

# 関連研究節自動生成に向けた引用論文の最適配置

大鹿雅史 笹野遼平  
名古屋大学大学院情報学研究科

oshika.masashi.f6@es.mail.nagoya-u.ac.jp  
sasano@i.nagoya-u.ac.jp

## 概要

関連研究節生成のためには、引用する論文を決定した上で、どのような段落構成や順序で論文を引用するかという論文配置を考える必要がある。しかし、関連研究節自動生成に関する既存研究では、引用すべき論文を推薦する引用推薦や、引用論文と引用順が与えられた場合に関連研究節を自動生成する研究は多く行われているものの、引用論文の最適配置には着目されてこなかった。本研究では、関連研究節の自動生成に向けた引用論文の最適配置に取り組む。具体的には、引用論文を段落ごとにまとめるクラスタリング、段落順の決定、段落内の引用順決定という3タスクに分解し、各タスクを解くことで引用論文の最適配置を実現する。

## 1 はじめに

学術論文における関連研究節は、その研究の立ち位置や新規性を示すために重要であるが、読者に研究を適切に理解してもらうためには、適切な論文を引用した上で、それらの論文の関係性を考慮し、適切な順序・段落構成で引用することが必要となる。関連研究節の自動生成に向けた既存研究として、引用すべき論文の推薦や節生成を行う研究がある。しかし、引用推薦の研究では基本的に候補論文集合から引用すべき論文の推薦のみに取り組んでおり、論文内での引用順の推薦は行われない [1, 2, 3]。また、節生成の研究では生成時に論文の引用順を与えた上で生成に取り組んだ研究が多く、引用順が既知でない場合はそれらを自動的に決める必要がある [4, 5, 6]。このように、既存研究はどのような段落構成や順序で論文を引用するかという論文配置に取り組んでおらず、完全な自動生成は実現できていないことが指摘されている [7]。

そこで、本研究では関連研究節の自動生成に向けて、引用論文の最適配置に取り組む。具体的には、

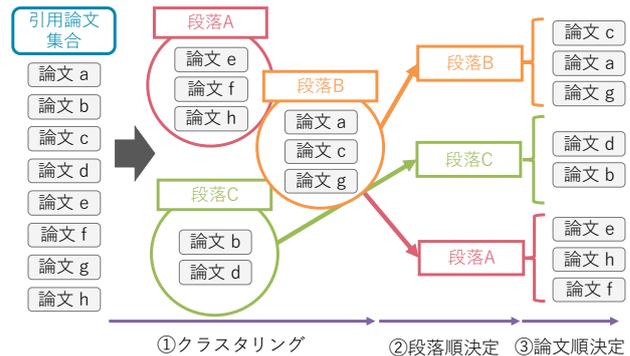


図1 最適配置に向けたタスクの概要図。

関連研究節で引用された論文一覧を与えられた場合に、それらを引用される段落ごとにまとめた上で論文の引用順を決定するタスクに取り組む。提案する手法では、引用論文の最適配置問題を、図1に示すように、引用論文を段落ごとにまとめるクラスタリング、段落順の決定、段落内の引用順決定という3タスクに分解し、各タスクを解くことで引用論文の最適配置を実現する。

## 2 データセット

モデルの学習や評価に使用するため、関連研究節で引用された論文の一覧と、それらの段落構成および引用順の情報をまとめたデータセットを作成した。データセットの作成には2022年8月までにACL anthologyに掲載された80,013本の論文をまとめたコーパスであるACL OCL[8], Semantic Scholar上の論文データが保持されたS2ORC[9], および、PDFを解析しテキスト化するツールであるPDFNLT<sup>1)</sup>を利用した。データセットの作成手順は以下の通りである。

1. ACL OCL から ACL, EMNLP, NAACL, TACL で2013年~2022年に発表された論文を抽出
2. PDFNLTを利用してテキスト化した後、“Related Work”という節が存在する論文を抽出

1) <https://github.com/KMCS-NII/PDFNLT-1.0>

表 1 データセットの統計値. () 内は平均を示す.

収集論文数	2,869
段落数	8,709 (3.0)
引用論文数	56,765 (19.8)

- “Related Work” に 2 つ以上の論文が引用されている段落が複数存在, かつ総引用数が 10 以上の論文を抽出
- 引用論文のタイトルをもとに S2ORC から著者情報や引用関係の情報を収集

作成したデータセットの統計値を表 1 に示す.

