A Neural Probabilistic Model for Context Based Citation Recommendation

Wenyi Huang[†], Zhaohui Wu[‡], Chen Liang[†], Prasenjit Mitra[†‡], C. Lee Giles[†‡]
[†]Information Sciences and Technology, [‡]Computer Sciences and Engineering
The Pennsylvania State University
University Park, PA 16802
{harrywy,laowuz}@gmail.com
{cul226,pmitra,giles}@ist.psu.edu

Abstract

Automatic citation recommendation can be very useful for authoring a paper and is an AI-complete problem due to the challenge of bridging the semantic gap between citation context and the cited paper. It is not always easy for knowledgeable researchers to give an accurate citation context for a cited paper or to find the right paper to cite given context. To help with this problem, we propose a novel neural probabilistic model that jointly learns the semantic representations of citation contexts and cited papers. The probability of citing a paper given a citation context is estimated by training a multi-layer neural network. We implement and evaluate our model on the entire CiteSeer dataset, which at the time of this work consists of 10,760,318 citation contexts from 1,017,457 papers. We show that the proposed model significantly outperforms other state-of-the-art models in recall, MAP, MRR, and nDCG.

Introduction

Citations are crucial for assignments of academic credit. Proper citations also help support claims in one's own work. However, with the growth in the number of research publications, researchers might find it hard to find appropriate and necessary work to cite. A citation recommendation engine can help check the completeness of citations when authoring a paper and find prior work related to the topic under investigation and to find missing relevant citations.

Most citation recommendation models fall into two categories: global recommendation (McNee et al. 2002; Strohman, Croft, and Jensen 2007; Nallapati et al. 2008; Bethard and Jurafsky 2010; Kataria, Mitra, and Bhatia 2010; Kucuktunc et al. 2012; Ren et al. 2014) which recommends a list of references for a given manuscript, and local recommendation (Tang and Zhang 2009; He et al. 2010; 2011; Huang et al. 2012b) which recommends citations for a particular context where a citation should be made. In this paper, we will focus on the latter: the local citation recommendation (or context-based recommendation).

A citation context c is defined as a sequence of words that appear around a particular citation. Usually a citation context contains words that describe or summarize the cited papers. Intuitively, the semantics of the cited documents should be close to the citation contexts. As shown in Fig. 1, words in red, such as "PageRank," "hyperlink," and "node ranking," should be semantically related to the cited paper. In addition, since all these words are used to describe the same citation, the semantics of these words should also be similar. This motivates us to learn the semantic embeddings for words in the citation contexts and cited documents and to recommend citations based on the semantic distance.

Citation Context Examples	Cited Paper
1. ...For example, PageRank [*] can be applied to the hyperlink structure on domains to obtain domain rank scores... 2. ..., and the unique solution vector r(i) can be expressed as the eigenvector of a matrix [*] or as the stationary probability of a random walk... 3. ... Rank sinks [*] are defined to be a set of nodes which have links between themselves but no links to the other nodes. 4. There is a lot of research work on static information network analysis, including ... , and node ranking [*, *], ...	L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. *The Pagerank Citation Ranking: Bringing Order to the Web.* Technical report, Stanford Digital Library Technologies Project, 1998.

Figure 1: Examples for citation contexts and the cited paper.

We propose to learn the distributed semantic representations of the words and the cited documents. Using the distributed representations of words and documents, we train a neural network model that estimates the probability of citing a paper given a citation context. The neural network model will tune the distributed representations of words and documents so that the semantic similarity between the citation context and the cited paper will be high. The model also ensures that key words used for citing similar documents will have high semantic similarity.

To evaluate our model, we conducted an experiment on a snapshot of the entire CiteSeer (Giles, Bollacker, and Lawrence 1998) dataset at Oct. 2013[1]. Our model outperformed other state-of-the-art models by achieving a recall@10 of 35.37%, a mean average precision of 18.35%, a MRR score of 0.1843, and an nDCG score of 0.2566.

The major contributions of this work are:

[1]The dataset is publicly available at http://refseer.ist.psu.edu/data/.

