WikiText-JA 構築による BERT 事前学習の効率化

小川 晃 [1,a)]　友利 涼 [1,a)]　亀甲 博貴 [2,b)]　森 信介 [2,b)]

1. はじめに

自然言語処理における様々なタスクにおいて、必要となるデータセットの整備は容易ではない。例えば感情分析を行う際には、テキストごとに「肯定的」「否定的」といったラベルが付与されたデータセットが必要となるが、そのデータセットは作成に多大なコストを要するため、用意が難しいのが現状である。そうした問題を解決するために、自然言語処理では入手が容易なラベル未付与のテキストを生コーパスとして活用する方法が研究されている。具体的には、Wikipedia や SNS といったデータに含まれる知識を事前学習することで、データセットの不足を補うといったアプローチである。Bidirectional Encoder Representations from Transformers (BERT) [1]はこうした事前学習の一つであり、自然言語処理の多くのタスクにおいて精度向上に大きく寄与することが報告されている。

一方で、BERT をベースとしたモデルはパラメータが非常に多く、学習に膨大な時間がかかることが知られている。例えば、京都大学大学院の黒橋研究室で公開されている BERT 日本語 pretrained モデルでは 1 GPU （GeForce GTX 1080 Ti を利用）で 30 epoch の学習に約 30 日を要したと報告されている[2]。そうした中、BERT の事前学習において、finetuning するタスクに合わせたドメインのコーパスを利用することでその性能が向上したという事例が報告されている[3][4]。

そこで本研究では、汎用性のある小規模なコーパスである WikiText-JA を用意する。その上で、そのコーパスに対象とするタスクのドメインに関わるコーパスを組み合わせたデータセットで BERT を事前学習することで、学習時間を削減しつつもモデルの性能を大規模コーパスで学習した際のそれとほぼ同等に維持することができるのかを調査する。実際に作成した BERT によって、テキスト内における固有な表現を抽出するタスクである固有表現認識（Named Entity Recognition; NER）を 2 種類のドメイン（ゲーム解説、レシピ）に対して実施したところ、日本語版 Wikipedia から無作為に抽出した記事で構成されたデータセット、および日本語版 Wikipedia 全記事で新たに構築した BERT に基づく NER の精度と同等以上の性能を示す結果となった。

1　京都大学大学院情報学研究科
2　京都大学学術情報メディアセンター
a)　{ogawa.akira.86c, tomori.suzushi.72e}@st.kyoto-u.ac.jp
b)　{kameko, forest}@i.kyoto-u.ac.jp

2. 関連研究

2-1. BERT

BERT は Transformer [5]をベースにしたモデルである。Transformer は self-attention のみを利用したモデルであり、長距離の依存関係を捉えることができるという特質を有する。この技術に基づく BERT は 2 つのステップ、具体的には大規模な生コーパスで事前学習を実施した後、各タスクで finetuning を行うことで構築される。

BERT の事前学習は 2 つのタスクを学習する形で実施される。1 つは文の一部の単語を[MASK]に置き換え、それに相当するトークンを推測する Masked Language Model、もう 1 つは 50 %の文を実際の次文とつなげた正例と残りの 50 %の文を無作為に抽出した文とつなげた負例を用意し、これらを識別する次文推定のタスク(Next Sentence Prediction)である。この 2 つのタスクで事前学習した BERT に基づき、その Transformer の上に各タスクに応じた最終層を加えることで、学習結果を他の異なるタスクに応用することが可能となる。その結果、文ペア分類問題、質問応答(SQuAD)、系列ラベリング問題など様々なタスクにおけるモデルの性能を向上させられることが報告されている[1]。