### 3 引用論文クラスタリング

まず, 与えられた論文の Related Work の節で引用された論文 (引用論文) 集合を, 引用される段落ごとにまとめる引用論文クラスタリングに取り組む. 本研究では全引用論文間の情報を考慮できるように, 図 2 に示すように引用論文をノードとしたグラフを作成し, クラスタリングを行う.

#### 3.1 グラフの構築

全引用論文間の情報を考慮するために引用論文をノードとして任意のノード間にエッジの存在する完全グラフ  $V$  を構築する. エッジの重みは二つの引用論文が同一段落内で言及されやすさを表すものとし, 大規模言語モデル (Large Language Model; LLM) を用いて算出する. 具体的には, LLM を二つの引用論文が引用元論文の同じ段落で言及されていれば 1, 異なる段落で言及されていれば 0 を出力する 2 値分類タスクでファインチューニングする. ファインチューニング後のモデルに二つの論文を入力し, 出力された値をエッジの重みとして利用する.

エッジの重みを算出する際の LLM への入力については 2 つの設定を比較する. 一つ目は, 引用論文のタイトル, 概要のみを入力するモデル ( $M_{base}$ ) である. 二つ目は, さらに引用元論文のタイトルと概要, 引用論文の出版年, 著者情報, 引用論文間の引用関係の有無も入力する拡張モデル ( $M_{extd}$ ) である.

#### 3.2 グラフクラスタリング

本研究ではグラフのクラスタリング手法として normalized cut (Ncut) [10] を利用する. ノード  $i, j$  間の重みを  $w_{ij}$  としたとき, 分割後のクラスタ  $A$  と

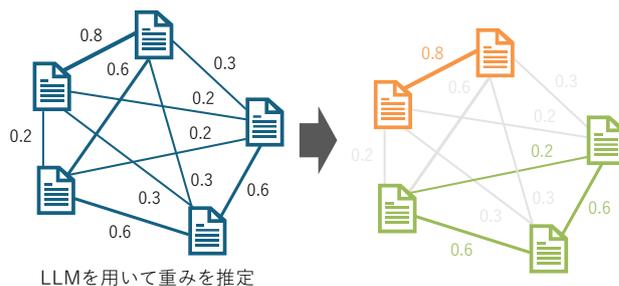


図 2 引用論文クラスタリングの概要図.

クラスタ  $B$  間のエッジの重みの総和を式 (1) で定義する. このとき, Ncut は式 (2) を最小化するクラスタリング手法であり, クラスタ内の重みの総和を大きく, クラスタ間の重みの総和を小さくすることを目的とした手法である. 3.1 節で作成したグラフ  $V$  に Ncut を適用することで, 引用される段落ごとにクラスタリングを行う.

$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \quad (1)$$

$$\text{Ncut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Cut}(A, V)} + \frac{\text{Cut}(A, B)}{\text{Cut}(B, V)} \quad (2)$$

#### 3.3 実験

**実験設定** 実験には 2 節で作成したデータセットを, 論文数が訓練データ, 検証データ, テストデータで 8 : 1 : 1 となるように分割し利用した. Ncut を利用する場合, クラスタ数は事前に与える必要があるが, 本研究では実際の引用元論文での段落数を利用した. 評価指標には B-Cubed 適合率, 再現率, およびその調和平均である F 値 [11] を利用した. また, LLM には事前学習済みの LLaMA3.1-8B-Instruct<sup>2)</sup> を利用し, モデルのファインチューニングには LoRA [12] を用いた<sup>3)</sup>.

また, 比較手法として LLM を用いたベースライン (LLM ベースライン) を構築した. LLM ベースラインは引用元論文と全ての引用論文を入力し, 各引用論文がどのクラスタに属するかを生成するタスクで LLM をファインチューニングするベースラインである. LLM ベースラインには LLaMA3.1-8B-Instruct を利用し, LoRA を用いてファインチューニングを行った. 論文情報として引用元論文のタイトルと概要, 引用論文のタイトル, 概要, 出版年, 著者情報を利用した<sup>4)</sup>.

2) <https://huggingface.co/meta-llama/LLama-3.1-8B-Instruct>

3) 実験設定の詳細を Appendix A に示す.

4) 実験設定の詳細を Appendix B に示す.

