グラフニューラルネットワークによる機械読解エンジン

高場 雄太　尾原 颯　宮下 知也　永田 基樹　森 遼太

株式会社 pluszero
{takaba.yuta, ohara.soh, miyashita.tomoya, nagata.motoki, mori.ryota}@plus-zero.co.jp

1　はじめに

ニューラルネットワークによって機械読解の性能は飛躍的に向上したが、ニューラルネットワークは言葉の関係を明示的・直接的に持つわけではなく、distract 文などを含む場合に精度が落ちるなどの不安定な一面がある [1]．そこで，意味表現の情報を機械読解に組み込めば，言葉の関係を保持しつつ，機械読解が行えるのではないかと考えた．本研究では，既存手法と意味表現の情報を組み合わせた読解エンジンを作成し，既存手法との比較を行った．

2　提案手法

本論文では，意味表現の 1 つである Abstract Meaning Representation[2](以下 AMR) から得られた文章のグラフ構造を Graph Convolutional Networks(以下 GCNs) の技術を用いて処理し，Bidirectional Encoder Representation from Transformers[3](以下 BERT) と組み合わせることで BERT 単体に比べて機械読解の精度が上がるかについて検証した．

2.1　Abstract Meaning Representation

AMR は，統語における句構造分析に対応するような意味表現である．AMR の基本的な特徴は以下の通りである．

- ルートがあるラベル付きグラフである．
- 文が異なる場合でも，同じ意味ならば同じ AMR に割り当てられる．
- 英語表現に偏ったものになっている．
- PropBank のアノテーションを採用している．

AMR は単語間の関係を示す relation というものを定めており，各 relation は受動態の意味のものも用意されている．従って，受動態の文を扱う時は、受動態の意味のものの relation を使用できる．また，AMR では平叙文だけでなく，否定文や，疑問文にも対応している．

2.2　Graph Convolutional Networks

GCNs とは画像処理などで利用される Convolutional Neural Networks の技術をグラフ構造に応用したものである．GCNs では隣接するノードの情報を利用して各ノードの特徴量を計算することで，周囲のノードの情報を考慮した特徴量の計算が行える．

2.2.1　Relational Graph Convolutional Networks[4]

Relational Graph Convolutional Networks(以下 RGCNs) は GCNs の一種である．RGCNs ではノード間の関係性毎に重み行列を用意することで，ノード間の関係性を考慮して各ノードの特徴量を計算できる．

ノード i の特徴量 $\boldsymbol{h}_i \in \mathbb{R}^F$ を次の層のノード i の特徴量 $\boldsymbol{h}_i^{'} \in \mathbb{R}^{F^{'}}$ に更新する式を以下に示す．

$$\boldsymbol{h}_i^{'} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_i^r} \boldsymbol{W}^r \boldsymbol{h}_j + \boldsymbol{W}^0 \boldsymbol{h}_i)$$

σ ： 活性化関数

R ： グラフに存在する関係性の集合

N_i^r ： ノード i と関係性 r で結ばれているノードの集合

c_i^r ： ノード i と関係性 r に関して固有に設定する規格化定数

$\boldsymbol{W}^r \in \mathbb{R}^{F^{'} \times F}$ ： 関係性 r における重み行列

$\boldsymbol{W}^0 \in \mathbb{R}^{F^{'} \times F}$ ： 自分自身への重み行列

2.2.2　Relational Graph Attention Networks[5]

Relational Graph Attention Networks(以下 RGAT) は RGCNs に注意機構 (Attention) を取り入れたものである．RGATs では各ノードにそのノードと隣接するノードの情報を足し合わせる際に，ノード間で Attention をとり，それに応じて重みづけを行っている．

関係性 r におけるノード i, j 間の Attention を $\alpha_{i,j}^r$ とすると，ノード i の特徴量 $\boldsymbol{h}^i$ の更新式を以下に示す．

$$\boldsymbol{h}_i^{'} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \alpha_{i,j}^r \boldsymbol{W}^r \boldsymbol{h}_j)$$