- We propose a neural probabilistic model that learns the probability of citing a paper given a citation context based on distributed representations of words and documents. The model is implemented in RefSeer[2] (Huang et al. 2014) ,a citation recommendation engine, for public uses.
- We evaluate the model on the entire CiteSeer dataset which consists of $10,760,318$ pairs of citation contexts and cited documents from $1,017,457$ papers. To the best of our knowledge, this may be the first work that evaluates citation recommendation on this large scale dataset.
- Compared to other state-of-the-art context-based methods, our model shows significant improvement on various performance metrics, with a 5% gain in recall@10, a 2% gain in MRR and MAP, and a 3% gain in nDCG.

Related Work

Citation Recommendation

There are two major strands of works on citation recommendation. The first is global citation recommendation that suggests a list of references for an entire paper manuscript. McNee et al. (2002) used a partial list of reference as the input query and recommended additional references based on collaborative filtering. Strohman et al. (2007) assumed that the input is an incomplete paper manuscript and recommended papers with high text similarity using bibliography similarity and Katz centrality measurement. Bethard and Jurafsky (2010) introduced features such as topical similarity, author behavioral patterns, citation count, and recency of publication to train a classifier for global recommendation. Topic modeling based approaches (Nallapati et al. 2008; Kataria, Mitra, and Bhatia 2010) were proposed to recommend papers that are topically relevant to the input query.

The second strand is local citation recommendation. The input query is a particular context where a citation should be made. Usually the context consists of one to three sentences. Local recommendation aims to recommend a short list of papers that need to be cited within the given context. This is the setting that we are focusing on in this paper.

Tang and Zhang (2009) proposed to use Restricted Boltzmann Machine to recommend citations based on candidate citation contexts. He et al. (2010) assumed that user has provided placeholders for citation in a query manuscript. A probabilistic model was trained to measure the relevance between the citation contexts and the cited documents. In another work (He et al. 2011), they first segmented the document into a sequence of disjointed candidate citation contexts, and then applied the dependency feature model to retrieve a ranked list of papers for each context. Statistical machine translation models were used to model the relation between citation contexts and cited documents. Huang et al. (2012b) proposed to treat citation context as the source language and the cited documents as target language. IBM translation model-1 was trained to learn the probability of citing a document given a word. Lu et al. (2011) used the translation model to learn the translation probabilities between words in the citation context and the cited document. Different from previous models that use statistical or topical information, our method is the first to leverage probabilistic neural network to model the relationship between citations and citation contexts.

[2]http://refseer.ist.psu.edu/

Distributed Representations

Neural network models have been applied to several NLP applications such as parsing (Collobert and Weston 2008; Socher et al. 2013a), sentiment analysis (Socher et al. 2013b), language modeling (Bengio et al. 2003; Mnih and Hinton 2007; Mikolov et al. 2010; Mnih and Teh 2012; Huang et al. 2012a; Mikolov et al. 2013), and machine translation (Gao et al. 2014). In all these works, distributed representations of words were used. The idea of distributed representations is to project words into multi-dimensional continuous-valued vectors. The advantage of using such representations is that they can encode both semantics and syntax of words, thus bridging the gaps between similar words.

In this work, we propose to use distributed representations for a different application – citation recommendation. The representations of words and documents are learnt simultaneously from citation context and cited document pairs.

Citation Recommendation Model

Problem Definition

Assume that the input query is one or two sentences in response to which users want the system to recommend citations. We propose to model the citation context given the cited paper $p(c|d)$. Given a dataset of training samples with $|C|$ pairs of citation context c_t and cited document d_t, the objective is to maximize the log-likelihood:

$$\text{Maximize} \qquad \sum_{t=1}^{|C|} \log p(c_t|d_t) \tag{1}$$

where $p(c_t|d_t)$ is the probability of using citation context c_t to describe the given document d_t. By assuming that words in the citation contexts are mutually conditional independent, $p(c_t|d_t)$ can be calculated by the product of the probability of using word w to describe the given document d_t:

$$p(c_t|d_t) = p(w_{t_1}, \cdots, w_{t_{|c_t|}}|d_t) = \prod_{i=1}^{|c_t|} p(w_{t_i}|d_t) \tag{2}$$

where $|c_t|$ is the number of words in the citation context c_t. The objective function can be written as:

$$\text{Maximize} \qquad \sum_{t=1}^{|C|} \sum_{i=1}^{|c_t|} \log p(w_{t_i}|d_t) \tag{3}$$

