Keyphrase Generation with Cross-Document Attention

Shizhe Diao^{♡*}, Yan Song[♠], Tong Zhang[♡]
[♡]The Hong Kong University of Science and Technology
{sdiaoaa, tongzhang}@ust.hk

♦Sinovation Ventures

songyan@chuangxin.com

Abstract

Keyphrase generation aims to produce a set of phrases summarizing the essentials of a given document. Conventional methods normally apply an encoder-decoder architecture to generate the output keyphrases for an input document, where they are designed to focus on each current document so they inevitably omit crucial corpus-level information carried by other similar documents, i.e., the cross-document dependency and latent topics. In this paper, we propose CDKGEN, a Transformerbased keyphrase generator, which expands the Transformer to global attention with crossdocument attention networks to incorporate available documents as references so as to generate better keyphrases with the guidance of topic information. On top of the proposed Transformer + cross-document attention architecture, we also adopt a copy mechanism to enhance our model via selecting appropriate words from documents to deal with outof-vocabulary words in keyphrases. Experiment results on five benchmark datasets illustrate the validity and effectiveness of our model, which achieves the state-of-the-art performance on all datasets. Further analyses confirm that the proposed model is able to generate keyphrases consistent with references while keeping sufficient diversity. The code of CDKGEN is available at https://github. com/SVAIGBA/CDKGen.

1 Introduction

Keyphrases summarize the essential ideas of a document with short and informative text pieces, which are beneficial to many downstream tasks such as text summarization (Liu et al., 2009; Qazvinian et al., 2010), sentiment analysis (Wilson et al., 2005), document categorization (Hammouda et al., 2005; Hulth and Megyesi, 2006), opinion mining (Berend, 2011), and so on.

Existing methods on keyphrase generation can be categorized into two types: extractive (Yang et al., 2017; Zhang et al., 2018; Sun et al., 2019) and generative methods (Meng et al., 2017; Chen et al., 2018; Yuan et al., 2018; Ye and Wang, 2018; Chan et al., 2019; Chen et al., 2019b,a). Compared to extractive methods, generative ones are more challenging since they need to produce, rather than extract, some phrases from the input document, where in most cases those phrases are absent.

Existing generative models for keyphrase generation mainly follow the encoder-decoder paradigm. Meng et al. (2017) firstly adopted the sequenceto-sequence (seq2seq) model for this task and many studies followed this methodology and utilized extra information (Chen et al., 2018; Ye and Wang, 2018; Chen et al., 2019a,b). However, these studies are limited in generating a fixed number of keyphrases. To alleviate this limitation, Yuan et al. (2018); Chan et al. (2019) employed a new training setup by joining all keyphrases to a delimiter-separated sequence and letting the seq2seq model decide the length of the output sequence so that it is able to produce a variable number of keyphrases for different documents. Although the aforementioned studies illustrate their effectiveness in keyphrase generation, they are expected to be enhanced in many aspects. First, in addition to the seq2seq architecture, one could use Transformer-based encoder-decoder for keyphrase generation because it has been proven useful in many similar tasks (Vaswani et al., 2017; Keskar et al., 2019; Khandelwal et al., 2019; Hoang et al., 2019; Liu and Lapata, 2019). Second, appropriately extracting and learning from external knowledge other than only the input document could provide essential help to keyphrase generation. Some

^{*}Work done during the internship at Sinovation Ventures.

¹For simplicity, in the following paper, we use 'keyphrase generation' to refer to generative methods for this task, in contrast to extractive methods or keyphrase extraction.

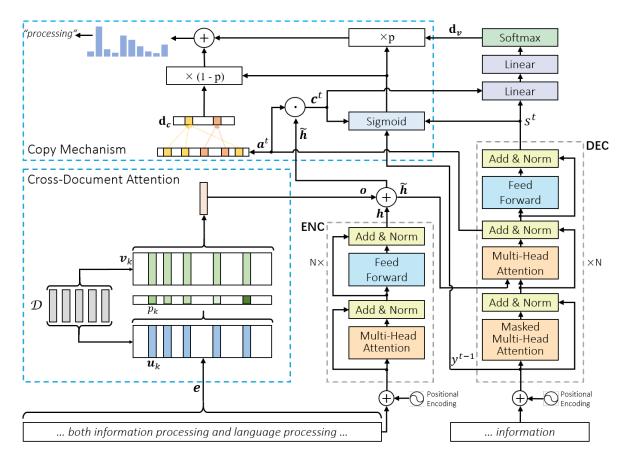


Figure 1: The overall architecture of CDKGEN, where the area marked by dashed box 'ENC' and 'DEC' denote the Transformer encoder and decoder parts, respectively. The figure with examples illustrates the last decoding step (generating the word 'processing') for a keyphrase 'information processing' with a given input document.

studies proved the idea by learning extra information from titles (Chen et al., 2019b), correlations among keyphrases (Chen et al., 2018) and other keyphrases from similar documents (Chen et al., 2019a). However, there is still huge room for improvement, even with this method. This is specially true for the scenarios where there are no titles or reference keyphrases provided. In addition, the self-attention mechanism of existing Transformer architecture is at the token-level for the current document, which is not effective for cross-document dependency. Recent studies find it beneficial to incorporate knowledge by using different level of attention mechanism, such as multi-scale attention (Guo et al., 2019), n-gram attention (Diao et al., 2019), knowledge attention (Zhang et al., 2019a,b) and so on. Therefore, we propose to expand the Transformer to the corpus-level attention.

