DisenCite: Graph-Based Disentangled Representation Learning for Context-Specific Citation Generation

Yifan Wang[1], Yiping Song[2]*, Shuai Li[1], Chaoran Cheng[1], Wei Ju[1], Ming Zhang[1]*, Sheng Wang[3]

[1] School of Computer Science, Peking University, Beijing, China
[2] National University of Defense Technology
[3] Paul G. Allen School of Computer Science, University of Washington
{yifanwang, chengchaoran, juwei, mzhang_cs}@pku.edu.cn, lishuai@stu.pku.edu.cn, songyiping@nudt.edu.cn, swang@cs.washington.edu

Abstract

Citing and describing related literature are crucial to scientific writing. Many existing approaches show encouraging performance in citation recommendation, but are unable to accomplish the more challenging and onerous task of citation text generation. In this paper, we propose a novel disentangled representation based model DisenCite to automatically generate the citation text through integrating paper text and citation graph. A key novelty of our method compared with existing approaches is to generate context-specific citation text, empowering the generation of different types of citations for the same paper. In particular, we first build and make available a graph enhanced contextual citation dataset (GCite) with 25K edges in different types characterized by citation contained sections over 4.8K research papers. Based on this dataset, we encode each paper according to both textual contexts and structure information in the heterogeneous citation graph. The resulted paper representations are then disentangled by the mutual information regularization between this paper and its neighbors in graph. Extensive experiments demonstrate the superior performance of our method comparing to state-of-the-art approaches. We further conduct ablation and case studies to reassure that the improvement of our method comes from generating the context-specific citation through incorporating the citation graph.

Introduction

The massive accumulation of scientific papers promotes advances in science. However, it also leads to the growing burdens for researchers to retrieve, review, and digest literature. Consequently, there is a pressing need to develop data-driven tools that can automate different tasks to accelerate literature reviews, such as literature summarizing (Luhn 1958; Mei and Zhai 2008; Yasunaga et al. 2019), citation recommendation (He et al. 2010; Huang et al. 2012), and paper representation (Beltagy, Lo, and Cohan 2019; Cohan et al. 2020). An important but unaddressed problem in this endeavor is to automatically generate citation text (Xing, Fan, and Wan 2020; Luu et al. 2021). Citation text generation aims to generate a short text, typically one or two sentences, to describe a cited paper in the context of the citing paper, making it an immediate step after citation recommendation and an inevitable step before finishing the paper.

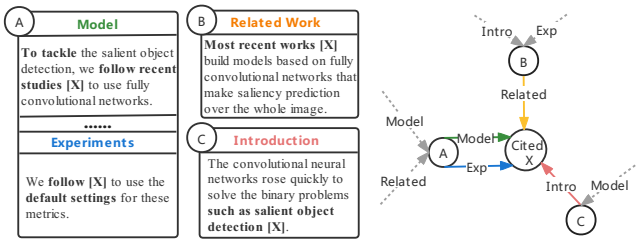


Figure 1: An illustration of different types of contextual citations for cited paper X and the constructed heterogeneous citation graph. The bold font indicates highly related text within each type-specific citation.

The most similar task to citation text generation is the well-studied task of paper summarization (Hoang and Kan 2010; Hu and Wan 2014; Chen and Zhuge 2019). However, citation text generation is more challenging than paper summarization since it is required to capture not only the content of the cited paper, but also the relationship between the cited and citing paper. One promising approach to capture such relationship is to leverage the citation graph, which enables us to resemble citation text from other similar papers.

Nevertheless, it remains challenging to integrate the structured citation graph and the unstructured paper text. CGSUM (An et al. 2021) leverages citation graph to incorporate the information of both paper and its references for the summarization model. SPECTER (Cohan et al. 2020) generates document-level embeddings of scientific papers for downstream tasks through pretraining a Transformer language model on citation graph. AutoCite (Wang et al. 2021) jointly learns citation recommendation and context generation based on the paper representations which are encoded with both citation graph structural and textual contexts. In addition to citation graph, some approach (Tian et al. 2021) utilizes social networks for few-shot text personalized conversation task and improves the generation performance.

Despite their encouraging performance, existing approaches only assume a single type citation relationship between two papers, which is over-simplified for modeling

*Corresponding authors

real-world citation text. Intuitively, citation text in different paper sections (e.g., introduction, related work, model, and experiment) could emphasize on the different aspects of the relationship between two papers. We demonstrate such examples in Figure 1, where the citation texts to the same cited paper present different semantic meanings. Existing approaches that encourage all citing papers, regardless of the context they belong to, to be closely embedded in the low-dimensional space. As a result, it could introduce bias and further harm citation text generation. For example, simply transferring the citation text from the model section to the experiment section could be problematic, as the former often discusses the technical content of the citing paper whereas the latter mainly focuses on the experiment settings.

Presented Work. Motivated by the above observations, we incorporate the citation graph into the citation text generation through considering the context and specific contained sections of each citation. Since all current citation graph datasets do not provide citation positions between papers, we first constructed a graph enhanced contextual citation dataset (GCite) based on papers from the Semantic Scholar Open Research Corpus (S2ORC) (Lo et al. 2020). Our dataset is a connected heterogeneous citation graph that contains 25K citation relationships with different types over 4.8K papers. We divide the context of papers into four sections (namely introduction, related work, model, and experiment) and extract citation context from each section. In this way, citation relationships between papers could be categorized by the citation contained sections in citing paper.

