Co-Citation Prediction with Graph Networks and Transformers

Samihan Dani, Gaurav Sett
Georgia Insitute of Technology
{sdani30, gauravsett}@gatech.edu

Abstract

A vast number of academic papers are published each year. Especially in fast-paced disciplines like computer science, it is impossible for researchers to develop a comprehensive understanding of the landscape. Citation networks have become a dominant approach to understanding the relationships between papers. Co-citation, where two papers are cited together, has been used to track the evolution of research topics. However, this work has been largely descriptive. We employ a novel deep learning techniques to predict co-citations. Our model is composed of an encoder to represent the semantic information of each paper, a graph network to represent the citation network, and a regression head to predict the number of co-citations for a pair of papers. Our approach is unable to significantly outperform a baseline. Our model learns to take a safe approach, predicting an average number of co-citations for most pairs of papers. This approach appears robust to tuning and architecture modifications. Further work is needed to determine the viability of the approach and opportunities for improvement.

1. Introduction

Even with advances in search engines and recommendation systems, it is difficult for researchers to keep up with the vast amount of literature published each year. In 2018, 2.6 million papers were published around the world [18]. While there are varying estimates of how many are read or cited, it appears that a significant and growing portion receives little attention [4]. In fast paced disciplines like computer science, it is impossible for researchers to keep track of all new developments.

However, the increasing scale of academic literature has been met by a growing number of tools to help researchers parse through the literature. Prior to the internet, researchers were bottlenecked by physical access. Search engines like Google Scholar have also made it much easier to find papers related to key terms. Semantic search, used by platforms like Semantic Scholar, have also improved this process by using machine learning techniques to understand the meaning of a query and the content of a paper.

In addition to querying papers by topic, researchers also want to find papers that are related in a more nuanced way. Citation networks have become a dominant approach to understanding these relationships, inspired by search algorithms like PageRank [10]. Citation networks are directed graphs where nodes represent papers and edges represent citations. For decades, researchers have investigated co-citations, where two papers are cited together, to track the evolution of research topics [13].

Work on co-citation has predominately employed descriptive analysis. However, deep learning techniques give an opportunity to build predictive models. We build a model to predict the likelihood of two papers being co-cited by subsequent work. We use a large language model to encode the semantic information of each paper, a graph convolutional network to incorporate the citation information, and a regression head to predict the number of co-citations expected for a pair of papers.

This work can help researchers manage the depth and breath of academic literature. For example, the model can help identify interdisciplinary research directions, or help consolidate past and present work.

Our code is available at `https://github.com/gauravsett/co-citation-prediction`.

2. Related Work

The co-citation measure was first introduced in 1973 by Small, taking advantage of newly created scientific citation indices to identify research development and relationships across disciplines [13]. This work has been mostly descriptive, analyzing the structure of academic literature and the usefulness in clustering documents. For example, co-citation has been found to outperform direct citations in the identification of coherent research clusters in biomedicine, but fell short of text-based clustering measures in computer science literature [1]. The techniques used to identify co-citation have also evolved. Some studies focus on the co-citation of authors, rather than papers, to identify research communities, and others incorporate temporal modeling [2].

Some work has used co-citation networks for link prediction. Inspired by PageRank, researchers have used Markov models from these networks to recommended papers [21]. Other work has used graph-based growth algorithms to discover subnetworks with a trend of new publications and co-citations [14].

However, it appears no work has combined semantic information about paper content with structural information from the citation network to predict co-citation. This work aims to fill the gap using deep learning techniques.

Advances in natural language processing have largely been driven by the development of large language models. These models are trained on large corpora of text to learn the semantic information of language. The most popular of these models are based on transformers, which have been shown to outperform recurrent neural networks [16]. These models have been used for a variety of tasks, including question answering, text generation, and text classification [3]. These models are encoder-decoder architectures, where the encoder learns the semantic information of the input and the decoder generates the output. The encoder produces a vector representation of the input, which can be used as input to other models.

Many papers have investigated the use of large language models for knowledge graph completion. One approach, involves using a pre-trained language model to learn entity representations from natural language descriptions. These representations are then used to complete the knowledge graph by inferring missing links [17]. Overall, while pre-trained language models have been shown to capture factual knowledge from massive texts, they are still quite behind specialized state-of-the-art models in terms of performance [9].

The impact of transformers on graph ML has lagged their transformation of natural language processing and computer vision. There are many initiatives to represent graph structures in embeddings for input to transformer models, hoping to leverage their compute efficiency [7, 20]. However, we aim to leverage both the graph structure of the citation network and semantic information of the paper. The Cascaded Transformers-GNN applies language modeling before the graph modeling [6], while the GraphFormer combines these into an iterative process [19]. Because we want to prioritize the semantic information for our task, and because we want our embeddings to generalize beyond our paper subset, we privilege the language model in our approach.

