Graph-to-Sequence アプローチによる学術論文のアブストラクト生成

近藤 雅芳　　新保 仁　　松本 裕治
奈良先端科学技術大学院大学 先端科学技術研究科
理化学研究所 AIP
{kondo.masayoshi.ki1, shimbo, matsu}@is.naist.jp

1 はじめに

学術論文のアブストラクトは、論文の情報を読者に端的に伝える上で極めて重要である．自然言語処理技術において、文書からその要旨を構成する方法を自動要約と呼ぶ．近年、自動要約はニューラルネットワークを用いた生成要約（以下、ニューラル生成要約）が活発に研究されている [3, 9, 10, 11, 12, 13, 19, 20].

ニューラル生成要約は、入力文書とその要約文を入出力ペアとする学習データを用いて、一般的に再帰型 Ses2Seq ニューラルモデルを訓練し、その学習済みモデルを用いて入力文書に対する要約文を生成する．ニューラル生成要約は、従来の抽出型手法に比べ高品質の要約を生成できる一方で、計算資源や時間の制約のため、学術論文のような非常に長い文書を訓練に用いることは難しい．従来、このような非常に長い文書に対しては、文書の前半の一定文字数のみを用いたり、重要文抽出を行う等の前処理によって圧縮し用いてきた．しかしながら、このような前処理を用いた場合、入力文書全体を考慮して要約を生成できないことや前処理コストが無視できないといった問題がある．

本研究では、このような長文データを入力とする際に生じるニューラル生成要約の問題に対して、**Graph-to-Sequence 生成アプローチを用いた学習方式を提案**する．具体的には、入力文書をグラフ化し、そのグラフを入力に要約文の生成を行う．Graph-to-Sequence 生成アプローチは、長文データをグラフ化することで、入力量を大幅に圧縮できるだけでなく、入力データの一部のみを用いる従来の前処理方法と異なり、データ全体を入力に用いることが可能である．また、入力文書からグラフへの変換には、bi-gram 統計に基づくため、グラフ変換の計算コストは小さい．提案手法の検証では、arXiv と PubMed の論文データから 19 万サンプルを超える大規模な新しいデータセットを構成し、実験を行った．

本研究の貢献は、以下である．

- Graph-to-Sequene アプローチによるニューラル生成要約の学習方式を提案する．提案手法は、論文データを単語をノードとするグラフに変換し、グラフニューラルネットワークと RNN を組み合わせたニューラルモデルによって、そのアブストラクトを生成する．
- 論文データのグラフを単純な bi-gram 統計から構成する．これにより、データ全体をグラフとして許容できるサイズに収められるだけでなく、グラフ変換の際の前処理による計算コストを大幅に軽減する．
- 提案手法を検証するために、新しい大規模な論文データセットとして arXiv-PubMedd データセットの構築を行った．

INPUTS-nodes: the / cosmic / acceleration / has / been / discovered / since / but / we / are / still / in / dark / about / nature / of / this / mystery / . / cosmological / constant / proposed / by / einstein / to / build / a / static / universe / solution / is / simplest / energy / model / ...(中略)... / parametrization / for / extract / information / most / wildly / used / called / polarski / linder / ...(中略)... / bounded / at / high / redshifts / direct / motivation / proposing / such / however / also / divergence / problem / when / approaches / corresponds / future / undoubtedly / nonphysical / ... (一部抜粋)

INPUTS-links: [the, cosmic], [cosmic, acceleration], [acceleration, has], [static, universe], [universe, solution], [solution, is], [is, the], [the, simplest], [simplest, dark], [dark, energy], [energy, model], [model, to], [to, explain], [explain, the], [the, acceleration], [alternative, dark], [dark, energy], [energy, models], [the, evolution], [evolution, of], [of, dark], [dark, energy], [energy, is], [is, determined], [determined, by], [by, its], [its, equation], a, parametrization], [parametrization, for], [the, dark], [dark, energy], [energy, model], ... (一部抜粋)

GOLD: in this paper , we propose a new class of parametrization of the equation of state of dark energy . in contrast with the famous cpl parametrization , these new parametrization of the equation of state does not divergent during the evolution of the universe even in the future . also , we perform a observational constraint on two simplest dark energy models belonging to this new class of parametrization , by using the markov chain monte carlo method and the combined latest observational data ... (省略)

BiLSTM: in this paper , we present the results of a global analysis of the cosmological constant @xmath0 . we show that the [UNK] of the model can be used to fit the power spectrum of the cosmic microwave background power spectrum . we find that the value of the derivative of the parameter @xmath0 is significantly larger than the value of the scale factor , which can be used to fit the power spectrum of the cosmic microwave background power spectrum . we also find that the value of the cosmological constant is significantly larger than the value of the cosmological constant .