Attention の値は以下の更新式で求められる．なお，$\boldsymbol{Q}^r \in \mathbb{R}^{1 \times F^{'}}, \boldsymbol{K}^r \in \mathbb{R}^{1 \times F^{'}}$ はそれぞれ，$\boldsymbol{h}_i$ を Attention のためのクエリ，キーに変換するための変換ベクトルである．

$$\begin{aligned} q_i^r &= \boldsymbol{Q}^r \boldsymbol{W}^r \boldsymbol{h}_i \\ k_i^r &= \boldsymbol{K}^r \boldsymbol{W}^r \boldsymbol{h}_i \\ E_{i,j}^r &= LeakyReLu(q_i^r + k_j^r) \\ \alpha_{i,\ j}^r &= \frac{exp(E_{i,j}^r)}{\Sigma_{k \in N_i^r} exp(E_{i,k}^r)} \end{aligned}$$

2.3 HotpotQA

HotpotQA はいくつかの文章 (以下 Paragraph) と質問文の組み合わせからなるデータセットである．Paragraph の中には，答を導くために必要になるものと回答には直接必要のない Paragraph が存在する．

HotpotQA の質問は yes, no で回答するもの，文章中にある語句や文章を回答とするものに大別される．

3 実験

3.1 HotpotQA の扱い方

今回の実験では，GCNs を用いた読解モデルがどれほど既存手法に寄与するかを明確に比較するために，タスクを単純化することにした．

前述したように，HotpotQA には読解には無関係な Paragraph が含まれているが，本研究では関連する Paragraph のみを抽出して実験を行った．

また，読解問題の解き方であるが，BERT と GCNs は別々で学習を進めた．

BERT においては質問文の答に該当する部分を本文中から指定する形式にした．本文中に答が存在しない質問文は答なしとしてデータに組み込んだ．HotpotQA には，yes か no で答る質問文が存在しているため，文章の文頭に”yes”と”no”の文字列を加え，本文中から、yes, no の単語を抜き出せるようにし学習を進めた．

GCNs においては，グラフのノードの単語と問題の答を比較し，答に含まれるノードのラベルを 1，含まれないノードのラベルを 0 として学習を進めた．

3.2 BERT の学習方法

BERT モデルは事前学習済みで公開されているもののうち，BERT_base をもとに，1 層の全結合層を加えたものを使用した．

入力には，質問文と問題文とを SEP トークンと呼ばれる Paragraph 同士の切れ目を示す特殊トークンで結んだものを使用した．あるトークン t に対して，答となる箇所の始まる確率である $p_{start}(t)$，答となる箇所の終点である確率である $p_{end}(t)$ の 2 つの値をモデルの出力とする．直前の全結合層の各トークン t に対して得られる 2 つの値をそれぞれ $start\_logit(t)$，$end\_logit(t)$ とし，入力全体のトークン数を T とするとこれら 2 つの値は以下で表される．

$$\begin{aligned} p_{start}(t) &= \frac{exp(start\_logit(t))}{\sum_{i=0}^{T} exp(start\_logit(i))} \\ p_{end}(t) &= \frac{exp(end\_logit(t))}{\sum_{i=0}^{T} exp(end\_logit(i))} \end{aligned}$$

損失は Cross Entropy，Optimizer は Adam Weight Decay Optimizer を使用した．GCNs の出力結果と統合する際は途中生成物の $start\_logit$ と end_logit を利用した．結合方法については 3.5 で議論する．

3.3 AMR グラフの GCNs への適用方法

AMR グラフを RGCNs，RGATs の入力にするにあたって，以下のような前処理を行った．

- 答となるノードを含まない問題を学習データから除外する
- AMR グラフのノードが持つ単語の内，amr_unknown など AMR に特有なものは unknown などの一般的な単語に変換する
- 各ノードが持つ単語を GloVe によって得られた事前学習済みの 300 次元のベクトルに embedding する．なお，事前学習済みのモデルの語彙にないものは 0 ベクトルで表現する．
- ノード A からノード B に能動の意味合いの relation のエッジがある時にはノード B からノード A へ受動の意味合いに変化させた relation のエッジを作成することで，相互に情報を更新できるようにする
- 学習するパラメータ数削減のために AMR にある約 200 個の relation を意味合いの近い 20 個の relation に分類し，その分類ごとに隣接行列を作成する

