き数学テキストを用いた訓練によってのみ改善する誤りについて分析した．その結果，①では，数式を含むことが多い構造に関する誤り，②では，注釈付き新聞テキストでは低頻度な構造の誤りが多く改善されることがわかった．

2　方法

本稿で比較する係り受け解析手法では全て東北大学が公開している Wikipedia での事前学習済み BERT-base モデル [3]（以下，東北大 BERT と呼ぶ）を用いた．そして，以下の 3 点について次節の表 3 の様に組み合わせた計 6 つの解析モデルを作成した:

1. 数学生テキストを用いた追加の事前訓練の有無
2. 4 つの数式表現方式のいずれを用いるか
3. 係り先予測の fine-tuning において，新聞に加えて数学係り受けデータを用いるか否か

2.1　使用データ

数学生テキストとして，1957 年〜2020 年の大学入試問題を収集し，数式を MathML 表記したもの [4]（以下，MRaw と表記）を用いた．注釈付き新聞

表 1 使用データのサイズ. 数式%は各数式を 1 つの形態素と数えたときの全形態素に占める数式の割合

	文数	文節数	形態素数	数式%
数学生テキスト	155,034	(不明)	2,867,847	16.35%
数学係り受けデータ	9,091	76,277	161,621	17.07%
新聞係り受けデータ	37,817	368,035	1,044,905	なし

表 2 4 つの方式による数式を含む入力の表現の例

入力文	$f(x) = 0$ の解は $x = 1$ である
[UNK]	[UNK], の, 解, は, [UNK], で, ある
[MATH]	[MATH], の, 解, は, [MATH], で, ある
Mixed	f, (, x,), = ,0, の, 解, は, x, = , 1, で, ある
ExprBERT	$\mathbf{e}_1$, の, 解, は, $\mathbf{e}_2$, で, ある [1)]

テキストとしては，京都大学テキストコーパス Ver.4（以下，KTC）[5] を用いた．注釈付き数学テキストとしては，1997 年〜2011 年のセンター試験と 1994 年〜2014 年の国立大学 2 次試験数学問題に KTC と同じ基準で係り受け構造を付与したもの（以下，MDep）を用いた．

2.2 数式の表現方法

数式の表現として，以下の 4 つを比較した．表 2 にそれぞれの例を示す．

[UNK] 数式全体をまとめて東北大 BERT の語彙の未知語トークン [UNK] で表す

[MATH] 新規追加した特殊トークン [MATH] で表す

Mixed 数式に現れる個々の記号（以下，数式トークン）を語彙に加え，日本語のサブトークンと数式トークンが混じった列に BERT を適用する

ExprBERT 数式トークン列を数式専用 BERT に入力したときの [CLS] ベクトルを数式に対応する単語埋め込みとして東北大 BERT に入力する

数式トークン列は，アルファベットや演算子，上付き・下付きを表す記号等からなる LaTeX 風の記法である．数式トークンの語彙サイズは 241 で，例えば，$a_n = n^2 + 100n$ であれば

$$a, \_\{, n, \}, =, n, \hat{}\{, 2, \}, +, 100, n$$

のように表す．数式専用 BERT の構成は BERT-base と同一である．

2.3 BERT の分野適応の手順

東北大 BERT を初期値とし，MRaw を訓練データとする Masked Language Modeling (MLM) タスクによって BERT の分野適応を行った．タスクの設定は Devlin ら [6] に従った．数式の表現として ExprBERT を用いる場合は，マスクしたトークンを予測する代わりに，東北大 BERT の単語埋め込みおよびミニバッチ中の数式に対する数式専用 BERT による埋め込みの中から，マスクしたトークンないし数式に対応するベクトルを選択するタスクとした．

2.4 係り受け解析モデル

Dozat と Manning[7] の Biaffine モデルを BERT と組み合わせて文節単位の係り受けに適用した．Biaffine 関数への入力ベクトルは，係り元の文節の語形および係り先候補の主辞に相当する形態素それぞれの先頭のサブトークンに対する BERT の出力ベクトルとした．主辞・語形の定義は文献 [8] に従った．

2.5 係り先予測 fine-tuning の手順

訓練データとして KTC，あるいは KTC と MDep を連結したものを使用して係り先予測タスクへの fine-tuning を行い，評価データとして MDep を用いた．訓練データとして KTC のみを用いる場合は，その 1 割を検証データとし，KTC と MDep を連結したものを用いる場合は，MDep の 1 割を検証データ，残りを訓練データとし，MDep を 5 分割した交差検証で評価した．

