Directed Criteria Citation Recommendation and Ranking Through Link Prediction

William Watson
S&P Global
william.watson@spglobal.com

Lawrence Yong
S&P Global
lawrence.yong@spglobal.com

ABSTRACT

We explore link prediction as a proxy for automatically surfacing documents from existing literature that might be topically or contextually relevant to a new document. Our model uses transformer-based graph embeddings to encode the meaning of each document, presented as a node within a citation network. We show that the semantic representations that our model generates can outperform other content-based methods in recommendation and ranking tasks. This provides a holistic approach to exploring citation graphs in domains where it is critical that these documents properly cite each other, so as to minimize the possibility of any inconsistencies.

CCS CONCEPTS

• Computing methodologies → Neural networks; **• Information systems → Retrieval models and ranking**.

KEYWORDS

graph neural networks, citation recommendation

1 INTRODUCTION

Deep learning has proven successful in creating high dimensional representations of data that can be leveraged to learn complex tasks. For instance, convolutions form the backbone for many image-based tasks, and text-based data has relied upon recurrent networks and attention mechanisms to produce high quality results for natural language processing and machine translation. Recently, the inception of transformer-based models by [11] has lead to a paradigm shift that is currently driving new state-of-the-art performance in many tasks, including graph-based analysis. This paper examines a graph-based approach using transformers to build document embeddings from a large citation network. We show that the resulting embeddings can be used to recommend citations for new documents, outperforming baselines on both precision and recall. In addition, we show that traditional graph tasks such as link prediction can be used as proxies for other important applications, such as search recommendation and recovery tasks.

Our approach operates on a corpus of documents known as *methodology* or *criteria*, maintained by a credit rating agency (CRA). These institutions cover thousands of entities globally. Each entity is rated according to a strict analytical framework that includes methodological criteria that guides every step of analysis. It is critical to keep this citation graph up-to-date, as any inconsistency can pose a risk to the accuracy of the ratings process. Hence, there is immense value in organizing, recommending, and ranking relevant criteria based on their joint semantic and graphical representations to resolve hidden and implicit citations.

2 RELEVANT BACKGROUND

Deep learning has proven to be versatile to a number of diverse tasks and datasets, and several attempts have been made to extend current architectures and frameworks to deal with data structured as graphs. Early attempts with graph data involved applying modified recurrent neural networks on a target node embeddings, propagating node states until an equilibrium is reached [3, 10]. Many authors have investigated spectral representations of graphs, simplifying earlier methods by restricting the scope of the filters with respect to the node's neighborhood [6]. Non-spectral approaches apply convolutions directly on groups of spatially close neighbors [4]. However, attention-based mechanisms introduced by [11] have been applied to graph problems by [12] and shown to outperform previous methods on citation datasets.

3 DATA: CRA CRITERIA CORPUS

Our dataset contains 2,247 criteria documents, publicly accessible via the CRA's website[1]. Citations can be expressed as inline hyperlinks or mentions of titles of other criteria documents. For the latter case, we used string matching to resolve explicit mentions. This method surfaced 13,959 directed citations within our dataset, an average rate of 6.2 citations per document. Our set originally featured 10,428 lemmatized nouns (with stop words removed). We reduced the set to the 300 most frequent words, and calculated TF-IDF vectors for each word-document pair. The vectors are normalized with mean 0 and standard deviation 1.

4 PROBLEM STATEMENT

Our approach employs link prediction as a proxy for citation recommendation and ranking. During training, a subset of nodes (i.e. criteria) are used to teach the model how to recover their missing links (i.e. citations). Linkages from validation nodes are masked, and our model *reconstructs* them. As each predicted link is associated with a probability, these *confidence scores* can be used as a ranking heuristic to order relevant citations. Key performance metrics are measured by the coverage and accuracy of the recovered missing linkages.

ICAIF '20, October 15–16, 2020, New York, NY, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://www.standardandpoors.com/en_US/web/guest/ratings/ratings-criteria

Figure 1: Augmented Transformer with Learned Residual.

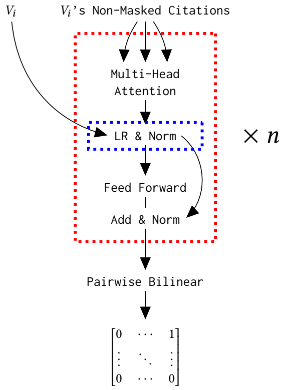
V_i
V_i's Non-Masked Citations
Multi-Head Attention
LR & Norm
× n
Feed Forward
Add & Norm
Pairwise Bilinear
$\begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$

