BERT を用いた Data Augmentation 手法の改善と JGLUE による評価

高萩恭介 [1]　新納浩幸 [2]
[1] 茨城大学大学院理工学研究科情報工学専攻
[2] 茨城大学大学院理工学研究科情報科学領域
22nm730l@vc.ibaraki.ac.jp
hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

概要

Data Augmentation は，教師あり学習においてその性能を改善させるために，訓練データを水増しする手法である．論文 [3] では，複数の BERT を使って文中の単語を別の単語に置換する Data Augmentation の手法を提案し，それを文書分類タスクに適用することでモデルの性能が改善することが示されている．本研究では，その手法における拡張文と置換対象単語の選択方法を変更したより適切にデータを拡張できる手法を提案する．実験では，複数のタスクに対して提案手法による Data Augmentation を試みた．その結果，提案手法は文書分類タスクの性能を改善できた．

1　はじめに

Data Augmentation は，教師あり学習の性能向上を図るために，訓練データを水増しする手法である．基本的には，ラベル付きデータに対して，ラベルを変えずにデータのみに簡単な変換を行い，それを新規のラベル付きデータとして訓練データに加える．

自然言語処理においては，テキスト中の単語をその単語の類似単語に置き換える Data Augemtation の手法がある [1]．しかし，BERT[2] を用いてタスクを解く場合に，同じ BERT の Masked Language Model を利用して単語置換を行っても効果はほとんどない．置換によって得られる単語の知識は，既にタスク処理を行う BERT に組み込まれているからである．

論文 [3] では，この問題を解消するために，タスクに利用する BERT とは別種の BERT を使って類似単語を得る手法を提案し，その手法が文書分類タスクの性能を改善させることが実験によって示されている．これは，タスクに利用する BERT に含まれない単語の知識を置換によって獲得できたからであると考えられる．

しかし論文 [3] で提案された手法は，一部のラベル付きデータに対する拡張ラベル付きデータを多く生成することがあり，それによって拡張後のデータセットに偏りが生まれる可能性が高い．そこで本研究では，論文 [3] で提案された手法を改善し，1 つのラベル付きデータに対して 1 つの拡張ラベル付きデータを生成するようにした．

また論文 [3] では，livedoor ニュースコーパス[1)] を用いた文書分類タスクのみで提案手法を評価しているが，他のタスクに対して手法が有効かどうかは不明である．そこで本研究では，日本語の言語理解ベンチマークである JGLUE[4] を用いて，文書分類，文ペア分類，QA の 3 つのタスクで提案手法を評価し，その有効性を検証した．その結果，提案手法は文書分類タスクのみに対して有効であることが示された．

2　関連研究

画像処理の分野では，画像を反転させる，画像の一部を切り取るといった Data Augmentation の手法がよく使われている．これらの手法は単純で，実装が簡単であるにも関わらず，Data Augmentation としての効果は高い．また現在は，ラベル付きデータに対する簡易的な変換による Data Augmentation だけではなく，様々なアプローチの手法が考案されている．

例えば，Mixup はデータとラベルを各々線形結合して，新たなデータを作成する手法である [5]．また，ラベル無しデータに対して何らかのラベルを付

1）https://www.rondhuit.com/download.html#ldcc

与して，訓練データを増やす手法である半教師あり学習や，GAN などの生成系ニューラルネットワークで画像を生成する手法も Data Augmentation に分類される [1].

