Automatic Generation of Citation Texts in Scholarly Papers: A Pilot Study

Xinyu Xing, Xiaosheng Fan, Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{xingxinyu,fanxiaosheng,wanxiaojun}@pku.edu.cn

Abstract

In this paper, we study the challenging problem of automatic generation of citation texts in scholarly papers. Given the context of a citing paper A and a cited paper B, the task aims to generate a short text to describe B in the given context of A. One big challenge for addressing this task is the lack of training data. Usually, explicit citation texts are easy to extract, but it is not easy to extract implicit citation texts from scholarly papers. We thus first train an implicit citation text extraction model based on BERT and leverage the model to construct a large training dataset for the citation text generation task. Then we propose and train a multi-source pointer-generator network with cross attention mechanism for citation text generation. Empirical evaluation results on a manually labeled test dataset verify the efficacy of our model. This pilot study confirms the feasibility of automatically generating citation texts in scholarly papers and the technique has the great potential to help researchers prepare their scientific papers.

1 Introduction

A scientific paper usually needs to cite a lot of reference papers and introduce each reference paper with some text. In this study, the text describing a reference paper is called citation text. A researcher usually needs to find relevant papers he wants to cite and write some text to introduce them when writing a scientific paper. However, the process of writing citation texts is tedious and time-consuming. In order to reduce the burden of researchers, we propose and try to address the task of automatic citation text generation.

Automatic generation of citation texts in scholarly papers is a challenging and meaningful task, however, there are very few studies investigating this problem. Given a cited paper B and the context in a citing paper A (i.e., the sentences before and after a specific position in paper A), the task aims to generate a short text to describe B with respect to the given context in A. The task is like the task of scholarly paper summarization (Luhn, 1958; Edmundson, 1969; Qazvinian and Radev, 2008; Mei and Zhai, 2008). Both of the two tasks aim to produce a text to describe the cited paper B. The major difference between the two tasks is that the citation texts reflect not only the salient content of B, but also the context of A. Different citing papers usually have different descriptions of the same cited paper. Sometimes one paper may cite another paper several times in different positions but give different descriptions because the specific contexts are different. Another difference between the two tasks is the length of the text. A citation text is usually much shorter than a paper summary. Generally, citation text generation can be considered as a task of generating a very short summary of paper B given the context of paper A. The difficulty lies in that given different A or different contexts of A, the task aims to produce different citation texts for the same B.

Most commonly, the citation text is a single sentence, but sometimes it may consist of several sentences (Jebari et al., 2018; Qazvinian and Radev, 2010; Sondhi and Zhai, 2014). Like (Small, 2011), we define citation text as a block of text composed of one or more consecutive sentences surrounding the reference sign. Each citation sentence can be classified as explicit or implicit (Qazvinian and Radev, 2010; Athar and Teufel, 2012; Yasunaga et al., 2019). Explicit citation is a citation sentence that contains explicit reference to the cited paper. An implicit (or non-explicit) citation sentence appears around the explicit citation sentence and it does not attach any explicit reference to the cited paper but supplies additional information about the cited paper. The citation text generation task in this study aims to generate both explicit and implicit

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6181–6190
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

citation sentences.

We build a citation text generation dataset based on the ACL Anthology Network corpus (AAN) (Radev et al., 2013). We first perform human annotation and get 1,000 citation texts (including explicit and implicit citation sentences). We randomly select 400 citation texts as test set, and use the other 600 citation texts to first train a citation text extraction model and then use the extraction model to automatically extract many more citation texts to build a large-scale training dataset.

With the training dataset we construct, we can train our citation generation model. In this paper, we use pointer-generator network (See et al., 2017) as the baseline model. We believe that the key to dealing with citation text generation problem is modelling the relationship between the context of citing paper A and the content of cited paper B. So we encode the context of paper A and the abstract of paper B separately, and add cross attention mechanism by making context and abstract attend to each other. We call our model multi-source pointer-generator network with cross attention mechanism. The evaluation results show that our model outperforms the baseline models.