Table 1: Ablation: Number of Hops

	Learned Residual						Recovery		
n hops	**L1**	**L2**	**L3**	**L4**	**L5**	**% Original**	**MAP@k**	**MAR@k**	**No. Params**
0-hop	-	-	-	-	-	100.0	0.214	0.545	23,360
1-hop	10.5	-	-	-	-	89.5	0.253	0.644	56,897
2-hop	12.8	6.4	-	-	-	81.6	**0.254**	**0.653**	90,434
3-hop	15.5	5.5	6.9	-	-	74.3	0.218	0.523	123,971
4-hop	16.2	7.7	6.0	6.7	-	67.8	0.181	0.398	157,508
5-hop	17.2	11.4	6.2	7.1	9.9	57.6	0.170	0.342	191,045

Table 2: Ablation: Effect of Different Components

	Similarity		Recovery		
Model (Embedding Size)	**MSE**	**Cosine**	**MAP@k**	**MAR@k**	**No. Params**
TF-IDF (300)	22.39	0.18	0.105	0.346	-
GT (64)	2.54	0.41	0.186	0.490	69,696
GT-LR (64)	2.55	0.47	0.198	0.556	86,338
Pairwise Bilinear (64)	7.28	0.71	0.214	0.545	23,360
GT + Bilinear (64)	2.45	0.74	0.219	0.548	73,792
GT-LR + Bilinear (64)	2.83	0.71	**0.254**	**0.653**	90,434

Table 3: Comparison for Different Citation Thresholds

	Citation Threshold						
Metric	**25%**	**50%**	**75%**	**90%**	**95%**	**99%**	**99.9%**
Total Citations	250,301	176,356	122,171	83,329	63,758	33,138	11,177
% Recommended	5.0	3.5	2.4	1.7	1.3	0.7	0.2
Within-Domain %	55.3	57.5	59.3	60.8	61.7	62.6	62.9
Out-of-Domain %	44.7	42.5	40.7	39.2	38.3	37.4	37.1
KL Divergence	1.540	1.147	0.817	0.534	0.392	0.186	0.340
Total EMD	0.405	0.363	0.321	0.253	0.212	0.134	0.130

5 RANKING EVALUATION

We measure embedding similarity and ranking recovery rates for the TF-IDF input, different model configurations, and the final logit predictions on the validation set. We also conduct several ablation studies as a reference. We rank candidates by their probability of citation for a given target criteria document through a pairwise dot product of their embeddings. We report the MAP and MAR scores for the top 20 results. Similarity scores are provided to display how *close* (MSE) and how *well oriented* (cosine similarity) our citation embeddings are to the target node's embedding.

6 METHODOLOGY & ABLATION

Training: Experiment Details. We split our nodes into a train and validation set to only include nodes with citations. The train set contains 1,472 nodes (65.5%), the validation set has 260 nodes (11.6%). Therefore 22.9% of nodes do not have outward citations, but may be cited. Following the transductive setup of [14], the training algorithm has access to all the node features. To prevent data leakage from our validation set, we void all of their outgoing connections in the original adjacency matrix, and we only predict on the training set. Therefore, we never correct errors in the validation set. Given the imbalance that is natural for link prediction tasks in sparse matrices, we employ negative sampling on a random subset of nodes known to not link together [9]. We use the Adam optimizer [5] with $\alpha = 0.001$, and trained for 1,920 updates. The model with the best validation recall score at $k = 20$ is saved.

Graph Attention Network. The model's core components are modular attention layers that follow the full transformer encoder structure [11], but inspired by the graph attention layers presented in [12]. We use a single linear embedding layer to compress the top 300 normalized TF-IDF features to a dense, 64-dimensional vector as our initial node embedding. Each graph layer (GT) operates on the updated node embedding, incorporating all direct neighbors which have been updated with information from their direct citations, and so on. We stack 2 graph layers to allow for each node to encapsulate the 2-hop sub-graph surrounding it. We apply a dropout layer with $p = 0.15$ on the adjacency matrix to simulate missing links, forcing the graph to attend on incomplete information. We use 8 heads for the multiheaded attention, ELU activation [2], and a feedforward dropout of $p = 0.1$. We conducted an ablation study in Table 1 that revealed the optimal number of hops as 2. Our embeddings became too noisy after this, as a 5-hop network can traverse 66.2% of the total paths, compared to 5.0% for the 2-hop network.

Learned Residual. We apply a learned attention mechanism on the first residual connection in the transformer, to control the influence of a node's neighborhood on the current node embedding (GT-LR) [1, 7]. This allows the model to control the influence of the graph structure on a node's embedding. The *additive* scoring function contextualizes local neighbors and their importance as a citation. The best models used only 20% of the network, suggesting that graph information is critical, albeit not as influential as the node's own embedding. The max residual weight observed for the graph structure for the *additive* attention was 89.4% for the first layer, and 83.2% for the second.

$$\begin{aligned} z_a &= \sigma\Big(v_a^T \tanh\big(W_a o_t + U_a n_t + b_a\big)\Big) \\ r_t &= z_a \odot o_t + (1 - z_a) \odot n_t \end{aligned} \tag{1}$$

Bilinear Scoring. To generate non-symmetric predictions, the final layer is the bilinear scorer $f(e_i, e_j) = e_i^T W_b e_j$; a pairwise dot product would produce a symmetric, undirected citation matrix for our asymmetric, directed matrix [13]. An ablation study revealed the GT-LR + Bilinear model saw an improvement of 17.5% in recall and 28.3% in precision over the pairwise scorer (GT-LR).