こうした様々なタスクで効果を発揮している BERT は、非常に多くのパラメータを学習時に必要とするモデルとして知られている。パラメータが多いということは学習に

多くのメモリと時間を要することに直結する。

こうした問題への対処の 1 つとして、A Lite BERT (ALBERT)というモデルが現在提案されている[6]。このモデルでは Attention を用いたレイヤーでパラメータを共有させる Cross-Layer Parameter Sharing、単語分散表現の埋め込みを分解する Factorized Embedding、そして正例として連続する二文、負例として正例の二文を入れ替えたものを用意し、それぞれの例で文の順序の正しさを評価する Sentence Order Prediction (SOP) が導入されている。ALBERT はパラメータ数が圧倒的に少なく、結果、学習におけるメモリの利用効率の向上が示されている。一方で、この手法は学習速度の向上にはあまり寄与しない。

2-2. 固有表現認識

固有表現認識 (NER) は、テキスト内における固有な表現を抽出する技術である。一般的な NER タスクは新聞記事を対象とし、人名クラス、地名クラス、組織名クラスなどの固有表現クラスを取得することを目的に実施される[7][8]。近年では将棋解説文における戦型や盤上の配置などの認識を目的としたゲーム解説 NER [9]や料理レシピのテキストにおける食材名や道具名などの認識を目的としたレシピ NER [10]など、分野特有の固有表現体系が定義された NER が提案され、それぞれ自動解説文[9]や手順書理解[11]などの基礎技術となっている。

こうした固有表現の認識器は一般に、対象とするタスクの単一データセットに基づいて学習が行われる。その一方で、異なるソースからの追加データを活用することで分類精度を向上させられることが知られている[12]。この手法は一般に転移学習と呼ばれ、転移学習と約 6,000 文程度のラベル付きデータを用いてバイオ医療 NER の精度向上を達成したことが報告されている[13]。

3. WikiText-JA

事前学習に用いるコーパスとして本研究では WikiText-JA を新たに作成する。図 1 にその概要を示す。

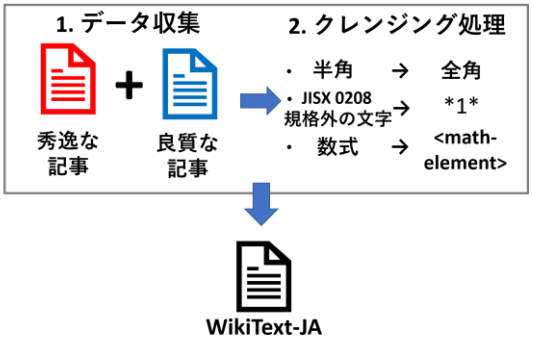

図 1 WikiText-JA の概要

WikiText-JA は、英語版 Wikipedia を活用した WikiText-103 [14]を参考に、日本語版 Wikipedia においてよくまとめられた記事として認定された秀逸な記事 (85 記事)、および良質な記事 (1,423 記事) を収集、後にクレンジング処理を実施したコーパスである。具体的なクレンジング処理として、文書内に現れる半角文字を全角に変換、JIS X 0208 の規格に沿わない文字や各種数式をアスタリスクや<math-element>でシンボル化などといった処理を行った。以上の過程を経たテキストデータをデータセットとしてまとめ、研究室のホームページにて現在公開している [1]。

WikiText-JA は日本語版 Wikipedia の様々なジャンルの記事を含んでおり、汎用的なデータセットとして利用することができる。加えて、文法的な正確さを保った長文を数多く含み、かつ文章間で表記のゆれが抑えられていることが期待される。したがって、正確な文法で記述された長文を偏りなく学習するにあたって、本コーパスの利用は適当であると考え、後述の BERT の事前学習に利用した。

4. 実験

4-1. 対象としたコーパス

実験に用いたコーパスの諸元を表 1、2 に示す。本研究では BERT の構築において、汎用的なコーパスとして前述の WikiText-JA を、特定のドメインに関わるコーパスとしてゲーム解説コーパス、およびレシピコーパスをそれぞれ用意した。ゲーム解説コーパス[9]は将棋の解説文からなり、分野特有の固有表現クラスとして人名クラスや戦型

