Dual Attention Model for Citation Recommendation

Yang Zhang
Graduate School of Informatics
Kyoto University
zhang.yang.33z@st.kyoto-u.ac.jp

Qiang Ma
Graduate School of Informatics
Kyoto University
qiang@i.kyoto-u.ac.jp

Abstract

Based on an exponentially increasing number of academic articles, discovering and citing comprehensive and appropriate resources has become a non-trivial task. Conventional citation recommender methods suffer from severe information loss. For example, they do not consider the section of the paper that the user is writing and for which they need to find a citation, the relatedness between the words in the local context (the text span that describes a citation), or the importance on each word from the local context. These shortcomings make such methods insufficient for recommending adequate citations to academic manuscripts. In this study, we propose a novel embedding-based neural network called "dual attention model for citation recommendation (DACR)" to recommend citations during manuscript preparation. Our method adapts embedding of three semantic information: words in the local context, structural contexts [1], and the section on which a user is working. A neural network model is designed to maximize the similarity between the embedding of the three input (local context words, section and structural contexts) and the target citation appearing in the context. The core of the neural network model is composed of self-attention and additive attention, where the former aims to capture the relatedness between the contextual words and structural context, and the latter aims to learn the importance of them. The experiments on real-world datasets demonstrate the effectiveness of the proposed approach.

1 Introduction

When writing an academic paper, one of the most frequent questions considered is: "Which paper should I cite at this place?" Based on the massive number of papers being published, it is impossible for a researcher to read every article that might be relevant to their study. Thus, recommending a handful of useful citations based on the contents of a working draft can significantly alleviate the burden of writing a paper. An example of the application scenario is demonstrated in Figure 1.

Currently, many scholars rely on "keyword searches" on search engines, such as Google Scholar [2] and DBLP [3]. However, keyword-based systems often generate unsatisfying results, because query words may not convey adequate information to reflect the context that needs to be supported (Jia and Saule, 2017; Jia and Saule, 2018). Researchers in various fields have proposed various methods to solve this problem. For example, studies in (McNee et al., 2002; Gori and Pucci, 2006; Caragea et al., 2013; Küçüktunç et al., 2013; Jia and Saule, 2018) considered recommendations based on a collection of seed papers, and (Alzoghbi et al., 2015; Li et al., 2018) proposed methods using meta-data, such as authorship information, titles, abstracts, keyword lists, and publication years. However, when applying such methods to real-world paper-writing tasks, there is a lack of consideration for the local context of a citation within a draft, which can potentially lead to suboptimal results. Context-based recommendations adopt a more practical concept that generates potential citations for an input context (He et al., 2010; He

[1] Cited papers other than the target citation in a citing paper, which are defined in (Zhang and Ma, 2020) and Definition 2 in Section 3.1 of this paper.

[2] https://scholar.google.com/

[3] https://dblp.uni-trier.de/

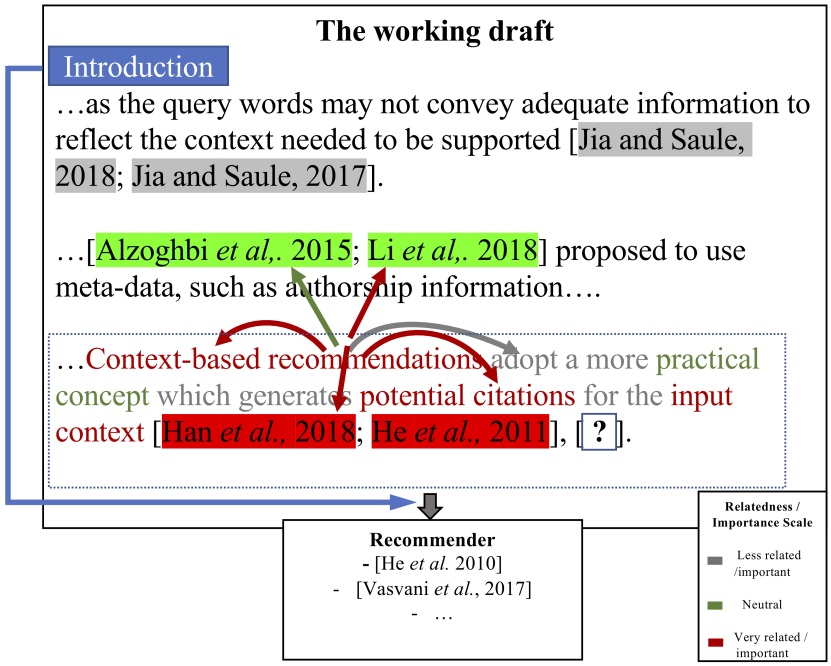

Figure 1: Concept of dual attention model for citation recommendation (DACR)

et al., 2011). Based on the context-based methodology, the HyperDoc2Vec (Han et al., 2018) proposed an embedding framework which further considers embedding with information of citation link between the local context in a citing paper and the content in a cited paper. Our previous study (Zhang and Ma, 2020) adapted the structural contexts in addition to citation link to further improve the recommendation performances. Context-based approaches could be potentially applicable in the real-world paper-writing process.

