Longitudinal Citation Prediction using Temporal Graph Neural Networks

Andreas Nugaard Holm[1], Barbara Plank[2], Dustin Wright[1] and Isabelle Augenstein[1]

[1]University of Copenhagen, Universitetsparken 1, 2100 København, Denmark
[2]IT University of Copenhagen, Rued Langgaards Vej 7, 2300 København, Denmark

Abstract

Citation count prediction is the task of predicting the number of citations a paper has gained after a period of time. Prior work viewed this as a static prediction task. As papers and their citations evolve over time, considering the dynamics of the number of citations over time seems the logical next step. Here, we introduce the task of sequence citation prediction. The goal is to accurately predict the trajectory of the number of citations a scholarly work receives over time. We propose to view papers as a structured network of citations, allowing us to use topological information as a learning signal. Additionally, we learn how this dynamic citation network changes over time and the impact of paper meta-data such as authors, venues and abstracts. To approach the new task, we derive a dynamic citation network from Semantic Scholar spanning over 42 years. We present a model which exploits topological and temporal information using graph convolution networks paired with sequence prediction, and compare it against multiple baselines, testing the importance of topological and temporal information and analyzing model performance. Our experiments show that leveraging both the temporal and topological information greatly increases the performance of predicting citation counts over time.

Keywords

citation count prediction, graph neural network, citation network, dynamic graph generation

1. Introduction

The problem of predicting citation counts of papers has been a long-standing research problem. Predicting citation counts allows us to better understand the relationship between a paper and its impact. However, prior research has viewed this as a static prediction problem, i.e. only predicting a single citation count at a static point in time. This ignores the natural development of the data as new papers are being published. Here, we propose to view the problem as a sequence prediction task, with models then having the ability to capture the evolving nature of citations.

This, in turn, requires a dataset to contain the papers' citation counts over a period of time, which adds a temporal element to the data, which can then be encoded by sequential machine learning models, such as Long short-term memory models (LSTM) [1]. Additionally, scholarly documents exhibit a natural graph-like structure in their citation networks. Given recent developments in modeling such data [2, 3] and prior research showing that modeling input as graphs can be beneficial, we hypothesize that modeling a paper's citation network is useful for predicting citation counts over time. In this paper, we consider citation networks, a dynamic graph which evolves over time as new citations and papers are added to the network. Leveraging the structured data in the graph allows us to discover complex relationships between papers. We want to tap into that knowledge and treat the citation data as a network, such that we can further exploit topological information and not just temporal information. By doing so, we investigate the hypothesis of paper citation counts being correlated with features such as authors, venue, and topics.

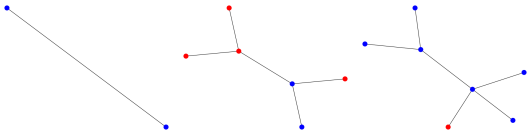

Figure 1: Illustration of the development of the dynamic graph through three time steps. Each node represents a paper; edges are citations between papers. Red nodes represent new papers in the current time step.

We use the well-established Semantic Scholar dataset [4] to construct our citation network. Its meta-data allows us to construct a dynamic citation network which covers a 42 year time-line, with an updated graph for each year. The Semantic Scholar dataset's meta-data also contains information about each paper's authors, venue, and topics, allowing us to study the correlation between these features and the citation count of a paper when considering the evolving

SDU@AAAI'22: Workshop on Scientific Document Understanding, March 01, 2022
aholm@di.ku.dk (A. N. Holm); bapl@itu.dk (B. Plank); dw@di.ku.dk (D. Wright); augenstein@di.ku.dk (I. Augenstein)
0000-0002-2006-5894 (A. N. Holm); 0000-0002-4394-1965 (B. Plank); 0000-0001-6514-8733 (D. Wright); 0000-0003-1562-7909 (I. Augenstein)

nature of the citation network. The correlation between these features and citation counts is well-known and studied by prior work [5]. Prior studies show that citations are correlated and there is a strong correlation between features such as authors, but are limited by only predicting a single citation, and not predicting the natural evolution of a papers growth.

We propose to use the constructed dynamic citation network (see Section 4.2) to predict the trajectory of the number of citations papers will receive over time, a new sequence prediction task introduced in this work. Furthermore, we propose an encoder-decoder model to solve the proposed task, which uses graph convolutional layers [6] to exploit the graphs' topological features and an LSTM to model the temporal component of the graphs. We compare our model against a vanilla graph convolutional neural network (GCN) and a vanilla LSTM, which individually incorporate either the topological information or the temporal information, but not both.