Our contributions are summarized as follows:

- We propose a new task of automatic citation text generation in scholarly papers.
- We annotate 1,000 citation texts and train a citation extraction model to automatically construct a large training dataset for the citation text generation task. The data are available at `https://github.com/XingXinyu96/citation_generation`.
- We propose the multi-source pointer-generator network with cross attention mechanism to address this challenging task. Evaluation results demonstrate the efficacy of our proposed model.

2 Related Work

Firstly, we introduce some studies on citation extraction. Kaplan et al. (2009) proposed a method based on coreference-chains for citation extraction. Sondhi and Zhai (2014) first independently trained a separate HMM for each citation in the article and then performed a constrained joint inference to label non-explicit citing sentences. Qazvinian and Radev (2010) proposed a framework based on probabilistic inference to extract implicit citations. Jebari et al. (2018) proposed an unsupervised approach which is based on topic modeling and word embedding for implicit citation extraction. Jebari et al. (2018) introduced method based on neural network but it did not give out convincing evaluation results.

A few studies have investigated the task of summarizing single scholarly paper, i.e., single document summarization in the scientific domain, which is relevant to the citation text generation task. Early works include (Luhn, 1958; Baxendale, 1958; Edmundson, 1969), and they tried to use various features specific to scientific articles for summary extraction. Later on, citation information has shown its usefulness for scientific paper summarization (Qazvinian and Radev, 2008; Mei and Zhai, 2008; Qazvinian and Radev, 2010; Cohan and Goharian, 2018; Yasunaga et al., 2019). Several benchmark tests have been set up for scientific summarization, including TAC 2014 Biomedical Summarization track and the CL-SciSumm Shared Task (Jaidka et al., 2016). A few other studies have investigated the task of summarizing multiple scholarly papers, i.e., multi-document summarization in the scientific domain (Mohammad et al., 2009; Yeloglu et al., 2011; Chen and Zhuge, 2014). Related work generation is a special case of multi-document scientific summarization (Hoang and Kan, 2010; Hu and Wan, 2014; Chen and Zhuge, 2019). However, the above related work about scholarly paper summarization is different from the task of citation text generation, which aims to generate a usually very short text to describe the cited paper in the given context of the citing paper.

3 Problem and Corpus

Formally, given a citing paper A, a cited paper B and the context C in A, the task aims to generate the citation text T to describe B. The context C refers to the sentences surrounding the target citation text in A and it is provided to distinguish different mentions of B in different positions of A. The following example shows a paragraph of (Lu et al., 2008) and this article cites paper (Wong and Mooney, 2006). In this example, A refers to (Lu et al., 2008) and B refers to (Wong and Mooney, 2006). The sentence underlined (i.e., the second sentence) is an explicit citation, and the sentence in italics (i.e., the third sentence) is an implicit citationand both of them compose the citation text. The remaining two sentences (i.e., the first and last sentences) compose the context C of A. The phrase in bold which indicates the explicit citation to paper B is called reference sign. And the explicit citation text can be defined as the sentence with a reference sign to the cited paper. The implicit citation text can be defined as the sentences that provide information about the cited paper but do not have any reference sign.

> ...SILT (Kate et al., 2005) learns deterministic rules to transform either sentences or their syntactic parse trees to meaning structures. WASP **(Wong and Mooney, 2006)** is a system motivated by statistical machine translation techniques. *It acquires a set of synchronous lexical entries by running the IBM alignment model and learns a log-linear model to weight parses.* KRISP (Kate and Mooney, 2006) is a discriminative approach ...

In this study, we build a citation generation dataset based on the ACL Anthology Network corpus (AAN) (Radev et al., 2013). The ACL anthology is a collection of papers from the Computational Linguistics journal, and proceedings from ACL conferences and workshops. In particular, we download and use the 2014 version of the AAN corpus which includes almost 23594 papers. After removing papers containing many garbled characters and papers without abstracts, there remains 16675 papers. The metadata of each paper and the paper citation network have been extracted and stored. We find all the mentions of each reference paper in a citing paper by using manually designed regular expressions to match the corresponding reference signs. Lastly, we extract 86052 explicit citations for further use.

3.1 Annotation Process

For each reference sign, we perform human annotation to get all citation sentences. We label a vector in which each dimension corresponds to a sentence. A sentence is marked with C if it is an explicit citation, and with 1 if it is an implicit citation. All other sentences are marked with 0. The label vector of the example we mentioned before is [0,C,1,0].