3.4 GCNs での学習

RGCNs，RGATs ともに二層重ね，各ノードの出力に sigmoid 関数をかけた後に binary cross entropy 誤差によって損失を計算することで学習する．また，Optimizer は Adam Optimizer を使用した．

また，AMR グラフのノードの単語と問題の答を比較し，答に含まれるノードのラベルを 1，含まれないノードのラベルを 0 として学習データを作成した．

ノード単位で学習を進めたため，出力は各ノードが答である確率となる．そのため，閾値を超えるような確率を持つノードが現れなかった場合は，答として何も出力されないことがある．

3.5 concat 方法

concat というのは新たな答を得るために，GCNs の結果と BERT の結果を組み合わせることを指すとする．具体的には以下の手法を用いた．

まず，3.2 の $start\_logit, end\_logit$ を高い順に並べ，上位 N 個 (今回の実験では $N = 5$) を答の候補とする．次に，$start\_logit, end\_logit$ の組み合わせ N^2 個の内，答の終了位置が答の開始位置よりも前に来ているものなど，おかしい組み合わせを取り除く．そして，それぞれの組み合わせの答となる範囲内に含まれる GCNs のノードの出力に sigmoid 関数をかけた結果の平均をとり $Graph\_score$ とする．最後に，それぞれの組み合わせにおいて以下を計算し，この値が最も高かった組み合わせを答とした．

$$start\_logit + end\_logit + \alpha_{[Graph]} \times Graph\_score$$

なお，今回の実験では $start\_logit, end\_logit$ の値がおよそ-10 から 10 の間にあるので，$\alpha_{[Graph]}$ を 0.0, 1.0，5.0，10.0，15.0，20.0 に変えて実験した．

4 実験結果

本章における score の定義は完全一致率とする．

本論文で行った実験は，RGATs と RGCNs でそれぞれ，5, 10, 100 エポック学習しそれぞれ BERT との concat を計算するというものである．$\alpha_{[Graph]}$ を 0.0, 1.0, 5.0, 10.0, 15.0, 20.0 に変化させたものに対して，エポック数と concat 後の score の関係を，RGCNs の時は表 1，RGATs の時は表 2 に示す．

RGCNs と RGATs の単独の学習において 100 エポック学習時の recall を図 1 に示す．

さらに，RGCNs と RGATs の単独の学習においてそれぞれ 10 エポック学習時と 100 エポック学習時における PR 曲線を図 2 として以下に示す．また，表 3 に RGCNs と RGATs の単独の学習におけるエポック数と PR 曲線の面積 (PR-AUC スコア) の関係を示す．

5 考察

図 2 と表 3 をみると，10 エポックの方では 100 エポックのものをに比べて，PR 曲線における面積が RGCNs, RGATs のどちらの手法でも大きくなっていることがわかる．ここから，GCNs 単独の回答能力をみたときは，10 エポックの方が 100 エポックに比べ良い結果であると言える．また，表 1 と表 2 から読み取れるように BERT に GCNs から得られる回答を concat をしても score の劇的な上昇は見られなかった．100 エポックのものの結果の方が 10 エポックのものより悪いにもかかわらず，concat を行った結果において，表 1 と表 2 をみると，10 エポックのものと 100 エポックのもので大きく結果に差が表れなかったことがわかる．このことから GCNs の結果の良し悪しによらず，concat 後の score に向上が見られなかったということになるため，concat 方法に問題があるのではないかと考えられる．また，concat 方法が適切でなかったことの他に，GCNs で解けている問題が BERT でも解けているため concat 後の score が向上しなかったのではないかとも考えられる．

concat 後の score の向上がなかった原因として，AMR の情報の抽出方法が適切でなかったことも考えられる．今回の実験では，AMR にパースしたときに答となる単語がノードとして現れているもののみを扱っているので，AMR からの情報抽出が十分になされたモデルを使用すれば，AMR からの情報をうまく抽出することができる．よって，GCNs の結果が BERT の score に大きな影響を与えなかった原因として，AMR の情報の抽出方法に問題があったと考えられる．

RGCNs と RGATs の recall を比較してみると，多くの step 数において，RGATs の方が良い結果が出ていることがわかる．そこから，RGATs の方が RGCNs と比較して，解くことのできている問題数が多くなっているということがわかる．RGATs は RGCNs にはない，Attension 機構が追加されており，AMR のエッジ情報の関係性の強弱が RGCNs に比べて強くつくような手法となっている．ここから AMR のエッジ情報における関係性の強弱は機械読解を行う上で必要な情報になってくるのではないかと考えることができる．