Based on our dataset, we propose a novel disentangled representation based model **DisenCite**, which integrates the document and relevant citation graph structural information for citation text generation. The key idea of our method is to infer the potential citation positions between the citing-cited paper pair and generate context in corresponding positions by leveraging disentangled factors. Specifically, we first encode the content of different sections from one paper to capture their specific context features. Given the node (paper) pair, we extract a local enclosing subgraph around the pair from the whole citation graph and then utilize the heterogeneous graph encoder to capture the information of subgraph structure, in which each features of node are initialized by the encoded section features and propagated based on the citation type. Furthermore, since the sections of one paper is more related to the neighbors cited within this section, we introduce a mutual information (MI) estimation strategy for the paper and its type-specific neighbors to disentangle the section-general and section-specific representation of the paper. And the disentangled factors represent different aspects of the paper which can be used in combination for the downstream multi-task decoder.

To summarize, in this paper we make the following contributions:

- *Dataset Contribution*: We release GCite[1], a graph enhanced contextual citation dataset containing 25K heterogeneous citation relationships over 4.8K papers. To the best of our knowledge, this is the first citation graph dataset with different types of citation relationships.
- *Conceptual*: We propose to learn the disentangled paper representations enhanced by the citation graph for contextual citation generation at different positions.
- *Methodological*: Our model captures the characteristic differences of representations acting in diverse roles via graphical MI estimation. It includes the MI maximization and minimization strategies among disentangled factors.

[1]https://github.com/jamesyifan/DisenCite

Related Work

Contextual Citation Generation. Neural text generation models have been widely used in machine translation (Bahdanau, Cho, and Bengio 2014), dialog systems (Song et al. 2018, 2020), and speech recognition (Graves, Mohamed, and Hinton 2013). For contextual citation generation, the task is similar to scholarly paper summarization, and some previous works formulate the related work generation as a special case of multi-document scientific summarization (Hoang and Kan 2010; Hu and Wan 2014; Chen and Zhuge 2019). Nevertheless, citation text generation is different from the related work generation based on paper summarization, where the generated text is much shorter and describe the cited paper according to the context of the citing paper. Recently, PTGEN-Cross (Xing, Fan, and Wan 2020) pilots the task of in-line citation generation which inserts a citing sentence into a particular context within a document. SciGEN (Luu et al. 2021) addresses the task of explaining relationships between two scientific documents using citation text generation. AutoCite (Wang et al. 2021) introduces a multi-task model to infer potentially related work and generates the citation context at the same time. However, these works fail to generate different types of citation texts according to the contexts and positions in the paper.

Disentangled Representation Learning. Disentangled representation learning, which aims to learn representations that separate explanatory factors of variations behind the data (Bengio, Courville, and Vincent 2013), has recently gained much attention. Not only such representations are demonstrated to be more robust, i.e., enhancing generalization ability as well as improving robustness to adversarial attack (Alemi et al. 2017), but also more compatible for the downstream applications, such as images generation (Chen et al. 2016; Higgins et al. 2017), recommendation (Ma et al. 2019b; Wang et al. 2020) and graph representation learning (Ma et al. 2019a). For text generation, some works (John et al. 2019; Cheng et al. 2020) disentangle the latent representation of style and content in language models for conditional text generation. DDS-VAE (Bao et al. 2019) generates sentences from the disentangled syntactic and semantic spaces. Instead of directly dividing the encoded contextual factors, our work focuses on learning disentangled representation enhanced by the citation graph for context-specific citation text generation.

Mutual Information Estimation. MI is a fundamental measurement of the dependence between two random variables, which has been applied to a wide range of tasks, including generative modeling (Chen et al. 2016; Cheng

et al. 2020) and the information bottleneck (Tishby, Pereira, and Bialek 2000). As exact value of MI is hard to calculate, MINE (Belghazi et al. 2018) makes the estimation of MI on deep neural networks feasible. For graph structure data, some works estimate MI for unsupervised or semi-supervised learning (Velickovic et al. 2019; Peng et al. 2020; Sun et al. 2020). VIPool (Li et al. 2020) leverages MI maximization to obtain an optimization for vertex selection by finding the vertices that maximally represent their local neighborhood. Recently, some approaches (Sanchez, Serrurier, and Ortner 2020; Cheng et al. 2020) perform representation disentanglement on images/texts based on MI estimation. Inspired by these observations, we utilize MI estimation to learn the disentangled paper representation in the citation graph.

Problem and Novel Dataset

Problem Definition. The goal of the contextual citation generation problem studied in this paper is to predict citation positions and generate corresponding citation text simultaneously. Formally, we define citation graph as a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ consists a set of scientific papers (nodes), $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of citation relations (edges) between nodes in $\mathcal{V}$. There is an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$ in citation graph, where each edge $e \in \mathcal{E}$ belongs to one particular edge type set $\mathcal{R} : \psi(e) \in \mathcal{R}$ with the property that $|\mathcal{R}| > 1$. Meanwhile, each node $v \in \mathcal{V}$ consists of four sections (namely introduction, related work, model and experiment) where the context of sections can be represented as a sequence of words. For each node pair (u, v) in $\mathcal{G}$, our task is to predict which sections that target paper u may cite related work v, and automatically generate the possible citation text.

Graph Enhanced Contextual Citation Dataset. Many scientific citation datasets have emerged in recent years. The most commonly used citation network datasets like Cora, Citeseer and PubMed (Sen et al. 2008), focus on paper category classification. AAN (Radev et al. 2013) and SSN (An et al. 2021) propose a graph-enhanced scientific summarization dataset but ignore the contextual citation relationships between papers. PTGEN-Cross (Xing, Fan, and Wan 2020) trains a citation text extraction model to construct a contextual citation generation dataset but without considering the graph structure and different citation relationships. In view of the above, we construct a graph enhanced contextual citation dataset GCite, consisting of 25K relationships with different types (7.5K introduction, 8.0K related work, 4.9K model and 4.6K experiment citations) over 4.8K papers extracted from computer science domain of S2ORC (Lo et al. 2020). We divide the body text of papers into four sections by the keywords, and extract citation texts within each section. All citation positions are labeled by the corresponding citation's contained section type.