Our approach: in this paper , we propose a new class of models for the evolution of the cosmic acceleration . we show that the model can be used to constrain the equation of state of the universe . in this paper , we propose a new parametrization of the equation of state of the model . the model can be used to constrain the equation of state of the universe . in this paper , we propose a new parametrization of the equation of state of the model . we find that the model can be used to constrain the equation of state of the universe .

図 1　arXiv-PubMed データセットのテストセットにおける実際の要約生成例．INPUTS はモデルへの入力を表す．上記の INPUT-links はノード対の一部を示しているが、モデルには隣接行列として与える．また、GOLD はサンプルの正解アブストラクト文、BiLSTM と Our approach はそれぞれのエンコーダモデルでの生成アブストラクト文を表す．

本稿の構成を述べる．第 2 章で関連研究を述べ、第 3 章では提案手法について説明を行う．第 4 章で実験を示し、第 5 章で結びを述べる．

2 関連研究

Neural Text Summarization：ニューラル要約の研究は、Rush ら [13] や Nallapati ら [10] が構築した大規模な要約データセッ

トにより進展した．ニューラル要約には、抽出要約と生成要約の２つのアプローチがある．ニューラル抽出要約においては、数百語程度の単語列を入力し 100 語程度の要約文を生成する「長文要約」タスクを対象に、ニューラルネットワークを入力文書からの文抽出 [9] や文並び替え [11] に適用する手法が一般的である．

ニューラル生成要約は、Rush ら [13] の 100 語程度の単語列を入力し 20 語程度の見出し文を生成する「短文要約」や、A.See ら [14] の「長文要約」の研究がある．最近では、ニューラルネットワークと強化学習を組み合わせたニューラル生成要約の研究 [3, 12] も進められている．ニューラル生成要約では、一般的に事前処理による入力単語列の選定を行い RNN-Seq2Seq モデルにより要約文生成を行う．また、ニューラル生成要約は、主にエンコーダとデコーダを接続するニューラルユニットを対象に研究がなされている [10, 14, 15].

一方で、本研究では入力データ全体をグラフ化して入力に用いる点で従来のニューラル生成要約とは異なる．また、本研究では、非再帰型のグラフニューラルネットワークを用いたエンコードにより高速処理が可能である点も、従来の一般的なアプローチと異なっている．

Graph/Data-to-Sequence Text Generation：Data-to-Sequence 生成アプローチは、構造データからテキストを生成する方法論である．近年、RDF データ [16] や知識ベース [18] からのテキスト生成の研究が進められている [2, 4, 5, 8]．例えば、Trisedya ら [16] は、グラフ構造を考慮できる再帰型ニューラルモデル GTR-LSTM を提案し、RDF データから短文を生成した．Marcheggiani ら [8] は、入力文章から依存項解析器や AMR 解析器を用いて得たグラフを入力として短い単文章を生成した．

しかしながら、これらの研究はグラフ構築に解析器を用いた事前処理を行っているが、本研究では、解析器を用いず bigram 統計からグラフを構築している点で異なっている．また、先行研究ではデータ数が数万のデータセットを用いて単一文の生成を対象としているのに対して、本研究では、10 万以上の大規模なデータセットを用いて２００語程度の複数文から構成される要約文を生成する点でも異なっている．

3 提案手法

3.1 問題設定

生成要約は、文書 X が与えられたとき、その要約文 Y を生成することが目的である．このとき、要約文 Y の文長 l_y は、文書 X の文長 l_x より短い ($l_y < l_x$)．本研究では、論文データ $X = \{w_{x1}, w_{x2}, ..., w_{xN}\}$ から得られるグラフ $G_X = (V, E)$ を入力として、要約文 Y を生成する．

3.2 グラフの定義

本研究のグラフ $G_X = (V, E)$ は、重み付き無向グラフである．V は X の語彙ノード集合であり、E は X からの bigram 統計に基づき抽出される単語対で構成されるリンク集合 ($|V|\times|V|$ の部分集合) である．また、E は、その単語対に未知語 (UNK) を含むものは除外するが、一般的な stopword に該当する語彙を含む単語対は含み、同単語対は重複を許して計数する．

