A Neural Citation Count Prediction Model based on Peer Review Text

Siqing Li[1,2], **Wayne Xin Zhao**[*1,2], **Eddy Jing Yin** and **Ji-Rong Wen**[1,2]
[1]School of Information, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
{lisiqing,jrwen}@ruc.edu.cn
batmanfly@gmail.com eddy_yin@hotmail.com

Abstract

Citation count prediction (CCP) has been an important research task for automatically estimating the future impact of a scholarly paper. Previous studies mainly focus on extracting or mining useful features from the paper itself or the associated authors. An important kind of data signals, *i.e.,* peer review text, has not been utilized for the CCP task. In this paper, we take the initiative to utilize peer review data for the CCP task with a neural prediction model. Our focus is to learn a comprehensive semantic representation for peer review text for improving the prediction performance. To achieve this goal, we incorporate the abstract-review match mechanism and the cross-review match mechanism to learn deep features from peer review text. We also consider integrating hand-crafted features via a wide component. The deep and wide components jointly make the prediction. Extensive experiments have demonstrated the usefulness of the peer review data and the effectiveness of the proposed model. Our dataset has been released online.

1 Introduction

In recent years, the number of scientific publications has been growing in a dramatic rate. For example, the numbers of submissions and accepted papers of EMNLP 2019 have increased to 2,877 and 684 respectively[1]. Given the huge volume of scholarly papers, a long-standing research challenge is how to effectively evaluate the impact of scientific literature (Garfield, 1999; Saha et al., 2003; Bornmann, 2013). A typical way to measure the impact of a scholarly paper is through the number of citations received after publication (Garfield, 1979; Aksnes, 2006), reflecting the influence in the research community.

Since citation count is an important evaluation measure for scientific impact, many researchers aim to develop automatic ways to predict the future citation of a paper (Castillo et al., 2007; Ibáñez et al., 2009; Davletov et al., 2014; Xiao et al., 2016). A typical approach is to casting the problem into a classification or regression task, focusing on extracting useful feature information (Yan et al., 2011; Chen and Zhang, 2015; Singh et al., 2015; Park et al., 2017) (*e.g.,* h-index and topic distribution). Although these studies have achieved important progress on this task, they mainly utilize information from the papers themselves or their associated authors. They have neglected an important kind of data signal for the prediction task, *i.e.,* peer review data.

Peer review is a widely adopted paper evaluation mechanism, in which three or more reviewers would be assigned to decide whether to accept or reject a paper. During the review process, the reviewers should assess the paper quality in terms of several important factors, including originality, correctness, substance and readability[2]. Intuitively, peer review data should be useful to predict future impact of a paper, since the review text contains assessment comments from domain experts. Fortunately, the mechanism of open review (Soergel et al., 2013) has made it possible to obtain peer review data for the citation count prediction (CCP) task.

Although it is appealing to leverage peer reviews for the CCP task, it is difficult to effectively extract supporting evidence and learn comprehensive semantic representations from peer review data. Reviews are usually written in natural language text, covering the assessment comments of a paper in multiple aspects. Some comments may not focus on the main contribution of a paper. For

*Corresponding author
[1]https://emnlp2019.org
[2]https://acl2018.org

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4914–4924,
Hong Kong, China, November 3–7, 2019.

example, a review typically contains the reminders for minor spelling errors or format problems. Another interesting observation is that different reviewers may focus on different aspects in their comments, and even raise divergent attitudes on the same aspect. Hence, it is important to consider both coverage and divergence of review comments for making a comprehensive decision on the paper impact.

In this paper, we take the initiative to study how to utilize the peer review data in the CCP task. We focus on how to learn a comprehensive semantic representation from peer review text for improving the prediction performance. To identify relevant evidence from long text, we utilize the abstract-review match method to learn abstract-aware review representations by using abstract text as an attentive query. In this way, we can reduce the influence of irrelevant content or noise. To further characterize the interaction among multiple reviews, we propose a novel cross-review match mechanism. With such a mechanism, a review representation will be decomposed into a parallel representation and an orthogonal representation by referring to the rest of the reviews. Our model can derive an effective semantic representation for capturing the comprehensive semantics of all the reviewers.

To evaluate our model, we have constructed two peer review datasets with citation counts. Extensive experiments have demonstrated the superiority of the proposed model over several competitive baselines. To our knowledge, it is the first time that peer review data has been utilized in the CCP task. Our work has shown that peer review data is important to improve the prediction performance. Our code and dataset have been released at `https://github.com/RUCAIBox/Citation-Count-Prediction`.

2 Related Work

Citation count prediction has been a hot research topic in the literature (Castillo et al., 2007; Ibáñez et al., 2009; Chakraborty et al., 2014). Early studies casted this task as a classification or regression task (Fu and Aliferis, 2008). Their focus was to identify features in a certain aspect to explore the factors of the impact of papers. Following works formally defined this task and thoroughly examined various possible factors correlated with citation counts (Yan et al., 2011; Bhat et al., 2015; Chen and Zhang, 2015; Singh et al., 2015; Chen and Zhang, 2015; Park et al., 2017). These studies mainly model the long-term scientific impact (Wu et al., 2019; Abrishami and Aliakbary, 2018; Yuan et al., 2018). Furthermore, some researchers casted the problem as a time series task, and focused on analyzing temporal features or patterns in the process of citation growth (Davletov et al., 2014; Xiao et al., 2016; Yuan et al., 2018). However, to the best of our knowledge, no work has utilized peer review data of scholarly papers for citation count prediction.