Our annotation process has two steps. First, we annotate the explicit citation sentences. Despite we have extracted explicit citations with rules, we cannot assure that the extraction is completely correct. In order to accurately evaluate the performance of our methods, the explicit citations in the test dataset should be human annotated. We randomly choose some automatically extracted explicit citations and highlight the reference signs we find. The annotators only need to judge if they think the extraction of reference sign is correct. We stop this step when we get 1,000 explicit citations which are ensured correct by human. The second step is to annotate implicit citation texts. For each explicit citation sentence, we take three sentences before it and three sentences after it as candidate sentences[1]. Note that all the candidate sentences must be in the same section as the explicit citation sentence. We provide candidate sentences, explicit citation sentence, abstract of citing paper and cited paper for every annotator. Explicit citation sentence has already been labelled with C, and the annotators just need to label other sentences with 1 or 0. Note that we require the citation sentences to be continuous, which means there cannot be non-citation sentences between two citation sentences. To make the data more reliable, we make sure that every annotation instance must be annotated by three different people. When they disagree with each other, we take the label chosen by majority.

After the annotation process, we get 1,000 annotated citation texts (including both explicit and implicit citation sentences) for further use. We randomly choose 400 citation texts as the final test dataset and the remaining citation texts are used for training.

4 Implicit Citation Text Extraction Model

After the annotation process, we have 400 citation texts as test dataset and 600 citation texts for training. However, we need large-scale training data to train a feasible citation text generation model. So we decide to use the 600 human annotated citation texts to train an implicit citation text extraction model to expand our training dataset.

We treat implicit citation text extraction as a sequence labeling problem and use BERT (Devlin et al., 2018) to deal with this problem. We add a classification layer on the final hidden representation of BERT and fine-tune the whole model on our dataset. We concatenate all the candidate sentences, the explicit citation sentence and the abstract of the cited paper as the input of BERT. We add a special tag '[s]' at the beginning of all sentences, a special tag '[explicit]' at the beginning of the explicit citation sentence and a special tag '[abs]' at the be-

[1] For simplicity, we do not consider the sentences with a long distance to the explicit citation.

	Precision	Recall	F-value	Acc
α=0.9	73.68	55.55	62.95	92.53
α=0.1	64.23	62.02	62.94	91.67

Table 1: Average test results for 10 fold cross-validation

	Precision	Recall	F-value	Acc
α=0.9	72.16	53.21	61.06	91.43
α=0.1	64.79	60.60	62.50	90.80

Table 2: Average test results on external test data

ginning of the cited paper's abstract. The abstract of cited paper does not need to be labelled but it can provide a lot of information to help label the candidate sentences. BERT gives out the probability of every sentence to be implicit citation. We set a threshold α to control the identification of implicit citation sentence. When the probability given out by BERT is greater than α, we take the corresponding sentence as an implicit citation sentence. It is obvious that the smaller α is, the more sentences will be recognized as implicit citation sentence. To ensure the citation text being continuous, we start to identify implicit citation sentences from the explicit citation sentence to both sides and stop when meeting the first non-citation sentence. We do 10 fold cross-validation on our training dataset and use the 400 test data as external test data. The 600 training data are split into 10 subsets. When training, we use 9 subsets for training and use the remaining one subset as test set. The average results for cross-validation are shown in Table 1. The average results on external test data are shown in Table 2.

Our model is compared with these baseline models:

All one: It labels all candidate sentences with 1.

Random: It labels all candidate sentences randomly.

Cosine sim: It first uses bag of words model to represent all texts as vectors. Then it calculates the cosine similarity between candidate sentence and cited paper's abstract, and the cosine similarity between candidate sentence and the explicit citation sentence. When the two similarities are both greater than the threshold, the sentence is labelled with 1.

W2v sim: This model is also based on similarity. The similarity in this model is calculated based on word2vec model. With two sequence of words, it first gets the corresponding two sequences of vectors $\{u_i\}$ and $\{v_j\}$ with word2vec model. Then it uses the two sequences of vectors to calculate a similarity matrix M. The element of the matrix $M_{i,j} = cos(u_i, v_j)$. Finally it keeps the max value of every row vector and takes the average value of the max value list as the final similarity.