To address the above aspects for enhancing keyphrase generation, we propose CDKGEN, a Transformer-based keyphrase generator with cross-document attention, where the Transformer serves as the encoder and decoder, and the cross-document attention leverages the latent topics from relevant

documents so as to help the decoder generate better keyphrases. As a result, our model is able to predict topic-dependent keyphrases, especially absent ones, resembling the way that humans might give keyphrases around the same topic. On top of the Transformer + cross-document attention design, we apply the copy mechanism (See et al., 2017) to provide the ability to generate out-of-vocabulary (OOV) ² words by directly selecting words from the input document. To generate a variable number of keyphrases for different documents via an end-to-end manner, we follow Yuan et al. (2018); Chan et al. (2019) to join all keyphrases into a sequence for training CDKGEN and its baselines.

Experimental results illustrate that CDKGEN outperforms all baselines on five benchmark datasets, where the state-of-the-art performance is observed on all datasets compared to previous studies. Particularly, CDKGEN performs well on both present and absent keyphrase prediction, where the comparisons among its different baselines reveal

²OOV herein refers to words which do not appear in the keyphrases in the training data.

the capability of cross-document attention and copy mechanism, respectively. Moreover, further analyses demonstrate that CDKGEN offers an effective solution to keyphrase generation with satisfactory keyphrase number and generation diversity.

2 The Approach

Our approach, CDKGEN, follows the encoder-decoder paradigm, where Transformer is used as the backbone model for encoding and decoding. In addition, we adopt cross-document attention networks in our approach to incorporate the latent topic information from relevant documents and interact with the Transformer. A copy mechanism is applied to enhance the results to tackle the out-of-vocabulary (OOV) problem. The entire architecture of CDKGEN is illustrated in Figure 1. Formally, the overall keyphrase generation process can be described as

$$Y = CDKGEN(d, \mathcal{M}(d, \mathcal{D})) \tag{1}$$

where $d=w_1w_2...w_i...w_n$ is the input document with w_i indicating its words and $Y=kp_1kp_2...kp_j...kp_m$ the output sequence that concatenates all keyphrases kp_j . \mathcal{M} refers to the cross-document attention networks that produce latent topic embedding for CDKGEN from a collection of documents \mathcal{D} according to d. The keyphrase generation is then enhanced with the latent topics provided in \mathcal{D} . Details of the cross-document attention networks and how we integrate it with the Transformer as well as the copy mechanism applied are described in the following subsections.

2.1 Cross-Document Attention

Given the input document d, relevant documents usually share similar topics, which are good references to help determine what could be the optimal keyphrases to describe d. For example, for a document about 'Travel Consultation System', the keyphrase 'Information Retrieval' may be absent in the given document but appear in other relevant documents, which could provide explicit information for keyphrase generation in this scenario.

To represent and exploit the latent topics from relevant documents, we firstly aggregate all documents from a collection (i.e., the union of both training and test set) to the set $\mathcal{D} = \{d_1, d_2, ..., d_k, ..., d_l\}$, and use two vector sets to represent them, i.e., key vectors $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k, ..., \mathbf{u}_l\}$, and value vectors $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k, ..., \mathbf{v}_l\}$ with \mathbf{u}_k and \mathbf{v}_k correspond-

ing to d_k . Specifically, \mathbf{u}_k is used to compute similarity with the input document while \mathbf{v}_k carries d_k 's encoding information for generating the final output, which acts as the latent topic embedding. Then for each input document d, we represent it through its sentential encoding \mathbf{e} and use it as the 'query' vector to address relevant documents. In detail, the addressing operation can be formalized as

$$p_k = \frac{\exp(\mathbf{e}^{\top} \cdot \mathbf{u}_k)}{\sum_{k=1}^{l} \exp(\mathbf{e}^{\top} \cdot \mathbf{u}_k)},$$
 (2)

and for the entire document set \mathcal{D} , we have

$$\mathbf{o} = \sum_{k=1}^{l} p_k \mathbf{v}_k,\tag{3}$$

where o is the output vector of the cross-document attention to represent the latent topics from relevant documents via a weighted encoding.

2.2 Integrating Cross-Document Attention with Transformer

Although RNN based sequence-to-sequence models are widely used for keyphrase generation task, we use Transformer (Vaswani et al., 2017) as the backbone encoder-decoder framework in this paper. This has been proved to have a more effective performance than sequence-to-sequence models in many generation tasks (Vaswani et al., 2017; Keskar et al., 2019; Khandelwal et al., 2019; Hoang et al., 2019; Liu and Lapata, 2019). Once the latent topic embedding o is obtained, we combine it with the Transformer encoding-decoding process via the following steps.