6 おわりに

本研究では，文章の意味情報を含むグラフ情報の付加による機械読解システムの高精度化を目標に，意味表現の 1 つである AMR から得られた文章のグラフ構造を GCNs の技術を用いて処理し BERT と組み合わせることで BERT 単体に比べて機械読解の精度が上がるかについて検証した．結果としては，今回採用したアプローチでは精度の向上を確認することはできず，その原因として以下を考えている．

- GCNs と BERT の concat の仕方に問題がある
- GCNs で解ける問題は BERT でも解けてしまっている
- グラフ構造からの情報が GCNs でうまく抽出し切れていない

表 1: RGCNs のエポック数と BERT と concat した結果の score の関係

	0.0(BERT のみ)	1.0	5.0	10.0	15.0	20.0
5 エポック	0.519815	0.519951	**0.520086**	0.519543	0.519679	0.519815
10 エポック	0.519815	0.519815	0.519815	0.519679	0.519543	0.519951
100 エポック	0.519815	0.519815	0.519815	0.519815	0.519272	0.519272

表 2: RGATs のエポック数と BERT と concat した結果の score の関係

	0.0(BERT のみ)	1.0	5.0	10.0	15.0	20.0
5 エポック	0.519815	0.519951	0.519951	0.519408	0.519679	0.519543
10 エポック	0.519815	0.519815	0.519543	0.519543	0.519272	0.519001
100 エポック	0.519815	0.519679	**0.520494**	0.519679	0.519408	0.518865

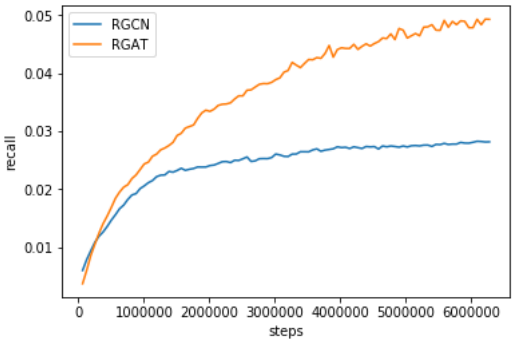

図 1: RGATs と RGCNs の recall

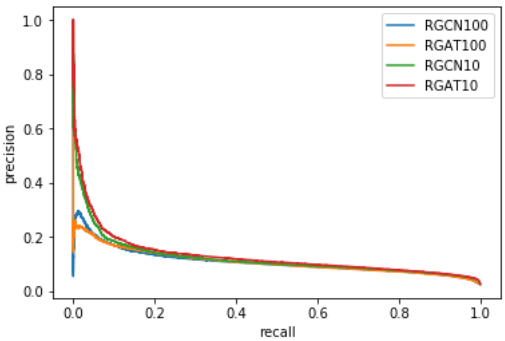

図 2: RGATs と RGCNs の PR 曲線

表 3: エポック数と PR-AUC スコアの関係

	10	100
RGCNs	0.12181676	0.10899657
RGATs	0.13039996	0.10859192

今後の取り組みとしては，今回精度が高まらなかったことの原因分析を深めることで，既存のニューラルネットベースの機械読解システムへ上手く明示的に意味情報を取り込む方法について模索したい．具体的には，BERT と GCNs における回答が相互に影響を及ぼすような concat 方法の検討を行うとともに，BERT, GCNs それぞれで得手・不得手とする設問の傾向を詳細に調べ，文章のグラフ構造から得られる情報のうち機械読解に大きく影響を与える情報がどのようなものかを調べることを計画している．

参考文献

[1] Jia, Robin and Liang, Percy Adversarial Examples for Evaluating Reading Comprehension Systems, 2017

[2] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer and Nathan Schneider Abstract Meaning Representation for Sembanking, 2013

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

[4] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling. Modeling Relational Data with Graph Convolutional Networks, 2017

[5] Dan Busbridge, Dane Sherburn, Pietro Cavallo, Nils Y. Hammerla. Relational Graph Attention Networks, 2019