Some work has leveraged large language models for link prediction. One framework called Bi-Link utilizes transformers for link prediction tasks [11]. Specifically, they used pre-trained language models to encode textual information about entities and relations in a knowledge graph. They then designed a contrastive learning approach with probabilistic syntax prompts to efficiently search for relational prompts according to learnt syntactical patterns that generalize to large knowledge graphs. This approach allows the model to comprehend the underlying semantics and logic patterns of relations, which is crucial for accurate link prediction.

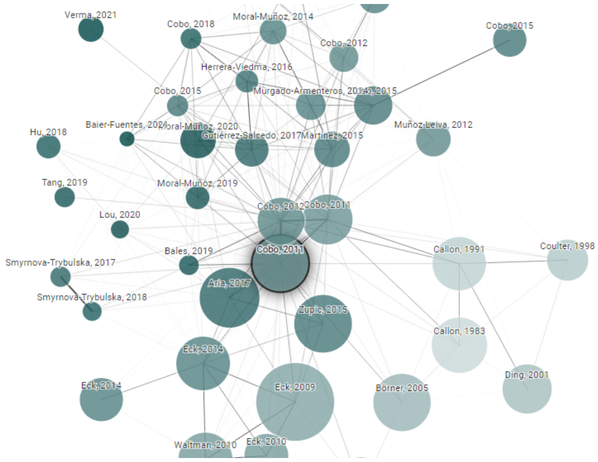

Figure 1. Example of citation graph

Feature	Description
id	Unique identifier
title	Published title of paper
authors	List of author names
venue	Publication venue (ex. conference, journal)
year	Year of publication
keywords	List of keywords
language	Language of paper
abstract	Abstract of paper
references	List of referenced paper identifiers

Table 1. Features of dataset

3. Data

We use the Citation Network Dataset as the basis for our work [15]. Specifically, we use the 14th edition of the DBLP citation network collected on January 1st, 2023. DBLP is a large computer science bibliography with 4.4 million publications from 2.2 million authors [8]. A diagram showed in Figure 1 illustrates how a citation graph should look like. Paper features can be found in Table 1. These papers date back to 1980.

We filter our dataset for our model. We select only English papers with complete information about abstracts, keywords, venues, and titles. From this set, we selected a random sample of 100,000 papers to reduce computational cost.

For each paper, we constructed a string combining the title, venue, keywords, and abstract to represent the content. This string is later used as input for our encoder model. Our graph model is based on a network where each paper is a node and each citation is an edge drawn from the citing paper to the cited paper. We use the references field to construct this network.

Some referenced papers from our dataset are not included in our sample. This is a result of the window of the dataset and our sample. In our graph network, these papers are represented by the average embedding of their neighbors, aiming to reduce the impact of missing data. For our regression model, we exclude these examples as we do not have the input data. However, the random sampling should prevent a selection effect.

4. Methods

Our approach aims to incorporate both semantic and structural information about papers to predict co-citation. Both intuitively offer important information about how a paper may contribute to literature. Semantic information can help identify topics and subfields. Structural information can represent the attention researchers have paid to different works and the relationships between them. Indeed, this deep learning based approach appears to be novel in the literature.

4.1. Architecture

Our model is composed of three components: an encoder, a graph network, and a regression head. The encoder is a language model that encodes the semantic information of each paper. The graph network incorporates the citation network. The regression head predicts the number of co-citations for a pair of papers. We describe each component in detail below.

For our training task, we selected 80% of co-citation pairs and estimate the number of co-citations for each pair. We use the remaining 20% of co-citation pairs for validation. We use a mean squared error loss function to train our model. We use the Adam optimizer with a learning rate of 0.00001, a weight decay of 0.001, train for 20 epochs. For our regression task, our batch size is 4096.

4.1.1 Transformer Encoder

We use the "MiniLM-L6-H384-uncased" language model from the Sentence Transformers library [12]. This model is based on the BERT architecture [3]. This model has been pre-trained on a large corpus of text and fine-tuned on a 1 billion sentence pairs dataset with a contrastive learning objective. The model outputs a 384-dimensional embedding for each paper.