Our contributions are as follows: 1) A dynamic citation network based on the Semantic Scholar dataset. The dynamic citation network contains 42 time-steps, with an updated graph at each time-step, based on yearly information. 2) We introduce the task of sequence citation count prediction. 3) A novel encoder-decoder model based on a GCN and LSTM to extract the dynamic graph's topological and temporal components. 4) A thorough study of the correlation between citation counts and temporal components.

2. Related Work

The task of predicting a paper's citations aims to predict the number of citations which a paper has obtained either by a given year or after n years. The task itself is not new and has been researched throughout the years, and multiple different approaches have been tried and shown to be effective. Some of these studies, have focused on feature vectors [5, 7] and explored distinct feature vectors' performance, where they primarily rely on meta-data, e.g. venue and authors. As peer review data has become available [8], recent research has focused on using non-meta-data information, such as peer-reviews [9, 10] to predict a paper's citation count.

What is common in existing research is the target: predicting a single citation count. This count can be set as one of the following years, or the citation count n years in the future. To predict these citation counts, we see a variety of different neural network models with distinct architectures [10, 11], as well as papers which focus on deeper feature vector analysis, where regression models are used [12, 7]. A side effect from prior research's focus on predicting single citation counts is that the utilized citation networks are static graphs, based on paper databases such as ArnetMiner [13], Arxiv HEP-TH [14] and CiteSeerX [15]. These static citation networks are not suitable for our proposed task because they only contain the topological information at a single point in time. As longitudinal citation datasets are rare, we derive a dataset from Semantic Scholar.

Citation networks are not exclusively used for citation count prediction. Other citation networks such as Cora [16], CiteSeer [17] or PubMed [16], all well known benchmark graphs, are used for node classification tasks, where the task is to predict a paper's topic. These networks are provided with minimal content. They consist of an adjacency matrix, the connections between citations, and a simple feature vector for each node of either 0/1-valued vector or a tf-idf vector, based on the dictionary of the paper content. These existing datasets do not fit our purpose, hence we derive our own, described in Sec. 4.2.

3. Temporal Graph Neural Network

Our model is an encoder-decoder model and therefore consists of two major components. The first component is the encoder, which takes an adjacency matrix of node connections and a node feature matrix as input, where the node feature matrix can e.g. consist of author information (illustrated in Figure 2). It uses the topological information from the graphs and creates feature vectors containing both the topological node features via a GCN. It should be noted that due to the use of dynamic graphs, the encoder generates a sequence of graph embeddings, one for each graph in the sequence. The second component, the decoder, utilizes the sequence of graph embeddings created by the encoder. By using an LSTM, we extract the temporal elements and create a sequence of citation count predictions (CCP) for each node in the dynamic graph.

3.1. Problem Definition

While the task of CCP has been researched before, in this paper, we are interested in predicting a sequence of citation counts, which to our knowledge is so far unexplored.

Let us start by introducing our graph notation. We denote our dynamic graph as $G = \{G_0 \dots G_{T-1}\}$, where G_t is a graph, at the given time t. Each graph in the dynamic graph set is defined as $G_t = (V_t, E_t)$, where V_t is the set of vertices at time t and E_t is the set of edges at time t. With a given dynamic graph, we aim to predict the sequence of citations for a given paper. We formalize this as $y^v = \{y_1^v \dots y_T^v\}$, where y_t^v is the number of citations for $v_t \in V_t$ and $y_t^v = |E_t^v|$. For our proposed task, we are given

the dynamic graph G, and are to predict the sequence of citation counts y.

3.2. Topological Feature Extraction

One of the central hypotheses we want to examine is if complex structural dependencies in a citation network can help predict the citation count of a paper. To test this, we employ a GCN to extract topological dependencies from the graphs. We choose a GCN over other methods as they work in Euclidean space, and are thus easy to use with other neural architectures such as convolutional neural networks (CNN) [3].

