単語と文書の関係によるグラフを用いた文書分類

中嶋寛武[1] 佐々木稔[1]
[1]茨城大学工学部情報工学科
{18t4064l, minoru.sasaki.01}@vc.ibaraki.ac.jp

概要

グラフ構造を利用した文書分類の手法の一つにRoBERTaGCNがある。RoBERTaGCNは、20NG、R8、Ohsumed、MR の４つのデータセットによる文書分類において、既存手法の中で最高の性能を誇っている。しかし、RoBERTaGCNは、文書-単語ノード間の関係と単語ノード間の関係は各ノード間の重みとして表現し考慮しているが、文書ノード間の関係は考慮できていない。そこで、文書の分散表現のコサイン類似度を文書ノード間の重みとしてグラフに追加することで、RoBERTaGCNの精度向上を試みた。実験の結果、RoBERTaGCNに対して、20NGで0.67%、Ohsumedで1.19%の分類性能の向上を確認できた。

1 はじめに

自然言語処理のタスクに文書分類がある。文書分類とは、与えられた文書に対して、事前に定義されたラベル群から適切なラベルを推定するタスクのことであり、その技術は、人間が文書を分類する作業を自動化するための技術として実社会でも応用されている。

文書分類の手法に RoBERTaGCN がある。RoBERTaGCN は BERT の大量ラベルなしデータを活用した大規模事前学習によって得られた知識と、GCN のトランスダクティブ学習の利点を組み合わせた手法である。この手法は、20NG、R8、Ohsumed、MR の４つのデータセットによる文書分類において、既存手法の中で最高の性能を誇っている。

しかし、RoBERTaGCN は、文書ノードと単語ノード間の関係と単語ノード間の関係は各ノード間の重みとして表現し考慮しているが、文書ノード間の関係は考慮できていない。そこで、文書ノード間の関係をノード間の重みとしてグラフに追加することでRoBERTaGCN の精度向上を試みた。具体的には、各文書を BERT モデルに入力し、その最終の隠れ層における CLS ベクトルを得る。そして各文書の CLS ベクトルのコサイン類似度を計算し、それらを文書ノード間の重みとして追加した。これにより、文書ノード間の関係を考慮したグラフが作成でき、RoBERTaGCN の各データセットにおける正答率を上回ることを期待した。

コサイン類似度の閾値を 0.5 と 0.95～0.995 までの間 0.005 刻みで設定して実験を行うことで、提案手法の性能を評価するとともに、各データセットにおける最適なコサイン類似度の閾値についても検討を行った。

2 関連研究

Kipf ら[1]は、畳み込み操作によって局所的に特徴量を抽出する畳み込みニューラルネットワーク(CNN）を、グラフデータに応用できるようにしたGCN を提案した。Kipf らの GCN による文書分類は2017 年当時、最高の精度で分類することができていた。

Yao ら[2]は、文書分類のタスクに、GCN を適用した手法である TextGCN を提案した。TextGCN は文書ノードと単語ノードを同一のグラフ(異種グラフ)上に表現し、それを GCN への入力としている。TextGCN は 2018 年当時、文書分類タスクにおいてstate-of-the-art を達成した。

Yuxiao ら[3]は、BERT の大量ラベルなしデータを活用した大規模事前学習によって得られた知識と、GCN のトランスダクティブ学習の利点を組み合わせた手法である BertGCN[i]を提案した。BertGCN では、各文書を BERT に入力して、その出力から文書ベクトルを抽出し、それを文書ノードの初期表現として文書と単語の異種グラフとともに GCN へ入力する。

[i] 事前学習済みモデルによってモデルの呼び名が区別されている。bert-base を使用したモデルが BertGCN、roberta-base を使用したモデルが RoBERTaGCN である。

BertGCN は現在、文書分類タスクにおいて state-of-the-art を達成している。

3 提案手法

提案手法の動作概要図を図 1 に示す。まず、文書を入力として単語と文書の異種グラフを構築する。次に、そのグラフ情報（重み行列と初期ノード特徴行列）を BERT と RoBERTaGCN に入力し、それぞれにおける予測結果を得る。最後に、それぞれの予測結果の線形補間を計算し、その結果を最終的な予測として採用する。尚、3.1 節の内容が本研究における RoBERTaGCN の改良点であり、3.2 節と 3.3 節の処理は RoBERTaGCN に準拠している。