However, the studies mentioned above still fail to take into consideration a number of essential characteristics of academic papers, which limits their usefulness.

1. Scientific papers tend to follow the established "IMRaD" format (introduction, methods, results and discussion, and conclusions) (Mack, 2014), where each section of an article has a specific purpose. For example, the introduction section defines the topic of the paper in a broader context, the method section includes information on how the results were produced, and the results and discussion section presents the results. Therefore, citations used in each section should comply with the specific purpose of that section. For example, citations in the introduction section should support the main concepts of the paper, citations in the methods section should provide technical details, and citations in the results and discussion section should aim to compare results to those of other works. Therefore, recommendations of suitable citations for a given context should also consider the purpose of the corresponding section.

2. Certain words and cited articles in a paper are much more closely related than other words and articles in the same paper. Capturing these interactions is essential for understanding a paper. For example, in Figure 1, the word "recommendation" is closely related to the words "context-based," "citations," and "context," but has a weak relationship with the words "adopt," "more," and "input." Additionally, a given word may have strong relatedness with some citations that appear in the paper. For example, the word "recommendation" has strong relatedness to citations "(Li et al., 2018)" and "(Han et al., 2018)" because both of these citations focus on recommendation algorithms.

3. Not every word or cited article has the same importance within a given paper. Important words and cited articles are more informative with respect to the topic of a paper. For example, in Figure 1, the words "context-based," "recommendations," "citations," and "context" are more informative than

the words “adopt,” “more,” or “generates.” The citation, “(Han et al., 2018),” may be more essential than “(Jia and Saule, 2018)” because the former is related to context-based recommendations, while the latter is related to a different approach.

Adequate recommendations of citations for a manuscript should capture the relatedness and importance of words and cited articles in the context which needs citations, as well as the purpose of the section on which the writer is currently working. To this end, we propose a novel embedding-based neural network called dual attention model for citation recommendation (DACR) that is designed to capture the relatedness and importance of words in the context which needs citations and structural contexts in the manuscript, as well as the section for which the user is working. The core of the proposed neural network is composed of two attention mechanisms, namely self-attention and additive attention. The former captures the relatedness between contextual words and structural contexts, and the latter learns the importance of contextual words and structural contexts. Additionally, the proposed model embeds sections into an embedding space and utilizes the embedded sections as additional features for recommendation tasks.

2 Related Work

2.1 Document Embedding

Document embedding refers to the representation of words and documents as continuous vectors. Word2Vec (Mikolov et al., 2013a) was proposed as a shallow neural network for learning word vectors from texts while preserving word similarities. Doc2Vec (Le and Mikolov, 2014) is an extension of Word2Vec for embedding documents with content words. However, these two methods generally treat documents as “plain texts,” meaning that when they are applied to scholarly articles. This can lead to some essential information being lost (for example, citations and metadata in scientific papers), which in turn can lead to suboptimal recommendation results. Some more recent studies have attempted to remedy this issue. HyperDoc2Vec (Han et al., 2018) is a fine-tuning model for embedding additional citation relations. DocCit2Vec (Zhang and Ma, 2020) proposed by our previous work considers both structural contexts and citation relations. Regardless, some vital information is still not considered, such as the semantic of section headers and the relatedness and importance of word in the context requiring support of citations, which are included in this study.

2.2 Citation Recommendation

Citation recommendation refers to the task of finding relevant documents based on an input query. The query could be a collection of seed papers (McNee et al., 2002; Gori and Pucci, 2006; Caragea et al., 2013; Küçüktunç et al., 2013; Jia and Saule, 2017), and the recommendations are then generated via collaborative filtering (McNee et al., 2002; Caragea et al., 2013) or PageRank-based methods (Gori and Pucci, 2006; Küçüktunç et al., 2013; Jia and Saule, 2017). Some studies (Alzoghbi et al., 2015; Li et al., 2018) have proposed using meta-data, such as titles, abstracts, keyword lists, and publication years as query information. However, in real-world applications, when providing support for writing manuscripts, these techniques lack practicability. Context-based methods (He et al., 2010; He et al., 2011; Han et al., 2018; Zhang and Ma, 2020) use a passage requiring support as a query to determine the most relevant papers, which can potentially enhance the paper-writing process. However, such methods may suffer from information loss because they do not consider sections within papers or the relative importance and relatedness of local context words.