[1]http://www.ar.media.kyoto-u.ac.jp/data/wikitext-ja/
(最終閲覧日: 2019.12.30)

データセット	行数	単語数	文字数
WikiText-JA	556,262	14,956,507	37,347,202
記事無作為抽出	835,595	13,972,443	37,344,631
ゲーム解説コーパス	741,405	11,676,046	25,952,440
レシピコーパス	11,788,955	214,058,693	529,672,344

表 1　BERT の事前学習に用いたコーパス諸元

データセット	文数	単語数	固有表現数	固有表現クラス数
レシピコーパス				
学習	1,614	34,802	14,058	
開発	66	1,326	527	21
テスト	67	1,345	597	
ゲーム解説コーパス				
学習	1,178	23,368	8,432	
開発	235	3,369	1,266	8
テスト	225	3,375	1,540	

表 2　NER に用いたコーパス諸元

クラス、プレイヤーの動作名クラスなど 21 種類が定義されている。レシピコーパス[10]は料理レシピのテキストであり、分野特有の固有表現クラスとして食材名クラスや道具名クラス、調理者の動作名クラスなど 8 種類が定義されている。これら 2 つのコーパスの NE は名詞句以外からも構成されており、調理者の動作名クラスやプレイヤーの動作名クラスなどは動詞で構成されている。

いずれのコーパスにおいても KyTea [15][2]を用いた単語分割を行い、さらに BPE [16]を適用してサブワードに分割、それを基本単位とした。また、WikiText-JA にレシピコーパス、またはゲーム解説コーパスを組み合わせて 2 種複合コーパスを作成、それらを BERT の事前学習に用いた。実験ではレシピコーパスとゲーム解説コーパスをそれぞれ用いて、レシピ NER とゲーム解説 NER を実施した。

4-2.　実験設定

本研究では BERT の事前学習において、前述の WikiText-JA、ゲーム解説コーパス、およびレシピコーパスを利用した。その上で、WikiText-JA と 2 つのドメイン特有のコーパスをそれぞれ組み合わせて 2 種複合コーパスを作成、そのデータセットに基づいて BERT の事前学習を行った。また、ベースラインとして前述の 3 つのコーパス単体、日本語版 Wikipedia 全記事（約 1,800 万文）、

[2] http://www.phontron.com/kytea/ (最終閲覧日: 2019.12.30)

データセット	ゲーム解説NER		
	Prec. (%)	Recall (%)	F-meas.
WixiText-JA	85.1	80.5	82.7
ゲーム解説コーパス	85.7	78.9	82.2
Wikipedia全体	**87.1**	81.9	84.4
記事無作為抽出+ゲーム解説	86.3	80.4	83.2
WikiText-JA+ゲーム解説	86.7	**82.6**	**84.6**

表 3 ゲーム解説 NER の実験結果（Precision Recall F-measure, 太字は最高精度の値を示す）

データセット	レシピNER		
	Prec. (%)	Recall (%)	F-meas.
WixiText-JA	87.4	89.9	88.6
レシピコーパス	89.9	92.2	91.0
Wikipedia全体	90.6	92.2	91.4
記事無作為抽出+レシピ	90.3	92.0	91.2
WikiText-JA+レシピ	**90.7**	**92.5**	**91.6**

表 4 レシピ NER の実験結果　(Precision Recall F-measure, 太字は最高精度の値を示す)

そして文字数に関して WikiText-JA と同サイズとなるように、日本語版 Wikipedia から無作為に記事を抽出したデータセット（以降、記事無作為抽出と呼称）でそれぞれ事前学習した BERT を用意した。ただし、日本語版 Wikipedia 全記事の BERT は KyTea で単語分割を行った後に 30 epoch で事前学習されたモデルを使用した。このモデルの事前学習には約 38 日の期間を要した。

事前学習における語彙サイズ（サブワードを含む）は 32,000 に設定し、入力の最大文長を 128 、バッチサイズを 32 として、200,000 step で実施した。2 種複合コーパスに基づく学習に要した時間は 1 GPU (GeForce GTX 1080 Ti を利用)を用いた場合、約 32 時間だった。