As an important paper evaluation mechanism, peer review has been widely adopted in various journals and conferences (Ross et al., 2006; Fisher et al., 1994). Based on private review data, researchers have explored the usefulness of peer reviews in several aspects, such as issue localization (Xiong et al., 2010), review utility (Xiong and Litman, 2011) and quality/tone (Ramachandran and Gehringer, 2011). More recently, to lower the barrier to studying peer reviews for the scientific community, a public dataset of peer reviews has been released for research purpose (Kang et al., 2018). Based on this dataset, Wang and Wan (2018) have employed peer review text to predict the overall decision status for a paper submission. Compared with (Wang and Wan, 2018), we focus on a different task by studying how to leverage peer review for future impact estimation instead of paper acceptance, which has its own technical challenges. Besides, we have released our dataset with citation counts online.

Our work is also related to the studies that analyze scientific literature or citation data, including concept or keyphrase extraction (Shen et al., 2018; Gordon et al., 2016; Luan et al., 2017; Caragea et al., 2014), citation or influence analysis (Lauscher et al., 2018; Chakraborty and Narayanam, 2016; Bonab et al., 2018; Chen et al., 2018), context modeling (Cohan and Goharian, 2017; Jin and Szolovits, 2018) and automatic paper rating (Yang et al., 2018).

3 Problem Formulation

Let d denote a scientific paper from a literature corpus $\mathcal{D}$. Following (Ibáñez et al., 2009), we only consider the abstract text for modeling paper content, which summarizes the main contributions of a paper. Let $\boldsymbol{a}_d$ denote the abstract text of d, which consists of multiple abstract sentences.

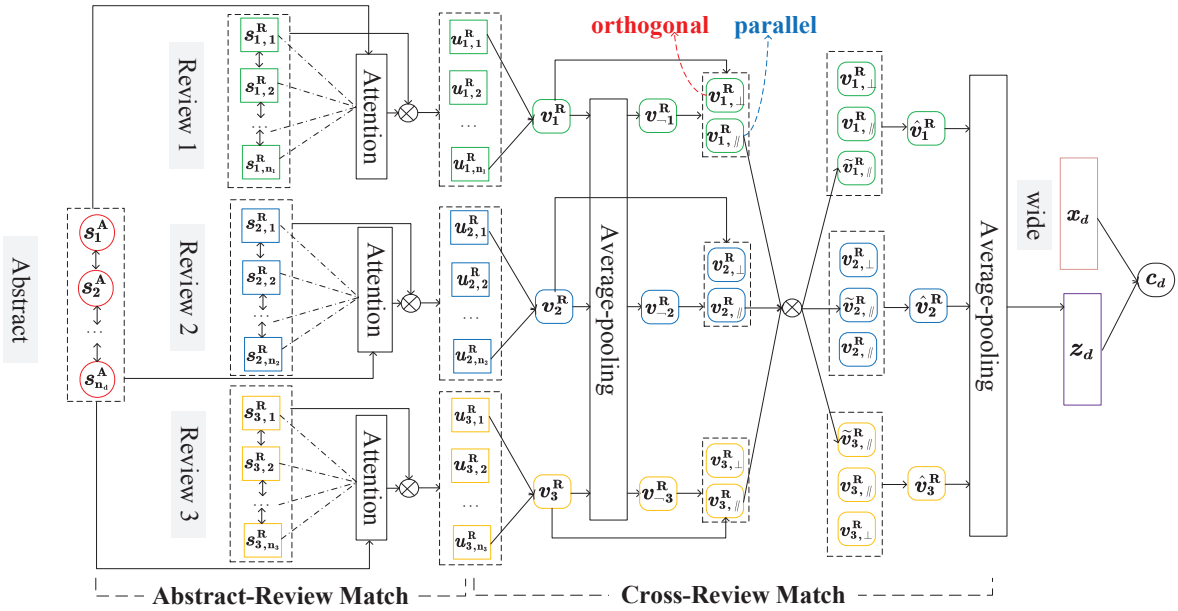


Figure 1: The overview of the proposed citation count prediction model.

We also assume that K peer reviews are available for paper d, characterized by $\{\boldsymbol{r}_k\}_{k=1}^{K}$, where $\boldsymbol{r}_k$ denotes the k-th review consisting of multiple review sentences. We assume that both abstract and review text share the same vocabulary $\mathcal{V}$. Besides these features, we also assume other types of information (*e.g.,* authors' h-index) are also available for our task. We use a vectorized representation $\boldsymbol{x}_d$ to encode all non-review features.

Based on the above preliminaries, we now define the Citation Count Prediction (CCP) task. We aim to learn an effective predictive function that takes as input the abstract text, review text and other available information and estimates the future citation count after a given time period:

$$f(\boldsymbol{x}_d, \boldsymbol{a}_d, \{\boldsymbol{r}_k\}_{k=1}^{K}) \rightarrow \hat{c}_d, \tag{1}$$

where $\hat{c}_d$ is the estimated citation count for d. To make the citation number more predictable, we normalize the value range of c_d within the interval $(0, 1)$. Here, we consider long-term citation count prediction in terms of years.