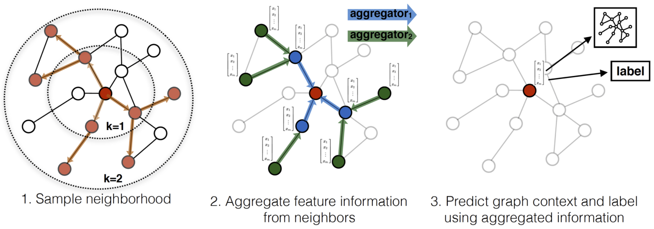
aggregator$_1$
aggregator$_2$
k=1
k=2
label
1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

Figure 2. Illustration of GraphSAGE Model

These embeddings are computed once prior to training and are used as input to the graph network. We do not fine-tune the language model during training.

4.1.2 Graph Convolutional Network

We use a graph convolutional network to incorporate the citation network. We use the PyTorch Geometric library [5] to implement this network. Specifically, we use the Sage-Conv layer, which is a graph convolutional layer based on the GraphSAGE algorithm [6]. This layer aggregates the embeddings of a node's neighbors and then concatenates this aggregation with it;s own previous embedding. Then, an update function is applied which performs a linear transformation on this vector. For these layers, we use 16 hidden channels, a dropout rate of 0.15, and a ReLU activation function. We apply this layer thrice in the graph network. The number of layers determined the size of the neighborhood for a node's aggregation. For instance, with our implementation, each node aggregates information from every node within a 3-hop radius of the node. This process can be seen in Figure 2 from [6].

This model produced convolved embeddings for each paper. We use these embeddings as input to the regression head.

4.1.3 Regression Head

Our regression head takes in a concatenated pair of convolved embeddings. The module is composed of a fully connected layer with a output dimension of 32, a ReLU activation function, and a final fully connected layer with a single output. This output is the predicted number of co-citations for a pair of papers.

5. Results

Our train loss can be seen in Figure 3. Our validation loss can be seen in Figure 4. We see that our validation loss plateaus after 10 epochs, suggesting that our model is not learning much after this point.

We evaluate our regression output using $RSME$ and R^2. We find that our model has an $RSME$ of 1.145 and an R^2

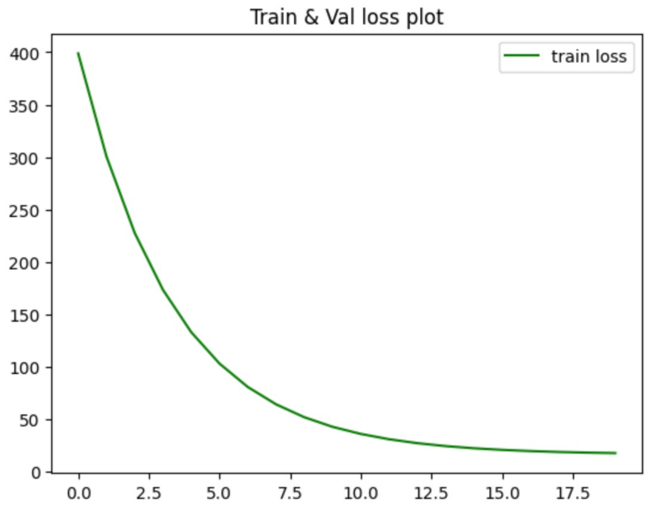

Figure 3. Training loss over epochs.

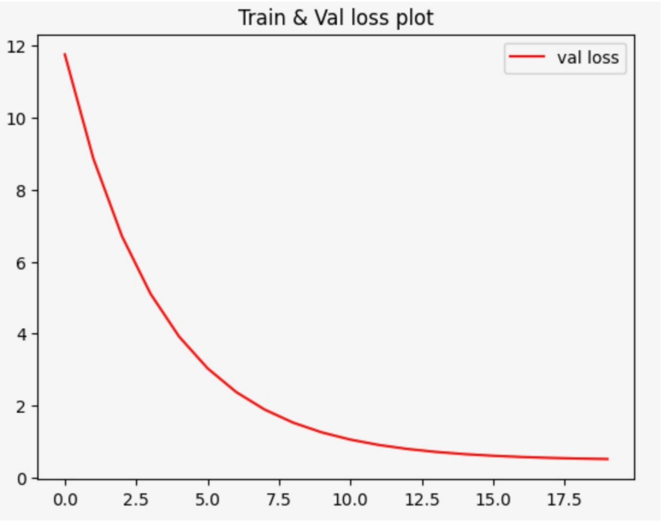

Figure 4. Validation loss over epochs.

of 0.002. This suggests that our model is not able to predict the number of co-citations well.

The decrease in our validation loss suggests our model is not overfitting. Upon investigating our model, we find that the model converging on a safe solution of predicting the average number of co-citations. These findings are robust to our tuning of optimizers, learning rates, batch sizes, loss functions, and hidden dimensions. We also experimented with different architectures for our regression head, changing the number of fully connected layers and the activation functions. For example, we tried SGD, SGD with momentum, and quantile loss.