After learning the probability $p(w_{t_i}|d_t)$, we apply Bayes' rule to get the citation probability for citing a document d_t given a word w_{t_i}: $p(d_t|w_{t_i}) = \frac{p(w_{t_i}|d_t)p(d_t)}{p(w_{t_i})}$ where $p(d_t)$ and $p(w_{t_i})$ can be calculated by observing the whole dataset.

Neural Probabilistic Model

In a neural probabilistic model, the conditional probability $p(w|d)$ can be defined using a softmax function:

$$p_\theta(w|d) = \frac{\exp\left(s_\theta(w,d)\right)}{\sum_{i=1}^{|V|}\exp\left(s_\theta(w_i,d)\right)} \quad (4)$$

where $|V|$ is the size of the vocabulary that consists of all words appearing in the citation contexts, and $s_\theta(w,d)$ is a neural network based scoring function with parameter θ.

We assume that the neural network parameter θ will project each word w into an n-dimensional continuous-valued vector v_w, and each cited document d into a n-dimensional continuous-valued vector v_d. The scoring function $s_\theta(w,d)$ is defined as:

$$s_\theta(w,d) = f({v_w}^\top v_d) \quad (5)$$

where $f(x) = \frac{1}{1+\exp(-x)}$ is a logistic function that rescales the inner product of the word representation v_w and the document representation v_d to $[0,1]$.

Computing the gradient of Equ. 4 is relatively expensive because it is proportional to the size of the vocabulary in the training set. In order to build an efficient model, we adapt two sampling methods to approximate the learning process: 1) negative sampling (Mikolov et al. 2013) and 2) noise-contrastive estimation (Gutmann and Hyvärinen 2010). Negative sampling is used to learn the distributed representations of words using the surrounding words. Noise-contrastive estimation is used to learn the distributed representations of both words and cited documents, and to estimate the probability of word w appears in the citation context given the cited paper d using the learnt representations.

Word Representation Learning

We use the negative sampling method proposed in skip-gram model (Mikolov et al. 2013). Given a pair of citation context c and cited document d, the skip-gram model will go through all the words in the citation context using a sliding window. For each word w_m that appears in citation context c, words that appear within M words before/after w_m are treated as positive samples. Suppose w_p is one of the positive sample for w_m. The training objective is defined as:

$$\ell_m(\theta) = \log s_\theta(w_p, w_m) + \sum_{i=1}^{k}\log\left(1 - s_\theta(w_{n_i}, w_m)\right) \quad (6)$$

where w_{n_i} is a negative sample randomly generated by noise distribution $p_n(\cdot)$, and k is the number of negative samples. The training objective is to learn the word representations that maximize Equ. 6.

The skip-gram model is able to learn high quality representations of words. However, it is not designed to estimate the conditional probability $p(w_p|w_m)$. In order to estimate the probability $p(w|d)$, we use noise-contrastive estimation to model the relationship between words and documents.

Document Representation Learning

Given a pair of citation context c and cited document d, we assume that each word w that appears in the context c and the cited document d meets the real data distribution $p_r(w|d)$. Any other random words are noise data generated from a noise distribution $p_n(w)$. We also assume that the size of noise data is k times of the real data size. Thus the posterior probabilities of a pair of word w and document d come from real/noise data distribution are:

$$p(1|w,d) = \frac{p_r(w|d)}{p_r(w|d) + kp_n(w)} \quad (7)$$

$$p(0|w,d) = 1 - p(1|w,d). \quad (8)$$

where $p(1|w,d)$ denotes the probability that w is generated from real data distribution $p_r(w|d)$, while $p(0|w,d)$ denotes the probability that w is generated by noise distribution.

