小規模コーパスを利用した領域特化型 ELECTRA モデルの構築

伊藤陽樹
茨城大学
工学部情報工学科
17t4014y@vc.ibaraki.ac.jp

新納浩幸
茨城大学大学院
理工学研究科情報科学領域
hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

1 はじめに

自然言語処理のタスクを機械学習のアプローチで解決する場合，訓練データの領域 (ソース領域) とテストデータの領域 (ターゲット領域) が異なるという domain shift の問題が深刻である．BERT[1] のような事前学習済みモデルは下流のタスクによりモデルを fine-tuning するため，domain shift の問題に対処できる．さらに近年はターゲット領域のコーパスを用いて BERT の追加学習を行い，追加学習できたモデルを fine-tuning することで更なる精度向上がなされている．ただし BERT の追加学習には多大な計算機資源が必要であり，簡単に行うことはできない．また利用するターゲット領域のコーパスも大規模なものが想定されるが，そのようなコーパスが現実には入手できないことも多い．

本研究では上記の問題への対処として ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately)[2] の利用を試みる．

ELECTRA は BERT で用いられる Masked Language Modeling (MLM) を Replaced Token Detection (RTD) という手法に置き換えた事前学習モデルである．MLM は入力文に対していくつかの単語を [MASK] に置き換え，その [MASK] を推定することでモデルを学習させている．しかし BERT では [MASK] の置き換えが全体の 15%であり効率が悪い．その点を改良したのが RTD である．RTD では Generative Adversarial Network(GAN) の考えに基づいて生成モデルと識別モデルの 2 つのモデルを用意し，生成モデルによって生成された各トークンが，識別モデルによって入力文に対して置換されているかどうかの 2 値判定で事前学習を行っている．RTD では全てのトークンを学習に扱うことができるためより計算効率性が良く，RTD を利用した ELECTRA モデルは同一サイズの BERT モデルと比較しても優れた性能を発揮することが知られている．

実験ではまず一般的な小規模の ELECTRA モデルを構築した．次にターゲット領域の小規模コーパスを利用し，その ELECTRA モデルの追加学習を行うことで領域特化型の ELECTRA モデルを構築した．構築した領域特化型の ELECTRA モデルは，BERT-base よりも小さなモデルではあるが，ターゲット領域の文書分類タスクに対して，BERT-base よりも性能がよいことを確認できた．

2 関連研究

2.1 系列ラベリングの教師なし領域適応

ELMo や BERT のような文脈が考慮された分散表現を生成するモデルは，ラベル付けされていない大規模コーパスを用いて事前学習を行うことで，幅広い自然言語処理タスクにわたって良い性能を発揮することが知られている． しかし，これらのモデルは学習に Wikipedia やニューステキストといったコーパスを使用しているため，ターゲットのテキストの領域や書かれ方が事前学習コーパスと大きく異なる場合，このアプローチが有効であるかどうかは不明である． そこで，論文 [3] では，ターゲット領域のテキストを使用して分散表現を fine-tuning することで性能向上を図っている．

先行研究では，初期近代英語と Twitter の 2 つのテキストをターゲット領域としてテストしている．どちらも既存の事前学習コーパスとは大きく異なるが，提案された手法により通常の BERT モデルよりも実質的な改善が得られたことを示している．

2.2 Domain Tuning と Task Tuning

一般に，ターゲット領域が事前学習コーパスと異なる場合，文脈に応じた単語分散表現はタグ付けタスクには効果がない可能性があると言われている．これは，ラベル付けされたデータもターゲットのテキストと大きく異なる可能性があるため，教師なし

の領域適応では特に深刻である． この問題に対処するために，論文 [3] 中では教師なしの領域適応のための AdaptaBERT モデルの手法を提案している．

具体的には，以下の 2 つのアプローチを適用している．

Domain Tuning ターゲット領域のテキストを使って教師なしで BERT の言語モデルをチューニングする方法．例えば，Wikipedia で学習した BERT を Twitter のテキストを使って再学習することがこれに当たる．

Task Tuning 教師データを使って BERT をチューニングする方法．例えば，固有表現認識であれば CoNLL 2003 を使って BERT を含むモデル全体を学習させることがこれに当たる．

実験設定は，Domain Tuning と Task Tuning の組み合わせにより以下の 4 つを定めている．

- Frozen BERT
- Task-tuned BERT
- AdaptaBERT
- Fine-tuned BERT

Frozen BERT は，BERT を学習させず特徴抽出器として用いる方法である．Task-tuned Bert は，ソース領域の教師付きデータを使って BERT を学習させる方法である．AdaptaBERT は，ターゲット領域の教師なしデータを使って BERT をチューニングした後，ソース領域の教師付きデータを使ってモデルを学習させる方法である．最後の Fine-tuned BERT というのは，ターゲット領域のデータを使って BERT を含むモデル全体を学習させる方法を指している．