	Precision	Recall	F-value	Acc
All one	12.67	100.00	22.49	12.67
Random	12.31	49.40	19.71	49.01
Cosine sim	16.87	54.62	25.78	60.15
W2v sim	19.43	54.62	28.66	65.55
SVM	34.39	26.10	29.68	84.33

Table 3: Test results on external test data for the baseline models

	Precision	Recall	F-value	Acc
α=0.9	73.66	60.64	66.52	92.26
α=0.1	66.02	68.67	67.32	91.55

Table 4: Test results on external test data when using full training data

SVM: It trains an SVM to classify if a sentence is implicit citation sentence. The features include sentence position feature, special pattern feature, similarity feature, etc.

Results of all these baseline models are shown in Table 3.

As shown in these tables, our extraction model outperforms all the baseline models. The F-value of our extraction models with α=0.1 and α=0.9 are very close. This indicates that they have close performance. The precision of extraction model with α=0.9 is higher, while the recall of extraction model with α=0.1 is higher. So we can get two different extraction models with two different α. And with the two different extraction models, we can construct two different datasets for further training citation generation model.

To get the two different datasets, we use all 600 data to train two final extraction models. We call the extraction model with α=0.1 $EXT_{\alpha=0.1}$ and call the extraction model with α=0.9 $EXT_{\alpha=0.9}$. The results on external test data when using full training data are shown in Table 4.

5 Final Evaluation Datasets

With the two implicit citation extraction models we trained in the previous section, we construct three datasets for experiments. In each dataset, a data example is a triple: [citing paper's context, cited paper's abstract, gold citation text]. The first dataset is an explicit citation text generation dataset

(**Explicit dataset**). The gold citation text in the training data and test data is single explicit citation sentence. Note that the explicit citation sentences in the training data are automatically extracted with rules and the explicit citation sentences of test data are human annotated. The second dataset is a full citation text generation dataset. The gold full citation texts of test data are human annotated. The gold full citation text of training data is constructed as follows: the gold explicit citation text is extracted with rules and the gold implicit citation text is extracted with $EXT_{\alpha=0.1}$. This extraction model gets higher recall, so we call this dataset high-recall full citation text generation dataset (**HR dataset**). The third dataset is also a full citation text generation dataset, and it is constructed in the same way with the second dataset except that the gold implicit citation text of training data is extracted with $EXT_{\alpha=0.9}$ and we call it high-precision full citation text generation dataset (**HP dataset**). The cited paper's abstract in all the three datasets refers to the abstract of the cited paper B. We use it to represent the content of paper B because the whole article is too long to encode. The citing paper's context in all the three datasets refer to the sentences around the gold citation text in citing paper A. we take three sentences before the gold citation text and three sentences after it as the context. Note that all the context sentences must be in the same section as the gold citation text.

Finally, we have three datasets for experiments:

- **Explicit dataset**: This dataset is built for explicit citation text generation. The test set contains 400 examples with human-annotated explicit citation texts and the training set contains 600 examples with human-annotated explicit citation texts and 85,052 examples with explicit citation texts extracted based on rules. The average lengths of explicit citation texts in the training and test sets are 29.64 words and 27.14 words, respectively.

- **HR dataset**: This dataset is built for full citation text generation. The test set contains 400 examples with human-annotated full citation texts and the training set contains 600 examples with human-annotated full citation texts and 85,052 examples with automatically extracted full citation texts (particularly using $EXT_{\alpha=0.1}$ to extract implicit citation sentences). The average lengths of full citation texts in the training and test sets are 43.50 words and 42.75 words, respectively.

- **HP dataset**: This dataset is similar to **HR dataset**, and $EXT_{\alpha=0.9}$ is used to automatically extract implicit citation sentences in the training dataset. The average lengths of full citation texts in the training and test sets are 39.77 words and 42.75 words, respectively.

