グラフニューラルネットワークによるアスペクトベースの感情分析

樊惠†，杉本徹‡
†芝浦工業大学大学院 理工学研究科，‡芝浦工業大学 工学部
{ma20079, sugimoto}@shibaura-it.ac.jp

1 はじめに

アスペクトに基づく感情分析（Aspect Based Sentiment Analysis 以下 ABSA）は感情分析タスクの一つであり，テキストを文レベルではなく単語レベルにタグ付けをする感情分析である．例えば「The food is delicious, but the service is too bad.」というレビューを考える．文レベルの感情分析手法はこのようなポジティブとネガティブの両方を含むレビューを分析しにくい．一方，ABSA は単語による分析を行う手法なので, このレビューに対し「food」と「service」を抽出し，タグを付けることができる．

ABSA のタグはアスペクト「B（はじめ）, I（中間）, E（終わり）, S（シングル）, O（無感情）」と感情「pos（ポジティブ），neu（中立的），neg（ネガティブ）」から構成される. 従来の研究は, 図 1 に示すように, タグを aspect と polarity に分けてそれぞれに予測する．あるいは，2 種類のタグをまとめ（unified），同時に予測する手法もある[1]. Li ら[2]は BERT モデルを使ってまとめたタグを予測し,「SemEval-2014 Task 4 Laptop」[3]において micro-f1 61.12%を得た．

	The	food	is	…	service	is	…
aspect	O	S	O		S	O	
polarity		pos			neg		
unified	O	S-pos	O		S-neg	O	

図 1　ABSA のタグ付け

しかし，ABSA に使用されるデータは短いテキストが多いので，タグの予測に使える情報量が少ないという問題が存在している．

本研究は，分析対象テキストにおける依存関係などの単語間の関係の情報を導入することで，情報量を増やすことを試みる．単語間の関係は通常グラフ構造なので，一般的なニューラルネットワークに適用できない．そのため，近年複数の分野でうまく機能することが示されているグラフニューラルネットワーク（以下 GNN）を用いる．この方法で，ABSA における情報量不足の問題を解決できる．

2 提案手法

本研究で提案するモデルの構造を図 2 に示す．まず，レビュー文をグラフ化し，モデルに入力する．そしてタグを予測する．最後に，従来手法により構築したモデルと比較して，結果を評価する．

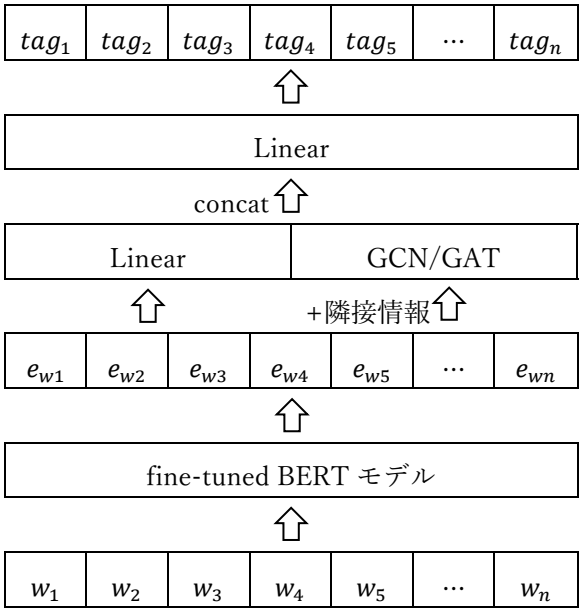

図 2 モデルの構造

2.1 テキストのグラフ化

GNN に入力するデータはグラフのノード特徴量とノードの隣接情報である．本研究は，文を単位としてグラフを構築する．各文の単語をグラフのノードとして扱う．そして，BERT モデルが出力したベクトルをノード特徴量とする．ノードの隣接情報を単語間の依存関係または self-attention により作る．