実験では Penn Parsed Corpus of Early Modern English(PPCEME) における品詞タグ付けと Workshop on Noisy User Text(WNUT) 2016 における固有表現認識について評価を行っている．品詞タグ付けでは，ソース領域のコーパスとして Penn Treebank (PTB) corpus of 20th century English，ターゲット領域のコーパスとして PPCEME を使っており，PTB は現代英語，PPCEME は 15 世紀から 17 世紀の英語をターゲットにしたコーパスとなっている．固有表現認識ではソース領域のコーパスとして Conference on Natural Language Learning(CoNLL) 2003，ターゲット領域のコーパスとして WNUT 2016 を使っており，CoNLL 2003 はニュースが対象で，WNUT は Twitter が対象となっている．

品詞タグ付けと固有表現認識の 2 つの実験結果から，Domain Tuning を使用してターゲット領域のコーパスで BERT を学習させることで性能が向上することが分かっている．ターゲット領域のラベル付きデータが大量に確保できない場合でも，ターゲット領域の教師なしデータを使って事前学習モデルをチューニングするだけで性能が向上するというは十分に実用的であるといえる．

3 ELECTRA モデルの利用

論文 [4] によると，事前学習モデルの学習時に用いられたコーパスによってバイアスを受けることが示されている．実際のシステムが対象にする領域に特化した事前学習モデルを構築すれば，それを利用することでその領域における精度を向上させることができる．しかし，現在の事前学習手法の多くは，効果を発揮するために莫大な計算資源を必要とする．計算量を増やして事前学習を行えば，下流タスクの精度が向上する場合がほとんどであるため，事前学習を行う際は下流タスクの精度だけでなく，計算効率性も考慮する必要がある．

そこで本論文ではより計算効率性の良い事前学習手法である RTD を採用した ELECTRA を利用する．

論文 [2] では，様々なサイズの ELECTRA モデルを学習し，計算量に対する下流タスクの性能を評価している．具体的には，自然言語理解のベンチマークの GLUE と質問応答技術のベンチマークの SQuAD に対する実験を行っている．それによると ELECTRA モデルは，他の最先端の自然言語処理モデルと比較しても，同じ計算量であれば，従来の手法よりも改善されることが示されている．例えば，RoBERTa および XLNet の 25%未満の計算量で，同等の性能を発揮することが分かっている．さらに効率化を進めると，単一の GPU で，4 日間で学習可能な ELECTRA-Small では，GPT よりも優れたパフォーマンスを発揮し，計算量は 30 分の 1 で済むことが分かっている．

4 実験

4.1 モデルの構築

日本語で事前学習を行った ELECTRA モデルを構築するために，公式 GitHub 上[1)]に公開されているプログラム `run_pretraining.py` を使用し，Google

1） https://github.com/google-research/electra

Colaboratory(Colab) 上の無料の TPU 資源と Google Cloud Storage(GCS) を利用した．Colab 上の TPU 資源を利用することで，GPU よりもさらに学習時間を短縮することが可能である．また，Colab 上の TPU は GCS を介してのみデータの入出力が可能であるため，GCS を利用する必要がある．

事前学習用のコーパスには比較対象である東北大学が公開した BERT(tohoku-BERT) と日本語 Wikipedia 全文を使用し，tokenizer も同じ Mecab-NEologd を使用する．tohoku-BERT の公式 GitHub[2] 上にあるプログラムを利用し，学習用コーパスの作成，テキストの前処理，語彙ファイルの作成，そして事前学習のための tensorflow データセットの作成を行っている．

4.2 モデルの評価

評価は，構築したモデルの小規模領域分野における文書分類タスクの性能により行った．fine-tuning 時に使用した評価用データは Livedoor-news コーパスを使用した．これは株式会社ロンウィットから公開されている livedoor ニュースの日本語ニュース記事を集めたデータセットである．各文書は URL，作成日時，タイトル，本文からなる構成だが，ここでは記事本文に属するカテゴリで数値ラベルを付けた．9 つのカテゴリに属する記事本文を訓練用データとテストデータに分け，訓練データでモデルを学習し，テストデータで記事本文からその記事のカテゴリを予測するという 9 値分類タスクを行い，その正解率で性能評価を行う．

各カテゴリの数値ラベルと含まれている記事数を表 1 に示す．

表 1 各カテゴリの数値ラベルと記事数

class	category	train	test
0	独女通信	87	696
1	IT ライフハック	87	696
2	家事チャンネル	86	692
3	livedoor HOMME	51	409
4	MOVIE ENTER	87	696
5	Peachy	84	674
6	エスマックス	87	696
7	Sports Watch	90	720
8	トピックニュース	77	616
sum		736	5895

2） https://github.com/cl-tohoku/bert-japanese

4.3 実験結果

作成したモデル ELECTRA-Small はパラメータが tohoku-BERT と同じ Base サイズではなく，Small サイズのパラメータで事前学習を行っている．これは事前学習における計算効率性も考慮しているためである．fine-tuning では学習は 50epoch まで行っている．epoch ごとに学習したモデルを保存し，各モデルでタスクの正解率が最も大きい値を選択した．結果を表 2 に示す．