The Proposed Model

Overview

The basic idea of our proposed model is to encode disentangled representation of documents based on the citation graph $\mathcal{G}$ to help predict which sections the citation could exist and generate the corresponding citation text. As shown in Figure 2, there are four component in DisenCite framework. Given a citing-cited node pair (u, v) and the extracted subgraph $\mathcal{G}_{u,v}$ consisting of its L hops neighbors, we encode the context of sections within each node into latent representation and divide it into two parts. Then we utilize a graph encoder to capture the information of the subgraph structure by aggregating neighborhood information and updating corresponding divided factors along with citation types. To disentangle the representation of the paper effectively, we introduce a MI estimation strategy among divided factors for each node in $\mathcal{G}_{u,v}$. Finally, a multi-task decoder jointly predicts the citation positions and utilizes specific disentangled factors to generate citation texts at corresponding positions.

Section Based Document Encoder

For each section of one paper, we encode its input document to get the corresponding section states, which represent different aspects of the paper. We first extract and remove the citation text from each section to prevent information leakage, and fliter out the related work section since most input of this section are citation contexts. Then, for each input document $D = \{x_1, x_2, \ldots, x_k\}$ of the remaining sections, we employ a single-layer recurrent neural network (RNN) with gated recurrent unit (GRU (Cho et al. 2014)) to convert the text to a sequence of hidden representations $H = \text{GRU}(x_1, \ldots, x_k)$ and perform a linear transformation of the final hidden state to obtain a hidden vector $h \in \mathbb{R}^d$ as section states. Here, d denotes the dimension of section states.

Considering document of sections can represent both features of the contained paper and specific characteristics of the section, we separate section states h into two parts $h_g \in \mathbb{R}^{\frac{d}{2}}$ and $h_c \in \mathbb{R}^{\frac{d}{2}}$, representing general and specific states of each section, i.e., $h = [h_g; h_c]$, where $[;]$ denotes concatenation. Then, by summing over all sections we integrate the general states of each section as the initial general states of paper $h_g^{(0)}$ and keep specific states of each section as paper's initial introduction, model, and experiment states, represented as $h_i^{(0)}$, $h_m^{(0)}$, and $h_p^{(0)}$. Notice that we concatenate all specific states of sections as this paper's initial related work states $h_r^{(0)}$.

Heterogeneous Graph Encoder

Since the citation relationships in graph are diverse, each aspect of paper is cooperatively characterized by its type-specific neighbors, we aggregate features from different types of neighbors to update this paper's corresponding aspect features. Given target node t and all its one hop neighbors, we group them by different citation relations to get $N_{\psi(e)}$, which denotes the same type of source nodes connecting to t with citation type $\psi(e)$. Inspired by the architecture of Transformer (Vaswani et al. 2017), we use L-layers of heterogeneous graph transformer layers to update target node's representation. For the l-th layer, we map the target node t into a Query vector and the source neighbors

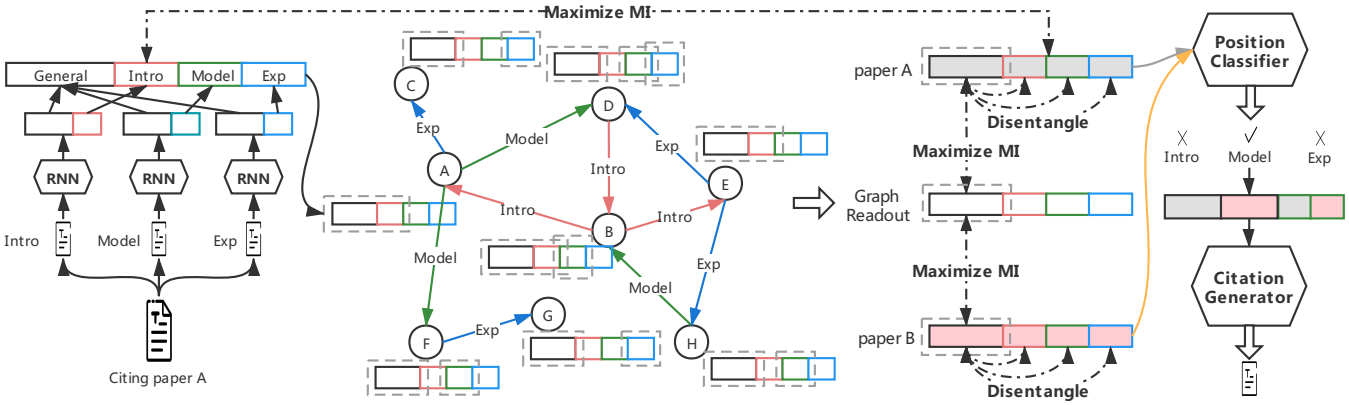


Figure 2: A schematic view of the DisenCite. We extract L hops neighbors from the target citing-cited paper pair as subgraph, the color and direction of edges in the subgraph represents different citation types and citing-cited relation of the paper pair. The MI-based disentangled encoding model are employed on the graph and we use corresponding disentangled factors of the target pair for the citation position and context generation tasks.

$s \in N_{\psi(e)}$ under each group into a Key vector, and calculate the attention weight $\alpha_{t,s}^{\psi(e)}$ on grouped neighbors $N_{\psi(e)}$.