The GCN uses the data flow between edges in the graph to create a graph embedding. As such, we can create an embedding influenced by all of the neighboring nodes in the graph. In this, we hypothesize that there is a relationship between the number of citations a given paper receives and that of its neighbors. The connections between the papers is described by an adjacency matrix A. Using our notation, we describe the GCN as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \tag{1}$$

where $\tilde{A} = A + I$; I is the identity matrix (which enables self-loops in $\tilde{A}$); $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, l is the l'th layer in the model; σ is an activation function; and $H^{(l+1)}$ is the output of the GCN layer $H^{(l)}$. We can then simplify the above equation:

$$H^{(l+1)} = \sigma\left(\hat{A}_t H_t^{(l)} W^{(l)}\right) \tag{2}$$

where $\hat{A}$ is defined as $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ and t is the time step in the dynamic graph. It should be noted that t has been left out in the first equation for simplicity. We also observe here that by adding multiple GCN layers, we allow the the graph embeddings to be affected by extended neighbours.

Since we work on a dynamic citation network, we have T distinct adjacency matrices, and we have to create a graph embedding for each graph in the sequence:

$$Z = \{Z_0 \dots Z_T\} = \{f(X, A_t) \mid A_t \in A\}, \tag{3}$$

where the function f is the GCN network, $Z_t \in \mathbb{R}^{m \times n}$ is a single graph embedding of dimensionality n with m nodes, and Z is the set of graph embeddings created by the GCN. It should be noted that X is shown as being independent of time, which is true for some of our node embeddings. However, some of our node embeddings are based on citations, which change through time, which makes X dependent on time. We will explore the distinct node embeddings in a later section. As shown in the equation, we also keep the same model over time, and do not change the GCN even though the graph changes. We instead try to generalize the model, working on all the graphs in the dynamic graph.

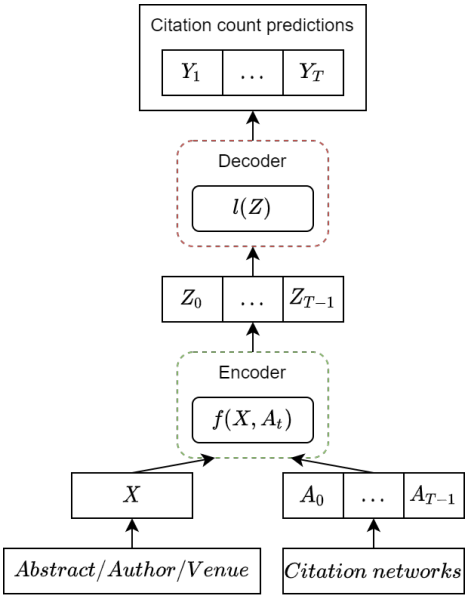

Figure 2: Our proposed encoder-decoder model

3.2.1. Temporal Feature Extraction

With the constructed graph embeddings, containing both topological information and node information. We want to extract the temporal information, which we use the sequence of graph embeddings to do. To extract the temporal information, we utilize an LSTM, where we can formalize the input and output as $Y = l(Z)$, where the function l is the LSTM and $Y \in \mathbb{R}^{m*T}$ are the CCPs.

3.2.2. Encoder-Decoder

In the final model, we combine the GCN and LSTM in an encoder-decoder model. The primary challenge in combining these two models though is that they operate on vastly different inputs. The GCN operates on entire graphs and needs all the nodes to appear in the graphs, including nodes which it intends to predict. The LSTM, however, does not have this requirement and can work on batches. To solve this issue in a simple yet effective approach, we embed the entire graph prior to the LSTM steps so that in the LSTM step, we can still split the data into batches for training, validation and testing. While other approaches have been researched, like embedding the GCN into the LSTM [18], we found the simple approach to perform better.

Figure 2 shows the architecture of our model. The GCN uses two layers to create the graph embedding. The LSTM is a single one-directional layer whose outputs are reduced to a sequence of scalars through a linear layer.

4. Dynamic Citation Count Prediction

As discussed earlier, we differentiate ourselves from prior work by predicting a sequence of citation counts over time as opposed to a single final citation count. Datasets for the latter exist, but are based on paper databases. However, existing citation networks are not usable for our task due to the graph of the citation network being static in those works, i.e., the citation network does not evolve over time. Given this, we construct a dataset, where we reconstruct the citation networks, at each time-step, for the purpose of studying citation count prediction over time.

4.1. Dataset

The dataset which we used to create our dynamic graph is based on Semantic Scholar [4].[1] The dataset is a collection of close to 200,000,000 scientific papers; the size of a graph of this size requires an immense system to run experiments on (recall the size of $Y \in \mathbb{R}^{m*T}$ where m is the number of papers). To reduce the dataset to a manageable size, we only kept papers from the following venues related to AI, Machine Learning and Natural Language Processing: *ACL*, *COLING*, *NAACL*, *EMNLP*, *AAAI*, *NeurIPS* and *CoNLL*. With the dataset only containing papers from the listed venues, we reduced the dataset's size to 47,091 papers. Furthermore, the Semantic Scholar dataset also holds an extensive collection of meta-data for each paper. We use this meta-data to construct our dynamic graph, as well as the graph's node embeddings.