When removing our encoder model, model performance decreases significantly. Node embeddings were initialized randomly and learnt through the graph network. These embeddings represent structural information about the paper's position in the citation network. This indicates the encoder is successfully capturing semantic information relevant to the task.

When removing our graph convolutions, model performance slightly decreases. Embeddings were passed straight from the transformer encoder to the regression head after concatenation. This indicates that the graph network is capturing structural information relevant to the task.

While this may lead us to believe the regression head is at issue, the model's failure may also be a combination of the encoder and graph network. Further analysis is needed to determine the exact cause of the model's failure.

6. Discussion

This paper employs a deep learning approach to predict co-citation. We start with a transformer encoder to capture the semantic information of each paper. We then use a graph convolutional network to incorporate the citation network. Finally, we use a regression head to predict the number of co-citations for a pair of papers.

We find that our model is not able to predict the number of co-citations well. This may be due to the complexity of the task. Co-citation is a complex phenomenon that is not well understood. Additionally, the number of co-citations is a noisy measure of the relationship between papers. For example, a paper may be cited for a variety of reasons, including criticism. We were also limited to paper abstracts which may not contain enough information for the task. For empirical papers, the methods and results sections may be more informative. These findings are difficult to represent in an abstract. Additionally, the technical matter such as equations and model architectures may be difficult to represent in text.

One additional problem may be that the dataset did not include embeddings for all of the papers referenced. Since we gave these embeddings, the average of all of the existing paper embeddings, it could be that they drowned out the notable papers during the aggregation in the convolutional layer. This explanation aligns with our results since the model seems to predict similar values regardless of the pairs shown.

Our work should be understood in the context of its limitations. First, our dataset is predominately computer science papers. Amongst those, we have filtered for English language papers only. Additionally, language model capabilities are known to change dramatically at different scales, so our findings may not generalize to other pre-trained encoders.

Future work may investigate different data, pre-trained encoders, and graph network architectures. Additionally, future work may investigate different tasks, such as classification or link prediction.

Team Contributions

Samihan Dani contributed to the model implementation, training, and evaluation. He also contributed to the writing of the literature review, methods, and results.

Gaurav Sett contributed to the collection and processing of data and the setup of model pipeline. He also contributed to the writing of the introduction, literature review, data section, and discussion.

References

[1] Kevin W. Boyack and Richard Klavans. Co-citation analysis, bibliographic coupling, and direct citation: Which citation approach represents the research front most accurately? *J. Assoc. Inf. Sci. Technol.*, 61:2389–2404, 2010. 1

[2] Chaomei Chen, Fidelia Ibekwe-SanJuan, and Jianhua Hou. The structure and dynamics of cocitation clusters: A multiple-perspective cocitation analysis. *Journal of the American Society for information Science and Technology*, 61(7):1386–1409, 2010. 2

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[4] James A Evans. Electronic publication and the narrowing of science and scholarship. *science*, 321(5887):395–399, 2008. 1

[5] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. 3

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 2, 3

[7] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35:14582–14595, 2022. 2

[8] Michael Ley. The dblp computer science bibliography: Evolution, research issues, perspectives. In *String Processing and Information Retrieval: 9th International Symposium, SPIRE 2002 Lisbon, Portugal, September 11–13, 2002 Proceedings 9*, pages 1–10. Springer, 2002. 2

[9] Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. In *Findings*, 2022. 2

[10] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. In *The Web Conference*, 1999. 1

[11] Bohua Peng, Shihao Liang, and Mobarakol Islam. Bilink: Bridging inductive link predictions from text via contrastive learning of transformers and prompts. *arXiv preprint arXiv:2210.14463*, 2022. 2

[12] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 3

[13] Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for information Science*, 24(4):265–269, 1973. 1

[14] Vladimir Smojver, Mario Štorga, and Goran Zovak. Exploring knowledge flow within a technology domain by conducting a dynamic analysis of a patent co-citation network. *Journal of Knowledge Management*, 25(2):433–453, 2021. 2

[15] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008. 2

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[17] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *ArXiv*, abs/2203.02167, 2022. 2

[18] Karen White. Publications output: Us trends and international comparisons. science & engineering indicators 2020. nsb-2020-6. *National Science Foundation*, 2019. 1

[19] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810, 2021. 2

[20] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021. 2

[21] Jianhan Zhu, Jun Hong, and John G Hughes. Using markov models for web site link prediction. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, pages 169–170, 2002. 2