Taking introduction citation for example, we consider that both general and introduction states are affected, and calculate the attention as:

$$\begin{aligned} \alpha_{t,s}^{\psi(e)} &= \underset{\forall s \in N_{\psi(e)}}{\text{Softmax}}\Big(\frac{(Q_{t,\psi(e)}^{(l-1)})(K_{s,\psi(e)}^{(l-1)})^{\mathrm{T}}}{\sqrt{d}}\Big), \\ Q_{t,\psi(e)}^{(l-1)} &= [W_{Q,g}^{(l-1)} h_{t,g}^{(l-1)}; W_{Q,i}^{(l-1)} h_{t,i}^{(l-1)}], \\ K_{s,\psi(e)}^{(l-1)} &= [W_{K,g}^{(l-1)} h_{s,g}^{(l-1)}; W_{K,i}^{(l-1)} h_{s,i}^{(l-1)}], \end{aligned} \tag{1}$$

where $h_{t,g}^{(l-1)}, h_{s,g}^{(l-1)} \in \mathbb{R}^{\frac{d}{2}}$ and $h_{t,i}^{(l-1)}, h_{s,i}^{(l-1)} \in \mathbb{R}^{\frac{d}{2}}$ are the general and introduction states of target and source papers respectively. $W_{Q,g}^{(l-1)}, W_{K,g}^{(l-1)} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ and $W_{Q,i}^{(l-1)}, W_{K,i}^{(l-1)} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ are transformation matrices for general and introduction states. After that, we can aggregate the general and introduction features of paper:

$$\begin{aligned} z_{t,g}^{\psi(e)} &= \sum_{s \in N_{\psi(e)}} \alpha_{t,s}^{\psi(e)} (W_{V,g}^{(l-1)} h_{s,g}^{(l-1)}), \\ z_{t,i} &= \sum_{s \in N_{\psi(e)}} \alpha_{t,s}^{\psi(e)} (W_{V,i}^{(l-1)} h_{s,i}^{(l-1)}), \end{aligned} \tag{2}$$

where $W_{V,g}^{(l-1)}, W_{V,i}^{(l-1)} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ are transformation matrices for the general and introduction states. Note that for model, experiment and related work sections, we also aggregate neighbor features to get $z_{t,m}$, $z_{t,p}$ and $z_{t,r}$ respectively. Moreover, through the mean pooling, we reduce the general states $z_{t,g}^{\psi(e)}$ aggregated from each type-specific neighbors to get the final aggregated general states $z_{t,g}$. Then, taking introduction states as an example, we apply a linear projection to the aggregated feature, followed by residual connection (He et al. 2016) as:

$$h_{t,i}^{(l)} = h_{t,i}^{(l-1)} + W_i^{(l-1)} z_{t,i}, \tag{3}$$

where $W_i^{(l-1)} \in \mathbb{R}^{\frac{b}{2}\times\frac{b}{2}}$ is the weight matrix, and l-th layer's output $h_{t,i}^{(l)}$ can be used as input for the next layer.

Representation Disentanglement with MI

MI captures non-linear statistical dependencies between variables. In this section, we maximize the MI of the same aspect features between target paper and its type-specific support neighbors to characterize each aspect while minimize the MI among different aspect features of one paper to enforce representation disentanglement.

General States Intra-MI. For general states of paper, we assume that the extracted subgraph $\mathcal{G}_{u,v}$ shares same general topics and encourage general states to carry information that is presented in all nodes of subgraph (and thus are globally relevant). Following Velickovic et al. (2019), we utilize a readout function to summarize the last layer patch representations of the subgraph into a graph-level representation:

$$h_{\mathcal{G}_{u,v}} = \text{READOUT}(\{h_{t,g}^{(L)} | t \in \mathcal{V}_{u,v}\}), \tag{4}$$

where $\mathcal{V}_{u,v}$ are all nodes in $\mathcal{G}_{u,v}$, READOUT can be a simple permutation invariant function such as the mean function. We define the general states MI estimator on global/local pairs, maximizing the estimated MI over nodes in $\mathcal{V}_{u,v}$:

$$L_{intra}^{g} = \frac{1}{|\mathcal{V}_{u,v}|} \sum_{t \in \mathcal{V}_{u,v}} I(h_{\mathcal{G}_{u,v}}, h_{t,g}^{(L)}), \tag{5}$$

where $I(h_{\mathcal{G}_{u,v}}, h_{t,g}^{(L)})$ is the MI estimator modeled by discriminator D_g. We use the binary cross-entropy (BCE) loss of the discriminator (following the formulation of Velickovic et al. (2019)):

$$\begin{aligned} I(h_{\mathcal{G}_{u,v}}, h_{t,g}^{(L)}) = \log \delta(D_g(h_{\mathcal{G}_{u,v}}, h_{t,g}^{(L)})) + \\ \mathbb{E}_{\tilde{H},\tilde{\mathcal{E}}}\big[\log\big(1 - \delta(D_g(h_{\mathcal{G}_{u,v}}, \tilde{h}_{t,g}^{(L)}))\big)\big], \end{aligned} \tag{6}$$

where $(\tilde{H}, \tilde{\mathcal{E}})$ are the negative samples obtained by an explicit (stochastic) corruption function. D_g is a bilinear scoring function.

Specific States Intra-MI. For specific states of paper, taking introduction states as an example, introduction aspect of target paper is the most related to neighbors cited in introduction section. Thus, to keep specific characteristics of introduction states, we maximize the MI between target node and its specific surrounding neighborhoods $N_{\psi(e)}$ on introduction states:

$$L^i_{intra} = \frac{1}{|\mathcal{V}_{u,v}|} \sum_{t \in \mathcal{V}_{u,v}} I(h^{(0)}_{t,i}, h_{N_{\psi(e)},i}), \tag{7}$$

where $I(h^{(0)}_{t,i}, h_{N_{\psi(e)},i})$ is the MI estimator modeled by discriminator D_i, $h_{N_{\psi(e)},i}$ is the aggregated introduction features of specific neighbors, namely, can be $h^{(L)}_{t,i}$. We use the BCE loss of the discriminator as in Equation 7 and the negative samples are obtained by the same corruption function. Similarly, we can also maximize the MI on model, experiment and related work states to get L^m_{intra}, L^p_{intra}, and L^r_{intra} respectively.