自然言語処理の分野においては，画像処理と比べると Data Augmentation に関する研究が少なく，これまでに行われた研究も画像処理分野での Data Augmentation の研究を基にした二次的なものが多い．これは，扱われるデータである言語が離散的であることが原因とされる．この原因により，単純な変換による Data Augmentation で自然なデータを生成することは難しい．しかし，自然言語処理の分野においても，Data Augmentation に関する試みはいくつか行われており，効果的な手法も考案されている．

Wei らは EDA (Easy Data Augmentation) という言語モデルや外部データを必要としない簡易な Data Augmentation の手法を提案した [6]．この手法では，訓練データのテキストに対して，同義語置き換え，ランダム挿入，ランダム交換あるいはランダム削除の 4 つの操作を複数回ランダムに適用することによって，訓練データを拡張する．また，ある言語で記述された文書を別の言語 に翻訳し，その翻訳文を更に元の言語に翻訳する，いわゆる逆翻訳を利用して Data Augmentation を行う試みも多い [7][8]．さらに，Gun らは画像処理の分野で利用される Mixup 手法を応用し，文の埋め込み表現あるいは単語の埋め込み表現を混合する Sentence Mixup や Word Mixup を提案している [9]．他にも，Chen らは BERT のある層を混合する TMix に半教師あり学習を併用した MixText を提案している [10].

3 提案手法

論文 [3] では，タスク処理用の BERT とは異なる BERT を用いて文中の一部の単語を別の単語に置き換える手法が提案されている．この手法は，タスク処理用の BERT に含まれない知識を単語置換によって獲得できるため，有効であると考えられている．

またこの手法では，データセットに含まれる全てのテキストの中から TF-IDF が高い単語を指定個数取り出し，各単語を含むテキストに対して単語置換を行っている．しかしこのやり方では，同じテキストを基にした拡張テキストが複数生成されることがあり，それによって拡張後のデータセットに偏りが生まれる可能性が高い．

今回は，論文 [3] で提案された手法を改善し，既存のデータセットに含まれる各テキストに対して単語を 1 つ選び，その単語を別の単語に置換して新たにテキストを生成するようにした．これによって，1 つのラベル付きデータに対して，1 つの拡張ラベル付きデータが生成されることとなる．

単語置換は，テキスト中の置換対象の単語を Mask トークンに置き換えて，その Mask トークンの位置に入る単語を BERT の Masked Language Model で予測することで行う．また，単語置換とタスク処理にはそれぞれ別の BERT を使う．今回は，タスク処理に東北大学の乾研究室が公開している'bert-base-japanese-v2'[2)]，単語置換にストックマーク株式会社が公開している BERT[3)] を利用する．東北大版 BERT は日本語の wikipedia を用いて，ストックマーク版 BERT は日本語のビジネスニュースのコーパスを用いて事前学習が行われている．そのため，2 つの BERT は事前学習の段階でそれぞれ異なる知識を獲得していると考えられる．

提案手法の詳細な手順について以下に示す．

1. データセットからラベル付きデータ（ラベル+テキスト）を 1 つ取り出す．
2. テキストをトークン化する．
3. 置換対象単語のトークンを Mask トークンに置き換える．
4. トークン列を BERT 入力用の ID 列に変換する．
5. ID 列を BERT に入力し，各トークンに対する予測結果を取得する．
6. 予測結果の中から，Mask トークンに入ると予測された上位 100 単語を取得する．
7. 上位 100 単語の中から，次の条件を満たす最上位単語を選択する．
   (a) 単語が名詞である．
   (b) 単語は置換対象単語とは異なる．
8. 置換対象単語を選択した単語で置き換えたテキストを生成する．
9. 生成されたテキストに元のラベルを付けたものを拡張ラベル付きデータとする．
10. データセットから取り出していないラベル付きデータがあれば，1 に戻る．
11. 元のデータセットに生成した拡張ラベル付きデータを全て追加する．

また，上記の手順における置換対象単語の決定は

2) https://huggingface.co/cl-tohoku/bert-base-japanese-v2
3) https://qiita.com/mkt3/items/3c1278339ff1bcc0187f

次のように行う．

1. データセットに含まれる全てのラベル付きデータから，テキストのみを取り出してリスト化する．
2. 分かち書きによって，テキストのリストを名詞のみのトークン列のリストに変換する．
3. 変換後のリストを TF-IDF ベクトルに変換する．
4. 各テキストにおける TF-IDF 値が最も高い名詞を選び，それらを置換対象単語とする．