2.2 Graph Convolutional Network

Graph Convolutional Network[4]（以下 GCN）の原理は各ノードが隣接ノードから特徴量を集約（aggregate）し，自身の特徴量を更新することである．これで，ニューラルネットワークの順伝搬を行う時に，更新された各ノードの特徴量に隣接ノードの特徴量も含まれ，計算に関連する情報が増えている．GCN の更新式を以下に示す．

$$h_i^{(l+1)} = \sigma(\sum_{j \epsilon N_i} \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} h_j^{(l)} W^{(l)} + b^{(l)})$$

$h_i^{(l+1)}$は第 l+1 層目のノード i の特徴量

σは非線形活性化関数

N_iはノード i 自身を含めた全ての隣接ノード

$\hat{A}$はグラフの隣接行列+単位行列 $\hat{D}$は$\hat{A}$の次数行列

$h_j^{(l)}$は第 l 層目のノード i 自身を含めた全ての隣接ノードの特徴量

$W^{(l)}$は第 l 層目の重み $b^{(l)}$は第 l 層目のバイアス

$\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$は隣接ノードの特徴量を集約するための正則化ラプラシアン行列である

2.3 Graph Attention Network

Graph Attention Network[5]（以下 GAT）は attention メカニズムを用い，特徴量を集約する．この際，集約するのは隣接ノードからの特徴量だけではなく，グラフの全ノードに対して attention をし，attention 係数を計算する. さらに, BERT モデルの self-attention のように multi-head を用い，よりロバストなモデルになるようにする．GAT の attention 係数の計算式を以下に示す.

$$a_{ij} = \frac{\exp\left(LeakyReLU(a[Wh_i || Wh_j])\right)}{\sum_{k \in N_i} \exp\left(LeakyReLU(a[Wh_i || Wh_k])\right)}$$

$$j \in N_i$$

h_iはノード i の特徴量

h_jはノード i と attention 係数を求めるノード j の特徴量

[·||·]はノード i と j の特徴量を feature transformation を行い，そして二つの新たな特徴量を concat する

a(·)は concat された特徴量を実数にマッピングする

LeakyReLu は非線形活性化関数

LeakyReLu を適用した後に, ノード i とノード j 間の attention 係数を取得できる．全ノードと attention する必要があるので，softmax を用い，各 attention 係数を正規化する．そして GAT の更新式を以下に示す.

$$h_i^{(l+1)} = \sigma(\sum_{j \epsilon N_i} a_{ij} W^{(l)} h_j^{(l)})$$

$h_i^{(l+1)}$は第 l+1 層目のノード i の特徴量

σは非線形活性化関数

N_iはすべてのノード

$h_j^{(l)}$は第 l 層目のすべてのノードの特徴量

$W^{(l)}$は第 l 層目の重み

3 評価実験

3.1 使用するデータとタグ

本研究は「SemEval-2014 Task 4 Laptop」データセット[3]を用いる. データセットの割合を表 1 に示す.

表 1　データセットの割合

Dataset	Train	Dev	Test
文	2741	304	800
アスペクト	2041	256	634

レビュー文とタグのフォーマットを表 2 に示す.

表 2　レビュー文とタグの例

w_n	First	the	screen	goes	completely	out	.
tag_n	O	O	S-neg	O	O	O	O

表 3 に示すように，タグの種類は感情タグ「pos, neu, neg」と位置タグ「BIES」の組み合わせ 12 種類と「O」と「eq」である.「O」は無感情単語を表す. BERT モデルは単語辞書による，長い単語を複数のトークンに分割する場合がある．分割された単語の中に「##」を含むトークンは「eq」を付ける.

表 3　タグの全 14 種類

B-pos	B-neu	B-neg	I-pos
I-neu	I-neg	E-pos	E-neu
E-neg	S-pos	S-neu	S-neg
O	eq		

3.2 BERT モデルの fine-tuning と埋め込み

本研究のデータセットによりふさわしい埋め込みを取得するために，事前学習済みの BERT モデル[6]を微調整する．エポック数は最大 10 として，訓練セット全体を 1 回訓練するごとに，検証セットでモデルを評価し，精度を計算する．また，毎回の精度を比較し，最良モデルを保存する．10 回の訓練を終えた後，保存したモデルが最適なモデルになる.