States Inter-MI. For the updated general, introduction, model and experiment states of one paper, namely $H^{(L)}_t = \{h^{(L)}_{t,g}, h^{(L)}_{t,i}, h^{(L)}_{t,m}, h^{(L)}_{t,p}\}$, mutual information minimization among them encourages them to learn different aspect information of the paper. Taking introduction and model states for example, the mutual information between them is defined as $I(h^{(L)}_{t,i}, h^{(L)}_{t,m})$, which reaches its minimum value zero when $h^{(L)}_{t,i}$ and $h^{(L)}_{t,m}$ are independent to each other. As orthogonality is a special case of linear independence of vector groups, instead of learning a discriminator between every two of states in one paper, we introduce the constraint of orthogonality between states. The constraint has also been demonstrated to be effective by many previous studies (Liang, Li, and Madden 2020).

$$L_{inter} = \frac{1}{|\mathcal{V}_{u,v}|} \sum_{t \in \mathcal{V}_{u,v}} |{H^{(L)}_t}^{\mathrm{T}} H^{(L)}_t - I|, \tag{8}$$

where $|\cdot|$ is the L_1 norm, I is the identity matrix.

Multi-Task Decoder

To realize the citation position classification and context generation simultaneously, we propose a multi-task decoder. Through integrating all last layer states of citing-cited node pair (u, v), we predict which sections the citation could exist and generate text through combining corresponding states.

Citation Position Classification. For the predicted node pair (u, v) in the graph, we take both two nodes as target, integrate the last layer states of nodes and define the probability that citation may exist within each section. Take introduction section as an example:

$$y^*_i = \sigma(w_i^{\mathrm{T}}[h^{(L)}_g; h^{(L)}_i; h^{(L)}_m; h^{(L)}_p; h^{(L)}_r] + b_i), \tag{9}$$

where $w_i \in \mathbb{R}^{\frac{5d}{2}}$ and b_i are the learnable parameters. $h^{(L)}_g$, $h^{(L)}_i$, $h^{(L)}_m$, $h^{(L)}_p$ and $h^{(L)}_r$ are the aggregated different aspect features of (u, v), taking $h^{(L)}_i$ for example, the aggregated aspect feature can be defined as:

$$h^{(L)}_i = W_i[h^{(L)}_{u,i}; h^{(L)}_{v,i}], \tag{10}$$

where $h^{(L)}_{u,i}, h^{(L)}_{v,i}$ are the updated introduction states of u and v, $W_i \in \mathbb{R}^{\frac{d}{2} \times d}$ is the transformation matrix. We define our object function for multi-label classification with BCE loss, and can be defined as:

$$L_1 = -\frac{1}{|4|} \sum_{x \in \{i,m,p,r\}} y_x \log y^*_x + (1 - y_x) \log(1 - y^*_x) \tag{11}$$

Context Generation. We adopt the frequently used GRU (Cho et al. 2014) to generate citation contexts. For the citation within a specific section, we incorporate the general and the specific states of (u, v) as the initial decoder hidden states d_0. Take introduction section for example:

$$d_0 = [h^{(L)}_g; h^{(L)}_i] \tag{12}$$

The hidden state d_m at time m is calculated recurrently:

$$d_m = \text{GRU}(d_{m-1}, \vec{x}_m), \tag{13}$$

where $\vec{x}_m \in \mathbb{R}^d$ is the word embedding generated at time m. At each step, d_m is transformed to produce the vocabulary distribution p_{vocab}:

$$p_{vocab} = \text{Softmax}(W_d d_m + b_d), \tag{14}$$

where $W_d \in \mathbb{R}^{|vocab| \times d}$ and $b_d \in \mathbb{R}^{|vocab|}$ are weight matrices, $|vocab|$ is the vocabulary size. And the objective of context generation for total M step is defined as:

$$L_2 = -\frac{1}{M} \sum_m \log\big(p_{vocab}(x_m)\big), \tag{15}$$

where x_m is target word at each step. We jointly train the multi-task objectives into a unified framework:

$$L(\Theta) = L_1 + \alpha L_2 + \beta L_{inter} - \gamma \sum_{x \in \{g,i,m,p,r\}} L^x_{intra}, \tag{16}$$

where Θ denotes all parameters of DisenCite. α, β, γ denote the hyper-parameters to balance different losses.

Experiments

We validate the proposed model on our graph enhanced contextual citation dataset GCite. Ablation and case studies are also provided to show the effectiveness of our model.

Experimental Settings

Dataset. The details of our GCite is provided in preliminary section, where the dataset consists of 25K citation relations over 4.8K papers. We random select 80% of citation relations to constitute the training set, and treat the remaining $10\%, 10\%$ as the validation and test set respectively.

Model Class	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
Extractive	Extradst-first	0.4767	0.3569	0.2347	0.1533	0.1106	0.0051	0.0910
	Extradst-random	0.3840	0.2899	0.1920	0.1254	0.1097	0.0056	0.0904
	Extracite-random	0.5162	0.4002	0.2855	0.2079	0.1699	0.0314	0.1375
Seq-based	Seq2seq	0.4883	0.3522	0.2448	0.1750	0.1574	0.0404	0.1394
	PTGEN-Cross	0.3139	0.2343	0.1669	0.1234	0.1641	0.0417	0.1454
	SciGEN	0.4959	0.3975	0.2885	0.2110	0.1556	0.0102	0.1348
Graph-based	AutoCite	0.4696	0.3348	0.2315	0.1650	0.1700	0.0375	0.1334
	GAT	0.5131	0.3818	0.2684	0.1937	0.1548	0.0382	0.1339
	HGT	0.5252	0.3920	0.2758	0.1982	0.1555	0.0388	0.1359
	DisenCite(Ours)	**0.5418**	**0.4109**	**0.2951**	**0.2175**	**0.1756**	**0.0446**	**0.1515**

Table 1: Context generation performance comparison on our graph enhanced contextual citation dataset GCite.