First, the input document is passed through the Transformer encoder which results in a hidden state \mathbf{h}_i for each input token w_i . Then we combine \mathbf{h}_i and \mathbf{o} via element-wise addition $\tilde{\mathbf{h}}_i = \mathbf{h}_i + \mathbf{o}$ and send it to the decoding process through each multihead attention layer to calculate the attention vector $\mathbf{a}^t = \alpha_1^t \alpha_2^t ... \alpha_i^t ... \alpha_n^t$ at each decoding step t. Next, \mathbf{a}^t is used to produce the context vector \mathbf{c}^t , a weighted sum of the encoding hidden states:

$$\mathbf{c}^t = \sum_{i=1}^n \alpha_i^t \tilde{\mathbf{h}}_i. \tag{4}$$

Later \mathbf{c}^t is concatenated with the decoder output \mathbf{s}^t and then fed into two linear layers, followed by a softmax function, to produce the vocabulary

DATASET	Ins	PEC	N	US	KRA	PIVIN	SEM	EVAL	KP	20к
	D#	T#	D#	T#	D#	T#	D#	T#	D#	T#
TRAIN	-	-	-	-	-	-	-	-	513,918	178
DEV	1,500	142	169	167	1,844	180	144	189	19,992	179
TEST	500	138	42	161	460	185	100	207	19,987	179
AVG. KPS	9	.6	1	1.5	5	5.2		5.7	5.3	
ABS. RATE	21.	5%	48	.7%	43.	8%	55	.5%	36.	7%

Table 1: The statistics of the experimental datasets, where '-' means that the particular parts are not used in our experiments; D# refers to document number and T# the average token number per document. AVG. KPS is the average number of target keyphrases per document, and ABS. RATE means the rate of absent ones in all target keyphrases.

distribution for the output word at step t

$$\mathbf{d}_v = \frac{\exp(\mathbf{z}^t)}{\sum_{V} \exp(\mathbf{z}^t)},\tag{5}$$

where $\mathbf{z}^t = W_1(W_2 \cdot (\mathbf{s}^t \oplus \mathbf{c}^t))$, a vector with |V| dimension and V is the predefined vocabulary providing word candidates for keyphrase generation. W_1 and W_2 are trainable parameters for the two aforementioned linear layers, respectively.

2.3 Copy Mechanism

In general, \mathbf{d}_v provides the reference for choosing words from a predefined vocabulary, however it is limited in its coverage of the OOV words excluded from the vocabulary. Copy mechanism offers a feasible solution to this limitation that enables the decoder to directly copy words from an input document, and has proven to be useful in many language generation tasks (Gu et al., 2016; Meng et al., 2017; Paulus et al., 2017; See et al., 2017; Chen et al., 2019b).

In our model, we apply the copy mechanism with a pointer-generator design (See et al., 2017) to leverage appropriate words from the input document. In detail, at time step t, the generation probability p is calculated by the context vector \mathbf{c}^t , the decoder output \mathbf{s}^t , and the last (t-1 step) prediction \mathbf{y}^{t-1} :

$$p = \sigma(\mathbf{W}_c \mathbf{c}^t + \mathbf{W}_s \mathbf{s}^t + \mathbf{W}_y \mathbf{y}^{t-1}), \quad (6)$$

where σ is the sigmoid function and \mathbf{W}_c , \mathbf{W}_s , \mathbf{W}_y are trainable parameters in the sigmoid module.

Therefore, the final prediction of the entire CD-KGEN model at time step t is obtained by:

$$\mathbf{y}^t = \arg\max(p\mathbf{d}_v + (1-p)\mathbf{d}_c), \qquad (7)$$

where \mathbf{d}_c is the copy distribution (a vector with |V'| dimension, where V' is the extended vocabulary³) with its each element calculated by $\sum_{\gamma:w_{\gamma}=w}\alpha_i^t$ $\forall:1\leq\gamma\leq|V'|$. This provides guidance to indicate important words (to be part of keyphrases) in the input document. Note that, to align with \mathbf{d}_c , zero padding is conducted at the end of \mathbf{d}_v to form a |V'|-dimension vector. Therefore in Eq. (7), p serves as a soft switcher to decide the preference of choosing a word from the predefined vocabulary by \mathbf{d}_v or copy a word from input document by \mathbf{d}_c .

3 Experiment Settings

3.1 Datasets

We conduct our experiments on five benchmark datasets, which are mainly from computer science domain and described as follows.

- INSPEC (Hulth, 2003), which contains 2,000 journal paper abstracts with corresponding keyphrases assigned by professional indexers.
- NUS (Nguyen and Kan, 2007), a scientific dataset consisting of 211 full papers with their keyphrases annotated by student volunteers.
- **KRAPIVIN** (Krapivin et al., 2009), consisting of 2,304 full papers from association for computing machinery (ACM) with keyphrases provided by their authors and verified by reviewers.
- SEMEVAL (Kim et al., 2010), which provides 244 full papers with corresponding keyphrases collected from ACM Digital Library.
- KP20K (Meng et al., 2017), which contains around 568K paper abstracts collected from several online resources including ACM Digital Library, ScienceDirect, Wiley, Web of Science, etc.