2.3 Attention Mechanisms

Attention mechanism is commonly applied in the field of computer vision (Tang et al., 2014) and detects important parts of an image to improve prediction accuracy. This mechanism has also been adopted in the recent researches in text mining. For example, (Ling et al., 2015) extended Word2Vec with a simple attention mechanism to improve word classification performance. Google’s BERT algorithm (Devlin et al., 2019) uses multi-head attention and provides excellent performance for several natural language

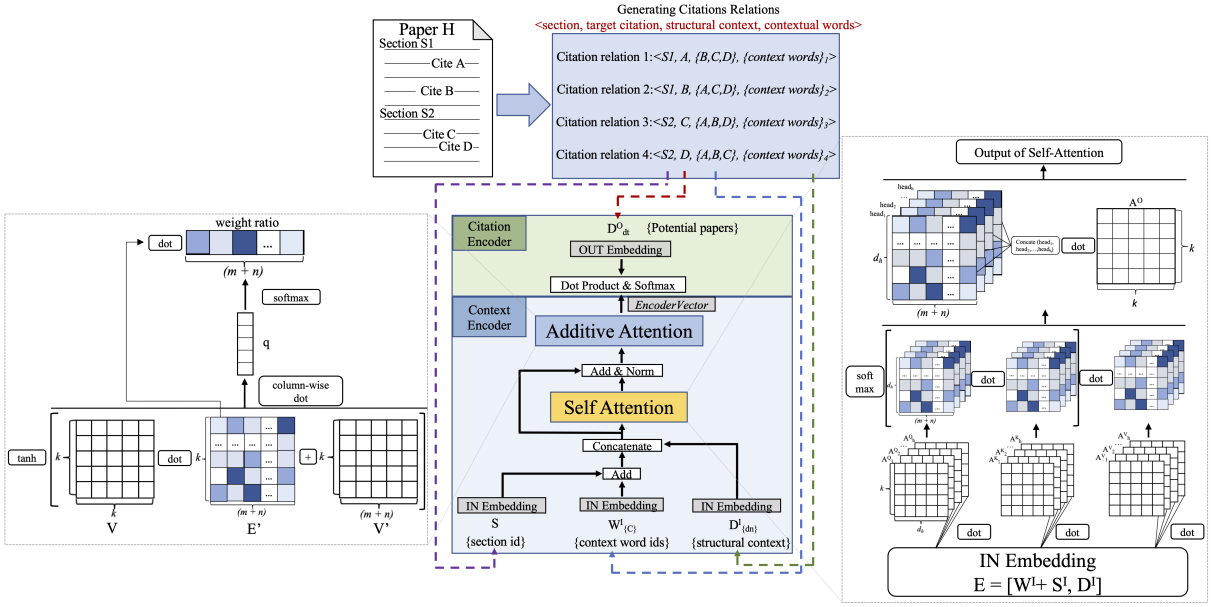

Figure 2: Architecture of DACR

processing tasks. The method introduced in (Wu et al., 2019) uses self-attention and additive attention to improve recommendation accuracy for news sources.

3 Preliminary

3.1 Notations and Definitions

Academic papers can be treated as a type of hyper-document, in which citations are equivalent to hyperlinks. Based on paper modeling with citations (Han et al., 2018) and modeling of citations with structural contexts (Zhang and Ma, 2020), we introduce a novel modeling with citations, structural contexts, and sections.

Definition 1 (Academic Paper)**.** *Let $w \in W$ represent a word from a vocabulary, W, where $s \in S$ represents a section from a section header collection, S, and $d \in D$ represents a document ID (paper DOI) from an ID collection, D. The textual information of a paper, H, is represented as a sequence of words, sections, and IDs of cited documents (i.e., $\hat{W} \cup \hat{S} \cup \hat{D}$, where $\hat{W} \subseteq W$, $\hat{S} \subseteq S$, and $\hat{D} \subseteq D$).*

Definition 2 (Citation Relationships)**.** *The citation relationships, $\mathcal{C}$, (see Figure 2) in a paper, H, are expressed by a tuple, $\langle s, d_t, D_n, C \rangle$, where $d_t \in \hat{D}$ represents a target citation, $\hat{D}$ represents the id of all the cited documents from H, $C \subseteq \hat{W}$ is the local context surrounding d_t, and $s \in \hat{S}$ is the title of the section in which the contextual words appear. If other citations exist within the same manuscript, then they are defined as "structural contexts" and denoted by D_n, where $\{d_n | d_n \in \hat{D}, d_n \neq d_t\}$.*

3.2 Problem Definition

Embedding matrices are denoted as $\mathbf{D} \in \mathbb{R}^{k \times |D|}$ for documents, $\mathbf{W} \in \mathbb{R}^{k \times |W|}$ for words, and $\mathbf{S} \in \mathbb{R}^{k \times |S|}$ for sections. The *i*-th column of $\mathbf{D}$, denoted by $\mathbf{d_i}$, is a k-dimensional vector representing document d_i. Additionally, the *j*-th column of $\mathbf{W}$ is a *k*-dimensional vector for word w_j, and the *s*-th column of $\mathbf{S}$ is a *k*-dimensional vector for section s.

