Pre-trained Transformer による 引用文脈を考慮した引用ネットワーク埋め込み

大萩雅也 [1]　相澤彰子 [2,1]
[1] 東京大学大学院　[2] 国立情報学研究所
ohagi-masaya999@g.ecc.u-tokyo.ac.jp　aizawa@nii.ac.jp

概要

論文間の引用関係を表した引用ネットワークをベクトル空間に投影することで得られる論文の埋め込みは論文推薦や論文分類など研究支援に関する様々なタスクに役立てることができる。この埋め込みに関する研究は多くあるが、引用関係に付随する引用の目的を取り込んだ埋め込みに焦点を当てた研究は多くない。そこで我々は、論文を引用する際に記述される文章である引用文脈を引用の目的の情報源であると捉える。そして Pre-trained モデルの一種である SciBERT を、被引用論文を引用元論文と引用文脈から予測するタスクで訓練することで引用文脈を捉えた引用ネットワーク埋め込みを作るモデルを作成する。実験を通して提案手法は引用論文推薦、論文分類上で既存手法を超える性能を示した。

1　はじめに

論文を引用するという行為は、その分野における研究の進展、そしてその中に自分の研究をどう位置づけるかを示すという点において論文執筆の際には不可欠なものである。そして論文間の引用関係はこの世に存在する数多くの論文の間にどのような関係性が存在するかを突き止めるための鍵の一つとなっており、それぞれの論文をノード、論文間の引用関係をエッジと捉えた時それらは引用ネットワークと呼ばれるグラフを構成する。さらにこのグラフをベクトル空間に投射することで得られたそれぞれの論文の埋め込みは論文推薦 [1]、論文の可視化 [2]、もしくは論文の分類 [3] など研究活動を支援するための様々なタスクに役立てることができる。

このベクトル空間への投射は引用ネットワーク埋め込みと呼ばれ数多くの先行研究 [4] が存在するが、どのような目的である論文が別の論文を引用しているかに注目した研究は数少ない。引用とは常に肯定的なものではなく、時には過去の研究に対する批判を目的として行われるものであり、さらに、ACL-ARC [5] データセットでは引用の目的を 6 種類に分類しているように単純な肯定/否定で捉えられるものでもない。これらの引用目的を考える上で重要となるのが引用文脈である。

引用を行う際に引用記号 ([5] など) とともに記述される文章である引用文脈は引用の目的を特定する上で重要な役割を果たしており [6]、その引用文脈をエッジの特徴として論文埋め込みに取り込んだ引用ネットワーク埋め込みは多様な引用の目的を取り入れたものとしてより正確に論文間の関係を捉えることが期待される。しかしながらその重要性、応用可能性にもかかわらず十分な数の既存研究が存在するとは言いがたい。

以上に基づき、我々は本研究で Pre-trained Transformer を活用した新たなモデルを提案する。我々は引用元論文とその論文内の引用文脈から被引用論文を予測させる Masked Paper Prediction(MPP) タスクを提案し、そのタスクを用いて SciBERT [7] を fine-tune することで引用文脈を考慮した引用ネットワーク埋め込みを作成した。さらに損失関数として、特定の引用文脈による被引用論文だけでなく引用関係にあるその他の近隣ノードにも注意を配る Structure-Aware Cross-Entropy Loss を提案する。

実験の結果、我々の手法は二つの引用ネットワーク上での引用論文推薦と論文分類において (1) 引用文脈を考慮した引用ネットワーク埋め込みの先行研究である hyperdoc2vec [8] を超える性能を示し、(2) 引用文脈を考慮しない引用ネットワーク埋め込みの state-of-the-art である RotatE [9] と匹敵する性能を出した。我々の手法は引用ネットワークだけでなくレビューサイトでの user2item のグラフ [10] などエッジに文章情報を持つその他のグラフへの応用可能性が期待される。