Model	Micro-F1↑	Macro-F1↑	HL↓
FastText	0.7673	0.7310	0.3436
CNN	0.7745	0.7398	0.3331
GRU	0.7713	0.7436	0.3390
SciBert	0.7644	0.7306	0.3046
Specter	0.7785	0.7541	0.2955
GAT	0.7730	0.7415	0.3287
HGT	0.7891	0.7675	0.3040
DisenCite(Ours)	**0.8004**	**0.7835**	**0.2923**

Table 2: Position prediction performance comparison on our graph enhanced contextual citation dataset GCite.

Model	Quality	Consistency	Section-Fit
Extradst-first	**1.60**	0.41	0.48
Extradst-random	1.55	0.38	0.50
Extracite-random	1.48	0.45	0.52
Seq2seq	0.80	0.30	0.38
PTGEN-Cross	0.81	0.31	0.46
SciGEN	1.38	0.69	0.67
AutoCite	0.83	0.48	0.62
GAT	1.06	0.65	0.62
HGT	1.13	0.76	0.93
DisenCite(Ours)	1.35	**1.01**	**1.50**

Table 3: Human evaluation compared with baselines. Average annotator agreement for three protocols: std=0.28, Fleiss' κ=0.33 (within reasonable range).

Baselines. To evaluate the performance of citation position prediction, we compare our model with the following three classes of models:

(A) Traditional Methods. (1) **FastText** (Joulin et al. 2017), (2) **CNN** (Kim 2014) and (3) **GRU** (Cho et al. 2014) which use average word embeddings, convolutional neural network and RNN with gated recurrent unit respectively to encode the document.

(B) Pretraining-based Methods. (1) **SciBert** (Beltagy, Lo, and Cohan 2019) and (2) **Specter** (Cohan et al. 2020) which are pretrained models using document and citation information respectively for downstream tasks.

(C) Graph-based Methods. (1) **GAT** (Velickovic et al. 2018) and (2) **HGT** (Hu et al. 2020) which are homogeneous and heterogeneous graph attention models. We initialize node embeddings by extracted document features.

Furthermore, to evaluate the performance of citation context generation, we compare DisenCite with the following three classes of methods:

(A) Extractive Methods. (1) **Extradst-first**, (2) **Extradst-random** and (3) **Extracite-random** which extract first sentence of cited paper, random sentence of cited paper, and random citation context from other citing papers for the target cited paper respectively.

(B) Seq-based Methods. (1) **Seq2seq** (Bahdanau, Cho, and Bengio 2014), (2) **PTGEN-Cross** (Xing, Fan, and Wan 2020) and (3) **SciGEN** (Luu et al. 2021) which leverage encoder-decoder framework, encoder-decoder with cross attention for the contexts of citing-cited paper pair and GPT2 language model respectively to generate citation context.

(C) Graph-based Methods. (1) **GAT** (Velickovic et al. 2018), (2) **HGT** (Hu et al. 2020) and (3) **AutoCite** (Wang et al. 2021) which are citation graph enhanced homogeneous, heterogeneous graph attention encoders, and multitask encoder respectively for citation generation.

Evaluation Metrics. We perform automatic evaluation metrics for citation position classification task, both automatic and human evaluation metrics for citation context generation task.

(A) Automatic Evaluation. For citation position classification, we employ three widely used metrics for measuring multi-label classification performance, including Micro F1, Macro F1 and Hamming Loss (HL), and notice that the smaller value of HL, the better performance of the learning algorithm. And for citation context generation, widely use metrics BLEU-1/2/3/4 and ROUGE-1/2/L are applied to measure the similarity between the generated context and the ground-truth.

(B) Human Evaluation. We invite 3 well-educated gradu-

	Introduction	Related Work	Experiment
Truth	**Detecting visual relations** between objects and stuffs is an emerging research problem that has drawn **significant attention** recently.	It is pointed out that such relation data would lead **scene graph generators** fitting to statistical counting based on **textual context** instead of understanding visual relations.	We train stacked motif networks on **vrr vg** and matches the results reported in [31] for object detector, **scene graph classification and scene graph detection**.
Extracite-random	The second approach jointly infers the objects and their relationships based on object proposals.	Such datasets are problematic in that they mainly contain **common relations** whose corresponding predicates can be easily detected using **statistical counting** based on the **text context**.	Nevertheless, we have not seen a notable improvement or comprehensive analysis in exploiting the **structured scene graph** for visual qa, despite the fact that several recent works have started to incorporate it.
SciGEN	The scene graph generation task has been widely used in visual relationship detection.	In the recent years , the most recent works heavily rely on **scene graphs**.	
HGT	Furthermore, we use of the **visual genome** vg dataset to evaluate the scene graphs in the visual genome and the coco.	Therefore, we use the same gcn based approaches and we use the top down vocabulary emerged.	
DisenCite	**Detecting visual relations** has drawn **significant attention** recently.	Scene graph, which applies gcn for vision level **scene graph generators**, is a common way to parse scene graph for **language features**.	In this work, we train on the **relation datasets** that consists of annotated scene graph, the images are used for **training and evaluations**.

Table 4: Comparison of generated citation contexts in different sections with ground-truth.

ate students to annotate the 100 generated citation texts for DisenCite and baselines. For each method, the annotators are requested to grade each generated citations in terms of three protocols: *Quality*, *Consistency* and *Section Fitness*. *Quality* measures the appropriateness of generated citations, and we refer 2 for fluent, 1 for few grammar mistakes and 0 for incomprehensible text. *Consistency* measures whether a generated citation is consistent with the topic of citing-cited paper pair, and we refer 2 for highly consistent, 1 for no conflicted and 0 for contradicted. By *Section Fitness* we mean: does the generated citation suitable for using in the corresponding section, and we refer 2 for perfectly fit, 1 for borderline and 0 for inappropriate. All annotators conduct the double-blind annotations on shuffled samples to avoid subjective bias.