4.2. Graph Construction

With the dataset reduced to a more manageable size, we search for an ideal dynamic graph of the citation network. We do this because working with graphs can be computationally heavy and the size of the graph based on the full semantic scholar dataset, can make some computations near unfeasible. We define an ideal dynamic graph as the sequence of graphs which has the largest connected graph in the final graph and has the most significant increase of nodes over time. We do not use the largest connected graph at each time step, as it can trick us into selecting a sub-optimal dynamic graph. A sub-optimal dynamic graph may present itself as the largest connected graph at a point in time, but will not stay as the largest connected graph through time, and will contain less nodes through time, compared to the ideal dynamic graph. To solve the issue of being tricked into selecting a less ideal dynamic graph, we have to probe each node in the data to observe the graphs' evolution. We define probing as the process of observing the evolution of the graph connected to the probed node. This process is automatically performed on all nodes of the largest connected graph in the final step. By probing all the nodes, we can choose the sequence of graphs which contains the most nodes over time. In Algorithm 1, we describe the process in the form of pseudo-code for a more precise insight in the process of constructing the ideal dynamic graph.

Algorithm 1: Dynamic Graph Construction

```
Input: data
Output: G
connected_graphs = dict()
for y ∈ years do
    gs ← find_connected_graphs(data[y])
    connected_graphs[y] ← sort(gs)
end
for paper ∈ data[min](years) do
    key_size[paper] = 0
    for y ∈ years do
        best = 0
        for g ∈ connected_graphs[y] do
            if paper ∈ g and |g| > best then
                best = |g|
            end
        end
        key_size[paper]+ = best
    end
end
best_paper = argmax(key_size)
G = dict()
for y ∈ years do
    for g ∈ connected_graphs[y] do
        if best_paper ∈ g then
            G[y] = g
            break
        end
    end
end
```

In Table 1, we show some of the properties of the last 10 graphs in the dynamic graph. It is clear how the graph is evolving over time, as can be seen in how both the number of vertices and edges increases, and how the degree D increases, indicating that the nodes in the graph obtains more citations over time. This indicates that the dynamic graph reflects the natural growth of a paper's citations.

By only using a subset of the nodes from the full graph to construct the dynamic graph, we ablate some of the full graph's properties. One notable property of the full graph is that the citation count of a paper is tied to the degree of a node; by using a subset of the full graph this property does not hold anymore, which leads to the following definition of the size of the set of edges $y_t^v = |E_t^v|$

[1] https://api.semanticscholar.org/

	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
$\|V\|$	14,584	16,603	18,529	20,760	23,327	26,529	29,293	33,759	38,080	38,168
$\|E\|$	103,519	127,277	152,869	181,666	217,807	267,940	308,186	387,738	475,007	476,015
Mean D	7.1	7.67	8.25	8.75	9.34	10.1	10.52	11.49	12.47	12.47
Max D	614	761	923	1,072	1,220	1,371	1,496	1,763	2,084	2,086
Max citation count	2,584	3,110	3,637	4,186	4,740	5,403	11,385	20,893	32,278	35,200
Avg. citation count	26.33	27.48	28.87	30.15	31.49	32.94	35.84	38.31	43.0	45.41

Table 1
Key values of the graphs.

changing to the following for a given node $y_t^v \geq |E_t^v|$. Another important point is that removing edges from the graph removes some of the information contained in the full graph (e.g. links to papers in other fields). Such edges are usually connected to more prominent papers because it is often the high impact papers, which obtain citations from papers outside the main field.

4.3. Feature Generation

The created dynamic graph nodes are not dependent on a set of specific features, and we can therefore select and create a set of features for each node containing our desired information. With a wide variety of meta-data fields available, we created a set of distinct features which we used for our predictions. Furthermore, we studied how each of these features affect the performance of the model.

