文章の一貫性を用いた自動生成文章検出手法の提案

原田 侑 [1]　Danushka Bollegala [2]　Naiwala P .Chandrasiri [1]
[1] 工学院大学 [2] The University of Liverpool.
j216232@ns.kogakuin.ac.jp
bollegala@liverpool.ac.uk
chandrasiri@kogakuin.ac.jp

1 はじめに

近年，自然言語処理における深層学習の活用が急速に拡大している．これは背景に深層学習（LSTM など）の技術の考案により，より複雑なタスクである翻訳や高度な質問応答を実現できるようになったことがある．そのような文章生成技術の進歩は目覚ましく，文章の導入部分を入力するだけで人間が書いたと誤認識される可能性のある文章を自動的に生成することが現実的になった．しかし，生成される高品質な文章が悪用された場合，様々な問題が発生すると考えられた結果，モデルの公開を遅らせた言語モデルに OpenAI の GPT-2[1] がある．そこで，本研究では文章の一貫性という観点からそのような高品質な文章を生成できる GPT-2 が生成した文章を人間が書いた文章と識別することを目的とする手法を提案した．また，提案手法が生成文章の識別タスクにおいて従来手法の精度を検出率の面において上回ったことを確認した．

2 関連研究

2.1 生成文章検出

現在の生成文章検出手法は深層学習を用いて行うのが盛んである．しかし，深層学習による検出には 2 点問題点がある．1 点目の問題は高精度の検出には学習が必要なことである．これは新しく高度な言語モデルが考案された際，対応に遅れが生じる問題が発生する可能性が考えられる．本研究では，その問題点に対して，深層学習自体は用いず，その成果の分散表現のみを用いることで学習をすることなく未知の言語モデルから生成された文章にも対応することを可能にすると同時に，非常に高速な実行速度を実現している．

2 点目の問題はこれらの先行研究は基本的に単一の文章を入力することを条件としている点である．現在の文章生成技術は，人間による会話応答の学習やハイパーパラメータの調節によって，文章の方向性をある程度特定することでより高精度な文章が出力される．そのため，実際にこの生成技術が利用される場合としては，文章の冒頭は人間が記述したものであったり，それに近い条件付けがされている可能性が高いと考えられる．しかし，従来手法では文章全てを用いて検出を行うため，冒頭の人間が書いた文章で誤検知してしまう可能性がある．そこで，本研究では人間が書いた文章と機械が生成した文章を組み合わせた文章を検出することを目的とすることでより実践的な手法になっている．加えて，本研究では研究の題材を従来の生成文章検出と異なり，ニュースだけでなく小説や Wikipedia 文章などといったものも対象とすることで汎用性が高い検出手法になっている．

2.2 文章の一貫性

Putra ら [2] は教師なし学習による手法であるグラフ構造を用いた評価式 PAV・SSV・MSV を提案している．特に式 (1) で一貫性を定義した PAV は検証タスクでそれまでの研究の一貫性の評価実験の精度を上回っている．式 (1) での S_i は i 番目の文の単語の分散表現の平均を表しており，文 S_i の一貫性は $0 \leq \mathrm{sim}$ を満たすまで $i-1$ から j の値を減少させていくことで決定する．

$$\mathrm{sim}(S_i, S_j) = \alpha \ \mathrm{uot}(\mathrm{Sw}_i, \mathrm{Sw}_j) + (1-\alpha)\cos(\vec{S_i}, \vec{S_j}) \quad (1)$$

$$\mathrm{tc} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{sim}(S_i) \quad (2)$$

Copyright(C) 2020 The Association for Natural Language Processing.
All Rights Reserved.

ここで uot は比較文章の単語の重複率を表す値で，計算には i 番目の文の単語群 Sw_i と j 番目の文の単語群 Sw_j を用い，0 ～ 1 の値をとる．cos はコサイン類似度で，単語の分散表現を平均化したものを文のベクトルとして扱い，文同士のコサイン類似度を計算している．また，α は uot とで cos の比率を変更するもので 0 ～ 1 の割合で示される．先行研究ではこの値を動かすことでタスクに応じた最適な値を求めている．最終的に，この sim の平均を文章内の文数 N を用いた式 (2) で求め，その値を文章全体の一貫性の評価値として用いている．しかし，これらの先行研究は全て，人間によって書かれた一つの文章を評価するものであり，機械によって生成された文章に対して最適化された評価指標ではない．そこで，評価指標として人間が書いた文章と機械が生成した文章を一貫性の面から識別することができる手法を提案する．