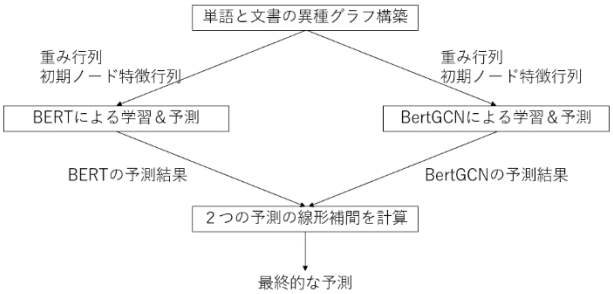

図 1：提案手法の動作概要図

3.1 異種グラフ構築

単語ノードと文書ノードを含む異種グラフを構築する。まず、ノードi,j間のエッジの重みを以下式(1)のように定義する。単語ノード間のエッジの重みには自己相互情報量（PPMI）を使用する。単語ノードと文書ノード間のエッジの重みには tfidf 値を使用する。RoBERTaGCN では、文書ノード間のエッジの重みは定義していなかったが、本研究では、各文書ベクトルの類似度を文書ノード間の重みとして定義する。具体的には、各文書を BERT に入力し、その最終の隠れ層から各文書の先頭に追加した特殊トークンである[CLS]における重みである CLS ベクトルを得る。そして各文書の CLS ベクトルのコサイン類似度を計算し、予め設定した閾値を超える類似度を文書ノード間の重みとして付加する。

$$A_{i,j}=\begin{cases} COS\_SIM(i,j), & i,j\ are\ documents\ (i\neq j) \\ PPMI(i,j), & i,j\ are\ words\ (i\neq j) \\ TF-IDF(i,j), & i\ is\ document, j\ is\ word \\ 1, & i=j \\ 0, & otherwise \end{cases} \quad (1)$$

次に、GCN に入力する初期ノード特徴行列を作成する。BERT を用いて文書埋め込み表現を取得し、それらを文書ノードの入力表現として扱う。文書ノードの埋め込み表現X_{doc}は、文書数n_{doc}と埋め込み次元数dを用いて$X_{doc}\in\mathbb{R}^{n_{doc}\times d}$で表される。総じて、初期ノード特徴行列は以下式(2)で与えられる。

$$X=\begin{pmatrix} X_{doc} \\ 0 \end{pmatrix}_{(n_{doc}+n_{word})\times d} \quad (2)$$

3.2 GCN への入力と学習

式(1)と(2)で示したノード間のエッジの重みと初期ノード特徴行列を GCN に入力し、学習を行う。第i層の出力特徴行列$L^{(i)}$は式(3)で計算される。

$$L^{(i)}=\rho\left(\tilde{A}L^{(i-1)}W^{(i)}\right) \quad (3)$$

ρは活性化関数であり、$\tilde{A}$は正規化された隣接行列である。$W^{i}\in\mathbb{R}^{d_{i-1}\times d_i}$は第$i$層における重み行列である。$L^{(0)}$は$X$であり、モデルの入力特徴行列になっている。GCN の出力は文書ノードの最終表現として扱われ、分類のためにその出力は Softmax 関数に入力される。GCN の出力による予測は以下式(4)で与えられる。

$$Z_{GCN}=softmax\left(g(X,A)\right) \quad (4)$$

3.3 BERT と RoBERTaGCN による予測の補間

BERT 埋め込み表現を直接扱う補助分類器で RoBERTaGCN を最適化することにより、より高速な収束と性能向上につなげている。具体的には、文書埋め込み表現Xと重み行列Wを softmax に直接送り込むことで BERT による補助分類器を作成する。補助分類器による予測は以下式(5)で与えられる。

$$Z_{BERT}=softmax(WX) \quad (5)$$

そして、RoBERTaGCN からの予測であるZ_{GCN}と BERT（補助分類器）からの予測であるZ_{BERT}で線形補間を行い、その線形補間の結果を最終的な予測として採用する。線形補間の結果は以下式(6)で与えられる。

$$Z=\lambda Z_{GCN}+(1-\lambda)Z_{BERT} \quad (6)$$

λは 2 つの予測の間のトレードオフを制御し、$\lambda=1$の場合は完全な RoBERTaGCN モデルを使い、$\lambda=0$の場合は BERT モジュールのみを使用することを意味する。$\lambda\in(0,1)$の場合、両方のモデルから予測のバランスを取ることができ、RoBERTaGCN のモデルをより最適化することができる。$\lambda=0.7$がλの最適値であることが Yuxiao らの実験によって明らかにされている。

4 実験

コサイン類似度の閾値を 0.5 と 0.95～0.995 までの間 0.005 刻みで設定して実験を行うことで、提案手法の分類性能を評価するとともに、各データセットにおける最適なコサイン類似度の閾値についても検討を行った。