³Such vocabulary is a combination of the predefined vocabulary and words from the current input document, which ensures the model to choose from more word candidates.

MODEL	Ins	PEC	N	US	KRA	PIVIN	SEM]	EVAL	KP	20к
MIODEL	$F_1@5$	$F_1@10$								
COPYRNN	0.292	0.336	0.342	0.317	0.302	0.252	0.291	0.296	0.328	0.255
CorrRNN	-	-	0.358	0.330	0.318	0.278	0.320	0.320	-	-
KG-KE-KR-M	0.257	0.284	0.289	0.286	0.272	0.250	0.202	0.223	0.317	0.282
MULTI-TASK	0.326	0.309	0.354	0.320	0.296	0.240	0.322	0.289	0.308	0.243
TG-NET	0.315	0.381	0.406	0.370	0.349	0.295	0.318	0.322	0.372	0.315
CATSEQ	0.290	0.300	0.359	0.349	0.307	0.274	0.302	0.306	0.314	0.273
CATSEQ-RL	0.250	-	0.364	-	0.287	-	0.285	-	0.310	-
CATSEQD	0.276	0.333	0.374	0.366	0.325	0.285	0.327	0.352	0.348	0.298
CATSEQD-RL	0.242	-	0.353	-	0.282	-	0.272	-	0.305	-
CATSEQCORR	0.227	-	0.319	-	0.265	-	0.246	-	0.289	-
CATSEQCORR-RL	0.240	-	0.349	-	0.286	-	0.278	-	0.308	-
CATSEQTG	0.229	-	0.325	-	0.282	-	0.246	-	0.292	-
CATSEQTG-RL	0.253	-	0.375	-	0.300	-	0.287	-	0.321	-
TRANSFORMER	0.288	0.291	0.348	0.325	0.292	0.276	0.291	0.302	0.318	0.257
TRANS+COPY	0.297	0.312	0.351	0.337	0.308	0.277	0.322	0.320	0.357	0.268
Trans+cd	0.299	0.334	0.349	0.347	0.311	0.284	0.310	0.321	0.322	0.266
CDKGEN	0.331	0.347	0.412	0.381	0.352	0.304	0.342	0.355	0.381	0.324

Table 2: Present keyphrase prediction results on five benchmark datasets from previous studies and our models. F_1 scores on the top 5 and 10 keyphrases are reported. '-' means the score is not reported (same below in other tables).

Following Meng et al. (2017), we remove duplicate documents from KP20K that appear in other datasets and conduct the same pre-processing on the remaining documents, such as tokenization, lowercasing, replacing all digits with *digit*, and so on. The statistics of the resulting datasets are reported in Table 1.

To ensure consistency with previous work (Meng et al., 2017; Yuan et al., 2018; Chen et al., 2018, 2019b), we only use the title and abstract of each document as the input text via their combination, and concatenate all its keyphrases into a single sequence as the output. Specifically, in the output sequence, keyphrases are organized in order: present keyphrases precede absent ones, where all present keyphrases are rearranged according to their first appearance in the input document, and all absent keyphrases keep their original order. According to the settings in the above studies, we use the training set from KP20K to train different models and evaluate them on the test sets of all five datasets.

3.2 Baselines

The following models are used as the main baselines in our experiments:

- TRANSFORMER: this is the baseline that we use as our backbone encoder-decoder only, that is, a four-layer Transformer model with 8 heads and 768 hidden units without other extensions.
- TRANS+COPY: the Transformer model with the same architecture of the previous one and

- equipped with the copy mechanism, to test how it performs with consideration of OOV words.
- TRANS+CD: the same Transformer model with cross-document attention to test how it helps in relevant documents for this task.

To further demonstrate the effectiveness of our model, we compare it with existing models from previous studies, including COPYRNN (Meng et al., 2017), CORRRNN (Chen et al., 2018), KB-KE-KR-M (Chen et al., 2019a), MULTI-TASK (Ye and Wang, 2018), TG-NET (Chen et al., 2019b) with their reported results on the benchmark datasets, as well as the performance of CATSEQ and CATSEQD from Yuan et al. (2018) and Chan et al. (2019), CATSEQCORR and CATSEQTG from Chan et al. (2019). In addition, we also compare with the reinforcement learning implementation (Chan et al., 2019) of the aforementioned CATSEQ* models⁴.

3.3 Evaluation Metrics

Following Meng et al. (2017) and Yuan et al. (2018), we adopt macro-averaged precision, recall and F-measure (F_1) as evaluation metrics by comparing the top k predicted keyphrases with the ground-truth keyphrases. In our experiments, k is set to be 5, 10, M and O, where M and O are variable cutoffs which are equal to the number of predictions and ground-truth keyphrases, respectively. Similar to previous work (Meng et al., 2017;

⁴Denoted with suffix '-RL' in the rest of the paper.