4 The Proposed Model

In this section, we present a neural citation count prediction model based on peer review text. Our model consists of two major components, namely the deep component and wide component, which model review-based text features and other hand-crafted features, respectively. Figure 1 presents an illustrative sketch for our model architecture. The notations and the descriptions are shown in Table 1.

Symbols	Descriptions
$\boldsymbol{x}_d$	the non-review features of paper d
$\boldsymbol{z}_d$	the final review representation of paper d
$\boldsymbol{c}_d$	the citation count of paper d
$\boldsymbol{a}_d$	the abstract text of paper d
$\boldsymbol{r}_k$	the k-th review consisting of multiple review sentences
$\boldsymbol{s}_j^A$	the learned representations of the j-th sentence in the abstract text
$\boldsymbol{s}_{k,j}^R$	the learned representations of the j-th sentence in the k-th review text
h_S	the dimension of sentence vectors
n_d	the number of sentences in abstract of paper d
n_k	the number of sentences in the k-th review of paper d
$\boldsymbol{u}_t^R$	the updated representation of the t-th sentence in a review after abstract-review match
h_H	the GRU hidden size
$\boldsymbol{v}_k^R$	the representation of the k-th review
$\boldsymbol{v}_{k,\parallel}^R$	the parallel representation of the k-th review
$\boldsymbol{v}_{k,\perp}^R$	the orthogonal representation of the k-th review
$\boldsymbol{v}_{\neg k}^R$	the representation of other reviews excluding the k-th review
$\hat{\boldsymbol{v}}_k^R$	the refined representation of the k-th review after cross-review match

Table 1: Notations used in the paper.

4.1 The Deep Component

The deep component is the core part of our model, which aims to extract important semantic charac-

teristics from peer review text for the CCP task. We first encode abstract and review sentences into embedding vectors, and then distill the relevant evidence from review text by referring to the abstract. To characterize the interaction of multiple reviewers, we further design a cross-review match mechanism to capture both consistency and divergence among different reviews.

4.1.1 Encoding Abstract and Review Sentences

We first pretrain the word embeddings using the word2vec model (Mikolov et al., 2013) using all the scientific corpus. To effectively encode the abstract and review sentences, we adopt the convolution-based method in (Kim, 2014) to model the sentences, sequentially consisting of a lookup layer, a convolution layer of 100 filters, and a max pooling layer. We denote the learned sentence representations of the abstract text as $\{\boldsymbol{s}_j^A\}_{j=1}^{n_d}$ and the k-th review text is denoted as $\{\boldsymbol{s}_{k,j}^R\}_{j=1}^{n_k}$, where each $\boldsymbol{s}_j^A$ or $\boldsymbol{s}_{k,j}^R$ is a h_S-dimensional vector for the j-th sentence in the abstract or the k-th review, and n_d and n_k is the number of sentences in abstract of paper d and its k-th review.

4.1.2 Improving the Review Representations with Abstract-Review Match

Review text reflects the subject assessment on a paper by the reviewers. A review is likely to cover the detailed comments from multiple aspects. It may contain irrelevant information for the prediction task, such as requesting source code release or pointing out minor spelling errors. It is key to identify relevant information focusing on the core contributions of a paper.

Intuitively, we can utilize the abstract information to purify the original review sentence representations, since it provides a good summary for the main contributions of a paper. Inspired by the recent progress on machine reading comprehension, we adopt the gated attention-based recurrent networks (Wang et al., 2017) for refining the review representations regarding to the abstract text. In our task, the abstract is considered as a question, and a review is considered as a passage. Similar to machine reading comprehension, we aim to learn a "question"-relevant review sentence representation that focuses on the core content from the abstract. For simplicity, we only consider the interaction between the abstract and an individual review, and omit the review index from the notations. Formally, we update the representation of the t-th sentence in a review as $\boldsymbol{u}_t^R \in \mathbb{R}^{h_H}$:

$$\boldsymbol{u}_t^R = \text{BiGRU}(\boldsymbol{u}_{t-1}^R, [\boldsymbol{s}_t^R, \boldsymbol{p}_t]^*), \tag{2}$$

where $\boldsymbol{p}_t \in \mathbb{R}^{h_S}$ is an attentional vector of a review computed based on the interaction between review and abstract sentence representations:

$$\begin{aligned} \boldsymbol{p}_t &= \sum_{i=1}^{n_d} \alpha_i^t \cdot \boldsymbol{s}_i^A, \\ \alpha_i^t &= \exp(\boldsymbol{h}_i^t) / \sum_{j=1}^{n_d} \exp(\boldsymbol{h}_j^t), \\ \boldsymbol{h}_j^t &= \boldsymbol{u}^\top \tanh(\boldsymbol{W}_s^A \boldsymbol{s}_j^A + \boldsymbol{W}_s^R \boldsymbol{s}_t^R + \boldsymbol{W}_u^R \boldsymbol{u}_{t-1}^R), \end{aligned} \tag{3}$$

where $\boldsymbol{W}$s $\in \mathbb{R}^{h_H \times h_S}$are parameter matrices to learn, $\boldsymbol{s}_j^A$ and $\boldsymbol{s}_t^R$ are the learned sentence representations for abstract and review text respectively in Sec. 4.1.1, αs are the attention coefficients, and $[\boldsymbol{s}_t^R, \boldsymbol{p}_t]^*$ is a gated update of the concatenation vector $[\boldsymbol{s}_t^R, \boldsymbol{p}_t]$:

$$[\boldsymbol{s}_t^R, \boldsymbol{p}_t]^* = \boldsymbol{g}_t \odot [\boldsymbol{s}_t^R, \boldsymbol{p}_t], \tag{4}$$

$$\boldsymbol{g}_t = \text{sigmoid}(\boldsymbol{W}_g [\boldsymbol{s}_t^R, \boldsymbol{p}_t]), \tag{5}$$

where " $\odot$ " is an element-wise product operation for vectors.