2 提案手法

2.1 引用文脈を考慮した引用ネットワーク

我々が対象とする引用ネットワークは {引用元論文, 引用文脈, 被引用論文} のトリプルの集合として捉えられる。それぞれの論文はネットワーク内の論文の集合 V に属し、論文間の引用関係は引用文脈という文章情報を特徴にもつ、論文ノード間のエッジとして捉えられる。本研究の目標はこの引用ネットワークのグラフ構造と、エッジの特徴としての引用文脈を捉えた論文埋め込みを作り出すことである。

2.2 既存手法 : hyperdoc2vec

引用文脈を考慮した引用ネットワーク埋め込みの代表的な既存手法としては word2vec [11] を応用した hyperdoc2vec [8] が挙げられる。この手法は後続の研究 [12] にとってもその基礎となる重要なものであるが、この手法には (1) 4.1 節での RotatE との比較からわかるようにグラフ構造を捉える力が弱く、そして (2) ネットワーク埋め込みのコーパスのみを訓練に用いているため大規模コーパスを用いて事前学習したモデルに比べて言語知識や学術知識に乏しいという 2 つの問題点が存在する。

2.3 提案手法

既存手法の問題を解決するために、Transformer [13] を用いナレッジグラフ埋め込みで高い性能を上げた CoKE Model [14] を参考に、これを引用文脈を考慮するように拡張した新たな手法を提案する。提案手法は学術論文上で事前学習された SciBERT を Masked Paper Prediction という我々の提案タスクで訓練することで引用ネットワークのグラフ構造と引用文脈の両方を学習することを目的とする。

Masked Paper Prediction は引用元論文と引用文脈から被引用論文を予測するタスクであり、概略を図 1 に示す。具体的な手順としては、引用ネットワーク内のそれぞれのトリプル {引用元論文, 引用文脈, 被引用論文} に対し、我々はまず引用文脈をトークナイズする。そして引用文脈内の引用記号の部分 ([1] や (Devil et al. 2019) など) をマスクトークンで置き換え、そのトークン列と引用元論文 id を連結したものを SciBERT に入力する。そして SciBERT から出力された埋め込み列のうち、マスクされたトークンの場所に相当する位置の出力を linear 層に入力するこ

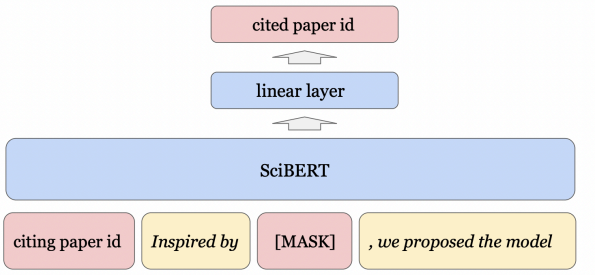

図 1 提案手法の概略

とで被引用論文を予測する。さらに、正解ラベルとの間の損失を用いてモデル全体を訓練することで、それぞれのトリプル内の関係をモデルに学習させていく。損失関数には Cross-Entropy Loss、もしくは我々が新しく提案した Structure-Aware Cross-Entropy Loss を使用する。

CoKE Model はネットワークのグラフ構造を捉える能力があることが [14] で示されており、我々の手法はその点で hyperdoc2vec のグラフ構造に対する弱さを補うことが期待される。さらに、SciBERT の学術論文に対する専門知識と言語知識を活用することで引用文脈のより良い埋め込みが期待される。

2.4 損失関数

我々のモデルで用いられる損失関数は Cross-Entropy loss (CE) と新たに提案する Structure-Aware Cross-Entropy Loss (SACE) の 2 種類である。まず通常の CE は以下の数式で定義される。

$$\begin{aligned} loss(x, d_{cited}) &= -log(\frac{exp(x[d_{cited}])}{\sum_j exp(x[j])}) \\ &= -x[d_{cited}] + log(\sum_j exp(x[j])). \end{aligned} \tag{1}$$