The proposed model initializes two embedding matrices (IN and OUT) for documents (i.e., $\mathbf{D^I}$ and $\mathbf{D^O}$), a word embedding matrix, $\mathbf{W^I}$, and a section embedding matrix, $\mathbf{S^I}$. A column vector from $\mathbf{D^I}$ represents the role of a document as a structural context and a column vector from $\mathbf{D^O}$ represents the role of a document as a citation (the implementation details of the experiment in Section 5.4 explain this in more detail). The word embedding matrix, $\mathbf{W^I}$, and section embedding matrix, $\mathbf{S^I}$, are initialized for all words of the word vocabulary and all sections of the section header collection.

The goal of this model is to optimize the following objective function:

$$\max_{\mathbf{D^I},\mathbf{D^O},\mathbf{W^I},\mathbf{S^I}} \frac{1}{|\mathcal{C}|} \sum_{\langle s,d_t,D_n,C\rangle\in\mathcal{C}} \log P(d_t|s,D_n,C). \quad (1)$$

4 Dual Attention Model for Citation Recommendation

An overview of the proposed DACR approach is presented in Figure 2. DACR is composed of two main components: a context encoder (Section 4.1) for encoding contextual words, sections, and structural contexts into a fixed-length vector and a citation encoder (Section 4.2) for predicting the probability of a target citation.

4.1 Context Encoder

The context encoder takes three inputs, namely, context words, sections, and structural contexts, from citation relationships. The encoder contains three layers: an embedding layer for converting words and documents (structural contexts) into vectors, a self-attention layer with an Add&Norm sub-layer (Vaswani et al., 2017) for capturing the relatedness between words and structural contexts, and an additive attention layer (Wu et al., 2019) for recognizing the importance of each word and structural context.

4.1.1 IN Embedding, Add and Concatenation layer

The IN embedding layer initially generates three embedding matrices $\mathbf{D^I}$, $\mathbf{W^I}$, and $\mathbf{S^I}$ for the document collection, word vocabulary and the section header collection. For a given citation relationship, the one-hot vectors of structural contexts, context words, and sections are projected with the three embedding matrices, denoted as $\mathbf{D^I}_{\{D_n\}}$, $\mathbf{W^I}_{\{C\}}$, and $\mathbf{S^I}_s$. The projected section vectors are then added to the word vectors (each word vector is added to a section vector), and the resultant matrix is denoted as $\mathbf{W'}$. $\mathbf{W'}$ and $\mathbf{D^I}_{\{D_n\}}$ are then concatenated column-wise and form one matrix, i.e., $[\mathbf{w'_1},...,\mathbf{w'_m},\mathbf{d^I_1},...,\mathbf{d^I_n}]$, and denoted as $\mathbf{E}$, where m is the number of input context words and n is the number of input structural contexts.

4.1.2 Self-attention Mechanism with Add&Norm

Self-attention (Vaswani et al., 2017) is utilized to capture the relatedness between input context words and structural contexts. It applies scaled dot-product attention in parallel for a number of heads, to allow the model to jointly consider interactions from different representation sub-spaces at different positions.

The k-dimensional embedding matrix, $\mathbf{E}$, from the last layer is first transposed and projected with three linear projections ($\mathbf{A}_i^Q$,$\mathbf{A}_i^K$, and $\mathbf{A}_i^V$) to a d_h dimensional space, where $d_h = k/h$, $i \in \{1...h\}$, and h denotes the number of heads. The $\mathbf{E}$ matrix is projected h times, and each projection is called a "head". At each projection (i.e., within a "head"), the dot products of the first two projected versions of $\mathbf{E}$ with $\mathbf{A}_i^Q$ and $\mathbf{A}_i^K$ are computed, and divided by $\sqrt{d_h}$. Subsequently, softmax is applied to obtain the resulting weight matrix with dimensions of $(m+n)*(m+n)$, i.e., $softmax(\frac{\mathbf{E}^T\mathbf{A}_i^Q\cdot(\mathbf{E}^T\mathbf{A}_i^K)^T}{\sqrt{d_h}})$, where $(m+n)$ is the total number of input context words and structural contexts. This weight matrix represents the relatedness between the input words and articles. The dot product of the weight matrix and the third projected version of $\mathbf{E}$, i.e., $\mathbf{E}^T\mathbf{A}_i^V$, is computed as the output matrix of the head, denoted as $\mathbf{head}_i$. The h numbers of the output head matrices are concatenated column-wise and projected again with $\mathbf{A}^O$ to yield the final output matrix. The computation procedure is represented as follows:

$$SelfAttention(\mathbf{E}) = Concat(\mathbf{head}_1,...,\mathbf{head}_h)\mathbf{A}^O, \quad (2)$$

$$\mathbf{head}_i = softmax(\frac{\mathbf{E}^T\mathbf{A}_i^Q\cdot(\mathbf{E}^T\mathbf{A}_i^K)^T}{\sqrt{d_h}})\cdot(\mathbf{E}^T\mathbf{A}_i^V), \quad (3)$$