In this way, for a review, we can obtain the abstract-aware review sentence representations $\{\boldsymbol{u}_j^R\}_{j=1}^{n_d}$, which encode more relevant information emphasized by the abstract. To learn the overall representation for the k-th review, we concatenate the sentence embeddings of the first and last sentences in it:

$$\boldsymbol{v}_k^R = [\boldsymbol{u}_1^R, \boldsymbol{u}_{n_k}^R], \tag{6}$$

where $\boldsymbol{u}_1^R$ and $\boldsymbol{u}_{n_k}^R$ are learned sentence embeddings using bidirectional Gated Recurrent Unit (BiGRU) in Eq. 2.

4.1.3 Improving Review Representations with Cross-Review Match

Previously, we have considered the interaction between the abstract and an individual review. The evaluation process of a paper typically requires multiple reviewers to make the final decision. According to (Hirschauer, 2010), coverage and divergence should be considered for the acceptance decision of a paper. Therefore, we propose to utilize cross-review match to learn a comprehensive semantic representation from different reviewers.

Given a review, we take the rest of the reviews as a reference source. We aim to learn the common semantics that are also discussed by other reviews (maybe with different attitudes), and identify unmentioned semantics by other reviews. To implement this idea, we adopt the orthogonal decomposition strategy proposed in (Wang et al., 2016). We decompose the original review representation into a *parallel* representation and an *orthogonal* representation. Formally, the representation of the k-th review $\boldsymbol{v}_k^R \in \mathbb{R}^{h_H}$ (Eq. 6) is decomposed to:

$$\boldsymbol{v}_{k,\parallel}^R = \frac{\boldsymbol{v}_k^{R\top} \cdot \boldsymbol{v}_{\neg k}^R}{\boldsymbol{v}_{\neg k}^{R\top} \cdot \boldsymbol{v}_{\neg k}^R} \cdot \boldsymbol{v}_{\neg k}^R, \quad (7)$$

$$\boldsymbol{v}_{k,\perp}^R = \boldsymbol{v}_k^R - \boldsymbol{v}_{k,\parallel}^R, \quad (8)$$

$$\boldsymbol{v}_{\neg k}^R = \text{avg-pooling}(\{\boldsymbol{v}_j^R\}_{j \neq k}). \quad (9)$$

where the parallel representation $\boldsymbol{v}_{k,\parallel}^R \in \mathbb{R}^{h_H}$ encodes common semantics also discussed by other reviews, and the orthogonal representation $\boldsymbol{v}_{k,\perp}^R \in \mathbb{R}^{h_H}$ encodes unmentioned semantics in other reviews. Such a decomposition is useful to extract more comprehensive semantics from multiple reviews. We use average pooling to construct the reference vector of other reviews.

We perform the above decomposition for each review associated with a paper. The parallel representation reflects the common semantics shared by other reviews. Since different reviewers may have divergent comments (or attitudes) towards the same content, we further introduce a corresponding attentional representation for enriching the semantics of the original representation:

$$\widetilde{\boldsymbol{v}}_{k,\parallel}^R = \sum_{j=1}^{K} \alpha_{kj} \cdot \boldsymbol{v}_{j,\parallel}^R, \quad (10)$$

$$\alpha_{k,j} = \exp(s_{k,j}) / \sum_{j'=1}^{K} \exp(s_{i,j'}),$$

$$s_{k,j} = \begin{cases} 0, & \text{if } k = j \\ (\boldsymbol{v}_{k,\parallel}^R)^\top \cdot \boldsymbol{v}_{j,\parallel}^R, & \text{otherwise.} \end{cases}$$

Then we combine the three representations and adopt a fully connected layer to obtain the refined representation for the k-th review $\hat{v}_k^R \in \mathbb{R}^{h_H}$:

$$\hat{v}_k^R = \boldsymbol{W}[\boldsymbol{v}_{k,\perp}^R, \boldsymbol{v}_{k,\parallel}^R, \widetilde{\boldsymbol{v}}_{k,\parallel}^R]. \quad (11)$$

It is able to capture the coverage and divergence in semantics for peer review text to some extent. Finally we use an average pooling operation over all review representations of paper d as its final representation $\boldsymbol{z}_d \in \mathbb{R}^{h_H}$:

$$\boldsymbol{z}_d = \text{avg-pooling}(\{\hat{v}_k^R\}_{k=1}^K). \quad (12)$$

4.2 The Wide Component

Besides review-based features, we consider directly integrating other important features for the prediction task, called *wide features*. Here, we use a vectorized representation $\boldsymbol{x}_d$ to represent all the wide features. We construct the vector by using the features proposed in previous studies (Yan et al., 2011; Bhat et al., 2015):

- Topic distribution: We utilize the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to learn the probability distribution over topics as the topic features.
- Diversity: We calculate the entropy of the paper's topic distribution to measure the topical breadth of a paper.
- Recency: We use the year of publication as the temporal feature to predict the citation count.
- Author influence: We use the number of authors and the average h-index as author features.