4 タスク

本研究では，提案手法の評価を行うために，日本語の言語理解ベンチマークである JGLUE[4] を用いる．JGLUE は文書分類タスクである Marc-ja と JCoLA，文ペア分類タスクである JSTS と JNLI，QA タスクである JSQuAD と JCommonsenseQA の計 6 つのタスクから構成される．今回は，この中から Marc-ja，JSTS，JCommonsenseQA の 3 つのタスクを利用し，評価を行うこととする．また，各タスクには，train（訓練）/dev（検証）/test（テスト）データの 3 つが用意されているが，論文執筆時点ではどのタスクのテストデータも公開されていない．そのため，各タスクでは，元の訓練データからいくつかデータを取り出すことで新たに訓練/検証データを作成し，元の検証データはテストデータとして使用することとする．今回の実験での各タスクの訓練/検証/テストデータの個数を表 1 に示す．なお，今回は各タスクの訓練データの個数を 100 個として，ファインチューニングに用いるデータが少ない状況を想定した実験を行う．また，各タスクでは訓練データを 5 パターン用意し，それら 5 つの評価平均を最終評価として使用する．

表 1 各データセットの構成

データセット	訓練	検証	テスト
MARC-ja	100	5,654	5,654
JSTS	100	1,457	1,457
JCommonsenseQA	100	1,126	1,126

4.1 Marc-ja

Marc-ja は文書分類用データセットであり，通信販売サイト「アマゾン」における商品レビューとそれに対する評価をまとめたコーパスである MARC (Multilingual Amazon ReviewsCorpus)[11] の日本語部分を元に構築されている．Marc-ja に含まれる各データは商品レビューと，ラベルである評価の二つからなる．ラベルは negative と positive の 2 種類であるため，タスクは 2 値分類となっている．評価指標には精度 (acc) を用いる．Data Augmentation では，各訓練データの商品レビューに対して単語置換処理を行う．

4.2 JSTS

JSTS は意味的類似度計算 (Semantic Textual Similarity, STS) のデータセットであり，YJ Captions Dataset[12] を利用して構築されている．STS は文ペアの意味的な類似度を推定するタスクで，正解の類似度は，0 (意味が完全に異なる) 〜 5 (意味が等価) の間の値として付与されるのが一般的である．JSTS の各データは文ペアとその類似度からなり，文ペアは YJ Captions Dataset のある画像に対する 2 つのキャプションで，類似度はクラウドソーシングによって決定されたものである．評価指標には，Pearson および Spearman 相関係数を用いる．Data Augmentation では，各訓練データの文ペアのうち，1 つめの文に対してのみ単語置換処理を行う．

4.3 JCommonsenseQA

JCommonsenseQA は，CommonsenseQA[13] という QA データセットの日本語版で，常識推論能力を評価することができる．JCommonsenseQA に含まれる各データは，問題文とそれに対する 5 つの選択肢，正解の選択肢を示すラベルから構成される．この選択肢のうち，問題文に対する正しい解答となるのは 1 つだけである．評価指標には精度 (acc) を用いる．Data Augmentation では，各訓練データの質問文に対して単語置換処理を行う．

5 結果

実験では，各タスクにおいて，BERT を訓練データでファインチューニングすることで，モデルの構築を行う．構築したモデルは，タスクに応じた評価指標でテストデータにより評価される．

各タスクにおいて，元の訓練データと Data Augmentaion を行った訓練データでそれぞれファインチューニングしたときのモデルの評価結果を図 1 に示す．結果としては，Data Augmentation を行った場合，MARC-ja のみモデルの性能が僅かに改善し，それ以外のタスクでは性能が悪化した．