3 提案手法

本研究では，より生成文章の検出に特化させた文章の一貫性の測定する 2 つの手法を提案した．提案手法では単語の分散表現を文単位で平均を取り，文ベクトルとした．各評価式は 2 つの一貫性評価式から構成され，ハイパーパラメータ α を 0 ～ 1 の値で変化させることでタスクに最適な値を求めた．また，文章の文・単語分割と見出し語変換には python の StanfordNLP パッケージ [3] を，単語の分散表現の獲得には事前学習済みの ELMo[4] を用いた．また，提案手法 2 は入力文章がどこまでなのかをという情報を既知のものという条件下で行った．

3.1 提案手法 1:CPCO

1 つ目の提案手法 CPCO(Consistency of against Preceding sentence using Cosine words Overlapping) は先行研究の評価式である PAV を改良したものである．CPCO では，PAV の基本的な概念である前文との文ベクトルの類似度と単語の重複率で文の一貫性を数値化することは同じである．しかし，単語の重複を検出する部分において，全く同じ単語のみを検出する uot をコサイン類似度を用いて同じ意味の単語を検出することのできる coswot に変更することでより曖昧な変化を捉えられるようにした．文 S_i の一貫性 $\mathrm{coh}(S_i)$ を式 (3) で定義し，その文章全体での一貫性を式 (4) で定義した．

$$\mathrm{coh}(S_i) = \alpha\ \mathrm{coswot}(\mathrm{Sw}_i, \mathrm{Sw}_{i-1}, \gamma) + (1-\alpha)\cos(\vec{S_i}, \vec{S_{i-1}}) \quad (3)$$

$$\text{text coherence} = \frac{1}{N}\sum_{i=0}^{N}\mathrm{coh}(S_i) \quad (4)$$

ここで，coswot は i と $i-1$ 番目の文の単語を分散表現に変換した Sw_i を Sw_{i-1} の各単語とコサイン類似度を用いて単語の類似度を計算し，閾値 γ を上回る単語を Sw_i の類似語と捉えることで単語の意味の重複を計算したものである．

3.2 提案手法 2:CICO

2 つ目の提案手法 CICO(Consistency of against Input sentences using Cosine words Overlapping) は，CPCO を更に生成タスクに特化させたものである．CICO では，入力文章がどこまでなのかという情報を追加で渡すことで入力文章とその後に続く文章を区別できるようにした．これにより，文章が自然な流れであるかを入力文章との文ベクトルの類似度や coswot による比較で検出することが可能にした．CICO の計算式は式 (5) のように定義し，提案手法 1 と同様にその文章全体の一貫性を平均を式 (4) で定義した．

$$\mathrm{coh}(S_i) = \alpha\ \mathrm{coswot}(\mathrm{Sw}_i, \mathrm{Sw}_{\mathrm{Input}}, \gamma) + (1-\alpha)\cos(\vec{S_i}, \vec{S_{\mathrm{Input}}}) \quad (5)$$

ここで，coswot は入力文章の単語の分散表現群 $\mathrm{Sw}_{\mathrm{Input}}$, 評価文章の単語の分散表現群 Sw_i を用いて計算し，コサイン類似度 cos は入力文章のすべての文ベクトルの平均 $\vec{S_{\mathrm{Input}}}$ と評価文章の文ベクトル $\vec{S_i}$ を用いて計算した．

4 実験環境

本研究では言語モデルに GPT-2 を使用し，実験時に公開されていた中で最大のパラメータ数 774M，その他の設定は GPT-2 の論文 [1] の Zero-shot の設定で生成される token 数を 300 に指定して用いた．実験文章作成では，人間が書いた文章の先頭 (平均 120 word) を入力文章として GPT-2 に入力し，その先の文章 (約 300 word) を GPT-2 で自動的に生成する形で行った．また，この生成文

Copyright(C) 2020 The Association for Natural Language Processing.
All Rights Reserved.