表2 引用論文クラスタリングの実験結果.

モデル	適合率	再現率	F 値
LLM ベースライン	0.696	0.698	0.697
$M_{base}$	0.702	0.711	0.706
$M_{extd}$	<b>0.708</b>	<b>0.726</b>	<b>0.717</b>

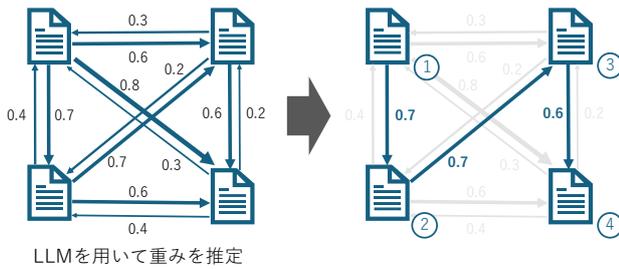


図3 順序決定の概要図.

**実験結果** 表2に引用論文クラスタリングの実験結果を示す. 実験結果より, 提案手法はLLM ベースラインよりも高いF値を達成しており, グラフを利用して論文間の関係を考慮してクラスタリングを行うことの有効性が確認できる. また,  $M_{extd}$  が  $M_{base}$  よりも高いF値を示していることから, 引用元論文のタイトルと概要や, 引用論文の出版年, 著者情報, 引用論文間の引用関係等の情報を追加することでエッジ間の重みをより正確に評価できるようになったと考えられる.

## 4 段落順決定

次に, 前節で作成した段落を関連研究節内で言及すべき順番に並び替える段落順決定に取り組む. 本研究では図3に示すように段落をノードとした有向グラフを作成し, 段落順を決定する.

### 4.1 グラフ構築

各段落をノードとして任意のノード間に両向きにエッジが存在する有向グラフを構築し, その後, エッジに二つの段落の順序関係を考慮した重みを付与する. エッジの重みは二つの段落の隣接しやすさと定義し, LLMを用いて算出する. 具体的には, 二つの段落  $A, B$  を LLM にこの順番で入力し, 引用元論文内で段落  $A$  が段落  $B$  の直前にあれば1, そうでなければ0を出力する2値分類タスクで LLM をファインチューニングする. ファインチューニングしたモデルに段落  $A, B$  を入力し, 出力された値をエッジの重みとする. LLM に入力する情報は引用

表3 段落順推薦の実験結果.

モデル	$\rho$
LLM ベースライン	0.513
$M_{base}$	0.470
$M_{extd}$	<b>0.554</b>

論文クラスタリングと同様に基本モデル  $M_{base}$  と拡張モデル  $M_{extd}$  の2つのモデルを比較する. ただし,  $M_{extd}$  は引用論文クラスタリングと異なり, 引用元論文のタイトルと概要, 引用論文の出版年, 著者情報を  $M_{base}$  に追加して入力する.

### 4.2 順序決定

グラフからの順序決定は, 式(3)で定義する目的関数を最大化する順列  $\pi$  を求めることで行う.

$$\arg \max_{\pi} \sum_{1 \leq i < j \leq n} w_{\pi(i)\pi(j)} \quad (3)$$

ここで,  $\pi(i)$  は順列  $\pi$  において  $i$  番目に配置される段落番号を示し,  $w_{\pi(i)\pi(j)}$  は段落  $\pi(i)$  から段落  $\pi(j)$  へのエッジの重みを表す. 式(3)を満たす順列  $\pi$  は, 重みの総和が最大となる段落順になる.

### 4.3 実験

**実験設定** LLM には LLaMA3.1-8B-Instruct を利用した. 評価指標にはスピアマンの順位相関係数  $\rho$  を用いた. 比較手法には, 引用元論文と段落ごとに分割した全ての引用論文を LLM に入力し, 段落順を生成するタスクでファインチューニングしたモデル (LLM ベースライン) を用いた. LLM ベースラインも同様に LLaMA3.1-8B-Instruct を利用し, 拡張モデル  $M_{extd}$  と同一の情報を入力した.

**実験結果** 表3に実験結果を示す.  $M_{extd}$  は同一の入力を使用している LLM ベースラインよりも0.041ポイント高いスコアとなり, 提案手法の有効性が確認できる.  $M_{base}$  と  $M_{extd}$  を比較すると後者の方が0.084ポイント高いスコアとなっており, エッジの重み推定のために LLM に入力する情報を増やすことの効果は非常に大きいと考えられる.