4.1 データセットについて

文書分類の実験において、よく用いられる英文コーパスである 20NG、R52、R8、Ohsumed、MR の 5 つをデータセットとして採用した。以下表 1 に各データセットの文書数や平均単語数、訓練データ数とテストデータ数を示す。

表 1：各データセットの情報

	文書数	平均単語数	訓練データ	テストデータ
20NG	18846	206.4	11314	7532
R8	7674	65.7	5485	2189
R52	9100	69.8	6532	2568
Ohsumed	7400	129.1	3357	4043
MR	10662	20.3	7108	3554

各データセットについて以下 2 つの項目の前処理を施し実験を行った。

・ストップワードの除去
・記号の除去

各文書を特徴づける単語のみを残すこと、また、単語数を減らしメモリを節約することの両方を目的として、ストップワードと記号を除去した。

4.2 実験結果

コサイン類似度の閾値ごとの実験結果を、もとの RoBERTaGCN での正答率と合わせて表 2 に示す。×とした項目は、メモリ不足によって実験が完了できなかった項目である。また、Ohsumed については、閾値 0.99 以上と 0.995 以上、閾値 0.975 以上と 0.98 以上の実験で、追加される CLS ベクトルのコサイン類似度のエッジの数が同じであったため、まとめて表記した。各データセットにおける CLS ベクトルのコサイン類似度の値の分布については付録に表形式で示した。どのデータセットにおいても特定の閾値において、元の RoBERTaGCN の分類性能を上回ることが確認できたが、R8、R52、MR については元の RoBERTaGCN の性能を上回っている閾値が 1 つから 2 つ程度となっており、安定的な分類性能の向上が見られなかった。一方、20NG では、0.95 から 0.995 までのどの閾値においても元の RoBERTaGCN の分類性能を上回っており、Ohsumed でも、大方の閾値において元の RoBERTaGCN の分類性能を上回った。際立った結果としては、閾値 0.975 において 20NG で 0.67%、閾値 0.965 において Ohsumed で 1.19%、元の RoBERTaGCN の分類性能を上回ることが確認できた。

表 2：コサイン類似度の閾値ごとの実験結果

	20NG	R8	R52	Ohsumed	MR
RoBERTaGCN	89.15	98.58	94.08	72.94	88.66
0.5	×	49.47	×	64.73	×
0.95	89.29	98.26	92.83	73.73	88.21
0.955	89.42	98.63	94.08	72.74	88.21
0.96	89.74	98.49	**94.16**	73.49	88.52
0.965	89.54	98.45	93.15	**74.13**	88.15
0.97	89.43	98.45	93.77	73.41	88.66
0.975	**89.82**	98.63	93.57	73.49	**89.00**
0.98	89.60	98.54	93.96		88.29
0.985	89.64	98.54	92.95	73.46	88.58
0.99	89.76	98.36	93.42	73.71	88.55
0.995	89.51	**98.81**	93.26		88.31

5 考察

以下表 3 に、各データセットにおいて追加された各種エッジの数とコサイン類似度の平均値を示す。

表 3：エッジの数とコサイン類似度の平均値

	pmi edge	tf-idf edge	cos_sim edge	cos_sim の平均
20NG	22413246	2276720	×	0.838
R8	2841760	323670	29441186	0.846
R52	3574162	407084	41400215	0.840
Ohsumed	6867490	588958	27376155	0.837
MR	1504598	196826	56674250	0.823

×とした 20NG のコサイン類似度のエッジはコサイン類似度を計算している途中でメモリ不足となり、

重みを付加し終えることができなかった。閾値を 0.5 に設定した実験では、20NG と R52、MR のデータセットにおいてメモリ不足により、実験が完了できなかった。また、実験が完了できた R8、Ohsumed においても、元の RoBERTaGCN の分類性能を大きく下回る結果となった。いずれの理由も追加する文書ノード間のエッジの重みの数が膨大になってしまったことが原因と考えられる。どのデータセットにおいても CLS ベクトルのコサイン類似度のエッジの数が、pmi エッジや tf-idf エッジの 2 倍以上となっている。また、CLS ベクトルのコサイン類似度の平均値がいずれのデータセットにおいても 0.8~0.85 の間にあることから、同じジャンルではない文書ノード間のエッジの重みも相当数追加され、それがノイズになってしまっていると考えられる。