6 Citation Generation Model

Our citation text generation model is a multi-source pointer-generator network with cross attention mechanism. Because the citation generation task has two input sequences, we use two encoders to encode them separately and allow the model to copy words from both input sequences. Such a multi-source pointer-generator network does not have the ability to model the relationship between two input sequences, so we add a cross attention mechanism on them. The cross attention mechanism calculates the attention distribution of every word to the other sequence of words. These attention distributions are used to help the decoder. We believe that the citing paper's context can tell the model what information in cited paper's abstract is important and vice versa. The structure of the whole model is shown is Figure 1.

6.1 Pointer-Generator Network

A typical seq2seq model with attention mechanism has three components: an encoder , a decoder and an attention network. The input text is seen as a sequence of words $\{w_1, w_2, ...w_n\}$. The encoder which is a single-layer bidirectional LSTM network receives input words one by one and produces a sequence of encoder hidden states $\{h_i\}$. At each decoding step t, the decoder which is a single-layer unidirectional LSTM receives the previous word and produces decoder state s_t. The attention distribution a^t is calculated as in (Bahdanau et al., 2014):

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \quad (1)$$

$$a^t = softmax(e^t) \quad (2)$$

where v, W_h, W_s and b_{attn} are learnable parameters. At each decoding step t, the attention vector a^t is used to calculate the context vector c_t:

$$c_t = \sum_i a_i^t h_i \quad (3)$$

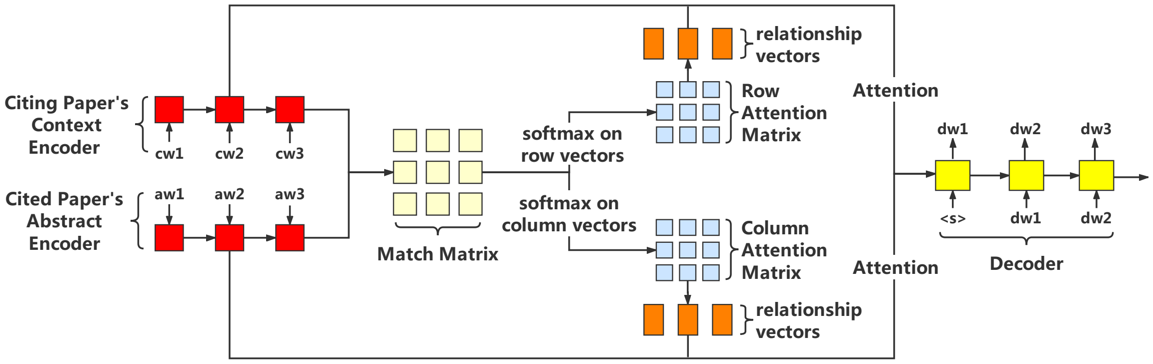

Figure 1: The structure of our generation model

The context vector c_t and the decoder state s_t are used to produce the vocabulary distribution P_v:

$$P_v = softmax(V_2(V_1[s_t, c_t] + b) + b') \qquad (4)$$

where V_1, V_2, b and b' are learnable parameters. P_v is a probability distribution over all words in the vocabulary. During training, we use P_v to calculate the cross entropy loss.

At each decoding step, this network can generate word like normal seq2seq model or copy word from the source sequence. The generation probability p_{gen} for timestep t is:

$$p_{gen} = \sigma(W_c^T c_t + W_s^T s_t + W_x^T x_t + b_{ptr}) \qquad (5)$$

where c_t is the context vector, s_t is the decoder state, x_t is the decoder input, W_c, W_s, W_x and b_{ptr} are learnable parameters and σ is the sigmoid function. p_{gen} is used as a soft switch to choose between generating a word from the vocabulary or copying a word from input sequence. For each text, we define an extended vocabulary which is the union of the vocabulary and all words appearing in the source text. We obtain the following probability distribution over the extended vocabulary:

$$P(w) = p_{gen} P_v(w) + (1 - p_{gen}) \Sigma_{i:w_i=w} a_i^t \qquad (6)$$

Note that if w is not in the vocabulary, $P_v(w)$ is zero. Then we use the probability distribution over the extended vocabulary to calculate the loss.