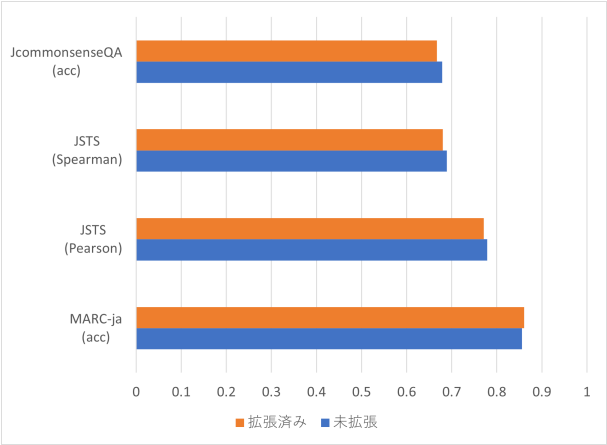

図 1　評価結果

6　考察

6.1　提案手法によるモデル性能低下の要因

JSTS と JCommonsenseQA の 2 つのタスクでは，提案手法を用いた場合にモデルの性能が低下してしまった．これは，新たに作成したデータの中に，テキストとラベルについての一貫性が失われたデータが多く含まれており，それらがノイズとなってモデルの性能を低下させたと考えられる．

例えば，JCommonsense において，問題文が「商品を確認するにはどこまで行くといい？」，正解の選択肢が「店頭」である訓練データが存在する．このデータに Data Augmentation を行うと，問題文が「自身を確認するにはどこまで行くといい？」に変更された．この場合，新たに生成されるデータは，問題文と正解の選択肢が矛盾したデータとなるため，訓練データとして適切ではない．

このように，QA や STS は，文書分類と比べて，取り扱うデータの一貫性が Data Augmentation によって失われやすく，それがモデルの性能低下に繋がったと考えられる．また，今回利用した 3 つのタスクのうち，MARC-ja はデータに含まれるテキストの文字数が全体的に多く，その他の 2 つのタスクはデータに含まれるテキストの文字数が少なかった．これも，JSTS と JCommonsense におけるモデル性能低下の要因の一つであると考えられる．

6.2　訓練データの量の影響

提案手法のような簡易的な Data Augmentation は，元の訓練データの量が少ない場合に効果的であり，量が十分な場合には効果がないと考えられている [6]．本節では，この点を確認するために，各タスクの訓練データの量を 100 個から 1000 個に増やして，同様の実験を行った．各タスクにおいて，元の訓練データと Data Augmentaion を行った訓練データでそれぞれファインチューニングしたときのモデルの評価結果を図 2 に示す．

結果としては，MARC-ja においても，Data Augmentation を行った場合にモデルの性能が悪化した．また，JSTS，JCommonsenseQA については，Data Augmentation の有無による性能の差が，訓練データが 100 個の場合と比べて大きくなった．この結果から，提案手法の効果は訓練データの量に影響し，量が大きくなるほど手法の効果が出にくくなると考えられる．

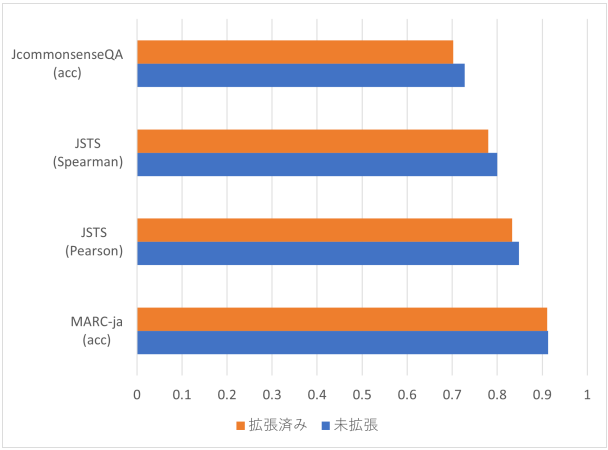

図 2　訓練データの量を 10 倍にしたときの評価結果

7　おわりに

本研究では，論文 [3] で提案された，BERT の Masked Lauguage Model を用いた単語置換による Data Augumentation の手法を改善した．論文 [3] の手法では，一部のラベル付きデータを基にした拡張ラベル付きデータが多く生成されることがあり，それによって拡張後のデータセットに偏りが生まれてしまう．本研究では，その問題を解消するために，1 つのラベル付きデータに対して 1 つの拡張ラベル付きデータを生成するようにした．また，改善した手法の効果を検証するために，日本語の言語理解ベンチマークである JGLUE に含まれる 3 つのタスクを利用して実験を行った．その結果，提案手法は文書分類タスクのみに対して効果があることが確認された．今後はこの手法に改良を加えて，より多くのタスクでモデルの性能を改善できるようにしたい．