where $\mathbf{A}^O \in \mathbb{R}^{k\times k}$, $\mathbf{A}_i^Q \in \mathbb{R}^{k\times d_h}$, $\mathbf{A}_i^K \in \mathbb{R}^{k\times d_h}$, and $\mathbf{A}_i^V \in \mathbb{R}^{k\times d_h}$ are projection parameters. d_h is the embedding dimension of the heads, h is the number of heads, and $k = d_h \times h$, where k is the dimension of the embedding vectors. The output matrix of the self-attention mechanism is then transposed and added

to the original $\mathbf{E}$ matrix. Next, dropout is applied (Hinton et al., 2012) to avoid over-fitting, and applied with layer normalization (Ba et al., 2016) to facilitate the convergence of the model during training. The final output matrix is denoted as $\mathbf{E}'$.

4.1.3 Additive Attention Mechanism

The additive attention layer (Wu et al., 2019) is utilized to recognize informative contextual words and structural contexts. It takes matrix $\mathbf{E}'$ from the last layer as input, whereby each column represents the vector of a word or document. The weight of each item is computed as follows:

$$\mathbf{Weight} = \mathbf{q}^T \cdot \tanh\left(\mathbf{V} \cdot \mathbf{E}' + \mathbf{V}'\right), \tag{4}$$

where $\mathbf{V} \in \mathbb{R}^{k \times k}$ is the projection parameter matrix, $\mathbf{V}' \in \mathbb{R}^{k \times (n+m)}$ is the bias matrix, and $\mathbf{q}$ (k-dimensional) is a parameter vector. The $\mathbf{Weight}$ vector is a row vector of dimension $(m+n)$, where each column represents the weight of a corresponding word or document. The $\mathbf{Weight}$ vector is applied with the dropout technique to avoid over-fitting.

The output, $\mathbf{EncoderVector}$, is the dot product of the softmaxed $\mathbf{Weight}$ vector and input matrix, $\mathbf{E}'$, where all rows of the embedding vectors are weighted and summed, as illustrated below:

$$\mathbf{EncoderVector} = \mathbf{E}' \cdot softmax(\mathbf{Weight}^T). \tag{5}$$

4.2 Citation Encoder

The citation encoder is designed to predict potential citations by calculating the probability score between an OUT document matrix, $\mathbf{D^O}$, and the $\mathbf{EncoderVector}$ from the context encoder, which is defined as follows:

$$\hat{\mathbf{y}} = \mathbf{EncoderVector}^T \cdot \mathbf{D^O}. \tag{6}$$

The scores are then normalized using the softmax function as follows:

$$\mathbf{p} = softmax(\hat{\mathbf{y}}). \tag{7}$$

4.3 Model Training and Optimization

We adopted a negative sampling training strategy (Mikolov et al., 2013b) to speed up the training process for DACR. In each iteration, it generates a positive sample (correctly cited paper) and n negative samples. Therefore, the calculated probability vector, $\mathbf{p}$, is composed of $[p_{positive}, p_{negative-1}, p_{negative-2}, ..., p_{negative-n}]$. The loss function computes the negative log-likelihood of the probability of a positive sample, as follows:

$$\mathcal{L} = -\log(p_{positive}) + \sum_{i=1}^{n} \log(p_{negative-i}). \tag{8}$$

Stochastic gradient descent (SGD) (Sutskever et al., 2013) is used to optimize the model.

5 Experiments

We evaluated the recommendation performance of our model and five baseline models on two datasets, namely DBLP and ACL Anthology (Han et al., 2018). The recall, mean average precision (MAP), mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG) are reported for a comparison of the models. These values are summarized in Table 2. Additionally, we proved the effectiveness of adding information about sections, relatedness, and importance, as shown in Figure 3.

Table 1: Statistics of the datasets

Overview of the dataset					Number of sections in the dataset								
		All	Train	Test	Generic Section	Abstract	Background	Introduction	Method	Evaluation	Discussion	Conclusions	Unknown
DBLP	No. of Docs	649,114	630,909	18,205	Train	617,402	9,589	452,430	3,226,521	153,737	19,738	435,514	155,777
	No. of Citations	2,874,303	2,770,712	103,591	Test	5,243	155	6,437	25,956	1,312	200	1875	58,975
ACL	No. of Docs	20,408	14,654	1,563	Train	11,725	114	9,973	42,749	4,186	442	9,456	847
	No. of Citations	108,729	79,932	28,797	Test	3,789	33	3,429	12,625	1,587	159	3,186	0

5.1 Dataset Overview

The larger dataset, DBLP (Han et al., 2018), contains 649,114 full paper texts with 2,874,303 citations (approximately five citations per paper) in the field of computer science. The ACL Anthology dataset (Han et al., 2018) is smaller, containing 20,408 texts with 108,729 citations; however, it has a similar number of citations per paper (about five per paper) to the DBLP dataset. We split the datasets into a training dataset, for training the document, word, and section vectors, and test dataset with papers containing more than one citation published in the last few years for recommendation experiments. An experimental overview is provided in Table 1.