モデルの比較対象として参考までに，株式会社シナモン AI から公開されている Senetence-Piece ベースの日本語 ELECTRA モデル (ELECTRA-SenetencePiece) の比較実験を行っている．このモデルは ELECTRA-Small と同じパラメータサイズで事前学習を行っているモデルである．

表からも明らかなように，ELECTRA-Small のモデルは，ELECTRA-SenetencePiece のモデルよりは高い正解率を出している．しかし，モデルのパラメータサイズが Base サイズより小さいため，比較対象の tohoku-BERT の正解率より約 4%ほど低い結果になっている．

表 2 fine-tuning の実験結果 (正解率)

model	正解率 (最高値)
tohoku-BERT	0.8835
ELECTRA-Small	0.8412
ELECTRA-SentencePiece	0.8024

4.4 追加学習

fine-tuning 時に領域に特化した小規模コーパスを使用して訓練を行っているが，ELECTRA モデルの事前学習の計算効率性に注目し，この小規模コーパスを使って追加で事前学習を行うことで，比較対象に匹敵するモデルが構築できる可能性がある．そこで，確認のために実験を行った．具体的には，先の実験で用いた Livedoor-news コーパスの記事本文を平文のまま取り出し，1 つの事前学習用データセットとして ELECTRA-Small の追加の事前学習を行った．

ELECTRA-Small は既に 1Mstep，学習時間にして 24 時間ほどの事前学習を行っている．このモデルにさらに 0.25Mstep，学習時間にして 10 時間ほどの追加学習を行った (ELECTRA-Small-1.25M)．

追加学習後のモデルで先ほどの fine-tuning を 5 回

実行し、5 回分の各々のモデルから最も高い正解率の値を取り出し、その平均と最高値を表 3 に表す．

表から明らかなように，比較的パラメータサイズの小さいモデルであっても，領域特化の小規模コーパスで追加学習を行うことで，tohoku-BERT を上回る ELECTRA モデルが構築できることを確認できた．

表 3 追加学習後の fine-tuning の実験結果 (正解率)

model	平均値	最高値
tohoku-BERT	0.8814	0.8835
ELECTRA-Small-1.25M	0.8834	0.8864

5 考察

5.1 モデルサイズ

以下の表 4 は，モデルサイズ別の TPU を利用した 1Mstep までの事前学習時間の予測値を表したものである．

表 2 の結果は事前学習モデルのパラメータサイズによる性能差があるため，厳密なモデル性能の比較とはいえない．しかし，tohoku-BERT と同じパラメータサイズの ELECTRA モデルを構築するには，表 4 のより 1 つの TPU 資源を利用しても 1 週間ほどの学習時間を要する．モデルサイズが大きくなればなるほど，事前学習には莫大な計算資源を必要とするため，事前学習モデルを構築することは困難になる．モデルの計算効率性と小さいモデルほどパラメータ効率の良い ELECTRA モデルであれば，Small サイズであってもそれなりの性能を発揮したがモデルサイズの違いによる性能差を覆すまでには至らない．より正確なモデル性能の差を確認するためには同一サイズの事前学習モデルを構築する必要がある．これらの調査は今後の課題である．

表 4 モデルの事前学習時間 (1Mstep)

model	学習時間
ELECTRA-Small	1d 22hs
ELECTRA-Base	7d 1h

5.2 追加学習の効果

表 2，表 3 の結果から，下流タスクの前に領域に特化した小規模コーパスでモデルの事前学習を行うことで，fine-tuning 後のモデルの精度向上につながることは明らかである．BERT や ELECTRA であってもモデルのパラメータサイズが大きくなれば，その分事前学習の時間がかかってしまうため，追加学習を行うことが困難である．しかし，Small サイズのモデルであれば，追加学習にかかる時間は Base サイズのモデルよりも遥に短くて済む．よって，ELECTRA-Small モデルに対して領域特化の小規模コーパスで追加学習を行うことは，より少ない計算量で性能を発揮することが可能であると言える．

6 おわりに

本論文では，ELECTRA モデルの計算効率性とスケールの際の性能の良さに着目し，ターゲット領域の小規模コーパスを用いて ELECTRA モデルの追加学習を行い，領域特化型の事前学習済みの ELECTRA モデルを構築した．構築した ELECTRA モデルは比較対象の tohoku-BERT よりも小さなモデルであるが，ターゲット領域の文書分類のタスクでは tohoku-BERT よりも高い性能を出すことができた．今後はより厳密なモデル同士の性能比較を行うために，同一サイズの事前学習モデルの構築を課題としたい．

謝辞

本研究は JSPS 科研費 JP19K12093 および 2020 年度国立情報学研究所公募型共同研究（2020-FC03）の助成を受けています．

参考文献

[1]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.

[2]Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

[3]Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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)*, pp. 4238–4248, Hong Kong, China, November 2019. Association for Computational Linguistics.

[4]新納浩幸, 白静, 曹鋭, 馬雯. Fine-tuning による領域に特化した distilbert モデルの構築. 人工知能学会全国大会論文集, Vol. JSAI2020, pp. 1E3GS902–1E3GS902, 2020.