3 実験結果

表 3 に，事前訓練データ・係り受け訓練データ・数式の表現方式の 6 つの組み合わせに対する評価結果を示す．全体に対する精度に加え，数式を含む文節を MATH，含まない文節を WORD と表すとき，係り元→係り先の 4 つのパターンごとの F_1 スコアを示した．Biaffine 関数の出力に基づく係り先予測の結果が交差する係り受け関係を生じるケースは極めて少なかったため，CKY 等の解析アルゴリズムは用いず，係り先予測の結果を評価した．

表 3 より，構文解析済みデータが存在しない場合であっても，数学問題データで MLM を行うことによってどの数式表現方式でも精度が向上していることがわかる．特に数式を [MATH] で表す場合は，数式を [UNK] で表す場合に比べて約 4 ポイントの精度向上を達成しており，とりわけ MATH → MATH 型の F_1 スコアは約 11 ポイントも向上している．

BERT の分野適応のみを行う場合，ExprBERT, Mixed, [MATH] の 3 つの数式表現方法を比較すると，[MATH] の結果が最も良好で，次いで Mixed, ExprBERT の順になっている．しかし，数式を

1) $\mathbf{e}_1$, $\mathbf{e}_2$ はそれぞれ対応する数式トークン列を数式専用 BERT に入力したときの [CLS] ベクトルを表す

表 3　事前訓練データ・係り受け訓練データ・数式の表現方式の 6 つの組み合わせに対する評価結果

事前訓練データ [2]	Wiki	Wiki+MRaw	Wiki+MRaw	Wiki+MRaw	Wiki	Wiki+MRaw	係り受け
係り受け訓練データ	KTC	KTC	KTC	KTC	KTC+MDep	KTC+MDep	関係
数式の表現	[UNK]	ExprBERT	Mixed	[MATH]	[MATH]	[MATH]	の数
全体 (精度)	88.69	91.60	92.17	92.93	96.07	96.25	58095
MATH → MATH (F_1)	82.32	91.82	90.10	93.36	96.79	97.46	6875
MATH → WORD (F_1)	89.91	92.76	93.40	93.74	96.13	96.55	15619
WORD → MATH (F_1)	85.30	88.51	89.60	91.34	96.06	96.24	9593
WORD → WORD (F_1)	90.73	91.98	92.89	92.91	95.86	95.75	26008
交差を含む文 (%)	0.77	1.66	0.55	0.79	0.25	0.31	

表 4　BERT の分野適応により改善した誤りの内訳

誤りタイプ	内訳 (%)
並列された数式との同格関係に関する誤り	30%
数式の並列の解析誤り	27%
ガ格を持たない命令形に関する誤り	8%
その他	35%

[MATH] と表す場合であっても BERT の分野適応のみ行う場合と，さらに MDep を用いて訓練する場合との間で約 3 ポイントの精度の差が生じている．

4　誤りの改善例の分析

BERT の分野適応による構文解析精度の向上の要因およびその限界を明らかにするために，誤りの改善例を分析した．具体的には，事前訓練データ：係り受け訓練データ：数式表現の組み合わせを

設定① = Wiki : KTC : [UNK]

設定② = Wiki+MRaw : KTC : [MATH]

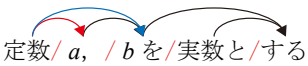
設定③ = Wiki+MRaw : KTC+MDep : [MATH]

と名付けるとき，設定①から設定②で改善される誤り (§4.1)，および設定②から設定③で改善する誤り (§4.2) について 5 分割交差検証のうちの 1 つ分から改善された誤りを 100 個サンプリングし，分析した．実際の改善例は付録 §6 に示す．

また，数式を [MATH] で表す単純な方式が，数式の構成要素まで考慮する方式を上回った理由を探るため，[MATH] 方式および ExprBERT による数式の埋め込みを観察した (§4.3).

4.1　BERT の分野適応による改善例の分類

設定①から設定②で改善した誤りを分類した結果を表 4 に示す．表中の並列された数式との同格関係の誤りとは，下図の様に，並列された数式のタイプを表す名詞が，正しい係り先である最後の並列要素に係らない誤りのことである．[3]



2)　Wiki は Wikipedia の略である．

数式の並列の解析誤りとは，並列要素が，右隣の並列要素に係らない誤りのことである:

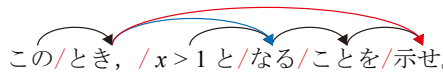
ガ格を持たない命令形に関する誤りとは，命令形にガ格あるいは無格の名詞が係る誤りである．特に並列された数式の一部が命令形に係る誤りが改善された例が複数見られた．