## 5 段落内論文順決定

最後に, 段落内の論文を言及順に並び替える段落内論文順決定に取り組む. 論文順決定においても図3のように引用論文をノードとした有向グラフを作成し, 順序を決定する.

## 5.1 グラフの構築

はじめに、各論文をノードとして任意のノード間に両向きにエッジの存在する有向グラフを構築する。その後、引用論文が段落内でどれくらい離れているかを考慮したエッジの重みを LLM を用いて算出する。具体的には、二つの引用論文  $a, b$  をこの順番で LLM に入力し、 $\frac{1}{\pi(b)-\pi(a)}$  を推定する回帰タスクでファインチューニングを行う。ここで、 $\pi(a)$  は段落内での論文  $a$  の言及順を表す。このモデルに二つの論文を入力することで、段落内での論文間の距離が取得でき、論文順の決定に利用する。LLM に入力する情報は引用論文クラスタリングと同様に、基本モデル  $M_{base}$  と拡張モデル  $M_{extd}$  の二つのモデルを比較する。

## 5.2 順序決定

論文順の決定は式 (4) で定義する目的関数を最小化する順列  $\pi$  を求めることで行う。

$$\arg \min_{\pi} \sum_{1 \leq i < j \leq n} \left| w_{\pi(i)\pi(j)} - \frac{1}{j-i} \right| \quad (4)$$

ここで、 $\pi(i)$  は順列  $\pi$  において  $i$  番目に配置される論文番号を示し、 $w_{\pi(i)\pi(j)}$  は論文  $\pi(i)$  から論文  $\pi(j)$  へのエッジの重みを表す。式 (4) は予測誤差を最小化する関数であり、これを満たす  $\pi$  は段落内での論文順を適切に予測した順列となる。

## 5.3 実験

**実験設定** LLM には LLaMA3.1-8B-Instruct を利用し、ファインチューニングにはデータセットから論文ごとに作成したペアを利用した。評価には、スピアマンの順位相関係数  $\rho$  を段落ごとに計算し、その平均値により評価した。論文順決定では、段落内の論文数が多かった際に全順列について式 (4) を求めるのは計算量の問題で困難である。そこで本研究では最適解を求めた上で評価を行うために、段落内論文数が 10 未満であるデータ (921 件中 777 件) のみを用いて評価を行った。

また、比較手法として LLM ベースラインと出版年をもとにしたベースライン (出版年ベースライン) を構築した。論文順決定の LLM ベースラインは引用元論文と段落内の引用論文すべてを LLM に入力し、論文順を生成するタスクでファインチューニングするモデルである。LLM には LLaMA3.1-8B-Instruct を利用し、論文情報には引用

表 4 段落内論文順推薦の実験結果.

モデル	$\rho$
出版年ベースライン	0.447
LLM ベースライン	0.409
$M_{base}$	0.354
$M_{extd}$	<b>0.452</b>

元論文のタイトルと概要、引用論文のタイトル、概要、出版年、著者情報を利用した。出版年ベースラインは段落内の引用論文を出版年が早い順に並べるベースラインである。

**実験結果** 表 4 に実験結果を示す。実験結果より、出版年ベースラインは LLM ベースラインよりも高いスコアを示しており、段落内論文順決定には出版年が大きく影響することが分かる。 $M_{base}$  と  $M_{extd}$  を比較すると、およそ 0.1 ポイントの差があり、段落内の論文順序を決める際は、引用論文のタイトルと概要だけでは不十分であると考えられる。一方で、出版年ベースラインと提案手法  $M_{extd}$  の差はわずかであり、性能のさらなる向上が課題となる。

## 6 おわりに

本研究では、関連研究節の自動生成に向けて引用論文クラスタリング、段落順決定、段落内論文順決定の 3 つのタスクを解き、引用論文の最適配置に取り組んだ。実験の結果から全てのタスクにおいて LLM を生成タスクでファインチューニングして利用するよりも、LLM を利用して重みを推定したグラフを利用することで高いスコアを示すことが確認できた。今後の展望として、特に段落内論文順決定の性能向上を目指すとともに、3 つのタスクを包括した評価を行う。また、実際に関連研究節の自動生成に取り組むことで引用論文の配置を行うことが節生成の品質に与える影響について検証を行う予定である。

## 謝辞

本研究は、JST ムーンショット型研究開発事業 JPMJMS2033 の支援を受けたものです。

## 参考文献

- [1] Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. Content-based citation recommendation.