4.3 The Joint Deep and Wide Model

Finally, we integrate the two components into a unified model. We take as input the deep and wide features respectively discussed in Section 4.1 and Section 4.2, and combine them as the prediction function (Eq. 1) :

$$\hat{c}_d = \tanh(\boldsymbol{w}_{deep}^\top \cdot \boldsymbol{z}_d + \boldsymbol{w}_{wide}^\top \cdot \boldsymbol{x}_d + b), \quad (13)$$

where $\boldsymbol{z}_d$ and $\boldsymbol{x}_d$ are the derived feature representations from the deep and wide components respectively, and $\boldsymbol{w}_{deep}$ and $\boldsymbol{w}_{wide}$ are the corresponding parameter vectors. Furthermore, we define the citation count prediction error over the training set with the Mean Squared Error (MSE):

$$L(\theta) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} (\hat{c}_d - c_d)^2. \quad (14)$$

where c_d is the normalized real citation count of each scholarly paper. We learn our model parameters via optimizing the $L(\theta)$ loss. The parameters in GRU and CNN are initialized by a normal distribution with zero mean and 0.01 variance, and the biases are initialized as zeros. To optimize our model, we adopt the Stochastic Gradient Descent (SGD) optimizer to learn the

Dataset	NIPS (2013-2016)		ICLR (2013-2017)	
	Abstract	Review	Abstract	Review
#Doc.	1,739	7,171	384	1,119
#Sent.	10,964	109,674	2,368	15,553
#Words	6,448	16,695	2,993	6,680
#Ave. Rev.	4.12		2.91	

Table 2: Statistics of our datasets after preprocessing.

model parameters. More implementation details can be found in Section 5.1.

5 Experiments

In this section, we first set up the experiments, and then present the results and analysis.

5.1 Experimental Setup

Datasets. Peer review data is not available for the majority of mainstream journals and conferences. Fortunately, ICLR and NIPS have provided the review text on their website. NIPS does not provide rating scores from the reviewers, and we only consider utilizing text data in this paper. For most of the published papers, it is difficult to accumulate a considerable number of citations in a short period. Hence, we only use the data ICLR 2013-2017 and NIPS 2013-2016 for evaluation, which has a two-year span to now for long-term impact prediction. The data of NIPS 2013-2016 and ICLR 2017 have been shared by Kang et al. (2018). The data of ICLR 2013-2016 was crawled from the OpenView website[3], including abstract text, review text and author data. Note that we only consider the accepted papers for citation prediction. We further crawl author data (*e.g.,* h-index) and paper citation from Google Scholar[4]. When encountering any ambiguity on author names or paper titles, a senior graduate student will manually collect the corresponding data. All the Google Scholar data has been accessed on March 31, 2019 to guarantee the recency of citation data. We remove the papers with only one reviewer. We perform basic text preprocessing using NLTK[5], including tokenization, lowercase, and stopword removal, and only retain the words that occur three times or more. In order to simulate the real situation, for both datasets, we take the data from the last year as test data, and the previous data as training set. The detailed statistics of the two datasets are summarized in Table 2. Since both datasets have a limited number of papers, we only hold out 5% training data as validation set. Our dataset and code are available at https://github.com/RUCAIBox/Citation-Count-Prediction.

Implementation Details. We pre-trained 300-dimensional word vectors as initial word embeddings using all the data in our dataset. The word vectors were fixed during the training process. The number of CNN filters h_S and GRU hidden size h_H are set to 128. For the wide component, we utilize the LDA to train a 100-topic model. All of the features in the wide component and the citation count are normalized within the interval $(0, 1)$. The dropout rate is set to 0.5 to prevent overfitting. For the hyper-parameters of SGD optimizer, we set the learning rate as 0.001.

Baseline Models. We compare our model against a number of baseline models:

• Linear Regression (LR), K-NearestNeighbor (KNN), Support Vector Regression (SVR) and Gradient Boost Regression Tree (GBRT): The four methods are commonly used regression models for citation prediction (Yan et al., 2011; Bhat et al., 2015). We adopt the same wide features from (Yan et al., 2011) as our wide component.

• Wide & Deep (Cheng et al., 2016): We borrow the Wide & Deep leaning framework from recommender systems to predict the citation count. We modify the deep component by implementing it as a feed-forward neural network on top of a bi-directional RNN component over review text.

• MILAM (Wang and Wan, 2018): It is a multiple instance learning network with a novel abstract-based memory mechanism to predict the overall decision (accept, reject, or borderline) based on review text. We modify the loss of this model as the MSE loss for citation regression. For a fair comparison, we also integrate the wide features in a similar way as our wide component.

Evaluation Metrics. To evaluate the performance of different methods on citation count prediction, following previous studies (Bhat et al., 2015; Yuan et al., 2018), we adopt five evaluation metrics, including MAE, RMSE, OR@30, OR@50, and Spearman's rank correlation coefficient. MAE and RMSE measure the difference between the real value and the predicted value for a regression task.