The choice of using authors and venues as features for our model is based on the hypothesis that authors listed on a paper have a major impact on the number of citations gained. We assume the same goes for venues: if a paper is published at a more highly ranked venue, it is more likely to gain a large amount of citations compared to a paper published at a lower ranking venue. We further motivate the choice of these two features based on prior work [5], who shows that author rank and venue rank are indeed two of the three features that are most predictive. We motivate the choice of using the abstract based on the assumption that the abstract of a paper contains information on the topics discussed in the paper, which can be used to identify if paper's topic is currently popular [19]. We further motivate the choice of using author and venue rank, as prior work shows them to be the most descriptive features [5]. The following sections provide short descriptions of the meta-data used to create these feature vectors and how each of them is calculated.
Abstract: To base our model on more than meta-data, we use the abstract of the papers to create a feature vector. To create an embedding of the abstract, we utilize BERT [20], specifically the pre-trained SciBERT [21] model. SciBERT is a contextualized embedding model trained using a masked language modeling objective on a large amount of scholarly literature. Representations from SciBERT have been shown to be useful for learning downstream tasks with scientific text, this is why we use them here. To obtain a feature vector of a given abstract, we tokenize the abstract text and pass this through SciBERT. SciBERT prepends a special `[CLS]` token for performing classification tasks, so we use the output representation of this token as the final feature vector for an abstract.
Author rank: To include the author information, we created a feature vector which ranks the authors based on their number of citations sorted by highest to lowest. Due to many authors having the same amount of citations, we allow authors to be of the same rank. As the final step for the feature calculation, we normalize the rankings by $X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$.
Venue rank: Together with the author rank, we also hypothesize that the venue has an impact on the number of citations of a paper. Therefore, we also created a feature ranking for the venues. The feature is calculated identically to the author rank. It should be mentioned that the meta-data contains a high amount of different labels for each of the venues which we are using. We reduce all the different labels of the same venue down to a single label for each venue, but keep each venue separated by year.

5. Experiments

In this section we present our experiments and results, and explore the importance of exploiting topological and temporal information.

5.1. Data

We use the constructed dynamic graph for our experiments and test each of the three distinct feature vectors. A detailed description of the feature vectors and the dynamic graph's construction can be found in Section 4. We split our data into a training, validation, and test set, with the following splits: 60%, 20%, and 20%. With the splits, we achieve a training set consisting of 22,900, and a validation and test set of 7,634. The training, validation and test sets are generated randomly, but are kept fixed throughout the experiments.

	GCN + LSTM	LSTM	GCN
Abstract	0.8284 ± 0.0162	1.0164 ± 0.0140	1.279 ± 0.1350
Author	0.7477 ± 0.0166	1.0184 ± 0.0273	1.1089 ± 0.0357
Venue	0.9259 ± 0.1161	1.0414 ± 0.0197	1.0828 ± 0.0030
Author + Venue	0.7572 ± 0.0131	1.0186 ± 0.0240	1.1248 ± 0.0271
All	0.7940 ± 0.0138	1.0152 ± 0.0157	1.3115 ± 0.1681

Table 2
The performance of our 3 models over a 10 year period. The results are reported as the MAE of the log citations. For the 10-year period, our deterministic approach have a MAE of 1.6378.

	GCN + LSTM	LSTM	GCN
Abstract	0.8001 ± 0.0147	1.0149 ± 0.0414	1.6690 ± 0.4404
Author	0.7462 ± 0.0911	1.0179 ± 0.0536	1.3756 ± 0.0334
Venue	0.8525 ± 0.1348	1.0156 ± 0.0388	1.3212 ± 0.0039
Author + Venue	0.7515 ± 0.0889	1.0132 ± 0.0480	1.3598 ± 0.0461
All	0.7803 ± 0.0167	1.0165 ± 0.0383	1.5177 ± 0.1892

Table 3
The performance of our 3 models over a 20 year period. The results are reported as the MAE of the log citations. For the 20-year period, our deterministic approach have a MAE of 2.0796.

Due to the large number of time-steps in the dynamic graph, we chose to create two different setups for our experiments. One which uses the last 10 years and another, which uses the last 20 years of the dynamic graph. We use the later years in the dynamic graph as these years contain the most papers and the graph has evolved the most.

While not mentioned in Section 4.3, we perform some further pre-processing of the data. For the feature vectors of author rank and venue rank, we perform a normalization of the values. We also perform pre-processing of the labels due to the high fluctuation of the number of citations. We take the $log(c+1)$ of the citation of a paper as the labels [22]. Taking the log of the citation increases the stability of the model during training.