CE はある論文 A が特定の引用文脈で引用している論文 B とそれ以外の論文を区別することができるが、論文 B 以外の論文 A と引用関係にある論文 C,D も完全な誤りと見なしてしまう。しかしながら我々の手法は引用ネットワークのグラフ構造を学習することが目的の一つであるため、隣接する論文に報酬を与えることはその学習に有用であると考えられる。hyperdoc2vec をベースとして Structure-Aware な引用論文推薦を提案した [12] は入力に隣接する論文を付け加えることでこれを達成したが、我々は損失関数で実現することを目的として以下の数式からなる SACE を提案する。d_{citing} は引用元論文、d_{cited} は被引用論文、$N(d_{citing})$ は引用元論文の隣接論文を表し、γ は被引用論文と隣接論文のどちら

を重視するかのハイパーパラメータである。なお、$N(d_{citing}) = 1$ の時は通常の Cross-Entropy Loss を用いた。今回の実験では $\gamma = 0.8$ とした。

$$loss(x, d_{cited}) = -\gamma log(\frac{exp(x[d_{cited}])}{\sum_k exp(x[k])}) - \frac{(1-\gamma)}{|N(d_{citing})| - 1} \sum_{j \in (N(d_{citing})/d_{cited})} log \frac{exp(x[j])}{\sum_k exp(x[k])} \quad (2)$$

3 実験

3.1 評価タスク

本研究では二つの評価タスクを用いた。一つ目は引用元論文と引用文脈から被引用論文を予測する引用論文推薦であり、提案手法の訓練タスクである Masked Paper Prediction と類似のタスクである。このタスクで我々はモデルが引用ネットワーク内のトリプルを正確に捉えられているかを評価する。二つ目はそれぞれの論文が扱っているタスク (機械翻訳、要約等) を引用ネットワークを用いて予測する論文分類である。このタスクはノードの埋め込みがネットワークのグラフ構造をどれほど捉えられているかを評価するために従来研究で用いられており [15, 4]、それぞれの埋め込みがどれほどグラフ構造を捉えられているかを評価する。

3.2 データセット

実験では二つの引用ネットワークを用いた。まず一つ目は FullTextPeerRead[16] と呼ばれる、ACL, ICLR, NIPS 上のネットワークであり、二つ目は ACL Anthology Sentence Corpus (https://github.com/KMCS-NII/AASC) と呼ばれる、ACL 上のコーパスから我々が作成したネットワークである。Appendix にそれぞれのデータセットの詳細を載せる。簡潔にいうと、AASC の方が FullTextPeerRead よりサイズが大きく、またネットワークの凝集度を表す Clustering Coefficiency が AASC は 0.135 であるのに対して FullTextPeerRead は 0.149 であることから後者の方がグラフ構造を活かした推論がしやすいと言える。それぞれのネットワークは訓練データ、評価データに分割され、訓練データを引用ネットワーク埋め込みの作成に、評価データを引用論文推薦タスクの評価に用いた。さらに、論文分類タスクのデータセットの作成にはそれぞれの論文が取り組んでいるタスクを抽出する必要があり、AASC のアノテーションには NLP-TDMS [17] 内で定義されている主要な NLP タスクのリスト、FullTextPeerRead のアノテーションには paperswithcode (https://paperswithcode.com/sota) 内の機械学習のタスクのリストを用いて論文のタイトルとアブストラクトから文字列一致でタスクを抽出した。

3.3 ベースライン

ベースラインとしては、hyperdoc2vec と RotatE[9] の 2 つの手法を用いた。前者は 2.2 節で紹介した、引用文脈を考慮した引用ネットワーク埋め込みの既存手法である。後者はは複素数空間上のベクトルとしてノードを、空間上の回転としてエッジを表現する手法であり、引用文脈を考慮しない引用関係の有無のみを含む引用ネットワークの埋め込みを作成するために用いた。我々は RotatE とその他の手法の比較によって引用文脈を特徴として用いることが埋め込みの性能向上につながるかどうかを検証する。