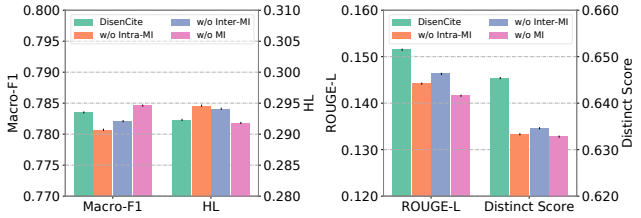
Implementation Detail. We implement our DisenCite model in Pytorch. The word embeddings are randomly initialized with dimension $d = 50$. We limit the input document length to 600 tokens with each section (introduction, method and experiment) less than 200 and citation context length less than 50. For our method, we sample 2 hops of neighborhoods for the target node pair as subgraph with each number of type-specific neighbors are 5 and 4 respectively. The hyper-parameter $\alpha = 1, \beta = 1e{-}1, \gamma = 1e{-}1$, and dropout with probability $p = 0.35$ is employed for all parameters to prevent overfitting. We optimize DisenCite with Adam optimizer by setting the initial learning rate $lr = 5e - 3$ and uses early stopping with a paticnce of 20, i.e. we stop training if ROUGE-L on the validation set dose not increase for 20 successive epochs. For baseline methods, we split exactly the same training, validation and test set as DisenCite and apply a grid search for optimal hyper-parameters.

Experimental Result

Table 2, Table 1 and Table 3 respectively show the results of citation position classification and context generation on GCite, where the best results are boldfaced.

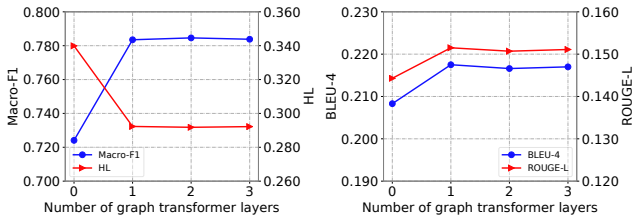
Position Prediction Result. As shown in Table 2, our model has achieved a significant improvement over the baseline methods and the heterogeneous graph attention model HGT gets sub-optimal results. Compared with traditional methods, pretraining-based models has a length limit of 512 and has not shown great advantages in the task. Meanwhile, homogeneous graph attention model GAT considers citation graph structure with single edge type but performs poor. It shows that the heterogeneous citation graph helps to get a better representation for citation position prediction.

Context Generation Result. For context generation, the automatic evaluation results are presented in Table 1, and we can see that our model has achieved the best performance. Compared with seq-based methods, graph-based methods especially HGT achieve slightly better results, which indicates that heterogeneous citation graph structure is critical for generating context. Moreover, for the same citing-cited paper pair, we report the degree of diversity by calculating the number of distinct n-grams (n=1) in generated citations among different sections, namely distinct score (Li et al.



(a) Position prediction (b) Context generation

Figure 3: Ablation study of the DisenCite, w/o means we remove the module from the original DisenCite.



(a) Position prediction (b) Context generation

Figure 4: Performance w.r.t different graph transformer layer numbers.

2015). Compared with 0 (namely exactly same) of other baselines (except extractive methods), DisenCite achieve 0.6454 which almost close to 0.7188 of ground truth.

Additionally, Table 3 shows the results of human evaluation, which is almost consistent with the automatic measurements. We can find that quality scores of the extractive methods are the best since these methods directly extract sentences from paper. However, for consistency scores, some seq-based and graph-based methods perform better. Compared with other baselines, DisenCite achieves the best performance especially on section fitness score, which demonstrates the effectiveness of achieving disentangled representation for context-specific citation generation.

Context Generation Patterns Analysis. We calculate BLEU-4 score of generated contexts within each section, and find that seq-based models (seq2seq) in introduction (0.19) and related work (0.18) perform slightly better than in model (0.16) and experiment (0.17). We think the reason is that citations in introduction and related work are always in the similar format of "XX et at al. . . . ". By integrating heterogeneous citation graph, DisenCite can encode not only textual features as other baselines but also different citation relationships of papers to get their disentangled representations for context-specific citation generation, which alleviates this problem with a higher score in introduction (0.21), related work (0.21), model (0.21) and experiment (0.22).

Ablation Study on MI Estimation. We conduct an ablation study to verify the effectiveness of disentangled representation via MI estimation. As shown in Figure 3, for the task of citation position prediction, the performance has slightly been affected with the removal of intra and inter MI estimations. In contrast, MI estimation has a more significant effect on citation context generation. The best results have been attained by considering both intra and inter MI estimations, which indicates that MI estimation can help to obtain the disentangled paper representation for different types of citation generation within corresponding section.

Effect of Graph Transformer Layer Number. By stacking different numbers of propagation layers, we investigate how the depth of DisenCite affects the performance. In particular, we stack the layer numbers in the range of $[0, 3]$. Figure 4 shows the experimental results and we can find that when there is no states propagation between papers (number of layers equals 0), our model becomes worse, which indicates that citation graph can greatly enhance the two citation generation tasks. Meanwhile, when the number of layers reaches to 1, the performance become stable, which demonstrates that one-hop neighbors is enough for the tasks. This is reasonable since authors always reference other citing papers (one-hop neighbors) to generate citation for the target cited paper in real world scenario.

Case Study. To better understand the disentangled paper representation for citation generation, we present some contexts generated by DisenCite. We randomly sample a source paper to generate its citations within each sections (related work and experiment sections are corresponding to the same cited paper) and bold font indicates highly related text within each section. As shown in Table 4, we can find that although extractive methods can generate different contexts for the paper, the generated texts are not suitable for the corresponding section. Other seq-based and graph-based methods can only generate the same citations for different sections of target paper and are not suitable for the corresponding section as well. DisenCite not only correctly predict the keywords (e.g., scene graph generators) and semantically related words (e.g., relation datasets) as the original text, but also help to generate reasonable citation context for different sections, especially when the same paper cited in different sections. The generation cases also demonstrate the advantage of achieving disentangled representation for context-specific citation generation.