参考文献

[1] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A Survey of Data Augmentation Approaches for NLP. **arXiv preprint arXiv:2105.03075**, 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] 恭介高萩, 浩幸新納. 複数の bert モデルを利用した data augmentation. Technical Report 4, 茨城大学工学部情報工学科, 茨城大学大学院理工学研究科情報科学領域, sep 2021.

[4] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. Jglue: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, 2022.

[5] H Zhang, M Cisse, Y Dauphin, and D Lopez-Paz. mixup: Beyond empirical risk minimization. iclr 2018. **arXiv preprint arXiv:1710.09412**, 2017.

[6] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. **arXiv preprint arXiv:1901.11196**, 2019.

[7] Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. Generalized data augmentation for low-resource translation. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5786–5796, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] Jiaao Chen, Yuwei Wu, and Diyi Yang. Semi-supervised models via data augmentationfor classifying interactive affective responses. **arXiv preprint arXiv:2004.10972**, 2020.

[9] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. **arXiv preprint arXiv:1905.08941**, 2019.

[10] Jiaao Chen, Zichao Yang, and Diyi Yang. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 2147–2157, Online, July 2020. Association for Computational Linguistics.

[11] Phillip Keung, Yichao Lu, György Szarvas, and Noah A Smith. The multilingual amazon reviews corpus. **arXiv preprint arXiv:2010.02573**, 2020.

[12] Takashi Miyazaki and Nobuyuki Shimizu. Cross-lingual image caption generation. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1780–1790, 2016.

[13] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. **arXiv preprint arXiv:1811.00937**, 2018.

A 付録

A.1 BERT の詳細設定

実験でのタスク処理に利用した BERT の詳細な設定は以下の通りである．

- モデル名：bert-base-japanese-v2
- 最大シーケンス長：512
- バッチサイズ：8
- 学習率：5e-05
- エポック数：4

A.2 実験結果の詳細

本節では，本研究で行った実験の評価結果の詳細について，それぞれ表 1，表 2 に示す．

表 2 5 章における評価結果の詳細

	MARC-ja		JSTS		JCommonsenseQA	
	acc		Pearson/Spearman		acc	
	未拡張	拡張済	未拡張	拡張済	未拡張	拡張済
No.1	0.863	0.871	0.770/0.689	0.754/0.683	0.682	0.674
No.2	0.855	0.856	0.794/0.694	0.791/0.685	0.674	0.664
No.3	0.854	0.857	0.779/0.681	0.741/0.637	0.670	0.656
No.4	0.855	0.874	0.764/0.683	0.783/0.694	0.676	0.660
No.5	0.855	0.847	0.786/0.695	0.787/0.699	0.692	0.684
平均	0.856	0.861	0.779/0.689	0.771/0.680	0.679	0.667

表 3 6 章における評価結果（訓練データ 10 倍）の詳細

	MARC-ja		JSTS		JCommonsenseQA	
	acc		Pearson/Spearman		acc	
	未拡張	拡張済	未拡張	拡張済	未拡張	拡張済
No.1	0.921	0.922	0.843/0.802	0.835/0.793	0.727	0.698
No.2	0.910	0.912	0.847/0.792	0.836/0.776	0.719	0.693
No.3	0.911	0.911	0.854/0.810	0.834/0.786	0.731	0.718
No.4	0.914	0.901	0.850/0.804	0.838/0.783	0.719	0.689
No.5	0.910	0.909	0.844/0.791	0.823/0.763	0.744	0.713
平均	0.913	0.911	0.848/0.800	0.833/0.780	0.728	0.702