Since we want to fit the neural probabilistic model $p_\theta(d|w)$ to the real distribution $p_r(d|w)$, we rewrite Equ. 7 using neural probabilistic model:

$$p(1|w,d,\theta) = \frac{p_\theta(w|d)}{p_\theta(w|d) + kp_n(w)}. \quad (9)$$

Note that the likelihood of binary classification is a Bernoulli distribution. In our scenario, given each word w_{t_i} that appears in the citation context ($i = 1, 2, \cdots, m_t$ is the index of words in context c_t), and the cited document d_t, we compute its contribution to the log-likelihood along with k randomly generated noise words $w_{n_1}, \ldots, w_{n_k}$ using the following objective function:

$$\ell_t(\theta) = \log\frac{p_\theta(w_{t_i}|d_t)}{p_\theta(w_{t_i}|d_t) + kp_n(w_{t_i})} + \sum_{i=1}^{k}\left[\log\frac{kp_n(w_{n_i})}{p_\theta(w_{n_i}|d_t) + kp_n(w_{n_i})}\right] \quad (10)$$

The training objective is to learn both the word representations and document representations that maximize Equ. 10. Noise-contrastive estimation(Gutmann and Hyvärinen 2010) treats the normalization constraint of p_θ to be a constant of $Z(s)$, so that Equ. 4 can be written as:

$$p_\theta(w|d) = \exp(s_\theta(w,d)) \cdot Z(w). \quad (11)$$

The parameters of the neural network are θ and $Z(w)$, where θ is the projection function that maps words and documents in to the n-dimensional space. When learning the model, the document representations and word representations will be tuned simultaneously.

Training

Fig. 2 shows the architecture of our proposed neural network model. At the bottom of the figure, we show an example that demonstrates how one pair of citation context and cited document is processed by the neural network. The size of the sliding window is 5 ($M = 2$), and word w_m is at the middle of the sliding window. Words in the sliding window around w_m are positive samples (w_p) and tuned by the left part of the neural network along with word w_m. The right part of the neural network will take the word w_m and the cited document d as inputs to tune both document representation matrix and word representation matrix to optimize the probability $p(w_m|d)$. In our experiment, we learn the parameters of the neural network with stochastic gradient descent.

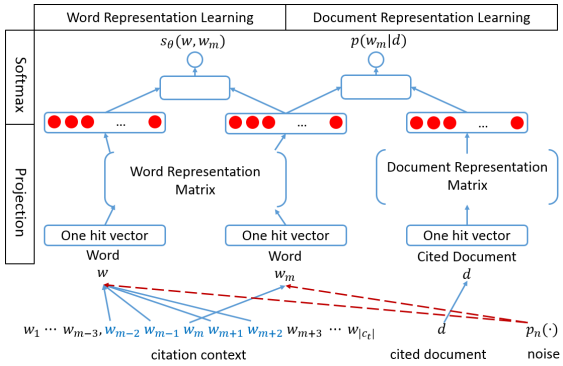

Figure 2: Neural network architecture for word and document representation learning.

Citation Recommendation

Using the fine-tuned word and document representations, we can get the normalized probability distribution $p(w|d)$ with Equ. 11. The table of $p(d|w)$ is pre-calculated using Bayes' rule and stored as an inverted index.

Given a query $q = [w_1, w_2, \cdots, w_{|q|}]$, the task is to recommend a list of document $R = [d_1, d_2, \cdots, d_N]$ that need to be cited. We go through all words in the query and assign the score for each document d_i using:

$$p(d_i|q) = \sum_{j=1}^{|q|} p(d_i|w_j)p(w_j|q) \tag{12}$$

where $p(w_j|q)$ describes the probability that w_j needs a citation. We use the term-frequency-inverse-context-frequency (TF-ICF) to measure $p(w_j|q)$. Given a query q, term-frequency TF is defined as the number of times a word w appears in query q. $\text{ICF}_w = \log \frac{|C|}{|\{c|c \in C, w \in c\}|}$, where C is the set of all of citation contexts, and $|\{c|c \in C, w \in c\}|$ indicates the number of citation contexts that contain w.