レビュー文を微調整した BERT モデルに入力し，「単語数+2, 768」次元のベクトルを取得する. BERT モデルの特性により, 出力するベクトルには「CLS」と「SEP」というトークンがある．この 2 つのトークンを除くとベクトルは「単語数, 768」次元になる．このベクトルをノード特徴量として用いる.

3.3 単語間の関係の構築

3.3.1 依存関係

「スタンフォード大学 NLP ライブラリ」[7]を用い，依存関係をノードの隣接情報とする．単語と単語の間に依存関係があれば，エッジを作る.

3.3.2 self-attention

テキストを BERT モデルに入力した時，self-attention 行列が生成される．self-attention 行列は単語と単語の関連性を示す．例文「The cat sat on the mat」に対する self-attention 行列の 1 つを図 3 に示す．

	[CLS]	The	cat	sat	on	the	mat	[SEP]
[CLS]	0.0550	0.1216	0.0463	0.0531	0.1045	0.1305	0.1063	0.3827
The	0.1150	0.1325	0.1035	0.0894	0.1535	0.1552	0.1272	0.1237
cat	0.0998	0.0812	0.1136	0.1840	0.0643	0.0829	0.2023	0.1718
sat	0.0819	0.1213	0.1661	0.0722	0.1056	0.1131	0.1243	0.2156
on	0.0899	0.0780	0.1188	0.0924	0.3052	0.0790	0.1081	0.1287
the	0.1022	0.1264	0.1097	0.0850	0.2118	0.1337	0.1282	0.1031
mat	0.0711	0.0292	0.3335	0.2328	0.0544	0.0279	0.1053	0.1459
[SEP]	0.1193	0.1274	0.0669	0.0545	0.1430	0.1398	0.1388	0.2103

図 3 self-attention 行列

12 層 12 ヘッドの BERT モデルに文を入力すると，このような行列が全部で 144 個生成される．self-attention の非線形活性化関数は softmax なので，行毎の合計は 1 である．各行で関連性の値が最大の成分（図 3 の赤字の所）に対応して，図 4 の統計行列の対応する位置にある成分に 1 を足す．これを 144 個の self-attention 行列すべてに対して行う．

	[CLS]	The	cat	sat	on	the	mat	[SEP]
[CLS]	~~47~~	~~5~~	~~4~~	~~1~~	~~0~~	~~1~~	~~8~~	~~78~~
The	~~41~~	~~5~~	18	9	3	6	4	~~58~~
cat	~~25~~	9	~~9~~	9	1	0	2	~~89~~
sat	~~29~~	5	7	~~12~~	7	2	5	~~77~~
on	~~30~~	3	3	21	~~6~~	8	10	~~63~~
the	~~27~~	10	2	9	12	~~3~~	21	~~60~~
mat	~~35~~	3	4	1	4	4	~~10~~	~~83~~
[SEP]	~~35~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~106~~

図 4 最大関連性の統計行列

統計行列において，「CLS」，「SEP」は BERT モデルのトークンであるので，無視する．また，GCN と GAT を計算する時に自己ループを追加するので，自身との関連性（対角成分）も無視する．残りの成分の中で各行で数値が最大の成分（図 4 の赤字の所）が単語間の強い関連性を表すと考え，対応する行と列の単語間にエッジを作る．

3.4 全結合層

特徴量を集約することによって，異なるノードの特徴量が同じ内容になる可能性がある．そのため，GCN/GAT 層とは別に全結合層を用意して concat することにする．これにより，埋め込みの情報をモデルの計算中に維持できる．

3.5 GCN と GAT の訓練

3.5.1 共通の部分

Batch-size は full-batch を設定する．学習率と過学習抑制「Weight Decay」は 0.001 と 0.0005 を設定する．Optimizer は Adam を選択する．エポック数は 500 を設定する．クロスエントロピーで損失を計算する．訓練セット全体を 1 回訓練するごとに，検証セットでモデルを評価し，micro-f1 が向上した場合にパラメータを保存する．最後に，保存したモデルをテストセットで評価する．