6.2 Multi-Source Pointer-Generator Network with Cross Attention

Then we introduce our generation model. Firstly we change the pointer-generator network to a multi-source pointer-generator network. The multi-source pointer-generator network has two encoders and one decoder. The two encoders encode the citing paper's context and cited paper's abstract separately. The input context of citing paper is seen as a sequence of words $\{cw_1, cw_2, ..., cw_n\}$ and the input cited paper's abstract is seen as a sequence of words $\{aw_1, aw_2, ..., aw_m\}$. We use the same notation to represent both a word and its embedding vector. The context is encoded by corresponding encoder to a sequence of encoder hidden states $\{ch_i\}$ and the cited paper's abstract is encoded to a sequence of encoder hidden states $\{ah_j\}$. At each decoding step t, we calculate attention vectors $\{ac_i^t\}$, $\{as_i^t\}$ and corresponding context vectors c_t^1, c_t^2 separately as described in equations (1), (2) and (3). To make the model copy words from both two encoders, we change equation (5) to:

$$\begin{aligned}[p_{gen}, p_{copy1}, p_{copy2}] = softmax(&W_{c1}^T c_t^1 + W_{c2}^T c_t^2 \\ &+ W_s^T s_t + W_x^T x_t + b_{ptr})\end{aligned} \qquad (7)$$

where p_{gen} is the probability of generating words, p_{copy1} is the probability of copying words from citing paper's context and p_{copy2} is the probability of copying words from cited paper's abstract. And equation (6) needs to be changed to:

$$\begin{aligned}P(w) = p_{gen} P_v(w) + p_{copy1} \Sigma_{i:cw_i=w} ac_i^t \\ + p_{copy2} \Sigma_{i:aw_i=w} as_i^t\end{aligned} \qquad (8)$$

Then we add the cross attention mechanism to the multi-source pointer-generator network. By making citing paper's context and cited paper's abstract attend to each other, we capture the relationships between them. First, we calculate a match matrix M between the sequence of context's states $\{ch_i\}$ and the sequence of cited paper's abstract's states $\{ah_j\}$. The element of the match matrix

$M_{i,j}$ is:

$$M_{i,j} = ch_i \cdot ah_j \tag{9}$$

Then we apply softmax function on the row vectors of the matrix and get an attention matrix A^{row}. The row vector A_i^{row} of the attention matrix is:

$$A_i^{row} = sotmax([M_{i,1}, M_{i,2}, ..., M_{i,m}]) \tag{10}$$

The vector A_i^{row} represents the attention of word cw_i to the sequence of words $\{aw_1, aw_2, ..., aw_m\}$. We also apply softmax function on the column vectors of the matrix and get another attention matrix A^{column}. The column vector of the attention matrix A_i^{column} represents the attention of word aw_i to the sequence of words $\{cw_1, cw_2, ..., cw_n\}$. With the two attention matrices, we calculate two special sequences of vectors. The first sequence of vectors $\{r_1, r_2, ..., r_n\}$ is calculated as:

$$r_i = \Sigma_{j=1}^{m} A_{i,j}^{row} * aw_j \tag{11}$$

The second sequence $\{q_1, q_2, ..., q_m\}$ is calculated as:

$$q_j = \Sigma_{i=1}^{n} A_{i,j}^{column} * cw_i \tag{12}$$

The vector r_i represent what the word cw_i thinks about the sequence of words $\{aw_1, aw_2, ..., aw_m\}$, while the vector q_j represents what the word aw_j thinks about the sequence of words $\{cw_1, cw_2, ..., cw_n\}$. We believe that the two sequences of vectors can model the relationship between the input citing paper's context and cited paper's abstract, so we call them relationship vectors. With these two sequences of relationship vectors, we calculate two new context vectors c_t^3 and c_t^4 separately at each decoding step t, by replacing the encoder hidden state h_i with the relationship vector r_i or q_j in equations (1) (2) and (3). Finally, we calculate the vocabulary distribution with all four context vectors. We just need to change equation (4) to:

$$P_v = softmax(V_2(V_1[s_t, c_t^1, c_t^2, c_t^3, c_t^4] + b) + b') \tag{13}$$

The final probability distribution over the extended vocabulary is still calculated as equation (8).

7 Experiments

7.1 Experimental Setup

The baseline models include:

RandomSen: It randomly selects a sentence from the abstract of paper B.