MODEL	Ins		NU		KRAI		SEMI		KP	
	$F_1@M$	$F_1@O$								
CATSEQ	0.262	0.307	0.397	0.383	0.354	0.324	0.283	0.310	0.367	0.319
CATSEQ-RL	0.300	-	0.426	-	0.362	-	0.327	-	0.383	-
CATSEQD	0.263	0.331	0.394	0.406	0.349	0.371	0.274	0.357	0.363	0.357
CATSEQD-RL	0.292	-	0.419	-	0.360	-	0.316	-	0.379	-
CATSEQCORR	0.269	-	0.390	-	0.349	-	0.290	-	0.365	-
CATSEQCORR-RL	0.291	-	0.414	-	0.369	-	0.322	-	0.382	-
CATSEQTG	0.270	-	0.393	-	0.366	-	0.290	-	0.366	-
CATSEQTG-RL	0.301	-	0.433	-	0.369	-	0.329	-	0.386	-
TRANSFORMER	0.256	0.247	0.369	0.371	0.357	0.318	0.277	0.329	0.334	0.287
TRANS+COPY	0.282	0.279	0.402	0.385	0.359	0.331	0.299	0.327	0.377	0.322
TRANS+CD	0.279	0.301	0.411	0.409	0.361	0.352	0.311	0.338	0.385	0.337
CDKGEN	0.305	0.334	0.435	0.412	0.372	0.375	0.329	0.359	0.398	0.361

Table 3: Present keyphrase prediction results on five benchmark datasets from previous studies and our models. F_1 scores on the top M and O keyphrases are reported, where M and O are variable cut-offs, which are equal to the number of predictions and ground-truth keyphrases, respectively. Note that we do not include COPYRNN, CORRRNN, KG-KE-KR-M, MULTI-TASK and TG-NET in the table since they did not conduct this evaluation.

Yuan et al., 2018; Chen et al., 2018, 2019b), we apply Porter Stemmer⁵ to obtain word stems for keyphrases to facilitate evaluation.

3.4 Implementation

We implement a Transformer structure similar to Vaswani et al. (2017), with 4 layers and 8 selfattention heads, 768 dimensions for hidden states, 768 for maximum input length, and random initialization. For cross-document attention, we utilize sentence-transformer (Reimers and Gurevych, 2019) to initialize key vectors \mathbf{u}_k and value vectors \mathbf{v}_k in order to guarantee reliable addressing as a warm start for those vectors and they are updated during the training process. Different from \mathbf{u}_k and \mathbf{v}_k , the sentential encoding \mathbf{e} of each input document d is represented as the average of its word representations which are randomly initialized to ensure their compatibility with the backbone Transformer's vector space during training. In the training stage, we choose the top 50,000 frequent words to form the predefined vocabulary and set the embedding dimension to 768. We adopt Adam as the optimizer with a learning rate of 0.0001 and a dropout rate of 0.5. We use beam search to generate multiple phrases and set beam size to 50 and maximum sequence length to 40.

4 Experimental Results

In this section, we compare our model with baselines and existing studies on the experimental datasets. The performance comparison for present and absent keyphrase prediction, as well as with existing studies, are presented in the following three subsections, respectively.

4.1 Present Keyphrase Prediction

The results on present keyphrase prediction are reported in Tables 2 and 3, with several observations.

First, CDKGEN achieves the best performance over all baselines, which indicates the advantage of incorporating cross-document attention and copy mechanism into the Transformer. For example, in both fixed and variable cut-off settings, CD-KGEN outperforms TRANSFORMER with significant improvements. Second, comparisons between TRANSFORMER and TRANS+COPY, as well as TRANS+CD and CDKGEN, confirm the effectiveness of the copy mechanism, similar to that observed in Meng et al. (2017); Yuan et al. (2018); Chen et al. (2018, 2019b), where TRANS+COPY and CDKGEN show a consistent improvement over TRANSFORMER and TRANS+CD, respectively. Since generating present keyphrases mainly requires a model having stronger 'extraction' abilities, a copy mechanism thus provides an effective solution to fulfilling this requirement especially for those present keyphrases with their words which also appear in the input document. When comparing TRANS+COPY v.s. TRANSFORMER, and CD-KGEN v.s. TRANS+CD, it is observed that the performance gains from copy mechanism on INSPEC and KP20K are larger than that of the other three datasets. This is because there are fewer present keyphrases in NUS, KRAPIVIN and SEMEVAL, so as their contained words; copy mechanism is not able to copy appropriate words to the output.