3.5.2 GCN

第 1 層目と第 2 層目は GCN 層である．第 3 層目は GCN 層と全結合層の concat である．第 4 層目はタグを予測する全結合層である．

3.5.3 GAT

第 1 層目は 4 ヘッド GAT 層である．第 2 層目は 1 ヘッド GAT 層と全結合層の concat である．第 3 層目はタグを予測する全結合層である．

3.6 ABSA の評価指標

本研究の評価指標は適合率，再現率と micro-f1 とする．出力されたタグの次元は「単語数，タグの種類数」である．活性化関数 log-softmax を適用して，タグを予測する．出力されたタグと正解のタグの中で，感情を持つ 12 種類のタグのみに着目して感情「pos, neu, neg」ごとに集計する[2]．評価指標を計算する数式を以下に示す．

$$precision = \frac{\sum_d TP_d}{\sum_d (TP + FP)_d} \; d\epsilon\{pos, neu, neg\}$$

$$recall = \frac{\sum_d TP_d}{\sum_d (TP + FN)_d} \; d\epsilon\{pos, neu, neg\}$$

$$micro - f1 = \frac{2 * precision * recall}{precision + recall}$$

4 実験結果

実験結果を表 4 に示す．参照モデル 3 種と提案モデル 4 種に対して，テストデータで各モデルを 5 回ずつ評価し，得られた結果の平均を記録した．

参照モデル[2]は，「BERT only」「BERT＋Linear（全結合層）」と「BERT＋GRU」である．Linear 層と GRU 層はそれぞれに BERT モデルと直接連結し，End-to-End モデルになる．

提案手法は，単語間の関係として依存関係（SD）と self-attention（SA），GNN として GCN と GAT の組み合わせ計 4 種類のモデルを試した．

図 5 は各モデルの訓練ステップと損失の関係を示す．図 6 は GCN と GAT の ROC 曲線を示す．

表 4　実験結果

Model	P	R	F1
BERT only	0.5298	0.6010	0.5889
BERT + Linear	0.5526	0.5972	0.5901
BERT + GRU	0.5861	0.6183	0.6013
BERT + GCN-SD	0.5902	0.6560	0.6209
BERT + GAT-SD	**0.5902**	**0.6627**	**0.6238**
BERT + GCN-SA	0.5830	0.6570	0.6172
BERT + GAT-SA	0.5880	0.6578	0.6204