7 DISCUSSION

7.1 Link Prediction as a Proxy for Citation Recommendation

We approached citation recommendation as a link prediction task, where our model attempts to reconstruct the true citations from our masked matrix. Our baseline was the 300-dimensional TF-IDF

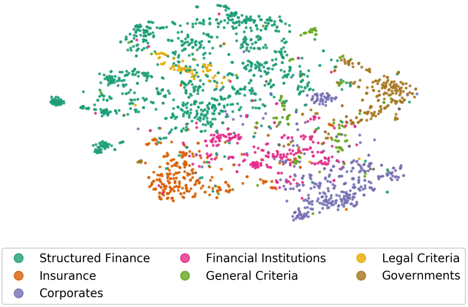

Figure 2: Embeddings, Colored by Subject Domain

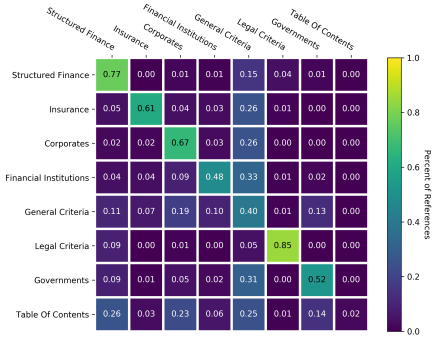

Figure 3: True Cross-Reference Matrix Organized by Subject Domain

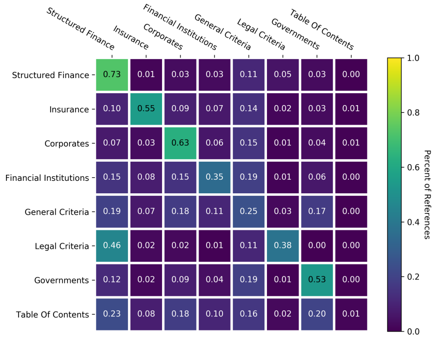

Figure 4: Predicted Cross-Reference Matrix Organized by Subject Domain, Threshold set at 50%

vectors that describe the content of the document. The idea was simple: documents that cite each other most likely share the same set of keywords and citations. The baseline performance for our validation set was only 10.5% MAP and 34.6% MAR for $k = 20$. However, using our methodology, where each link is a *citation*, we see that our results are maximized at 25.4% MAP and 65.3% MAR. Table 2 highlights the effectiveness of using graph transformers for link prediction.

7.2 Cross-pollination of Domain Areas

The model self-organizes embeddings to have the same orientation as its citations for a positive prediction. Non-cited documents orient in opposing directions, to create a negative prediction. The utility of this approach lies in what the model recommends along side the ground truth. An analysis of our citation matrix from Figure 3 shows that some domain areas are highly self contained, but others tend to cross-pollinate with other areas. We know of 13,959 citations, and by setting the prediction threshold at 50%, the model recommends 176,356 citations, out of a possible 5,049,009 (3.5%). Our predicted recommendations in Figure 4 are 57.5% within-domain, 42.5% out-of-domain. In comparison, the true citation matrix is 62.6% within-domain, 37.4% out-of-domain. We provide several metrics to compare citation threshold levels in Table 3.

7.3 Quality of Embeddings

A qualitative representation of the embeddings is generated by t-SNE projections [8]. From Figure 2, we see that domains self-organize into their respective clusters, indicating a strong desire to cite within-domain over out-of-domain articles. Legal criteria tends to cluster along a single axis, due to it's strong within-domain citation preference at 85%. General Criteria, as the domain with the most diffuse cross-references, is scattered about the manifold without a strong cluster.

8 CONCLUSION AND FUTURE WORK

In this study, we presented the utility for using link prediction as a proxy for a citation recommendation engine. Here, we can organize a citation network of criteria documents and generate relevant citations to new documents. We hope to expand our approach to other domains such as link prediction in business relationships and supply chain networks.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:cs.CL/1409.0473

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv:cs.LG/1511.07289

[3] M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 2. 729–734 vol. 2.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. arXiv:cs.SI/1706.02216

[5] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

[6] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:cs.LG/1609.02907

[7] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. arXiv:cs.CL/1508.04025

[8] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. arXiv:cs.CL/1310.4546

[10] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:cs.CL/1706.03762

[12] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. arXiv:stat.ML/1710.10903

[13] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. arXiv:cs.CL/1412.6575

[14] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. arXiv:cs.LG/1603.08861