表 1 の各データセットの平均単語数と表 2 の実験結果を分析すると、提案手法は元の RoBERTaGCN と比較して、平均単語数が多いデータセットほどより高い分類性能を得やすいことが分かる。これは、平均単語数が多いほど、文書の CLS ベクトルにそれらの文書の特徴がよく反映され、同じジャンルの文書ノード間により多くコサイン類似度の重みが追加されたためと考える。反対に、平均単語数が少ないほど、文書の CLS ベクトルに差が表れにくくなり、違うジャンルの文書の CLS ベクトルのコサイン類似度が高く出てしまい、違うジャンルの文書のノード間の重みにコサイン類似度の重みが追加されてしまったことが、分類性能が思うように上がらなかった要因であると考えられる。

表 2 において、分類性能の最高値を示している CLS ベクトルのコサイン類似度の閾値における、追加されたエッジの数の割合を計算した。以下表 4 に計算結果を示す。

表 4：追加されたエッジの数の割合

	文書ノードの組み合わせ総数	追加された cos_sim edge の数	追加されたエッジの割合
20NG	177576435	753	0.0004240
R8	29441301	175	0.0005944
R52	41400450	28890	0.0697818
Ohsumed	27376300	15	0.0000547
MR	56833791	921	0.0016205

追加されたエッジの数の割合と分類性能に関係性は見えてこないため、今後は、文書ノード間に追加する重みを CLS ベクトルのコサイン類似度の閾値で決定するのではなく、コサイン類似度の値の上位○%といった基準での実験を行い、文書ノード間のエッジの数と分類性能の関係を明らかにする必要がある。

6 おわりに

本論文では、文書ノード間のエッジの重みとして文書の CLS ベクトルのコサイン類似度を付加することで RoBERTaGCN を改良でき、元の RoBERTaGCN の分類性能を上回ることを確認した。特に、提案手法は長い（平均単語数の多い）文書に対して特に有効であることが実験によって示された。

今後は、提案手法と BertGCN の相性や、GCN ではなく代わりに GAT を用いた場合の分類性能についても、実験を行って確認していきたい。

参考文献

1. Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In ICLR.
2. Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In AAAI-19.
3. Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li and Fei Wu. 2021. BertGCN: Transductive Text Classification by Combining GCN and BERT.

A 付録

	20NG	R8	R52	Ohsumed	MR
0.805~0.810	4781087	719057	1104948	766536	1800011
0.810~0.815	5229099	776635	1185426	823930	1922686
0.815~0.820	5690520	836643	1264104	878885	2040467
0.820~0.825	6143451	895164	1341940	934015	2148834
0.825~0.830	6587173	952113	1413448	982846	2237196
0.830~0.835	7006646	1006325	1476608	1030365	2308212
0.835~0.840	7387380	1056332	1536423	1067766	2359851
0.840~0.845	7709918	1103151	1583754	1106943	2386007
0.845~0.850	7961954	1143746	1625226	1131000	2381791
0.850~0.855	8129401	1173718	1647367	1146919	2344264
0.855~0.860	8184962	1196986	1656101	1152006	2278550
0.860~0.865	8129401	1207685	1652681	1145095	2178369
0.865~0.870	7926526	1206162	1628293	1120880	2049195
0.870~0.875	7598358	1191391	1585389	1081346	1885799
0.875~0.880	7135041	1157781	1519101	1028785	1702723
0.880~0.885	6554026	1112037	1438473	960676	1500465
0.885~0.890	5867506	1052726	1337900	879214	1290734
0.890~0.895	5114362	977491	1222742	782603	1078874
0.895~0.900	4309005	893612	1100173	678577	879914
0.900~0.905	3508252	798168	966449	571299	694353
0.905~0.910	2739662	697300	830496	465663	532612
0.910~0.915	2038573	592527	693525	361593	394964
0.915~0.920	1432510	488917	563682	267719	285562
0.920~0.925	939363	389946	441614	186487	201794
0.925~0.930	571941	297674	331644	121112	138133
0.930~0.935	316277	215911	236698	71941	94084
0.935~0.940	157878	149205	160526	37921	63581
0.940~0.945	69846	97523	103226	17596	41997
0.945~0.950	28428	60929	63680	7224	28096
0.950~0.955	10641	37172	37985	2232	18661
0.955~0.960	4523	22302	22619	596	11965
0.960~0.965	2134	13197	13374	121	7052
0.965~0.970	1145	7804	7840	9	3829
0.970~0.975	619	4054	4079	3	1765
0.975~0.980	320	1946	1963	0	689
0.980~0.985	162	821	873	1	204
0.985~0.990	103	304	338	1	22
0.990~0.995	75	122	161	1	4
0.995~	93	175	261	0	2