Third, the cross-document attention is proved

⁵https://www.nltk.org/_modules/nltk/ stem/porter.html

Model	INSPEC			NUS		KRAPIVIN		SEMEVAL		KP20K	
WIODEL	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50	
COPYRNN	0.051	0.101	0.078	0.144	0.116	0.195	0.049	0.075	0.115	0.189	
CorrRNN	-	-	0.059	-	0.108	-	0.041	-	-	-	
CATSEQ	0.028	0.029	0.037	0.031	0.070	0.074	0.025	0.025	0.060	0.062	
CATSEQD	0.052	0.071	0.084	0.110	0.120	0.145	0.046	0.063	0.117	0.151	
MULTI-TASK	0.022	-	0.013	-	0.021	-	0.006	-	0.021	-	
TG-NET	0.063	0.115	0.075	0.137	0.146	0.253	0.045	0.076	0.156	0.268	
TRANSFORMER	0.044	0.098	0.067	0.132	0.077	0.189	0.044	0.055	0.109	0.155	
TRANS+COPY	0.053	0.106	0.081	0.147	0.082	0.193	0.051	0.079	0.121	0.178	
Trans+cd	0.067	0.108	0.081	0.152	0.119	0.199	0.053	0.081	0.133	0.211	
CDKGEN	0.068	0.117	0.087	0.155	0.151	0.257	0.056	0.088	0.166	0.273	

Table 4: Absent keyphrase prediction results on five benchmark datasets from previous studies and our models. Recall on the top 10 and 50 generated keyphrases are reported.

to be useful for keyphrase generation where TRANS+CD and CDKGEN present significantly improved results over their counterparts without using the cross-document attention. For example, the cross-document attention brings an average of 3.70% $F_1@O$ improvement on all datasets for TRANS+CD over TRANSFORMER, and 3.94% F_1 @10 improvement for CDKGEN over TRANS+COPY, respectively. Particularly, CD-KGEN shows higher performance gain on INSPEC and KRAPIVIN than on NUS, which may be the result of the different annotation quality among different datasets. INSPEC and KRAPIVIN are annotated by more professional annotators than NUS, and those annotators may consider more information according to relevant documents to ensure the consistency of the keyphrases for different documents. Overall, the above observations illustrate that using both cross-document attention and copy mechanism gives a synergistic effect over baselines and the two components are effectively connected to complement each other for present keyphrase generation.

4.2 Absent Keyphrase Prediction

The ability to generate absent keyphrases is the essential feature of generative models. We compare CDKGEN with its baselines on five benchmark datasets with respect to two evaluation metrics, namely, recall and F_1 , and report their results in Tables 4 and 5, respectively. Several observations can be made. First, CDKGEN outperforms all baselines in terms of both recall and F_1 , with obvious improvement over the best baseline model (i.e. TRANS+CD). It is observed that on the KRAPIVIN dataset, CDKGEN achieves the biggest improvement. The underlying reason is that KRAPIVIN contains many more candidate words

for absent keyphrases in relevant documents, so that the cross-document attention helps with their support. Second, removing the copy mechanism generally does not hurt the performance. The reason is rather straightforward because the copy mechanism is only able to choose present words in the input document while those words may not be included in absent keyphrases.⁶ Third, similar to the present keyphrase generation, cross-document attention provides significant improvement for absent keyphrases. It is a direct evidence that relevant documents help the decoding process choose appropriate words to form keyphrases that do not directly correspond to the current document. These results further demonstrate the generalization capability of CDKGEN.

4.3 Comparison with Existing Studies

We also compare CDKGEN and its baselines with existing models on the same datasets, with their results property at the upper parts of Tables $2\sim5$. There are several comparisons drawn from different aspects. First, Transformer confirms its superiority to sequence-to-sequence structures in this task. The comparison between TRANS+COPY and COPY-RNN clearly illustrates that the encoding-decoding process implemented by Transformer has better results on both the present and absent keyphrases. This is aligned with the observations in other studies also using Transformer (Vaswani et al., 2017; Keskar et al., 2019; Khandelwal et al., 2019; Hoang et al., 2019; Liu and Lapata, 2019). Second, it is proved that directly using cross-document at-

⁶Many absent keyphrases are from the reorganization of existing words from the predefined vocabulary.

⁷The results are directly cited from their papers. Note that not all evaluation metrics are available for each model because they are tested with different criteria in each paper. For example, in Table 5, existing models evaluated with F_1 do not report their recall scores as those in Table 4.

MODEL	INSPEC		NU	NUS		KRAPIVIN		SEMEVAL		КР20к	
MODEL	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	$F_1@5$	$F_1@M$	
CATSEQ	0.004	0.008	0.016	0.028	0.018	0.036	0.020	0.028	0.015	0.032	
CATSEQ-RL	0.009	0.017	0.019	0.031	0.026	0.046	0.018	0.027	0.024	0.047	
CATSEQD	0.007	0.011	0.014	0.024	0.018	0.037	0.016	0.024	0.015	0.031	
CATSEQD-RL	0.010	0.021	0.022	0.037	0.026	0.048	0.021	0.030	0.023	0.046	
CATSEQCORR	0.005	0.009	0.014	0.024	0.020	0.038	0.018	0.026	0.015	0.032	
CATSEQCORR-RL	0.010	0.020	0.022	0.037	0.022	0.040	0.021	0.031	0.022	0.045	
CATSEQTG	0.005	0.011	0.011	0.018	0.018	0.034	0.019	0.027	0.015	0.032	
CATSEQTG-RL	0.012	0.021	0.019	0.031	0.030	0.053	0.021	0.030	0.027	0.050	
TRANSFORMER	0.002	0.007	0.018	0.022	0.019	0.033	0.021	0.021	0.011	0.035	
Trans+copy	0.005	0.010	0.019	0.029	0.022	0.037	0.022	0.025	0.018	0.037	
Trans+cd	0.007	0.015	0.024	0.037	0.023	0.039	0.022	0.030	0.021	0.044	
CDKGEN	0.015	0.022	0.024	0.038	0.033	0.057	0.024	0.033	0.031	0.052	