Models	ROUGE-1	ROUGE-2	ROUGE-L
RandomSen	15.18	1.37	11.35
MaxSimSen	15.65	1.64	11.45
EXT-ORACLE	22.60	4.21	16.83
COPY-CIT	20.54	3.25	14.79
PTGEN	24.60	6.16	19.19
PTGEN-Cross	**26.28**	**7.50**	**20.49**

Table 5: Comparison results on Explicit dataset

	ROUGE-1	ROUGE-2	ROUGE-L
RandomSen	15.65	1.36	10.98
MaxSimSen	17.70	1.80	12.20
Ext-ORACLE	22.59	3.88	15.97
COPY-CIT	19.32	2.71	13.02
PTGEN	22.83	5.17	18.37
PTGEN-Cross	**24.54**	**5.44**	**19.21**

Table 6: Comparison results on HP dataset

	ROUGE-1	ROUGE-2	ROUGE-L
RandomSen	15.65	1.36	10.98
MaxSimSen	17.70	1.80	12.20
Ext-ORACLE	22.59	3.88	15.97
COPY-CIT	20.08	2.67	13.01
PTGEN	23.26	5.12	18.83
PTGEN-Cross	**24.22**	**6.04**	**19.38**

Table 7: Comparison results on HR dataset

MaxSimSen: It selects a sentence from the abstract of paper B, which has the largest similarity with the context of A.

EXT-ORACLE: It can be viewed as an upper bound for extractive models. It creates an oracle citation text by selecting the best possible sentence from the abstract of paper B that gives the highest ROUGE with respect to the gold text.

COPY-CIT: It randomly copies one citation text from the papers in the training dataset which also cite the paper B.

PTGEN: It is a pointer-generator network which allows both copying words via pointing and generating words from a fixed vocabulary. When using this model, we concatenate the citing paper's context and the cited paper's abstract as the input sequence.

Our proposed model is called **PTGEN-Cross**. Both our model and the PTGEN has 256-dimensional hidden states and 128-dimensional word embeddings. The vocabulary size is set to 50k. At test time the citation texts are produced using beam search with beam size 4.

7.2 Results

7.2.1 Automatic Evaluation

We evaluate our models with ROUGE (Lin, 2004), reporting the F_1 scores for ROUGE-1, ROUGE-2

Context	...They include entity approaches for local coherence which track the repetition and syntactic realization of entities in adjacent sentences [otherrefer] and content approaches for global coherence which view texts as a sequence of topics, each characterized by a particular distribution of lexical items [otherrefer]. **[cit]** Early theories [otherrefer] posited that there are three factors which collectively contribute to coherence: intentional structure (purpose of discourse), attentional structure (what items are discussed) and the organization of discourse segments...
Abstract	We combine lexical, syntactic, and discourse features to produce a highly predictive model of human readers judgments of text readability ... Our experiments indicate that discourse relations are the one class of features that exhibits robustness across these two tasks.
Gold	Other work has shown that co-occurrence of words [otherrefer] and discourse relations [refer] also predict coherence.
PTGEN	Recently, approaches [refer] have been suggested to predict the quality of discourse relations.
PTGEN-Cross	Other work has shown that co-occurrence of sentences [otherrefer]; [refer] and discourse relations [otherrefer] discourse can be used to predict the coherence of sentences in texts.

Table 8: Example output citation texts

	Gold	PTGEN	PTGEN-Cross
Readability	4.89	3.77	**3.79**
Content	4.42	2.76	**2.77**
Coherence	4.41	2.70	**2.85**
Overall	4.55	2.84	**2.91**

Table 9: Human evaluation results

and ROUGE-L. The test results on three datasets are shown in Tables 5, 6 and 7, respectively.

On all three datasets, extractive models perform poorly. Our baseline generation model PTGEN outperforms EXT-ORACLE which can be seen as a 'perfect' extractive system. This is completely different from how these models preform on other summarization tasks like news document summarization. We believe it shows the particularity of this task. It not only requires the model to capture the important content of the cited paper, but also requires the model to capture the attitude of the citing paper to the cited paper. The model not only needs to generate fluent and informative text, but also needs to ensure the contextual coherence.