Conclusion

In this paper, we augment different types of citation relations and propose a disentangled paper representation based model DisenCite for citation text generation. Specifically, we use not only the document information of citing-cited pairs, but also the useful document information of the corresponding research community from citation graph to generate the final citation. We construct a graph enhanced contextual citation dataset GCite with 4.8K papers and 25K citation edges with different types. In addition, we design a citation graph enhanced disentangled encoding model via MI estimation strategies for the downstream citation position prediction and context generation tasks. Experiments show the effectiveness of our proposed model and the important role of disentangled representations for citation generation. For future work, we plan to explore heterogeneous graph structure and extend DisenCite model to general text generation tasks, such as dialog and machine translation systems.

Acknowledgments

This paper is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101902 as well as the National Natural Science Foundation of China (NSFC Grant No. 62106008 and No. 62106275).

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *ICLR*.

An, C.; Zhong, M.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X. 2021. Enhancing scientific papers summarization with citation graph. In *AAAI*, 12498–12506.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bao, Y.; Zhou, H.; Huang, S.; Li, L.; Mou, L.; Vechtomova, O.; Dai, X.; and Chen, J. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *ACL*, 6008–6019.

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual information neural estimation. In *ICML*, 531–540.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP*, 3613–3618.

Bengio, Y.; Courville, A. C.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8): 1798–1828.

Chen, J.; and Zhuge, H. 2019. Automatic generation of related work through summarizing citations. *Concurrency and Computation: Practice and Experience*, 31(3): e4261.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NeurIPS*, 2172–2180.

Cheng, P.; Min, M. R.; Shen, D.; Malon, C.; Zhang, Y.; Li, Y.; and Carin, L. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *ACL*, 7530–7541.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. S. 2020. Specter: Document-level representation learning using citation-informed transformers. In *ACL*, 2270–2282.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*, 6645–6649.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, Q.; Pei, J.; Kifer, D.; Mitra, P.; and Giles, L. 2010. Context-aware citation recommendation. In *WWW*, 421–430.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.

Hoang, C. D. V.; and Kan, M.-Y. 2010. Towards automated related work summarization. In *COLING*, 427–435.

Hu, Y.; and Wan, X. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *EMNLP*, 1624–1633.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *WWW*, 2704–2710.

Huang, W.; Kataria, S.; Caragea, C.; Mitra, P.; Giles, C. L.; and Rokach, L. 2012. Recommending citations: translating papers into references. In *CIKM*, 1910–1914.

John, V.; Mou, L.; Bahuleyan, H.; and Vechtomova, O. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *ACL*, 424–434.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of tricks for efficient text classification. In *EACL*, 427–431.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, 1746–1751.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Li, M.; Chen, S.; Zhang, Y.; and Tsang, I. W. 2020. Graph cross networks with vertex infomax pooling. In *NeurIPS*.

Liang, X.; Li, D.; and Madden, A. 2020. Attributed Network Embedding based on Mutual Information Estimation. In *CIKM*, 835–844.

Lo, K.; Wang, L. L.; Neumann, M.; Kinney, R.; and Weld, D. S. 2020. S2ORC: The semantic scholar open research corpus. In *ACL*, 4969–4983.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2): 159–165.

Luu, K.; Wu, X.; Koncel-Kedziorski, R.; Lo, K.; Cachola, I.; and Smith, N. A. 2021. Explaining Relationships Between Scientific Documents. In *ACL*, 2130–2144.

Ma, J.; Cui, P.; Kuang, K.; Wang, X.; and Zhu, W. 2019a. Disentangled graph convolutional networks. In *ICML*, 4212–4221.

Ma, J.; Zhou, C.; Cui, P.; Yang, H.; and Zhu, W. 2019b. Learning Disentangled Representations for Recommendation. In *NeurIPS*, 5712–5723.

Mei, Q.; and Zhai, C. 2008. Generating impact-based summaries for scientific literature. In *ACL*, 816–824.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*, 259–270.

Radev, D. R.; Muthukrishnan, P.; Qazvinian, V.; and Abu-Jbara, A. 2013. The ACL anthology network corpus. *Language Resources and Evaluation*, 47(4): 919–944.

Sanchez, E. H.; Serrurier, M.; and Ortner, M. 2020. Learning disentangled representations via mutual information estimation. In *ECCV*, 205–221.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Song, Y.; Liu, Z.; Bi, W.; Yan, R.; and Zhang, M. 2020. Learning to customize model structures for few-shot dialogue generation tasks. In *ACL*, 5832–5841.

Song, Y.; Yan, R.; Feng, Y.; Zhang, Y.; Zhao, D.; and Zhang, M. 2018. Towards a neural conversation model with diversity net using determinantal point processes. In *AAAI*, 5932–5939.

Sun, F.-Y.; Hoffmann, J.; Verma, V.; and Tang, J. 2020. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*.

Tian, Z.; Bi, W.; Zhang, Z.; Lee, D.; Song, Y.; and Zhang, N. L. 2021. Learning from My Friends: Few-Shot Personalized Conversation Systems via Social Networks. In *AAAI*, 13907–13915.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*, 5998–6008.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *ICLR*.

Wang, Q.; Xiong, Y.; Zhang, Y.; Zhang, J.; and Zhu, Y. 2021. AutoCite: Multi-Modal Representation Fusion for Contextual Citation Generation. In *WSDM*, 788–796.

Wang, Y.; Tang, S.; Lei, Y.; Song, W.; Wang, S.; and Zhang, M. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM*, 1605–1614.

Xing, X.; Fan, X.; and Wan, X. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *ACL*, 6181–6190.

Yasunaga, M.; Kasai, J.; Zhang, R.; Fabbri, A. R.; Li, I.; Friedman, D.; and Radev, D. R. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*, 7386–7393.