5.2. Experimental Setup

We perform experiments with three distinct models: 1) Our proposed model, consisting of a GCN and LSTM; 2) a standard LSTM; 3) a standard GCN. All hyper-parameters are shared across the models. We evaluate models at specific times and over time.

For our selected models, we used the Adam [23] optimizer, with a learning rate of 0.001. For the GCN we used two layers, with each layer consisting of 256 hidden units. Both the GCN and the GCN with LSTM used this setup. The LSTM was set to have a single uni-directional layer of 128 hidden units, with the output being reduced to 1 dimension by a linear layer. For the models using an LSTM, we its batch size to 256. We ran the models for 1000 epochs and if no update to the best validation score have been observed over 10 epochs, we terminate the training early. As mentioned, we used SciBERT to encode the abstracts, with an output vector of size 768. The models have been run using random seeds, and each of the experiments have been executed 10 times. In the results section, we report the mean and the standard deviation of the 10 runs.

We compare to a simple deterministic baseline: predicting the mean citation count of the training and validation at each time step.

5.3. Evaluation Metric

To evaluate the performance of the models, we measure the mean absolute error, defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y - \hat{y}|, \tag{4}$$

where Y are the citation counts and $\hat{Y}$ are the predicted values. We also use the MAE to optimize the model. We chose to use MAE, instead of mean squared error (MSE), to mitigate outlier papers which have a high amount of citations. We additionally use MAE as the training objective for the same reason.

5.4. Results

As previously mentioned, we ran our experiments on dynamic graphs of 10 years and 20 years. The results of the 10 year experiment is shown in Table 2, and the results of the 20 years experiment is shown in Table 3. The tables show that our models outperform the simple deterministic approach. Figure 3 shows results over time.

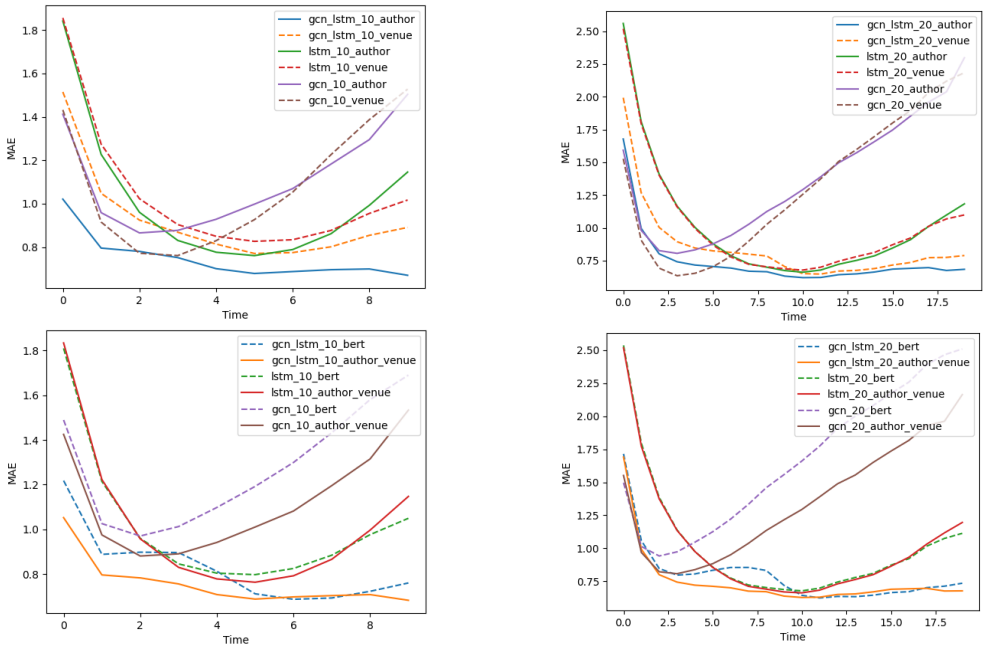

Figure 3: MAE at each time-step, where **left** show the MAE for our 10 year experiments and **right** show the MAE at each time-step for our 20 year experiments, where the x-axis shows the time and y-axis the MAE.