NER タスクでは、対象データセットを訓練データ、開発データ、テストデータに分割し、それぞれ 8 : 1 : 1 の割合になるようにした。その上で、そのタスクに対する finetuning を 4 epoch で実施した。

4-3. 実験結果

表 3、4 に各データセットに基づく事前学習による NER タスクの精度を示す。訓練、評価にはゲーム解説コーパスとレシピコーパスの 2 種類のコーパスを用いた。訓練、評価共にゲーム解説、レシピで個別に行った。

表 3、4 より、WikiText-JA とゲーム解説コーパス、もしくはレシピコーパスで事前学習した BERT は各コーパス単体、および記事無作為抽出と各ドメインのコーパスで事前学習した BERT のいずれよりも高い精度を示した。加えて、ゲーム解説 NER、レシピ NER ともに、WikiText-JA と対象ドメインで事前学習した BERT は日本語版 Wikipedia 全体で構築した BERT のそれと同等以上の精度を示していた。したがって、WikiText-JA とタスクのドメインに関わるコーパスを BERT の事前学習に利用することにより、小規模なコーパスサイズ、および短い事前学習の時間で日本語版 Wikipedia 全体に基づくそれと同等以上の性能を得られることが示唆された。

5. おわりに

本論文では汎用性のある小規模コーパスとして WikiText-JA を構築し、そのコーパスとタスクで対象とするドメインに関わるコーパスを組み合わせることで、BERT の事前学習を効率化することを目指した。特定のドメインにおける NER を実施した結果、その 2 種複合コーパスに基づいて構築された BERT は日本語版 Wikipedia から無作為抽出した記事のデータセット、および日本語版 Wikipedia 全体で事前学習した BERT による精度と同等、もしくはそれを上回る結果であった。今後は、WikiText-JA とドメイン特有のコーパスで事前学習した BERT の学習効率をさらに検証するため、学習が進んでいった際の各時点における、提案手法で構築した BERT と日本語版 Wikipedia 全体で事前学習した BERT との性能比較を実施する予定である。

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: 事前学習 of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] 柴田 友秀, 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, pp.205-208, 2019.

[3] Iz Beltagy, Kyle Lo, Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv preprint arXiv:1903.10676v3*, 2019.

[4] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, Jaewoo Kang. Bio-BERT: a pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NIPS2017)*, pp. 5998-6008, 2017.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942v3*, 2019.

[7] Erik F. Tjong Kim Sang and Fine De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL 2003)*, pp. 142-147, 2003.

[8] Lev Ratinov and Dan Roth. Design Challenges and Misconceptions in Named Entity Recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009)*, pp. 147-155, 2009.

[9] Shinsuke Mori, John Richardson, Atsushi Ushiku, Tetsuro Sasada, Hirotaka Kameko, and Yoshimasa Tsuruoka. A Japanese Chess Commentary Corpus. *Proceedings of the Tenth International Conference on Language Resources and Evaluation Conference (LREC 2016)*, pp.1415-1420, 2016.

[10] Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. Flow Graph Corpus from Recipe Texts. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pp. 2370-2377, 2014.

[11] Yoko Yamakata, Shinji Imahori, Hirokuni Maeta, and Shinsuke Mori. A method for extracting major workflow composed of ingredients, tools, and actions from cooking procedural text. *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW 2016)*, pp. 1-6, 2016.

[12] Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. Tuning multilingual transformers for language-specific named entity recognition. *In Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pp. 89–93, 2019.

[13] John M. Giorgi and Gary D. Bader. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics, 34*, pp. 4087-4094, 2018.

[14] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. *arXiv preprint arXiv:1609.07843,* 2016.

[15] Graham Neubig, Yosuke Nakata and Shinsuke Mori. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pp. 529-533, 2011.

[16] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with sub-word units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 1715-1725, 2016.