Complexity Analysis

Suppose that the training sample size is $|C|$, the average number of words in each citation context is $\overline{|c|}$, n is the dimension of the word and document representation vector, $2M + 1$ is the size of the sliding window, and k_w, k_d are the numbers of negative/noise samples used for learning word and document representation respectively.

For word representation learning, computing the gradient for one positive sample with k_w negative samples takes $O(k_w n)$. For document representation learning, the complexity of the gradient of Equ. 10 is $O(k_d n)$. Suppose that the gradient descent algorithm takes I iterations until converges, the training complexity is $O\left(I|C| \cdot \overline{|c|}(2Mk_w + k_d)n\right)$.

For testing, we will preprocess the learnt words and documents representations to build an inverted index of $p(d|w)$. The computation cost for preprocessing is $O(|V||D|n)$, where $|V|$ is the number of words in the vocabulary and $|D|$ is the number of cited documents in the training set. After the inverted index is built, the recommendation complexity for one query q is $O(|D| \cdot |q|)$, where $|q|$ is the number of words in the query. Note that when building the inverted index, word w with $p(d|w)$ smaller than 0.01% will be filtered out since it makes little contribution for recommendation.

Experiments

Data and Metrics

A snapshot of CiteSeer paper and citation database was obtained at Oct. 2013. The dataset is split into two parts: (1) papers crawled before 2011 (included) as the training set and (2) papers crawled after 2011 as the testing set. Citations are extracted along with their citation contexts. One citation context consists of the sentence where a citation appears, as well as the sentences that appear before and after. As a result, the training set contains $|C|$=8,992,476 pairs of citation contexts and citations and the testing set contains 1,628,698 pairs. For text normalization, rare words that appear less than 5 times are filtered out and we did not distinguish between uppercase/lowercase words. The size of the vocabulary is $|V|$=281,817. The number of unique documents that have been cited in the training set is $|D|$=329,365.

In all experiments, we use the citation contexts and cited papers extracted from the test set as ground truth. The number of recommendations is limited to 10 for each query. We reported the results of standard measures for information retrieval and recommendation (Mean average precision (MAP), Recall, Mean Reciprocal Rank (MRR), and normalized Discounted Cumulative Gain (nDCG)) on the test set.

Baselines and Parameter Settings

- **Cite-PLSA-LDA** (**CP-LDA**) (Kataria, Mitra, and Bhatia 2010): We use the original implementation provided by the authors. The number of topic is set to 600.
- **Restricted Boltzmann Machine** (**RBM**) (Tang and Zhang 2009): We trained a two layer RBM as suggested by the authors. We set the size of the hidden layer to 600.
- **Citation Translation Model** (**CTM**) (Huang et al. 2012b): We use the GIZA++ toolkit [3] to learn a citation translation model. The training iteration is set to 20.
- **Word2vec Model**(**W2V**) (Mikolov et al. 2013): We use word2vec model to learn both word and document representations. Cited documents are regarded as "words" (one document uses a unique token when cited by different papers). The dimension of the word and document representation vectors is set to $n = 600$.
- **Neural Probabilistic Model** (**NPM**): Our proposed model. The dimension of the word and document representation vectors is set to $n = 600$. For negative sampling, we set the number of negative samples to $k = 10$. For noise-contrastive estimation, we set the number of noise samples to $k = 1000$.

For window size used in learning word representation, we follow the word2vec paper by fixing $M = 5$.

[3] http://code.google.com/p/giza-pp/

Baseline Comparison

In Fig. 3 and Table 1, we show the performances of citation recommendation on the whole CiteSeer dataset.