これらの誤りが改善された理由として以下が予想される．まず，Wikipedia では未知語トークンは点在して出現しやすいのに対し，特殊トークンを用いた MLM により，数式同士が近くに出現しやすいことが学習される．さらに，KTC による係り受け訓練で，近接する同種のトークンは，並列関係になりやすいことが学習されることで数式の並列を含む構造が正しく解析されやすくなる．

4.2　数式係り受けデータによる訓練でのみ改善した例の分類

設定②から設定③で改善した誤りを分類した結果を表 6 に示す．「A は B とする」型の誤りは，「A は」の文節が，「する」に係ってしまい，「A は B と (ともに何かを) する」という誤った解釈に対応する構造となるものである: [4]

条件が命令形に係る誤りとは，下図の様に「条件が成り立つとき (のみ)〜せよ」という誤った解釈に対応する構造を出力する誤りである:



3)　この項の図では，黒矢印は正解，赤矢印は設定①，青矢印は設定②で出力される係り受け関係を表す．

4)　この項の図では，黒矢印は正解，赤矢印は設定②，青矢印は設定③で出力された係り受け関係を表す．

表 5　異なる数式表現による数式埋め込みの比較

2 次関数 $y = ax^2 + bx + c$ に似ている数式				点 $(0, 4)$ に似ている数式			
ExprBERT	類似度	[MATH]	類似度	ExprBERT	類似度	[MATH]	類似度
$y = ax^2 + bx + c \cdots$	0.952	$y = x^2$	0.978	$(4, 0)$	0.961	$R(b + 1, (b + 1)^2)$	0.949
$y = a(x - b)^2 + c$	0.943	$y = x^2 - ax + b$	0.958	$(0, 3)$	0.959	([アイウ]	0.948
$y = x^3 + ax^2 + bx + c$	0.940	$y = x^2 + (p - 1)x + 3$	0.954	$(0, 5)$	0.956	$P(-1, 3)$	0.940
$y = x(x - a)(x - b)(x - c)$	0.937	$y = (9/4)x^2 + ax + b$	0.954	$(0, (14/3))$	0.956	Q	0.940
$y = -ax^3 + bx + c$	0.933	$y = x^2$	0.950	$(-(7/2), 0)$	0.955	A	0.938

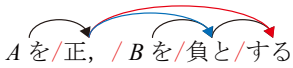
表 6　数式係り受けデータを用いた訓練で改善した誤りの内訳

誤りタイプ	内訳 (%)
数式の並列の解析誤り	22%
並列された数式との同格関係に関する誤り	14%
「A は B とする」型の誤り	11%
条件が命令形に係る誤り	9%
部分並列の解析誤り	8%
その他	38%

表 7　データの並列要素数

	KTC		MDep	
	個数	内訳	個数	内訳
2 個	964	78.95%	1650	62.43%
3 個	191	15.64%	743	28.11%
4 個	45	3.69%	203	7.68%
5 個	14	1.15%	22	0.83%
6 個以上	7	0.57 %	25	0.95%
合計	1221		2643	

部分並列の解析誤りとは，「父は山，母は海が好きだ」の様な構造における解析誤りである:



§4.1 と §4.2 で共通して多く見られた並列に関する誤りの改善例を比べると，§4.1 の改善例の多くでは並列要素が 2〜3 個であるのに対し，§4.2 の改善例では並列要素をそれ以上含むものが多かった．表 7 に KTC と MDep における読点で区切られた名詞並列中の要素数の分布を示す．この形の並列は KTC では MDep に比べ少数で，かつ並列要素が少ないものが主である．このため，多数の要素からなる並列構造に関する誤りは MRaw での MLM と KTC での訓練のみでは十分に改善できなかったと考えられる．

「A は B とする」型の誤りおよび条件が命令形に係る誤りは，正解・誤りいずれの構造も文法的にありうるが，数学テキストで主である解釈に対応する方の構造が，新聞では出現しにくいため，MDep での学習によりはじめて改善されたと考えられる．

また，部分並列の解析誤りは，この構造が KTC では 1%未満の文しか出現しないのに対し，数学テキストでは頻出すること，さらに「母は海が好きだ」に対する正しい構造が「父は山,」が付加されることで誤りになるという特殊性により，KTC のみでの係り受け訓練では改善されにくかったと考えられる．

4.3　異なる数式表現による数式埋め込みの比較

数式を [MATH] で表す単純な方式が，数式の構成要素まで考慮する方式を上回った理由を探るため，[MATH] 方式および ExprBERT による数式の埋め込みを観察した．表 5 に，(1)“2 次関数 $y = ax^2 + bx + c$ の... ”および (2)“ ... 2 点 $(0, 4)$, $(2, k)$ を... ”の下線部の数式を，文脈とともに設定②の BERT に入力した時と，文脈なしで ExprBERT に入力した時に得られる埋め込みと cos 距離が近くなる入力を示した．