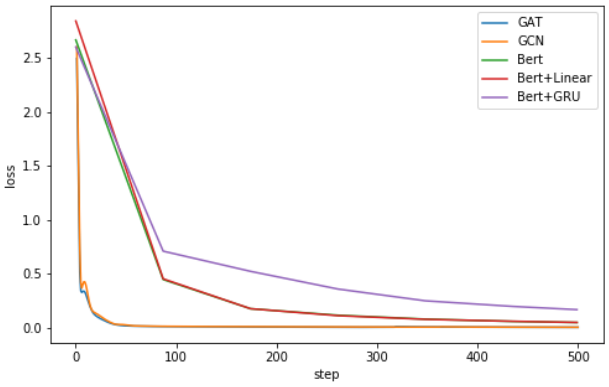

図 5 ステップと損失の関係

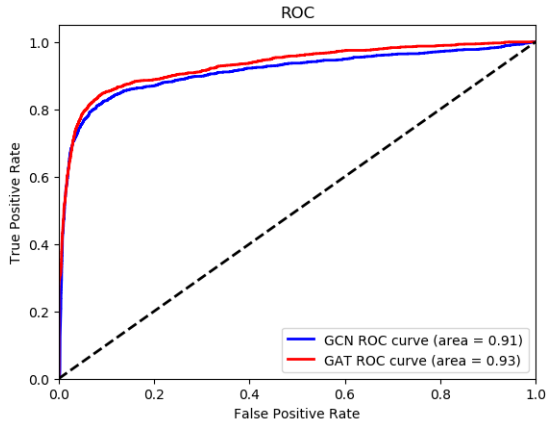

図 6 ROC 曲線

5 考察

表 4 を見ると，「BERT+GAT+依存関係」の組合せが最も高い micro-f1 の値を得た．GCN/GAT を用いたモデルの micro-f1 が「BERT only」より高いということは，ABSA タスクにおける GCN/GAT の有効性を示している．また，BERT モデルと他のニューラルネットワークの組合せよりも高いので，GCN/GAT を用いたモデルは ABSA タスクによりふさわしいと判断できる．ノードの隣接情報を生成する 2 つの手法（依存関係と self-attention）を比較すると，結果にほとんど差が無かった．レビューの単語数は多くないので，単語間の関係はあまり複雑ではない．そのため，ノードの隣接情報を生成する手法を変えても，結果にあまり影響がなかったと考えられる．

次に図 5 を見ると，GCN/GAT を用いたモデルの収束が最も速い．GCN/GAT は固定された BERT モデルの埋め込みを使ったので，モデルの計算中に埋め込みを更新しない．一方，linear または GRU を使う場合は BERT モデルと直接に連結するので，batch ごとに BERT モデルのパラメータも更新する．そのため，毎回埋め込みを改めて計算する必要があるため，GCN/GAT よりも計算に時間がかかり収束も遅くなる．

最後に，図 6 の ROC 曲線が示すように，ABSA タスクに対して GAT モデルは GCN より良い結果が得られた．本研究は inductive learning であるので，訓練セットはテストデータを含まない．そのため，テストセットをモデルに入力する時に，グラフの構造の変化が生じた．GCN モデルのパラメータを更新する時に，すべてのノード特徴量を更新する必要があるため，GCN モデルはグラフの構造に影響される．一方，GAT モデルは単にノード間の attention 係数の計算を行うため，グラフの構造の影響を受けないので本研究により適している．

6 おわりに

本研究は，グラフニューラルネットワークによるアスペクトベースの感情分析手法を提案した．GCN/GAT を用いることで，BERT を用いる既存手法よりも良い結果が得られた．グラフニューラルネットワークモデルはグラフ構造のデータを処理できるので，単語間の依存関係などの情報をモデルに入力できる．そのため，モデルの学習における情報量が増えたので，精度が向上したと考えられる．

グラフニューラルネットワークによるアスペクトベースの感情分析は複数のグラフで構成されたノード分類タスクである．しかし，レビューの単語数が異なるので，各グラフのノード数も異なる．今後，グラフのノード数が一致するかどうかが結果にどのくらい影響を及ぼすかを研究する．また，エッジの重みを含むグラフニューラルネットワークもあるので，エッジの重みを付ける手法を考えつつ，このようなグラフニューラルネットワークを研究することも今後の課題である．

参考文献

1. Knowing What, How and Why: A Near Complete Solution for Aspect-based Sentiment Analysis. **Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, Luo Si .** AAAI 2020, 2019 年.

2. Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. **Xin Li, Lidong Bing, Wenxuan Zhang, Wai Lam.** Association for Computational Linguistics, 2019 年.

3. **SemEval-2014 Task 4.** https://alt.qcri.org/semeval2014/task4/data/uploads/laptops-trial.xml, 2020-8 閲覧.

4. Semi-Supervised Classification with Graph Convolutional Networks. **Thomas N. Kipf, Max Welling.** ICLR 2017, 2017 年.

5. Graph Attention Networks. **Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio.** ICLR 2018, 2018 年.

6. **BERT-base-uncased.** 12-layer, 768-hidden, 12-heads, 110M parameters. https://pytorch.org/hub/huggingface_pytorch-transformers, 2020-4 閲覧.

7. **Stanza: A Python NLP Library for Many Human Languages.** https://github.com/stanfordnlp/stanza/, 2020-9 閲覧.