5.2 Document Preprocessing

The texts were pre-processed using ParsCit (Councill et al., 2008) to recognize citations and sections. In-text citations were replaced with the corresponding unique document IDs in the dataset. Section headers often have diverse names. For example, many authors name the "methodology" section using customized algorithm names. Therefore, we replaced all section headers with fixed generic section headers using ParsLabel (Luong et al., 2010). Generic headers from ParsLabel are *abstract, background, introduction, method, evaluation, discussions, and conclusions*. If ParsLabel is not able to recognize a section, we label it as "*unknown*." Detailed information for each section is listed in Table 1.

5.3 Implementation and Settings

DACR was developed using PyTorch 1.2.0 (Paszke et al., 2019). In our experiments, word and document embeddings were pre-trained using DocCit2Vec with an embedding size of 100, a window size of 50 (also known as the length of the local context, i.e. 50 words before and after a citation), a negative sampling value of 1000, and 100 iterations (default settings in (Zhang and Ma, 2020)). The word vectors for generic headers, such as "introduction" and "method," were selected as pre-trained vectors for the section headers. DACR was implemented with 5 heads, 100 dimensions for the query vector, and a negative sampling value of 1000. The SGD optimizer was implemented with a learning rate of 0.0001, a batch size of 100, and 100 iterations for the DBLP dataset, or 300 iterations for the ACL Anthology dataset. To avoid over-fitting, we applied a 20% dropout in the two attention layers.

Word2Vec and Doc2Vec were implemented using Gensim 2.3.0 (Řehůřek and Sojka, 2010), and HyperDoc2Vec and DocCit2Vec were developed based on Gensim. All baseline models were initialized with an embedding size of 100, a window size of 50, and default values for the remaining parameters.

5.4 Recommendation Evaluation

We designed three usage cases to simulate real-world scenarios:

- Case 1: In this case, we assumed the manuscript was approaching its completion phase, meaning the writer had already inserted the majority of their citations into the manuscript. Based on the leave-one-out approach, the task was to predict a target citation, by providing the contextual words (50 words before and after the target citation), structural contexts (the other cited papers in the source paper), and section header as input information for DACR.

- Case 2: Here, we assumed that some existing citations were invalid because they were not available in the dataset, i.e., the author had made typographical errors or the manuscript was in an early stage of development. In this case, given a target citation, its local context and section header, we randomly selected structural contexts to predict a target citation. The random selection was implemented using the build-in Python3 *random* function. All case 2 experiments were conducted three times to determine the average results to rule out biases.

Table 2: Citation recommendation results (** statistically significant at 0.01 significance level)

Model	DBLP				ACL			
	Recall@10	**MAP@10**	**MRR@10**	**nDCG@10**	**Recall@10**	**MAP@10**	**MRR@10**	**nDCG@10**
W2V (case 1)	20.47	10.54	10.54	14.71	27.25	13.74	13.74	19.51
W2V (case 2)	20.46	10.55	10.55	14.71	26.54	13.55	13.55	19.19
W2V (case 3)	20.15	10.40	10.40	14.49	26.06	13.21	13.21	18.66
D2V-nc (case 1)	7.90	3.17	3.17	4.96	19.92	9.06	9.06	13.39
D2V-nc (case 2)	7.90	3.17	3.17	4.96	19.89	9.06	9.06	13.38
D2V-nc (case 3)	7.91	3.17	3.17	4.97	19.89	9.07	9.07	13.38
D2V-cac (case 1)	7.91	3.17	3.17	4.97	20.51	9.24	9.24	13.68
D2V-cac (case 2)	7.90	3.17	3.17	4.97	20.29	9.17	9.17	13.58
D2V-cac (case 3)	7.89	3.17	3.17	4.97	20.51	9.24	9.24	13.69
HD2V (case 1)	28.41	14.20	14.20	20.37	37.53	19.64	19.64	27.20
HD2V (case 2)	28.42	14.20	14.20	20.38	36.83	19.62	19.62	27.18
HD2V (case 3)	28.41	14.20	14.20	20.37	36.24	19.32	19.32	26.79
DC2V (case 1)	44.23	21.80	21.80	31.34	36.89	20.44	20.44	27.72
DC2V (case 2)	40.31	20.16	20.16	28.69	33.71	18.47	18.47	25.17
DC2V (case 3)	40.37	19.02	19.02	26.84	31.14	16.97	16.97	23.20
DACR (case 1)	**48.96****	**23.25****	**23.25****	**33.93****	**42.43****	**22.92****	**22.92****	**31.64****
DACR (case 2)	**45.39****	**22.32****	**22.32****	**31.98****	**40.13****	**21.93****	**21.93****	**30.04****
DACR (case 3)	**42.32****	**21.39****	**21.39****	**30.22****	**38.01****	**20.84****	**20.84****	**28.45****

- Case 3: It is assumed that the manuscript was in an early phase of development, where the writer has not inserted any citations or all existing citations are invalid. Only context words and section headers were utilized for the prediction of the target citation (no structural contexts were used).