By inspecting the results, one can clearly observe that the GCN-LSTM has the best performance among the three models. We further observe that the GCN-LSTM improves on the performance of the pure GCN and LSTM individually, indicating that it learns from both the temporal and the topological information provided by the dynamic citation network. Furthermore, the GCN increases in error going from a 10 year interval to a 20 year interval, where we see the other models slightly improve. To further study this, we plot the error of the different time steps in Figure 3, which show the models' performances over time. By inspecting the plots, we observe a trend of the pure models i.e. the GCN and LSTM models, struggle and deteriorate over time, compared to the combined GCN-LSTM model, which keeps improving over time until it starts plateauing. Comparing the 10-year and 20-year plots, one can observe that the deterioration continues, where the 10-year plot stops. It can also be seen, that the GCN-LSTM keeps improving up until year 10, where it levels out. All of the models decrease drastically in error up until two time-steps; afterward, the pure models start deteriorating.

5.5. Discussion

Tables 2 and 3 show the impact of single feature types. We hypothesize that author information is very predictive, as shown by prior work. Inspecting the results from the different feature ablations, we can observe that the author features performs the best, confirming our hypothesis. Figure 3 further confirms this, showing that large parts of the gain of the model over time stems from author information.

The feature vector created by the venues performs the worst in both experiments. We hypothesize that the venues' performance could be increased if a more generalized notation for venue meta-data were available. They are noisy (also due to OCR errors) and have many spelling variants.

To further study the impact of features, we calculate the average MAE for each distinct author and venue, where we use the predictions made by the GCN-LSTM, trained on the author feature vectors over 20 years. We show the result for the venues in Table 4 and the ones for authors in Table 5. One can observe that the difference between the top and the bottom venue is drastically lower than the difference between the top and bottom author. This further indicates that author features are a strongly predictive feature for citation counts.

We also show the average degree and the number of papers for each of the venues in Table 4. With a higher representation of papers in the collection, we expect a more reliable prediction. This is indeed the case – we observe the top venues often have a higher number of papers in their collection. To further analyse this, we

	Venue	MAE	Avg. degree	n
1	COLING 1973	0.04295	1	20
2	AAAI 2020	0.06397	4.67	240
3	NAACL 2019	0.0863	15.25	2160
⋮				
185	ACL 1983	0.7714	2	20
186	ACL 1988	0.7794	19.6	100
187	EMNLP 1998	0.8917	4.5	40

Table 4
The top 3 and bottom 3 venues, sorted by the mean MAE, going from lowest to highest.

	Author ID	MAE	Avg. degree	n
1	32968	0.0131	14	1
2	22037	0.0131	14	1
3	32969	0.0131	14	1
⋮				
24536	1375	2.6356	5	1
24537	807	2.6356	5	1
24358	4290	2.6356	5	1

Table 5
The top 3 and bottom 3 authors, sorted by the mean MAE, going from lowest to highest.

observe the average degree of the papers in the collection, however, we do not notice a higher performance where the degree is higher. This indicates that the model is better at predicting papers with higher citation counts, because the degree of a node is tightly bound to the number of citations.

6. Conclusions

In this paper, we propose the task of citation sequence prediction. We introduce a new dataset of scholarly documents for this task based on a dynamic citation graph evolving of 42 years, starting from a single node growing to a large graph. We further study the effect of temporal and topological information, and propose a model to benefit from both information (GCN+LSTM). Our results show that utilizing both the temporal and topological information is superior to only utilizing either the temporal or topological information. Using the proposed model, we study the effect of different features, to identify which information is most predictive of a paper's citation count over time. We find author information to be the most predictive and informative over time.

In future work, the impact of training a single GCN on the dynamic graph could be explored, since the error over time of the GCN is deteriorates fast.

Acknowledgments

We like to thank Johannes Bjerva for the fruitful discussions in the early stages. This work is partly funded by Independent Research Fund Denmark under grant agreement number 9065-00131B.

References

[1] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780. URL: https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735.

[2] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012. doi:10.1016/j.aiopen.2021.01.001.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A Comprehensive Survey on Graph Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 4–24. doi:10.1109/TNNLS.2020.2978386.

[4] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. A. Ha, R. M. Kinney, S. Kohlmeier, K. Lo, T. C. Murray, H.-H. Ooi, M. E. Peters, J. L. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. v. Zuylen, O. Etzioni, Construction of the Literature Graph in Semantic Scholar, in: NAACL-HLT, 2018. doi:10.18653/v1/N18-3011.

[5] R. Yan, J. Tang, X. Liu, D. Shan, X. Li, Citation count prediction: learning to estimate future citations for literature, in: Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11, ACM Press, Glasgow, Scotland, UK, 2011, p. 1247. URL: http://dl.acm.org/citation.cfm?doid=2063576.2063757. doi:10.1145/2063576.2063757.