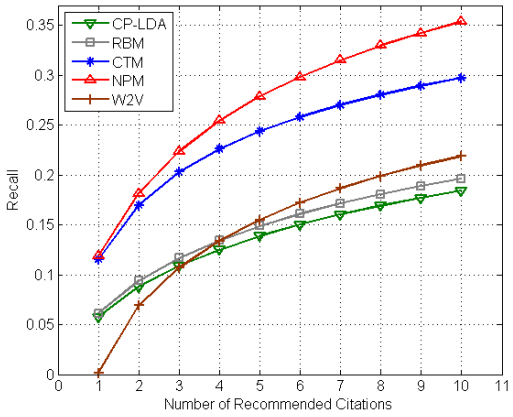

Figure 3: Recall as the number of recommended citations ranges from 1 to 10.

Model	MRR	MAP	nDCG
CP-LDA	0.0916	0.0912	0.1288
RBM	0.0997	0.0982	0.1476
CTM	0.1687	0.1681	0.2261
W2V	0.0662	0.0663	0.1356
NPM	**0.1843***	**0.1835***	**0.2566***

Table 1: MRR, MAP, nDCG scores for top 10 recommendations; * indicates when NPM better than CTM is statistically significant ($p < 0.001$).

We observe that NPM outperforms all other baselines on every evaluation metric. The proposed model improves the overall recommendation with a 5% gain on Recall@10 and 2% gain on MAP compared to the second best model CTM. The recall curve of the proposed model is consistently above all the other baseline methods. This shows that the improvement of our proposed model is very robust. NPM also has a roughly 2% gain on MRR and a 3% gain on nDCG compared to the second best model CTM. This indicates that our proposed model generates better ranked recommendation lists compared to the baseline methods. We also performed a Fisher's randomization test and a Student's t-test which show that our proposed model outperformed all baselines with p-value $p < 0.001$. The significant tests show that the improvements are statistically significant.

In Fig. 4, we show the performances of different models with respect to papers' citation counts (i.e cited frequency). According to the citation counts of the cited documents, we split the test set into four intervals: <100, 100∼500, 500∼1000, and >1000. For each interval, we plot the number of test data that are correctly recommended by each model. From the figure, we observe that compared to other models, our proposed model is particularly good at recommending papers that are not frequently cited (less than 100 citations). When it comes to papers that are well cited, all methods have roughly similar performances. This result indicates that our model can learn a good representation for documents, even with a small number (less than 100) of training data. Other baseline methods fail to model the relationship between words and documents if there are not enough learning samples.

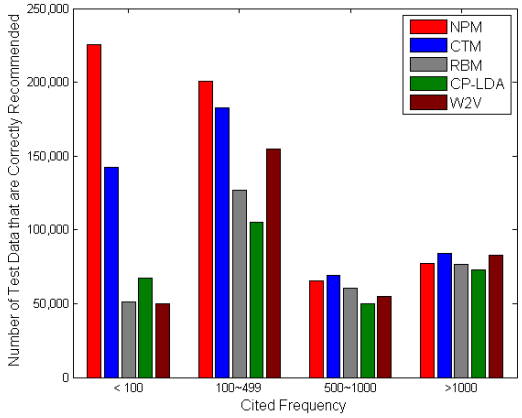

Figure 4: Performance versus papers' cited frequency.

Recommendation Examples To better understand why our proposed model outperforms baselines, in Fig. 5 we show an example of top 5 recommended citations with different models. The query is a citation context extracted from paper *Fast computation of simrank for static and dynamic information networks* [4]:

> There is a lot of research work on static information network analysis, including..., and node ranking[*, *]...

Papers that are not correctly recommended are marked with symbol "[X]", while the correct ones are marked with "[O]".

From Fig. 5, we observe that only the proposed model successfully recommended the 2 correct citations. CTM, RBM and CP-LDA missed correct recommendations because they failed to capture the semantics of "node ranking". W2V correctly recommended the second citation. The ground truth citations are *The PageRank citation ranking*[5] and *Authoritative sources in a hyperlinked environment*[6], which are usually cited with the word "pagerank." However, when it comes to "node ranking," only the proposed model captures the semantic similarity between "node ranking" and "pagerank," because using word representation, $Vector(\text{node}) + Vector(\text{ranking})$ results in a vector that is very close to $Vector(\text{pagerank})$. The proposed method succeeded in building the semantic bridge between the words "node ranking" and the two papers that were cited.