3.4 実験設定

提案手法においては、バッチサイズを 16、学習率を 5e-5 に設定した上で、5 エポックの訓練を行った。論文埋め込みはランダムに初期化したものを用いた。引用文脈の長さとしては、引用記号が含まれる文章とその前後一文を抜き出した上で、トークナイズした際のサブワードが引用記号の前後それぞれ 125 トークンに収まるように文章を切り出した。ベースラインの実験設定はそれぞれの論文に従った。また、論文分類タスクに関しては先行研究 [8] に従いそれぞれのモデルから論文の埋め込みを取り出した上で、SVM を用いて訓練、評価を行った。

4 結果

4.1 実験結果

表 1, 2 に示す通り、提案手法は hyperdoc2vec に比べて両タスク、両データセットにて高い性能を出しておりその有用性が確認できる。特に引用論文推薦における性能の向上は顕著であり、SciBERT を用いたことによって我々の手法が二つの論文の間の橋渡しとしての引用文脈をより正確に捉えることができるようになったことがわかる。さらに、論文分類における性能の向上からは提案手法がより正確にグラフ構造を捉えることに成功していることがわかる。

次に、Cross-Entropy Loss (CE) と我々の提案した

Structure-Aware Cross-Entropy Loss (SACE) を比べてみると、引用論文推薦において SACE を用いたモデルは CE を用いたものに比べて FullTextPeerRead では性能が下がっており、AASC では性能が上がっている。これは二つのネットワークの違いに起因すると考えられる。性能向上した AASC では、3.2 節に書いたように、グラフ構造を活かした推論がしやすく、SACE は引用文脈だけでなく周囲のグラフ構造を意識した訓練を行う損失関数であるためその効果が出たものと考えられる。ただ FullTextPeerRead においても SACE を活用することによる埋め込みの変化はあるものと思われ、論文分類における SACE を用いた提案手法の性能向上はグラフ構造をより意識した訓練方法の有用性を示している。

最後に我々の手法と RotatE を論文分類上で比較すると、FullTextPeerRead では我々の手法がより高い性能を収めており、ここからは引用文脈を考慮することによって論文の埋め込みがより他の論文との関係、そして論文が取り組むタスクに関する情報を捉えることに成功していることが見てとれる。ただ AASC においては RotatE の方がわずかではあるが高い性能を出しており、全ての条件に一貫する有用性まで示すことはできなかった。

表 1 引用論文推薦の結果

	FullTextPeerRead		AASC	
	MAP	MRR	MAP	MRR
hyperdoc2vec	0.190	0.204	0.105	0.111
CE+Proposed Model	**0.454**	**0.466**	0.308	0.323
SACE+Proposed Model	0.345	0.357	**0.335**	**0.347**

表 2 論文分類の結果

	FullTextPeerRead		AASC	
	Macro	Micro	Macro	Micro
hyperdoc2vec	0.329	0.391	0.701	0.804
CE+Proposed Model	0.395	0.484	0.740	0.835
SACE+Proposed Model	**0.438**	**0.495**	0.738	0.838
RotatE	0.324	0.415	**0.743**	**0.842**

4.2 Ablation Study

前節の通り、提案手法は既存手法である hyperdoc2vec に対して有意な性能向上を果たしたが、提案手法は既存手法に対して (1) Transformer の活用、(2) 事前学習の活用という二つの違いが存在し、どちらがより性能向上に貢献したかは定かでない。よって我々は (1) 事前学習なしの Transformer Model、(2) 学術知識を持たない Pre-trained Transformer Model である BERT [18]、 (3) SciBERT の 3 つを比較する Ablation Study を行った。引用論文推薦、論文分類における実験結果を表 3、4 に示す。まず BERT base と SciBERT base を比べると、FullTextPeerRead では SciBERT base の方が性能が上がっているものの、AASC では BERT base のモデルの方が性能が高くなっており、二つのモデルの間にはどちらかが全ての条件において優れていると言えるほどの差は存在しない。よって、提案手法の性能向上は論文に関する学術知識というよりは、言語一般に関する知識によるものであると考えられる。さらに、事前学習をしない場合とその他のモデルを比べてみると、事前学習をしない場合はかなり性能が落ちており、ベースラインである hyperdoc2vec にも劣る結果となっている。ここからは、提案手法の性能向上は Transformer Model からだけではなく、それを事前学習による言語知識と組み合わせた際にのみ得られるということが見てとれた。