Our proposed model PTGEN-Cross obviously outperforms the baseline model PTGEN. This proves the effectiveness of the cross attention mechanism. We think the cross attention mechanism helps the model capture the relationship between the citing paper's context and the cited paper's abstract. The results on explicit citation text generation dataset are all higher than the results on the other two datasets, which means the task of explicit citation text generation is easier than the task of full citation text generation. We think it is because the context of explicit citation sometimes contains some implicit citation sentences and these sentences can be very helpful to the generation of explicit citation text. Another possible reason is that the quality of the training dataset for explicit citation generation is higher than the other two training datasets. Because the test data of the two full citation text generation datasets is the same, we can compare the results of our model training on the two datasets. The model trained on the high-recall dataset performs slightly better. This tells us the coverage ability of the implicit citation extraction model is more important when constructing training dataset for citation generation.

7.2.2 Human Evaluation

We randomly sample 50 instances from the high-recall test set and perform human evaluation on them. Three graduate students are employed to rate the citation text produced by each method in four aspects: readability (whether the citation text is fluent), content (whether the citation text is relevant to the cited paper's abstract), coherence (whether the citation text is coherent with the citing paper's context) and overall quality. The rating score ranges from 1 to 5, and 1 means very bad and 5 means very good. Note that every text is scored by three judges and we take take the average of three scores. The results are shown in Table 9.

As is shown in the table, our model outperforms the baseline model, especially with respect to the coherence and overall aspects. This further demonstrates the efficacy of our proposed model. We show an example of generation in Table 8. Note that all reference signs to the cited paper are masked as '[refer]' and all reference signs to other papers are masked as '[otherrefer]'. The '[cit]' in bold in context indicates the position the citation text should be. We can see that the citation text generated by our model is more contextual coherent because it can capture the relationship between context and the cited paper's abstract better.

8 Conclusion and Future Work

In this paper we investigate the challenging task of automatic generation of citation texts in scholarly papers. We annotate a dataset and train an implicit citation extraction model to automatically enlarge the training data. we then propose the multi-source pointer-generation network with cross attention mechanism to deal with this task. Empirical evaluation results on three datasets verify the efficacy of our proposed method. In future work, we will consider introducing more information like the citation texts to the cited paper in other papers to help the generation.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Tencent AI Lab Rhino-Bird Focused Research Program (No.JR201953) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Awais Athar and Simone Teufel. 2012. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 18–26, Jeju Island, Korea. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Phyllis B Baxendale. 1958. Machine-made index for technical literaturean experiment. *IBM Journal of research and development*, 2(4):354–361.

Jingqiang Chen and Hai Zhuge. 2014. Summarization of scientific documents by detecting common facts in citations. *Future Generation Computer Systems*, 32:246–252.

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

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2-3):287–303.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 427–435. Association for Computational Linguistics.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2016. Overview of the cl-scisumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 93–102.

Chaker Jebari, Manuel Jesús Cobo, and Enrique Herrera-Viedma. 2018. A new approach for implicit citation extraction. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 121–129. Springer.

Dain Kaplan, Ryu Iida, and Takenobu Tokunaga. 2009. Automatic extraction of citation contexts for research paper summarization: A coreference-chain based approach. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLPIR4DL)*, pages 88–95.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 783–792. Association for Computational Linguistics.

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

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of ACL-08: HLT*, pages 816–824.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of human language technologies: The 2009 annual conference of the North American* *chapter of the association for computational linguistics*, pages 584–592. Association for Computational Linguistics.

Vahed Qazvinian and Dragomir R Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 689–696. Association for Computational Linguistics.

Vahed Qazvinian and Dragomir R Radev. 2010. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 555–564. Association for Computational Linguistics.

Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The acl anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Henry Small. 2011. Interpreting maps of science using citation context sentiments: a preliminary investigation. *Scientometrics*, 87(2):373–388.

Parikshit Sondhi and ChengXiang Zhai. 2014. A constrained hidden markov model approach for non-explicit citation context extraction. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 361–369. SIAM.

Yuk Wah Wong and Raymond J Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 439–446. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Ozge Yeloglu, Evangelos Milios, and Nur Zincir-Heywood. 2011. Multi-document summarization of scientific corpora. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 252–258. ACM.