- In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, pp. 238–251, 2018.
- [2] Yang Zhang and Qiang Ma. Dual attention model for citation recommendation. In **Proceedings of the 28th International Conference on Computational Linguistics (COLING)**, pp. 3179–3189, 2020.
- [3] Sayar Ghosh Roy and Jiawei Han. ILCiteR: Evidence-grounded interpretable local citation recommendation. In **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 8627–8638, 2024.
- [4] Cong Duy Vu Hoang and Min-Yen Kan. Towards automated related work summarization. In **Proceedings of the 23rd International Conference on Computational Linguistics (COLING)**, pp. 427–435, 2010.
- [5] Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. Capturing relations between scientific papers: An abstractive model for related work section generation. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)**, pp. 6068–6077, 2021.
- [6] Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. Target-aware abstractive related work generation with contrastive learning. In **Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)**, pp. 373–383, 2022.
- [7] Xiangci Li and Jessica uyang. Related work and citation text generation: A survey. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 13846–13864, 2024.
- [8] Shaurya Rohatgi, Yanxia Qin, Benjamin Aw, Niranjana Unnithan, and Min-Yen Kan. The ACL OCL corpus: Advancing open science in computational linguistics. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 10348–10361, 2023.
- [9] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 4969–4983, 2020.
- [10] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 22, No. 8, pp. 888–905, 2000.
- [11] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In **36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING)**, pp. 79–85, 1998.
- [12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In **The Tenth International Conference on Learning Representations (ICLR)**, 2022.

表 5 引用論文クラスタリングの  $M_{extd}$  で利用したプロンプト.

---

```

**Seed Paper**
Title: {Title of Seed Paper}
Abstract:{Abstract of Seed Paper}
**Paper1**
Title: {Title of Paper1}
Abstract:{Abstract of Paper1}
Author: {Author of Paper1}
Year of Publication: {Year of Publication of Paper1}
**Paper2**
Title: {Title of Paper2}
Abstract:{Abstract of Paper2}
Author: {Author of Paper2}
Year of Publication: {Year of Publication of Paper2}
**Citation linkage**
{True or False}

```

---

## A 実験設定：提案手法

LoRA は全ての線形層に対して適用し、ハイパーパラメータは  $r$  を 8,  $\alpha$  を 16 に設定した. ファインチューニングを行う際のパラメータは3つのタスク全てで初期の学習率を  $1e-4$  とし線形に減衰させ最適化手法に AdamW を利用した. 引用論文クラスタリングではバッチサイズを 8, エポック数を 1 に、段落順決定ではバッチサイズを 2, エポック数を 3 に、論文順決定ではバッチサイズを 8, エポック数を 1 としてファインチューニングを行った.

引用クラスタリングの拡張モデル  $M_{extd}$  で利用したプロンプトを表 5 に示す. 引用元論文 (Seed Paper) と二つの引用論文を入力し同一段落での言及されやすさを取得する.

表 6 引用論文クラスタリングの LLM ベースラインで利用したプロンプト.

---

```

You are a researcher specializing Natural Language
Processing.
I am writing scientific paper with the following title and
abstract.
Title: {Title of Seed Paper}
Abstract:{Abstract of Seed Paper}
Please cluster the following {number of references} references
for {number of clusters} clusters by topic.
**references**
Title: {Title of Reference1}
Abstract:{Abstract of Reference1}
Author: {Author of Reference1}
Year of Publication: {Year of Publication of Reference1}
. . .

```

---

## B 実験設定：LLM ベースライン

LLM ベースラインも提案手法と同様に LoRA を利用してファインチューニングを行った. LoRA は全ての線形層に対して適用し、ハイパーパラメータは  $r$  を 16,  $\alpha$  を 32 に設定した. ファインチューニングを行う際のパラメータは初期の学習率を  $1e-4$  とし線形に減衰させ、バッチサイズを 1, エポック数を 3, 最適化手法に AdamW を利用した.

実験に利用した引用クラスタリングにおけるプロンプトの例を表 6 に示す. **\*\*references\*\***に続いて、クラスタリングする全ての引用論文の情報を入力し、段落ごとに引用論文タイトルを生成する.