表 5 より，入力 (1) に対しては，ExprBERT による数式埋め込みでは，2 次関数に加え 3 次関数の埋め込み，[MATH] 方式では，2 次関数との類似度が高いことが分かる．入力 (2) に対しては，ExprBERT による数式埋め込みでは，座標の埋め込み，[MATH] 方式では，点を表す変数との類似度が高いことが分かる．[5)]

これらの例より，[MATH] 方式による数式埋め込みは，係り受け解析を目的とする場合，数式の特徴を十分に捉えていることがうかがえる．よって，数式の表現が単一の特殊トークン [MATH] であっても，文脈によって数式同士の類似度が予測できるため，より複雑な ExprBERT による数式埋め込みと同等以上の効果があったものと思われる．

5　おわりに

BERT の分野適用を行う場合，数式を専用の特殊トークンで表す方式が最も構文解析精度が高いことがわかった．さらに，BERT の分野適用のみで，数式を含む構造に関する誤りが多く改善されることがわかった．しかし，fine-tuning に用いた注釈付き新聞テキストで低頻度な構造は改善されにくいことがわかった．

5) [アイウ] はセンター試験での穴埋め表記であり，([アイウ] はある点の x 座標のみの数式である．

謝辞

数学問題テキストを提供頂いた広松芳紀様および数学テキスト係り受けデータを提供頂いた「ロボットは東大に入れるか」プロジェクトの皆様に深く感謝いたします.

参考文献

[1] 柴田知秀, 河原大輔, 黒橋禎夫. Bert による日本語構文解析の精度向上. 言語処理学会 第 25 回年次大会, 2019.

[2] Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. Dependency parsing of biomedical text with bert. In **BMC Bioinformatics**, 2020.

[3] https://github.com/cl-tohoku/bert-japanese.

[4] 広松芳紀. 大学入試数学問題集成, 2022. https://mathexamtest.web.fc2.com/index.html.

[5] 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト. 言語処理学会 第 3 回年次大会, pp. 115–118, 1997.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In **North American Chapter of the Association for Computational Linguistics (NAACL)**, 2019.

[7] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In **5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings**, 2017.

[8] 工藤拓, 松本裕治. チャンキングの段階適用による係り受け解析. 情報処理学会, 2002.

6　付録

6.1　BERT の分野適応の手順

パラメータ最適化の最大エポック数は 50 で，検証データに対するロスが最小になるエポック数を選んだ．ミニバッチサイズは 16 とした．最適化手法として AdamW を用い，学習率は 5×10^{-5} とした．

6.2　係り先予測 fine-tuning の手順

最大エポック数を 5 とした以外は §6.1 と同じ設定でパラメータ最適化を行った．

6.3　BERT の分野適応により改善した誤りの実例

以下では，黒矢印は正解，赤矢印は設定①，青矢印は設定②で出力された係り受け関係を表す．

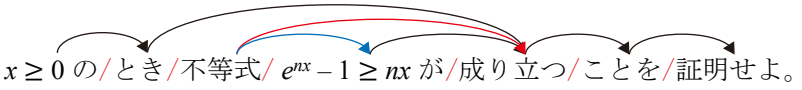


図 1　並列された数式との同格関係に関する誤りの実例 1

図 2　並列された数式との同格関係に関する誤りの実例 2

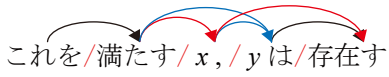


図 3　数式の並列の解析誤り

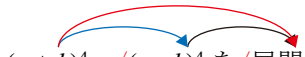


図 4　ガ格を持たない命令形に関する誤り

6.4　数学係り受けデータを用いた訓練で改善した誤りの実例

以下では，黒矢印は正解，赤矢印は設定②，青矢印は設定③で出力された係り受け関係を表す．

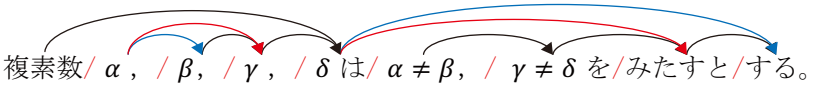


図 5　「A は B とする」型の誤り，部分並列の解析誤りの実例

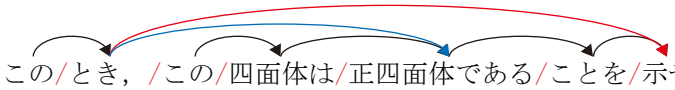


図 6　条件が命令形に係る誤りの実例