Table 5: Absent keyphrase prediction results on five benchmark datasets from previous studies and our models. F_1 scores on the top 5 and M keyphrases are reported, where M is a variable cut-off equal to the number of predictions.

tention is more effective than other complicated training and decoding strategies. Compared with those models trained by reinforcement learning, CDKGEN outperforms most of them, especially on NUS and SEMEVAL datasets which have more target keyphrases per document. Considering that reinforcement learning approaches perform much better than other existing models because they are encouraged to generate more keyphrases with an adaptive reward, CDKGEN achieves the same goal with a much more efficient solution. Similarly, compared with CATSEQ, CATSEQD which uses several decoding techniques such as semantic coverage mechanism, CDKGEN shows a consistent improvement on all five datasets.

Third, CDKGEN outperforms the models utilizing extra information, e.g., CORRRNN, CATSEQ-CORR, TG-NET and CATSEQTG. This indicates that integrating relevant documents has some advantages over keyphrase correlations or titles. Compared with MULTI-TASK which uses the information of both labeled data and large-scale unlabeled data, CDKGEN shows remarkable improvements over it and the observation demonstrates that the end-to-end design of learning from relevant documents is better than that of tagged unlabeled documents with propagated errors.

Compared with KG-KE-KR-M which uses the keyphrases of similar documents, the performance of CDKGEN suggests that exploiting documents rather than their labels is more effective and has less annotation requirements.

5 Analyses

We analyze several aspects of CDKGEN and its baselines regarding their generation results. The details are illustrated in this section.

MODEL	PRES	SENT	ABS	ENT
	MAE	AVG.#	MAE	AVG.#
ORACLE	0.000	2.837	0.000	2.432
CATSEQ	2.271	3.781	1.943	0.659
CATSEQD	2.225	3.694	1.961	0.629
CATSEQCORR	2.292	3.790	1.914	0.703
CATSEQTG	2.276	3.780	1.956	0.638
CATSEQ-RL CATSEQD-RL CATSEQCORR-RL CATSEQTG-RL	2.118	3.733	1.494	1.574
	2.087	3.666	1.541	1.455
	2.107	3.696	1.557	1.409
	2.204	3.865	1.439	1.749
TRANSFORMER TRANS+COPY TRANS+CD CDKGEN	2.477	3.766	1.798	1.125
	2.335	3.696	1.667	1.247
	2.233	3.689	1.543	1.453
	2.004	3.655	1.411	1.797

Table 6: Evaluations of predicting the correct number of keyphrases on the KP20k validation set. MAE denotes the mean absolute error and Avg. # the average number of generated keyphrases. For both MAE and Avg. #, the closer a model is to ORACLE the better it performs.

5.1 Number of Generated Keyphrase

In addition to the performance evaluation by F_1 or recall scores, an important criterion for generative models is to investigate how many keyphrases are generated, especially when one uses keyphrase sequence as the decoding target. In doing so, we follow Chan et al. (2019) to use mean absolute error (MAE) to calculate the difference between the prediction and ground-truth (oracle) keyphrase numbers, where a lower MAE refers to better generation performance. We also list the average number of generated keyphrases to evaluate how close such a number is with respect to the oracle one. The results from different models on the KP20K validation set are shown in Table 6. In general, CD-KGEN has the lowest MAE on both present and absent keyphrases, where it outperforms all base-

MODEL	Ins	NUS	KR	SE	KP
TRANSFORMER	10.11	12.10	12.44	14.33	10.10
TRANS+COPY	12.49	13.11	13.40	15.22	12.33
Trans+cd	25.82	26.75	22.88	26.71	19.22
CDKGEN	32.74	33.48	26.48	29.09	23.94

Table 7: The average unique predictions from different models on all datasets. INS, KR, SE and KP denote INSPEC, KRAPIVIN, SEMEVAL and KP20K, respectively.

line models as well as the previous best models (i.e. CATSEQD-RL and CATSEQTG-RL). As for the average number, CDKGEN also shows the closest number to the oracle one, especially on absent keyphrases where models with cross-document attention show significantly better results and are comparative to the reinforcement learning methods, which are designed particularly to encourage the model to generate the correct number of diversified keyphrases. Compared to such methods, our model is much more efficient without requiring a complex training procedure.

5.2 Generation Diversity

Another important criterion to evaluate generative models is the diversity of generated keyphrases. To assess with respect to such criterion, we follow Yuan et al. (2018) to calculate the average unique predictions and visualize the decoding states from different models on all experimental datasets.

The results of the average unique predictions are reported in Table 7. In general cross-document attention helps to generate more diversified keyphrases so that TRANS+CD has more unique predictions than TRANSFORMER and TRANS+COPY, which is not surprising because cross-document attention enlarges the reference by relevant documents. Of all models, CDKGEN has the most unique predictions, which is a further diversified decoding process via combining cross-document attention and copy mechanism.