表 3 引用論文推薦における Ablation Study

	FullTextPeerRead		AASC	
	MAP	MRR	MAP	MRR
no Pre-train	0.074	0.063	0.032	0.027
BERT base	0.412	0.425	**0.325**	**0.344**
SciBERT base	**0.454**	**0.466**	0.308	0.323

表 4 論文分類における Ablation Study

	FullTextPeerRead		AASC	
	Macro	Micro	Macro	Micro
no Pre-train	0.033	0.022	0.048	0.407
BERT base	**0.427**	0.466	0.732	0.833
SciBERT base	0.395	**0.484**	**0.740**	**0.835**

5 おわりに

本論文では Pre-trained Transformer を用いて引用文脈を考慮したネットワーク埋め込みを作成する新たな手法を提案した。この手法は引用論文推薦と論文分類に対して高い性能を発揮し、既存手法に対する優位性を示した。本研究では埋め込みの汎用的な評価を目指して論文分類での評価を行ったが、引用文脈の直接的な有用性に焦点を当てた評価タスクの導入は我々の手法の実用性を示す上で重要な一歩になると考える。さらに、引用ネットワーク以外にエッジに文章情報を特徴として持つネットワークにその適用範囲を広げることも重要な指針の一つである。

参考文献

[1] Chanathip Pornprasit, Xin Liu, Natthawut Kertkeidkachorn, Kyoung-Sook Kim, Thanapon Noraset, and Suppawong Tuarob. **ConvCN: A CNN-Based Citation Network Embedding Algorithm towards Citation Recommendation**, p. 433–436. Association for Computing Machinery, New York, NY, USA, 2020.

[2] Han Tian and Hankz Hankui Zhuo. Paper2vec: Citation-context based document distributed representation for scholar recommendation. **CoRR**, Vol. abs/1703.06587, , 2017.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In **International Conference on Learning Representations**, 2018.

[4] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, Vol. 32, No. 1, p. 4–24, Jan 2021.

[5] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the evolution of a scientific field through citation frames. **Transactions of the Association for Computational Linguistics**, Vol. 6, pp. 391–406, 2018.

[6] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In **Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing**, EMNLP '06, p. 103–110, USA, 2006. Association for Computational Linguistics.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[8] Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. hyperdoc2vec: Distributed representations of hypertext documents. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2384–2394, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[9] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In **International Conference on Learning Representations**, 2019.

[10] Lu Lin and Hongning Wang. Graph attention networks over edge content-based channels. In **Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**, pp. 1819–1827, 2020.

[11] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, **1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings**, 2013.

[12] Yang Zhang and Qiang Ma. Doccit2vec: Citation recommendation via embedding of content and structural contexts. **IEEE Access**, Vol. 8, pp. 115865–115875, 2020.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

[14] Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. Coke: Contextualized knowledge graph embedding. **arXiv preprint arXiv:1911.02168**, 2019.

[15] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, KDD '14, pp. 701–710, New York, NY, USA, 2014. ACM.

[16] Chanwoo Jeong, Sion Jang, Hyuna Shin, Eunjeong Park, and Sungchul Choi. A context-aware citation recommendation model with bert and graph convolutional networks, 2019.