To conduct recommendation via DACR, an encoder vector was initially inferred using the trained model with inputs of cases 1, 2, and 3, and then, the OUT document vectors were ranked based on dot products.

Five baseline models were adapted for comparison with DACR. As the baseline models do not explicitly consider section information, information on the section headers were neglected in the inputs.

1. **Citations as words via Word2Vec (W2V)** This method was presented in (Berger et al., 2017), where all citations were treated as special words. The recommendation of documents was defined as ranking OUT word vectors of documents relative to the averaged IN vectors of context words, and structural contexts via dot products. The word vectors were trained using the Word2Vec CBOW algorithm.

2. **Citations as words via Doc2Vec (D2V-nc)**(Berger et al., 2017). The citations were removed in this method, and the recommendations were made by ranking the IN document vectors via cosine similarity relative to the vector inferred from the learnt model by taking context words and structural contexts as input (this method results in better performance than the dot product). The word and document vectors were trained using Doc2Vec PV-DM.

3. **Citations as content via Doc2Vec (D2V-cac)** (Han et al., 2018). In this method, all context words around a citation were copied into the cited document as supplemental information. The recommendations were made based on cosine similarity between the IN document vectors and inferred vector from the learnt model. The vectors were trained using Doc2Vec PV-DM.

4. **Citations as links via HyperDoc2Vec (HD2V)** (Han et al., 2018). In this method, citations were treated as links pointing to target documents. The recommendations were made by ranking OUT document vectors relative to the averaged IN vectors of input contextual words based on dot products. The embedding vectors were pre-trained by Doc2Vec PV-DM using default settings.

5. **Citations as links with structural contexts via DocCit2Vec (DC2V)** (Zhang and Ma, 2020). The recommendations were made by ranking OUT document vectors relative to the averaged IN vectors of input contextual words and structural contexts based on dot products. The embedding vectors were pre-trained by Doc2Vec PV-DM with default settings.

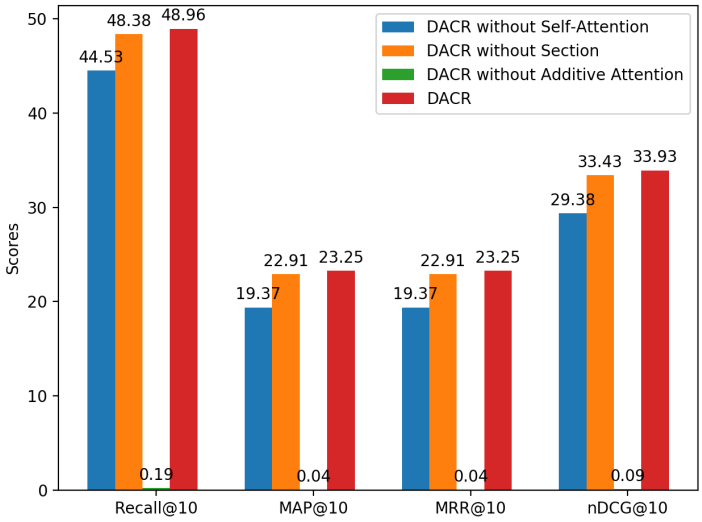

Figure 3: Effectiveness of adding sections, relatedness, and importance

There are three main conclusions that can be drawn from Table 2. First, DACR outperforms all baseline models at 1% significance level across all evaluation scores for all cases and datasets. This implies that the additionally included combined information: namely sections, relatedness, and importance, are essential for predicting useful citations. The effectiveness of each added information is presented in Section 5.5.

Second, performance increases when additional information is preserved in the embedding vectors. When comparing Word2Vec, HyperDoc2Vec, DocCit2Vec, and DACR, Word2Vec only preserves contextual information, HyperDoc2Vec considers citations as links, DocCit2Vec includes structural contexts, and DACR exploits the internal structure of a scientific paper to extract richer information. The evaluation scores increase with the amount of information preserved, indicating that overcoming information loss in embedding algorithms is helpful for recommendation tasks.

Third, DACR is effective for both the large (DBLP) and medium (ACL Anthology) sized datasets. However, we also realized that the smaller dataset requires higher iterations for the model to produce effective results. It is presumed that more iterations of training can compensate for a lack of diversity in the training data.

The performance of DACR could be further improved by more accurately recognizing section headers. Moreover, we determined that some labels were incorrectly recognized or unable to be recognized by ParsLabel. Therefore, we will work on improving the accuracy of section recognition in future work.