With respect to the topic-based methods, CP-LDA and RBM, we observed that both methods inferred the topic of

[4]Li, Cuiping, et al. *"Fast computation of simrank for static and dynamic information networks."* In Proceedings of EDBT, 2010.

[5]Page, Lawrence, et al. *"The PageRank citation ranking: Bringing order to the web."* (1999).

[6]Kleinberg, Jon M. *"Authoritative sources in a hyperlinked environment."* Journal of the ACM (JACM) 46.5 (1999): 604-632.

Query and Ground Truth	NPM	CTM	RBM	CP-LDA	W2V
There is a lot of research work on static information network analysis, including ... , and node ranking [1, 2], ... [1] The PageRank citation ranking [2] Authoritative sources in a hyperlinked environment	[O] Authoritative sources in a hyperlinked environment [X] The anatomy of a large-scale hypertextual Web search engine [X] Topic-sensitive PageRank [X] Improved Algorithms for Topic Distillation in Hyperlinked Environments [O] The PageRank citation ranking	[X] Modern Information Retrieval [X] An Efficient Boosting Algorithm for Combining Preferences [X] Optimizing search engines using clickthrough data [X] Learning to rank using gradient descent [X] Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications	[X] A survey of active network research [X] Statistical mechanics of complex networks [X] How to model an internetwork [X] Network information flow [X] SNORT - lightweight intrusion detection for networks	[X] Network Information Flow [X] Linear network coding [X] Polynomial Time Algorithms for Network Information Flow [X] An algebraic approach to network coding [X] Network Coding for Large Scale Content Distribution	[X] Modern Information Retrieval [X] The anatomy of a large-scale hypertextual Web search engine [X] Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications [X] A Scalable Content-addressable Network [O] Authoritative sources in a hyperlinked environment

Figure 5: A comparison of ground truth with the top 5 recommended citation lists.

the query as "network," which is a much higher level concept of "node ranking." The papers recommended by CTM are more targeted to "ranking" but not "node ranking." W2V model failed to link the semantics of the PageRank paper with "node ranking" and "network analysis." Due to space limitation, we only show one example that illustrates the strength of our proposed model, and reveals the weakness of the baseline models. In the testing set, there are more samples that support this finding.

Model Parameters

Noise Distribution p_n and Number of Samples k Both negative sampling and noise-contrastive estimation use a noise distribution $p_n(\cdot)$ to randomly generate negative or noise samples. Following the word2vec paper (Mikolov et al. 2013), we use the frequency-based distribution to generate negative samples for word representation learning. The number of negative samples is fixed to 10. For document representation learning, we also have the two free parameters: the choice of noise distribution $p_n(\cdot)$, and the number of noise samples k for noise-contrastive estimation.

We chose a uniform distribution and frequency-based distribution as the generators for the noise samples. The uniform distribution assumes that each word will be randomly selected as a noise sample with equal probability, while the frequency-based distribution assumes that noise words will be randomly selected according to the frequency they appear in the training set. For the number of noise samples k, we trained models with $100, 300, 500$ and $1,000$ samples per data point for each noise distribution.

# of Noise k	Uniform Noise	Frequency-based Noise
100	0.3096	0.2913
300	0.3275	0.3165
500	0.3491	0.3297
1,000	**0.3537**	0.3450

Table 2: Recall@10 versus noise distributions and sample size.

The compared results are shown in Table 2. For the choice of noise distribution, we observe that noise generated by a uniform distribution trained a better model than frequency-based noise. With respect to the number of noise samples k, we find that as k increases, the performance for both noise distributions generally increases. We would expect that if we set k larger, better performance will also be achieved. However, the training time will increase linearly with k. For a trade-off between performance and model training time, we choose k=1,000 to be our proposed model for comparison with other baselines.