[17] Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. Identification of Tasks, Datasets, Evaluation Metrics, and Numeric Scores for Scientific Leaderboards Construction. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5203–5213, Florence, Italy, July 2019. Association for Computational Linguistics.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

A　提案手法における入力埋め込み

本節では提案手法における入力埋め込みの作成方法をモデルに関する詳細として説明する。我々がベースとしている SciBERT の入力埋め込みはトークン埋め込み、位置埋め込み、セグメント埋め込みの合計によって構成されるが、提案手法の入力埋め込みはトークン埋め込み、位置埋め込み、トークンタイプ埋め込みの 3 つの埋め込みの合計によって構成される。トークン埋め込みは引用文脈内の単語埋め込み、もしくはネットワーク内の論文埋め込みに対応するものであり、入力の際にトークンの埋め込みの辞書として機能する。単語埋め込みに関しては SciBERT のそれを流用し、論文埋め込みに関してはランダムに初期化したものを用いた。位置埋め込みはトークンの位置を表すためのものであり、これも SciBERT のものを流用した。最後に、トークンタイプ埋め込みは単語埋め込みと論文埋め込みを区別するためのものでありそのトークンが単語の場合はトークンタイプ 0 に対応する埋め込みが、論文の場合はトークンタイプ 1 に対応する埋め込みが足し合わされる。この埋め込みもまたランダムに初期化したものを用いた。

B　論文分類における論文埋め込み

本節では実験における論文分類の詳細設定としてそれぞれのモデルに対する論文埋め込みの抽出方法を説明する。まず既存研究である hyperdoc2vec に関して説明すると、hyperdoc2vec には IN vector と OUT vector と呼ばれる 2 種類の論文埋め込みが存在し、今回は実験による検討の結果、それぞれの論文に対応する OUT vector を論文埋め込みとして用いた。また、ベースラインとして用いた RotatE は静的な論文埋め込みを訓練する手法であり、その埋め込みを用いた。最後に我々の提案手法における抽出方法を説明する。提案手法は SciBERT をベースとしており、例えば入力に用いるトークン埋め込みを論文埋め込みとして用いることは可能であるが今回は複数の手法を検討した結果以下の工程を用いて埋め込みを取得することとした。ここで埋め込みを抽出したい論文を論文 A とすると、まず我々は埋め込みの訓練に用いたデータから論文 A を被引用論文とするデータ (そのようなデータがない場合は論文 A が引用元論文であるデータ) を集める。そしてそれぞれのデータを訓練である Masked Paper Prediction と同様の変換方法を用いて入力系列としたのちに SciBERT に入力する。そして SciBERT から出力された埋め込みのうち入力系列内の論文 A に対応する位置の出力埋め込みの平均を全てのデータに対して取ることでこれを論文 A の埋め込みとしてみなした。

C　データセット間の比較

表 5 に今回我々が用いた 2 つの引用ネットワークの詳細が記載されている。まず AASC が ACL の論文のみで構成されているのに対し、FullTextPeerRead(以下 FTPR) は ACL だけでなく ICLR,NIPS の論文も含んでおり、機械学習全般の論文をターゲットとしている。ノード数、エッジ数からは FTPR の方が AASC よりサイズが小さいことがわかるが、連結成分の数からは FTPR がより分離されたネットワークであることがわかる。平均最短経路長 (今回は連結なノード間でのみ計測した) は AASC の方が FTPR より長く、さらにネットワークの凝集度を表すクラスタリング係数は AASC の方が高い。ここからは AASC が FTPR に比べてノード間の距離が近く、そしてネットワークにおける関係性の密度が高く凝集性が高いということが見て取れる。ネットワーク上の推論はノード A、B、そしてノード B、C の間に関係があるならノード A、C の間にも関係が成り立ちやすいという特性を活かしたものであり、AASC はこの特性が強いためネットワーク構造を用いた推論を行いやすいデータセットであると考えられる。

	FullTextPeerRead	AASC
会議の種類	ACL,ICLR,NIPS	ACL
ノード数	4,838	39,455
エッジ数	16,652	267,951
平均次数	5.23	13.55
最大次数	536	2,627
連結成分の数	276	186
平均最短経路長	4.89	4.25
クラスタリング係数	0.135	0.149

表 5　ネットワーク分析の指標によるデータセットの比較

D　謝辞

本研究の一部は JST CREST JPMJCR1513 の支援を受けたものである。