[3]https://openreview.net/
[4]https://scholar.google.com
[5]https://www.nltk.org

Datasets	Models	MAE ("↓")	RMSE ("↓")	OR(@30) ("↑")	OR(@50) ("↑")	Spearman's Rank ("↑")
NIPS	LR	0.1776	0.1903	0.27	0.33	0.4776
	KNN	0.1701	0.1900	0.33	0.36	0.4848
	SVR	0.1677	0.1856	0.33	0.40	0.5279
	GBRT	0.1863	0.1974	0.23	0.34	0.5310
	Wide&Deep	0.1470	0.1848	0.30	0.38	0.5351
	MILAM	0.1426	0.1792	0.37	0.38	0.5458
	Our model	**0.1349**	**0.1726**	**0.4**	**0.42**	**0.5561**
ICLR	LR	0.2395	0.2723	0.40	0.70	0.1475
	KNN	0.2293	0.2674	0.40	0.72	0.1874
	SVR	0.2226	0.2578	0.40	0.70	0.1328
	GBRT	0.2223	0.2607	0.43	0.70	0.1469
	Wide&Deep	0.2182	0.2607	0.47	0.72	0.2440
	MILAM	0.2093	0.2510	0.47	0.72	0.2510
	Our model	**0.1866**	**0.2279**	**0.50**	**0.76**	**0.3026**

Table 3: Performance comparisons of different methods for citation count prediction using two datasets. "↑" ("↓") indicates that a larger (smaller) value corresponds to a better performance.

Spearman's Rank measures the overall correlation between the predicted list and the ground-truth list sorted by the citation number descendingly. OR@k measures the overlapping ratio between top k predicted results and the real ordered list.

5.2 Results and Analysis

In this subsection, we construct a series of experiments on the effectiveness of the proposed model for the citation count prediction task.

Main Results. Table 3 presents the performance of different methods on citation count prediction. We can make the following observations. First, the four traditional baselines (LR, KNN, SVR and GBRT) perform worse than the two deep learning baselines (W&D, MILAM). These four baselines only utilize the wide features with traditional machine learning models. Second, MILAM performs consistently better than W&D, since it has designed a more elaborate architecture to model the review text. Finally, our model outperforms all the baselines with a substantial margin, especially for the ICLR dataset. Our model is able to integrate the wide features and learn the comprehensive representation for peer review text, which is the key of the performance improvement over baselines. Overall, the two datasets show the similar findings. In what follows, we will report the results on ICLR dataset due to space limit.

Ablation Analysis. The major novelty of our model is that it utilizes abstract-review match and cross-review match to learn a comprehensive abstract-aware representation for peer review text.

Models	MAE	SR
Our full model	**0.1866**	**0.3026**
w/o abstract-review match	0.1983	0.2861
w/o cross-review match	0.2025	0.2735
w/o wide component	0.2065	0.2697

Table 4: Ablation analysis on ICLR dataset (SR = Spearman's Rank).

Models	MAE		SR	
	original	+review	original	+review
LR	0.2395	0.2289	0.1475	0.1700
KNN	0.2293	0.2176	0.1874	0.2155
SVR	0.2226	0.2174	0.1328	0.1768
GBRT	0.2223	0.2157	0.1469	0.2239
Our Model	**0.1866**		**0.3026**	

Table 5: Analysis of the usefulness of peer review text on ICLR dataset (SR = Spearman's Rank).

Moreover, we integrate the wide features for the prediction task. To examine the contribution of the three parts, we examine the performance of the model variants by removing each module from the complete model. We present the MAE results of our model and its three variants in Table 4. As we can see, all components are useful to improve the final performance.

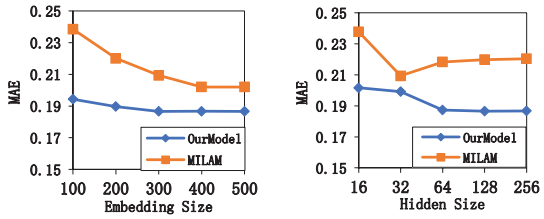
Usefulness of Peer Review Text. A major motivation of this paper is that peer review text is useful to the citation prediction task, which has been seldom studied in previous studies. Hence, we examine whether peer review text is also

Test Datasets	NIPS		ICLR	
Models	MAE	SR	MAE	SR
MILAM	0.1530	0.5227	0.2056	0.26
Our Model	0.1425	0.5348	0.1889	0.2933

Table 6: Analysis on cross-venue prediction (SR = Spearman's Rank).

Abstract								
· · · we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task · · · Additionally, the LSTM did not have difficulty on long sentences. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly.								
Reviewer #1			Reviewer #2			Reviewer #3		
Sentences	$\boldsymbol{v}_{\parallel}$	$\boldsymbol{v}_{\perp}$	Sentences	$\boldsymbol{v}_{\parallel}$	$\boldsymbol{v}_{\perp}$	Sentences	$\boldsymbol{v}_{\parallel}$	$\boldsymbol{v}_{\perp}$
[0.31]-I think your last translated-sentence example kind of shows the kind of weirdness that can result.	0.15	0.85	**[0.21]**-This paper presents the elegant idea of translating from source to target languages with an LSTM.	0.93	0.07	**[0.19]**-The ideas represented in this work are extremely interesting, and I love the elegance and simplicity of the proposed RNN architecture.	0.88	0.12
[0.24]-A solution has to be scalable in principle to long sentences, and I think it's clear that your method cannot.	0.38	0.62	**[0.15]**-This is an elegant model, and I am inclined to accept it, despite the fact that it only "works" for sentences that do not have infrequent words.	0.63	0.37	**[0.17]**-The idea of the paper is good and very interesting, providing an elegant neural solution to machine translation.	0.91	0.09
[0.21]-I am skeptical that this idea could be a practical solution to MT.	0.44	0.56	**[0.09]**-The paper does have some major holes in the experiments.	0.69	0.31	**[0.11]**-I'm sceptical that common RNNs would find it hard to model "long-term dependencies"· · ·	0.14	0.86
[0.14]-I believe the experimental investigation is competent and complete.	0.73	0.27	**[0.05]**-The experimental results are not convincing	0.71	0.29	**[0.04]**-It would be more convincing to have some more examples.	0.54	0.46