3.3 Graph-to-Sequence Model

本研究は、エンコーダとデコーダをそれぞれ Graph Neural Networks (GNN) と LSTM-RNNs で構成する Graph2Seq モデルを用いて、グラフを入力し自然文を生成する．すなわち、論文 X から得られるグラフ $G_X = (V, E)$ を入力として、要約文 Y を生成する．また、エンコーダとデコーダとの接続に、pointer-generator メカニズム [14] を適用する．

$$\begin{cases} H_V = GNNEncoder(G_X; \Theta_{enc}) \\ \hat{y}_t = LSTMDecoder(H_V, \hat{Y}_{<t-1}, Z; \Theta_{dec}) \\ Z = PtrGenFunction(H_V, \hat{y}_{t-1}; \Theta_{other}) \end{cases}$$

ここで、H_V は、ノード集合 V に一対一対応するエンコード特徴量ベクトルの集合 $H_V = \{h_v\}_v^V$(node embeddings) である．このとき、$H_V \in \mathbf{R}^{|V|\times|h_v|}$ であり、$|V|$ と $|h_v|$ は、それぞれノード集合の大きさと node embeddings の特徴量次元を示す．また、Z は、Pointer-Generator メカニズムによって H_V と Y から得られる特徴量ベクトルを表す．

3.3.1 Encoder: Graph Neural Networks

GNN は、グラフ $G = (V, E)$ を入力として、一般的に、そのノード集合 V の node embeddings H_V を出力する．本研究では、以下の T. N. Kipf らのグラフニューラルモデル [6] をエンコーダに用いる．

$$H_V^{out} = GraphNN(H_V^{in}; G, \Theta) \quad (1)$$
$$= Relu([\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}]H_V^{in}W) \quad (2)$$

このとき、H_V^{in}, H_V^{out} は、それぞれ前層から出力される node embeddings と当該層が出力する node embeddings である．なお、第 1 層目の入力には、ノード集合 V に対応する単語埋め込み表現を用いる．また、$\tilde{A} = A + I_V$ であり、A はグラフ G の隣接行列、I_V は $V \times V$ 単位行列、$\tilde{D_{ii}} = \sum_j \hat{A}_{ij}$ である．W は重みパラメータであり、$W \in \mathbf{R}^{|h_v|\times|h_v|}$ である．

さらに、本研究では、入力グラフ $G = (V, E)$ に対して、そのエンコード特徴量 graph embedding e_G を下記のように定める．

$$e_G = \sum_{v\in V} h_v^{out} \quad (3)$$

このとき、graph embedding の次元は、$e_G \in \mathbf{R}^{|h_v|\times 1}$ である．得られた node embeddings は、pointer-generator メカニズムを介して LSTMDecoder と接続される．一方で、graph embedding は、LSTMDecoder の初期隠れ状態として用いる．

3.3.2 Decoder: LSTM-RNNs と Pointer-Generator Mechanism

我々は、デコーダとして下記のような単層の LSTM-RNN ユニット[*1]を用いる．

$$h_{t+1},\ s_{t+1} = LSTM(w_t, s_t; \Theta_{lstm}) \quad (4)$$
$$P_{Dec}(w_{t+1}) = softmax(W_{out}h_{t+1}) \quad (5)$$

h_{t+1} と s_{t+1} は、それぞれ、LSTM の出力ベクトルと隠れ状態ベクトルである．$P_{Dec}(w_{t+1})$ は、タイムステップ $t+1$ での単語

[*1] LSTM 関数の詳細な数式は本稿の紙面の関係上、割愛した．

	source	target
1 サンプルあたりの平均単語数	4328.21	179.70
1 サンプルあたりの平均文章数	166.77	6.36
1 文章あたりの平均単語数	25.95	28.22
1 サンプルあたりの平均ノード数	763.64	-
1 サンプルあたりの平均リンク数	8312.63	-

表 1　arXiv-PubMed データセットの Trainset 詳細.

の生成確率で、W_{out} は、重みパラメータであり、デコーダの隠れ状態ベクトルの次元数を $|h_{dec}|$ とすると、$W_{out} \in \mathbf{R}^{|V|\times|h_{dec}|}$ である.