Dimension n Another important parameter for the proposed model is the dimension of the distributed representations of words and documents. Table 3 shows the performance changing when we set different dimensions for word and document representations. The overall performance generally increases with the dimension n. Although better results may be achieved by with a larger dimension n, the training time also increases linearly as n. As such, we choose $n = 600$ as our best model.

dimension n	MRR	MAP	nDCG	Recall@10
100	0.1681	0.1692	0.2297	0.3118
300	0.1709	0.1702	0.2372	0.3247
600	**0.1843**	**0.1835**	**0.2566**	**0.3537**

Table 3: Recall@10 versus word and document representation dimension.

Conclusion and Future Work

We used the distributed representations of words and documents to build a neural network based citation recommendation model. The proposed model then learns the word and document representations to calculate the probability of citing a document given a citation context. A comparative study on a snapshot of CiteSeer dataset with existing state-of-the-art methods showed that the proposed model significantly improves the quality of context-based citation recommendation.

Since only neural probabilistic models were considered for context-based local recommendations, one could explore a combined model that takes the entire content of a manuscript as an input for both local and global citation recommendations.

Acknowledgments

We gratefully acknowledge partial support from the NSF.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

Bethard, S., and Jurafsky, D. 2010. Who should i cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, 609–618. ACM.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 160–167. ACM.

Gao, J.; He, X.; Yih, W.-t.; and Deng, L. 2014. Learning semantic representations for the phrase translation model. In *The 52nd Annual Meeting of the Association for Computational Linguistics*. ACL.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. Citeseer: an automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, DL '98, 89–98. New York, NY, USA: ACM.

Gutmann, M., and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 297–304.

He, Q.; Pei, J.; Kifer, D.; Mitra, P.; and Giles, L. 2010. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, 421–430. ACM.

He, Q.; Kifer, D.; Pei, J.; Mitra, P.; and Giles, C. L. 2011. Citation recommendation without author supervision. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, 755–764. ACM.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012a. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 873–882. Association for Computational Linguistics.

Huang, W.; Kataria, S.; Caragea, C.; Mitra, P.; Giles, C. L.; and Rokach, L. 2012b. Recommending citations: Translating papers into references. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 1910–1914. ACM.

Huang, W.; Wu, Z.; Mitra, P.; and Giles, C. L. 2014. Refseer: A citation recommendation system. In *Proceedings of 14th ACM/IEEE Joint Conference on Digital Libraries*.

Kataria, S.; Mitra, P.; and Bhatia, S. 2010. Utilizing context in generative bayesian models for linked corpus. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 1340–1345.

Kucuktunc, O.; Kaya, K.; Saule, E.; and Catalyurek, U. V. 2012. Fast recommendation on bibliographic networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, 480–487. IEEE Computer Society.

Lu, Y.; He, J.; Shan, D.; and Yan, H. 2011. Recommending citations with translation model. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, 2017–2020. New York, NY, USA: ACM.

McNee, S. M.; Albert, I.; Cosley, D.; Gopalkrishnan, P.; Lam, S. K.; Rashid, A. M.; Konstan, J. A.; and Riedl, J. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, 116–125. ACM.

Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, 3111–3119.

Mnih, A., and Hinton, G. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, 641–648. ACM.

Mnih, A., and Teh, Y. W. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, 1751–1758.

Nallapati, R. M.; Ahmed, A.; Xing, E. P.; and Cohen, W. W. 2008. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD*, KDD '08, 542–550. New York, NY, USA: ACM.

Ren, X.; Liu, J.; Yu, X.; Khandelwal, U.; Gu, Q.; Wang, L.; and Han, J. 2014. Cluscite: Effective citation recommendation by information network-based clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 821–830. ACM.

Socher, R.; Bauer, J.; Manning, C. D.; and Ng, A. Y. 2013a. Parsing with compositional vector grammars. In *ACL*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Association for Computational Linguistics.

Strohman, T.; Croft, W. B.; and Jensen, D. 2007. Recommending citations for academic papers. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, 705–706. ACM.

Tang, J., and Zhang, J. 2009. A discriminative approach to topic-based citation recommendation. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, 572–579. Springer-Verlag.