Table 7: Samples of the abstract and reviews from NIPS dataset. For each review, we present four comment sentences. The similarity weights *w.r.t.* the abstract text have been shown in red font.



(a) Varing the word embedding size (b) Varing the RNN hidden size

Figure 2: The impact of embedding size and hidden size in terms of MAE scores on ICLR dataset.

useful to improve traditional prediction models (LR, KNN, SVR and KNN). Note that our focus is to verify the general usefulness of peer review text instead of the most suitable text features for the baselines. Here, we adopt the simple yet classic doc2vec model (Le and Mikolov, 2014) to encode peer text into a vectorized representation. Then, we integrate these text features into the four baseline methods. As shown in Table 5, the performance of all the four methods have been improved with the text features. The results have shown that the peer text is indeed generally useful for the citation prediction.

Cross-venue Evaluation. To examine the robustness of our model, we further perform a cross-venue evaluation. For ICLR (NIPS) test set, we apply the models trained on the full NIPS (ICLR) dataset. We only select the best baseline MILAM as a comparison. As shown in Table 6, we are able to see that the performance decreases compared with the results in Table 3, since we use a training set from a different venue. But the decline is not obvious. In the future we will consider how to improve the performance. Our model is still better than the baseline MILAM for both datasets.

Parameter Sensitivity. Next, we investigate the performance with respect to two major parameters in our model, *e.g.,* the word embedding size and the GRU hidden size. As shown in Figure 2(a) and Figure 2(b), our model is consistently better than the best baseline MILAM with all the parameter values. An embedding size of 300 and a hidden size of 128 yield the best results for our model.

5.3 Qualitative Analysis

Previously, we have shown that both the review-abstract match and cross-review match are useful in the prediction performance. In Table 7, we perform the qualitative analysis on a sample paper with three reviews for understanding how the two mechanisms work.

We first compute the similarity weight between the abstract representation and a review sentence, and sort the review sentences according to such weights. As we can see, the comments on model design have been overall ranked in a higher position than those on experiments. With the abstract-review match, our model indeed identifies

more relevant content regarding to the abstract text.

Then, we analyze the corresponding semantic explanation of the cross-review match, in which we decompose an overall review representation into a parallel representation $\boldsymbol{v}_{\parallel}$ and an orthogonal representation $\boldsymbol{v}_{\perp}$. It is difficult to directly understand the two vectors. Instead, we compute the similarity between a comment sentence and the review parallel (or orthogonal) representation. Then we normalize the two similarity values into a distribution over two representations (parallel or orthogonal). It can be seen that the comments focusing on the common aspect have a larger weight on $\boldsymbol{v}_{\parallel}$. For example, for the comments from the first row, reviewer #2 and reviewer #3 have similar general comments about the model architecture, and both comments have very large weights on the parallel representation. While, reviewer #1 raises a different comment on other model detail, corresponding to a large weight on the orthogonal representation. Interestingly, the fourth row corresponds to the comments on the experiments. The three comment sentences are more related to the parallel representation (*i.e.,* the common issue), while they have conveyed different attitudes. This phenomenon can be partially captured by the representation $\widetilde{\boldsymbol{v}}^{R}_{k,\parallel}$ (Eq. 10) by attending to the parallel representations of other reviews.

6 Conclusion

In this paper, we studied how to utilize the peer review text to improve the citation prediction task. We developed a joint deep and wide model that was able to integrate both deep and wide features into a unified predictive function. The deep features were learned from peer review text by applying the abstract-review and cross-review match mechanisms. We constructed two evaluation datasets with peer review text. Extensive results have demonstrated the effectiveness of our proposed model. Currently, we only consider the semantic-level representations from peer review. As future work, we will consider how to extend our work by modeling sentiments of review text.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful and constructive comments. This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872369, 61832017 and 61572059, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China under Grant No. 18XNLG22 and 19XNQ047.

References

Ali Abrishami and Sadegh Aliakbary. 2018. Nncp: A citation count prediction methodology based on deep neural network learning techniques. *arXiv preprint arXiv:1809.04365*.

Dag W Aksnes. 2006. Citation rates and perceptions of scientific contribution. *Journal of the American Society for Information Science and Technology*, 57(2):169–185.

Harish S Bhat, Li-Hsuan Huang, Sebastian Rodriguez, Rick Dale, and Evan Heit. 2015. Citation prediction using diverse features. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 589–596. IEEE.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Hamed Bonab, Hamed Zamani, Erik G Learned-Miller, and James Allan. 2018. Citation worthiness of sentences in scientific reports. In *SIGIR*, pages 1061–1064.