さらに、エンコーダとデコーダの接続に、pointer-generator メカニズム [14] を用いる. pointer-generator メカニズムは、タイムステップ t での単語の生成確率 $P(w)$ を下記の式により定める.

$$P(w) = p_{gen} \cdot P_{Dec}(w) + (1 - p_{gen}) \cdot P_{EncAttn}(w) \quad (6)$$

$P_{Dec}(w)$ は、Decoder による単語の生成確率であり、$P_{EncAttn}(w)$ は、エンコード特徴量との attention 操作により計算される単語の予測確率である. また、p_{gen} は、下記の式によって計算される確率値 ($p_{gen} \in [0, 1]$) である.

$$p_{gen} = \sigma(W_c c + W_s s + W_w w + b) \quad (7)$$

ここで、c, s, w, b は、それぞれ、エンコーダとの attention 操作によるコンテクスト特徴量ベクトル、、デコーダの隠れ状態ベクトル、デコーダへの入力単語ベクトル、バイアス項を示す. W_c, W_s, W_w は、それぞれのベクトルに対する重みパラメータである. また、σ はシグモイド関数である.

以上の構造で提案モデルは構成され、クロスエントロピー誤差関数を最小化するように学習をすることで最適化を行う.

4　実験

実験では、入力量比較、精度比較と生成サンプル評価の 3 つの検証を行う. 各比較評価については、入力量比較はサンプルあたりの入力グラフノード数と従来の入力単語列の長さを比べる. 精度比較は、提案モデルとベースラインモデルを ROUGE 指標 (ROUGE-1/2/L/SU4)[7] に基づいて評価を行う.

4.1　データセット

構成と詳細: 本研究には、新しく我々が構築した arXiv-PubMed データセットを用いる. arXiv と PubMed はオープンアクセスの論文サイトであり、学術論文のデータを収集することができる.

データセット構成は、訓練データが 181,264 対、開発データが 5,000 対、テストデータが 5,000 対となっている. 語彙としては頻出 30,000 語を用い、そこに含まれない単語に対しては未知語処理 (UNK 変換) を行った. 表 1 に、arXiv-PubMed データセットの詳細を示す. また、データの収集と補正には、以下の規則に基づいて行った.

1. introduction と conclusion のセクションを備え、全体で 3 セクション以上で構成される論文データを抽出する.
2. 抽出データ内の各センテンスは、最大 60 文字で打ち切る.

規則 2 は、ノイズ除去を目的とした補正処理である. 論文サンプルは tex データで構成されており、図表や数式のスクリプトコードを含んでいる. このため、要約生成に用いない不必要な文字・記号を多く含んでいることからノイズ除去を行った.

グラフへの変換: 各論文サンプルに対して、単語 bigram 抽出を行った. また、UNK が含まれる bigram は排除し、得られた bigram 関係データから無向グラフの隣接行列を構成した. さらに、各ノードには、入力テキストに現れる語彙の順序に応じた position embedding を与えた.

4.2　実験設定

実験は、エンコーダのみを比較するために、その他の構成部は比較対象のモデルで共通とした. 精度評価には、ROUGE 指標を用いた. 実験環境は、理研 AIP の深層学習用大型計算機 RAIDEN 上で行った.

モデル設定：提案モデルは、3 層の GraphNN を用いた. また、各層の入出力には residual 接続と layer-normalization を適用した. 入力部の単語埋め込み表現ベクトルとエンコード特徴量の次元は、いずれも 256 とした. また、position embedding として fourier-positional encoding [17] を用いた.

学習設定：モデル訓練は、clip-gradient の大きさを 2 として、最適化法に Adagrad を用いて行なった. 初期学習率は 0.1 とした. 提案モデルのパラメータ初期化には、xavier 初期化を用いた. 訓練は、300,000 iters 程度行った. 学習には、一般的なクロスエントロピー誤差関数を用いた. モデルの最大出力長は、学習時とテスト時で、それぞれ 200 語と 220 語とした. また、テスト時は、ビーム幅を 4 とするビー ムサーチを用いて予測系列を得た. 実装には、python(2.7 系) と Tensorflow(ver.1.2) [1] を使用した.

比較モデル：比較モデルは、BiLSTM をベースラインモデルとして用いた. モデルの設定は、提案法と同様の設定を行った. また、入力は、グラフノードを position の順序に従って与えるものとした.