[6] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[7] T. Yu, G. Yu, P.-Y. Li, L. Wang, Citation impact prediction for scientific papers using stepwise regression analysis, Scientometrics 101 (2014) 1233–1252. URL: http://link.springer.com/10.1007/s11192-014-1279-6. doi:10.1007/s11192-014-1279-6.

[8] D. Kang, W. Ammar, B. Dalvi, M. van Zuylen,

S. Kohlmeier, E. Hovy, R. Schwartz, A Dataset of Peer Reviews (PeerRead): Collection, Insights and NLP Applications, arXiv:1804.09635 [cs] (2018). URL: http://arxiv.org/abs/1804.09635, arXiv: 1804.09635.

[9] B. Plank, R. v. Dalen, CiteTracked: A Longitudinal Dataset of Peer Reviews and Citations, in: Proceedings of BIRNDL ACM SIGIR, Paris, France, July 25, 2019, volume 2414, CEUR-WS.org, 2019, pp. 116–122.

[10] S. Li, W. X. Zhao, E. J. Yin, J.-R. Wen, A Neural Citation Count Prediction Model based on Peer Review 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 4913–4923. URL: https://www.aclweb.org/anthology/D19-1497. doi:10.18653/v1/D19-1497.

[11] J. Wen, L. Wu, J. Chai, Paper Citation Count Prediction Based on Recurrent Neural Network with Gated Recurrent Unit, in: 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2020, pp. 303–306. doi:10.1109/ICEIEC49280.2020.9152330, iSSN: 2377-844X.

[12] F. Davletov, A. S. Aydin, A. Cakmak, High Impact Academic Paper Prediction Using Temporal and Topological Features, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14, ACM Press, Shanghai, China, 2014, pp. 491–498. URL: http://dl.acm.org/citation.cfm?doid=2661829.2662066. doi:10.1145/2661829.2662066.

[13] J. Tang, D. Zhang, L. Yao, Social Network Extraction of Academic Researchers, in: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM '07, IEEE Computer Society, USA, 2007, pp. 292–301. URL: https://doi.org/10.1109/ICDM.2007.30. doi:10.1109/ICDM.2007.30.

[14] J. N. Manjunatha, K. R. Sivaramakrishnan, R. K. Pandey, M. N. Murthy, Citation prediction using time series approach KDD Cup 2003 (task 1), ACM SIGKDD Explorations Newsletter 5 (2003) 152–153. URL: https://doi.org/10.1145/980972.980993. doi:10.1145/980972.980993.

[15] C. Caragea, J. Wu, A. Ciobanu, K. Williams, J. Fernández-Ramírez, H.-H. Chen, Z. Wu, L. Giles, CiteSeerx: A Scholarly Big Dataset, in: M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, K. Hofmann (Eds.), Advances in Information Retrieval, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2014, pp. 311–322. doi:10.1007/978-3-319-06028-6_26.

[16] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective Classification in Network Data, AI Magazine 29 (2008) 93–93. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/2157. doi:10.1609/aimag.v29i3.2157, number: 3.

[17] C. L. Giles, K. D. Bollacker, S. Lawrence, CiteSeer: an automatic citation indexing system, in: Proceedings of the ACM International Conference on Digital Libraries, ACM, 1998, pp. 89–98. URL: https://pennstate.pure.elsevier.com/en/publications/citeseer-an-automatic-citation-indexing-system.

[18] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction, IEEE Transactions on Intelligent Transportation Systems (2019) 1–11. doi:10.1109/TITS.2019.2935152.

[19] S. M. Gerrish, D. M. Blei, A language-based approach to measuring scholarly impact, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, pp. 375–382.

[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805 [cs] (2019). URL: http://arxiv.org/abs/1810.04805, arXiv: 1810.04805.

[21] I. Beltagy, K. Lo, A. 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://www.aclweb.org/anthology/D19-1371. doi:10.18653/v1/D19-1371.

[22] G. Maillette de Buy Wenniger, T. van Dongen, E. Aedmaa, H. T. Kruitbosch, E. A. Valentijn, L. Schomaker, Structure-Tags Improve Text Classification for Scholarly Document Quality Prediction, in: Proceedings of the First Workshop on Scholarly Document Processing, Association for Computational Linguistics, Online, 2020, pp. 158–167. URL: https://www.aclweb.org/anthology/2020.sdp-1.18. doi:10.18653/v1/2020.sdp-1.18.

[23] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.