5.5 Effectiveness of Adding Sections, Relatedness, and Importance

In this section, we explore the effectiveness of adding the following information: section headers, relatedness, and importance. We run three modified DACR models without the corresponding layer, for example, removing the section embedding layer for verifying the effectiveness of section information, removing the self-attention layer for determining the relatedness between contextual words and articles, and removing additive attention for demonstrating the importance of context. We present the scores of recall, MAP, MRR, and nDCG at 10 for case 1 on the DBLP dataset for comparison, which are illustrated in Figure 3.

Both models, DACR without self-attention and DACR without additive attention perform significantly worse than the full model of DACR, whereas the performance of DACR without section information drops by a minor margin. Three conclusions can be drawn from these facts.

Firstly, all modified models performed poorer than the full model, which supports our hypothesis: sections, relatedness, and importance between contextual words and articles are important for recommending useful citations. The relatedness information is more beneficial than section information, which is

evident when comparing DACR without section and DACR without self-attention.

Secondly, the primary reason for the 0-close scores of the model without additive attention is that the losses of the model did not converge without the additive attention layer. Therefore, we consider that the additive attention has a two-fold purpose: ensuring convergence and learning the importance of context.

Lastly, only appropriate combinations of information and neural network layers lead to optimal solutions, as deficits in any of the three types of information (section, relatedness, and importance, or attention layers) result in low performance.

6 Conclusions and Future Work

In this study, we proposed a citation recommendation model with dual attention mechanisms. This model aims to simplify real-world paper-writing tasks by alleviating the issue of information loss in existing methods. Our model considers three types of essential information: section for which a user is working and need to insert citations, relatedness between the local context words and structural contexts, and their importance. The core of the proposed model is composed of two attention mechanisms: self-attention for capturing relatedness and additive attention for learning importance. Extensive experiments demonstrated the effectiveness of the proposed model in designed scenarios intended to mimic the real world scenarios as well as the efficiency of the proposed neural network.

In future work, we will first attempt to improve the accuracy of recognizing section headers to improve the usability and performance of the algorithm. Second, we will include additional paper-related information in the model, such as word positions. Third, we will explore more sophisticated neural network architectures to improve accuracy and reduce the training time of the model.

Acknowledgements

This research has been supported in part by JSPS KAKENSHI under Grant Number 19H04116 and by MIC SCOPE under Grant Numbers 201607008 and 172307001.

References

Anas Alzoghbi, Victor Anthony Arrascue Ayala, Peter M. Fischer, and Georg Lausen. 2015. Pubrec: Recommending publications based on publicly available meta-data. In *LWA 2015 Workshops*, pages 11–18.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Matthew Berger, Katherine McDonough, and Lee M. Seversky. 2017. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Trans. Vis. Comput. Graph.*, 23(1):691–700.

Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra, and C. Lee Giles. 2013. Can't see the forest for the trees?: a citation recommendation system. In *JCDL*, pages 111–114.

Isaac G. Councill, C. Lee Giles, and Min-Yen Kan. 2008. Parscit: an open-source CRF reference string parsing package. In *LREC*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Marco Gori and Augusto Pucci. 2006. Research paper recommender systems: A random-walk based approach. In *WI*, pages 778–781.

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. 2018. hyperdoc2vec: Distributed representations of hypertext documents. In *ACL*, pages 2384–2394.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and C. Lee Giles. 2010. Context-aware citation recommendation. In *WWW*, pages 421–430.

Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C. Lee Giles. 2011. Citation recommendation without author supervision. In *WSDM*, pages 755–764.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Haofeng Jia and Erik Saule. 2017. An analysis of citation recommender systems: Beyond the obvious. In *ASONAM*, pages 216–223.

Haofeng Jia and Erik Saule. 2018. Local is good: A fast citation recommendation approach. In *ECIR*, pages 758–764.

Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V. Çatalyürek. 2013. Towards a personalized, scalable, and exploratory academic recommendation service. In *ASONAM*, pages 636–641.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

Shuchen Li, Peter Brusilovsky, Sen Su, and Xiang Cheng. 2018. Conference paper recommendation for academic conferences. *IEEE Access*, 6:17153–17164.

Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W. Black, Isabel Trancoso, and Chu-Cheng Lin. 2015. Not all contexts are created equal: Better word representations with variable attention. In *EMNLP*, pages 1367–1372.

Minh-Thang Luong, Thuy Dung Nguyen, and Min-Yen Kan. 2010. Logical structure recovery in scholarly articles with rich document features. *IJDLS*, 1(4):1–23.

Chris A. Mack. 2014. How to Write a Good Scientific Paper: Structure and Organization. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 13(4):1 – 3.

Sean M. McNee, István Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. 2002. On the recommending of citations for research papers. In *CSCW*, pages 116–125.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8024–8035.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147.

Yichuan Tang, Nitish Srivastava, and Ruslan Salakhutdinov. 2014. Learning generative models with visual attention. In *NIPS*, pages 1808–1816.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*, pages 6388–6393.

Y. Zhang and Q. Ma. 2020. Doccit2vec: Citation recommendation via embedding of content and structural contexts. *IEEE Access*, 8:115865–115875.