4.3　実験結果

4.3.1　入力量：系列長 vs ノード数

図 2 は、arXiv-PubMed データセットの訓練サンプルに対する系列長とノード数の散布図である. 特に、総単語数が 1 万語からなるサンプルではノード数が 700 から 1500 語の程度の範囲に収まっているのが分かる. その対比は約 7 分の 1 程度であり、入力量の大幅な圧縮が可能である. 従来法では 1 万語からなるサンプル全体を入力として扱うことは難しかったが、本研究の学習方式では、グラフを入力として扱うことにより入力文全体の情報を扱うことが可能である.

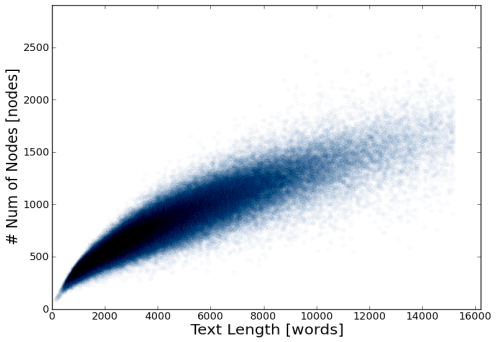

図 2　文長 (系列長) とノード数を軸とする arXiv-pubmed データセットの訓練サンプルの散布図.

MODEL	ROUGE			
	1	2	L	SU4
BiLSTM	26.22	6.58	15.48	7.65
GraphNN	30.17	7.51	16.55	9.83

表 2　各モデルの ROUGE(F1) で評価した値を示す。

4.3.2　精度評価

表 2 に精度評価の結果を示す．ベースラインの BiLSTM に対して、大きく上回る結果となっている．特に、ROUGE-1 と ROUGE-SU4 で 2 ポイント以上の向上が見られる．

4.3.3　要約生成

図 1 に、提案手法による実際の生成例を示す．提案法の生成した要約文は、正解要約文 (GOLD) と似た文章を生成していることが分かる．一方で、提案法の生成文中頃に、”in this paper , we propose a new parametrization of the equation of state of the model . の文章単位の繰り返し生成が生じており、グラフを入力とするモデルにおいても従来のニューラル生成要約で生じる問題が同様に生じていることが分かる．

5　おわりに

本稿では、グラフを入力として要約文を生成する新しいニューラル生成要約の学習フレームワークを提案した．提案法は、論文全体を bigram のみによる単純なグラフ構造に置き換え、Graph Neural Networks を用いてエンコードを行う．また、これら提案法を検証するための新しいデータセットを構築し、実験でその効果を検証した．今後は、より多くの比較モデルとの精度検証や従来の系列を入力とするニューラルモデルとの組み合わせた実験、繰り返し生成の問題への対処や高品質な生成要約に向けた手法の改善を行っていきたいと考えている．

参考文献

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] D. Beck, G. Haffari, and T. Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Long Papers*, 2018.

[3] A. Çelikyilmaz, A. Bosselut, X. He, and Y. Choi. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018 (Long Papers)*, 2018.

[4] P. Fernandes, M. Allamanis, and M. Brockschmidt. Structured neural summarization. *CoRR*, 2018.

[5] B. Hachey, W. Radford, and A. Chisholm. Learning to generate one-sentence biographies from wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Long Papers*, 2017.

[6] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. 2017.

[7] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[8] D. Marcheggiani and L. Perez-Beltrachini. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation, 2018*, 2018.

[9] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI 2017*, pages 3075–3081, 2017.

[10] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL, 2016*, 2016.

[11] S. Narayan, S. B. Cohen, and M. Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018 (Long Papers)*, 2018.

[12] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *ICLR*, 2018.

[13] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389, 2015.

[14] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL 2017*, pages 1073–1083, 2017.

[15] J. Tan, X. Wan, and J. Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1171–1181, 2017.

[16] B. D. Trisedya, J. Qi, R. Zhang, and W. Wang. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Long Papers*, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[18] P. Vougiouklis, H. ElSahar, L. Kaffee, C. Gravier, F. Laforest, J. Hare, and E. Simperl. Neural wikipedian: Generating textual summaries from knowledge base triples. *J. Web Sem.*, 52-53:1–15, 2018.

[19] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. R. Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), 2017*, 2017.

[20] Q. Zhou, N. Yang, F. Wei, and M. Zhou. Selective encoding for abstractive sentence summarization. In *ACL 2017*, pages 1095–1104, 2017.