Lutz Bornmann. 2013. What is societal impact of research and how can it be assessed? a literature survey. *Journal of the American Society for Information Science and Technology*, 64(2):217–233.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446.

Carlos Castillo, Debora Donato, and Aristides Gionis. 2007. Estimating number of citations using author reputation. In *International Symposium on String Processing and Information Retrieval*, pages 107–117. Springer.

Tanmoy Chakraborty, Suhansanu Kumar, Pawan Goyal, Niloy Ganguly, and Animesh Mukherjee. 2014. Towards a stratified learning approach to predict future citation counts. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 351–360. IEEE Press.

Tanmoy Chakraborty and Ramasuri Narayanam. 2016. All fingers are not equal: Intensity of references in scientific articles. *arXiv preprint arXiv:1609.00081*.

Chengyao Chen, Zhitao Wang, Wenjie Li, and Xu Sun. 2018. Modeling scientific influence for research trending topic prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Junpeng Chen and Chunxia Zhang. 2015. Predicting citation counts of papers. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*, pages 434–440. IEEE.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM.

Arman Cohan and Nazli Goharian. 2017. Contextualizing citations for scientific summarization using word embeddings and domain knowledge. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136. ACM.

Feruz Davletov, Ali Selman Aydin, and Ali Cakmak. 2014. 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*, pages 491–498. ACM.

Martin Fisher, Stanford B Friedman, and Barbara Strauss. 1994. The effects of blinding on acceptance of research papers by peer review. *Jama*, 272(2):143–146.

Lawrence D Fu and Constantin Aliferis. 2008. Models for predicting and explaining citation count of biomedical articles. In *AMIA Annual Symposium Proceedings*, volume 2008, page 222. American Medical Informatics Association.

Eugene Garfield. 1979. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 1(4):359–375.

Eugene Garfield. 1999. Journal impact factor: a brief review.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 866–875.

Stefan Hirschauer. 2010. Editorial judgments: A praxeology of ąővotingąŕin peer review. *Social Studies of Science*, 40(1):71–103.

Alfonso Ibáñez, Pedro Larrañaga, and Concha Bielza. 2009. Predicting citation count of bioinformatics papers within four years of publication. *Bioinformatics*, 25(24):3303–3309.

Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. *arXiv preprint arXiv:1808.06161*.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Kai Eckert. 2018. Investigating the role of argumentation in the rhetorical analysis of scientific publications with neural multi-task learning models. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. *arXiv preprint arXiv:1708.06075*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ho-Min Park, Yenewondim Biadgie Sinshaw, and Kyung-Ah Sohn. 2017. Temporal citation network-based feature extraction for cited count prediction. In *International Conference on Mobile and Wireless Technology*, pages 380–388. Springer.

Lakshmi Ramachandran and Edward F Gehringer. 2011. Automated assessment of review quality using latent semantic analysis. In *2011 IEEE 11th International Conference on Advanced Learning Technologies*, pages 136–138. IEEE.

Joseph S Ross, Cary P Gross, Mayur M Desai, Yuling Hong, Augustus O Grant, Stephen R Daniels, Vladimir C Hachinski, Raymond J Gibbons, Timothy J Gardner, and Harlan M Krumholz. 2006. Effect of blinded peer review on abstract acceptance. *Jama*, 295(14):1675–1680.

Somnath Saha, Sanjay Saint, and Dimitri A Christakis. 2003. Impact factor: a valid measure of journal quality? *Journal of the Medical Library Association*, 91(1):42.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A web-scale system for scientific knowledge exploration. *arXiv preprint arXiv:1805.12216*.

Mayank Singh, Vikas Patidar, Suhansanu Kumar, Tanmoy Chakraborty, Animesh Mukherjee, and Pawan Goyal. 2015. The role of citation context in predicting long-term citation profiles: An experimental study based on a massive bibliographic text dataset. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1271–1280. ACM.

David Soergel, Adam Saunders, and Andrew McCallum. 2013. Open scholarship and peer review: a time for experimentation.

Ke Wang and Xiaojun Wan. 2018. Sentiment analysis of peer review texts for scholarly papers. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 175–184. ACM.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*.

Ziming Wu, Weiwei Lin, Pan Liu, Jingbang Chen, and Li Mao. 2019. Predicting long-term scientific impact based on multi-field feature extraction. *IEEE Access*, 7:51759–51770.

Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M Chu, and Hongyuan Zha. 2016. On modeling and predicting individual paper citation count over time. In *IJCAI*, pages 2676–2682.

Wenting Xiong and Diane Litman. 2011. Automatically predicting peer-review helpfulness. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 502–507. Association for Computational Linguistics.

Wenting Xiong, Diane Litman, and Christian Schunn. 2010. Assessing reviewers' performance based on mining problem localization in peer-review data. In *Educational Data Mining 2010-3rd International Conference on Educational Data Mining*, pages 211–220.

Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. 2011. Citation count prediction: learning to estimate future citations for literature. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1247–1252. ACM.

Pengcheng Yang, Xu Sun, Wei Li, and Shuming Ma. 2018. Automatic academic paper rating based on modularized hierarchical convolutional neural network. *arXiv preprint arXiv:1805.03977*.

Sha Yuan, Jie Tang, Yu Zhang, Yifan Wang, and Tong Xiao. 2018. Modeling and predicting citation count via recurrent neural network with long short-term memory. *arXiv preprint arXiv:1811.02129*.