章を各入力文章に 5 サンプル生成した．実験に使用した文章は BBC Dataset[5] のジャンル 5 種類 (BBC_business, BBC_Entertainment, BBC_politics, BBC_sports, BBC_tech), Gutenberg Dataset[6] (Story), Wikipedia Database dumps[7](Wikipedia) の計 7 カテゴリより 1 カテゴリ 5 個の文章を選択し，図 1 のように選択された文章から GPT-2 を用いて文章を生成した．

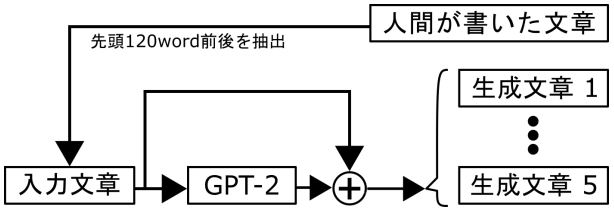

図 1: 実験データの作成方法

5 実験結果

提案手法を評価する方法として，人間が書いた文章を正解，GPT-2 が生成した文章を不正解と定義し，人間が書いた文章の一貫性の評価値が GPT-2 により生成された文章の評価値より高かったとき，生成文章の検出が可能であるとすることで提案手法がどれだけ生成文章を検出できるかを評価した．実験では全ての手法に対して $\alpha = [0.0 \sim 0.9]$, $\gamma = [0.0 \sim 0.9]$ の各組合せを計算した．表 (1) はその組み合わせの中で全カテゴリの平均が最も良い結果を得られた値のみを記載した．(PAV : $\alpha = 0.0$, CPCO : $\alpha = 0.3, \gamma = 0.1$, CICO : $\alpha = 0.2, \gamma = 0.0$) 結果より，すべてのカテゴリにおいて，提案手法 2 つが従来手法の PAV を共に上回ることが確認できた．

表 1: 各手法の最良結果正解率

Category	PAV	CPCO	CICO
BBC_Business	0.85	**1.00**	**1.00**
BBC_Entertainment	0.76	0.99	**1.00**
BBC_politics	0.83	0.98	**1.00**
BBC_sports	0.88	0.99	**1.00**
BBC_tech	0.86	**0.99**	0.98
Story	0.62	0.76	**0.80**
Wikipedia	0.94	**1.00**	**1.00**
Mean	0.820	0.958	**0.968**

6 考察

従来手法の PAV と提案手法 1 の結果の比較より提案したコサイン類似度を用いて単語の重複率を計算する手法 (coswot) が先行研究の手法 (uot) より，生成文章の識別タスクでは有効であることが確認できた．これは，uot では同じ単語の繰り返しのみしか評価できなかったが，coswot では言い換えや似ている単語の評価を可能にし，人間特有の高度な言い換えや語の移り変わりを捉えることが可能になったからであると考えられる．また，Story カテゴリは他のカテゴリと比較すると顕著に検出率が低かった．これには様々な要因があると考えられるが，代表的なものとして文体が他のカテゴリと大きく傾向が異なることが考えられる．物語は場面転換や感動詞・口語表現が多い傾向にあり，単純な単語の重複率や文ベクトルの類似度ではそのような文学的な表現を一貫性として評価することが難しいことが生成文章の検出率に影響したと考えられる．

7 おわりに

本研究では一貫性を用いて生成された文章を識別することが可能な評価手法 CPCO と CICO を提案した．また評価実験により，提案手法の 2 つは生成タスクにおいては従来手法より優れた検出力があることが確認できた．また，提案手法 2(CICO) は Story 以外のカテゴリにおいて 100%に近しい識別が可能であり実用できる高い性能があることが分かった．今後の課題として，本研究では入力文章と生成文章を組み合わせた文章を研究の対象としたが，提案手法では検出する文章群が全て生成文章だった場合，正しく検出することができない．そこで，入力する文章の種類に依存せず，全て生成文章だった場合などにも対応できる手法を考案したい．

参考文献

[1] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[2] Jan Wira Gotama Putra and Takenobu Tokunaga. Evaluating text coherence based on se-

Copyright(C) 2020 The Association for Natural Language Processing.
All Rights Reserved.

mantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pp. 76–85, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 160–170, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[4] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[5] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pp. 377–384. ACM Press, 2006.

[6] Shibamouli Lahiri. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 96–105, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[7] enwiki dump progress on 20191120. `https://dumps.wikimedia.org/enwiki/20191120/`. (Accessed on 11/22/2019).

Copyright(C) 2020 The Association for Natural Language Processing.
All Rights Reserved.