The visualization for CDKGEN and its baseline models are presented in Figure 2. Following Yuan et al. (2018), we randomly sample 2,000 input documents in the KP20K validation set and run different models on them. We then use t-SNE (Maaten and Hinton, 2008) to produce the 2D plots of the decoding states (vectors) from the last layer of the decoder at the first, second and third steps. It is clearly shown that the states from TRANS+CD and CDKGEN tends to be clustered into several groups while there is no obvious cluster for that

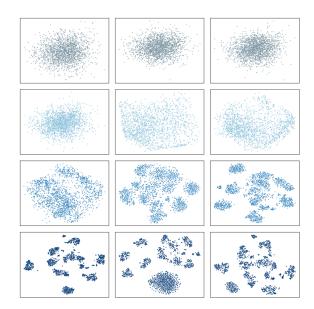


Figure 2: t-SNE plots of decoding states for 2,000 documents from the KP20k validation set. Rows from top to bottom represent the results from TRANSFORMER, TRANS+COPY, TRANS+CD, and CDKGEN, respectively. In each row, plot 1 to 3 demonstrate the decoding states from the 1st, 2nd and 3rd step, respectively.

from TRANSFORMER and TRANS+COPY. This suggests that cross-document attention provides useful information to diversify the decoding process so as to generate different keyphrases.

5.3 Case Study

To further analyze how keyphrases are generated, we perform a case study on an example document about 'travel consultation system'. Figure 3 shows the input document, the most relevant documents (according to p_k from the cross-document attention), target the present and absent keyphrases for the input document and the predictions from different models. It is observed that the relevant documents contain the target keyphrases (i.e. 'language processing' and 'information retrieval') which are highly related to the topic. Models with crossdocument attention are able to generate them and others cannot. For present keyphrases, CDKGEN can generate more targets than others, which shows its ability to capture the right keyphrases on the same topic (i.e. information processing) with the help of cross-document attention. Specifically, for TRANSFORMER and TRANS+COPY, their predictions on both present and absent keyphrases are not satisfactory. This illustrates that only using Transformer and the input document is not enough for effective keyphrase generation.

Title: A study on meaning processing of dialogue with an example of development of travel consultation system.

Abstract: This paper describes an approach to processing meaning instead of processing information in computing. Human intellectual activity is supported by linguistic activities in the brain. Therefore, processing the meaning of language ... user and retrieve information through dialogue. Through a simulation example of the system, we show that both information processing and language processing are integrated.

Ground-truth Keyohrases:

linguistic activities; human intellectual activity; meaning processing; information processing;

travel consultation dialogue system; language processing; information retrieval; user utterance understanding;

Relevant Document 1:

Title: A new approach to intranet search based on information extraction.

Abstract: This paper is concerned with 'intranet search'. By intranet search, we mean searching for information on ... persons, experts, and homepages. Traditional information retrieval only focuses on search of relevant documents, ... This paper describes the architecture, features, component technologies, and evaluation results of the system.

Relevant Document 2:

Title: An effective statistical approach to blog post opinion retrieval.

Abstract: Finding opinionated blog posts is still an open problem in information retrieval, as exemplified by the recent trec blog tracks. Most of the current solutions involve the use of external resources and manual efforts in identifying subjective ... an effective performance in retrieving opinionated blog posts, which is as good as a computationally expensive approach using natural language processing techniques.

Present Keyphrases

Target: meaning processing; information processing; language processing; travel consultation dialogue system;

linguistic activities; human intellectual activity;

TRANSFORMER: meaning processing; dialogue system; intellectual activity; TRANS+COPY: information processing; dialogue system; human intelligence;

TRANS+CD: meaning processing; information processing; information;

CDKGEN: meaning processing; information processing; language processing; consultation dialogue system;

Absent Keyphrases

Target: information retrieval; user utterance understanding; TRANSFORMER: learning framework; dialogue processing; TRANS+COPY: travel dialogue system; travel time;

TRANS+CD: information retrieval; natural language processing; dialogue system;

CDKGEN: information retrieval; natural language processing;

Figure 3: Examples of an input document about 'travel consultation system' with its two most relevant documents (selected by the cross-document attention) and the target keyphrases with predictions from different models. The keyphrases presented in blue are successful predicted results and their corresponding source text.

6 Related Work

Keyphrase generation mainly consists of two methodology streams, extractive and generative approaches. There is a large body of research focusing on extracting keyphrases from documents (Hulth, 2003; Mihalcea and Tarau, 2004; Witten et al., 2005; Wu et al., 2005; Nguyen and Kan, 2007; Medelyan et al., 2008, 2009; Wan and Xiao, 2008; Grineva et al., 2009; Liu et al., 2011; Wang et al., 2016; Le et al., 2016; Zhang et al., 2016; Luan et al., 2017). Compared to extractive approaches, generative ones have attracted more attention in recent years for their ability to predict absent keyphrases for an input document. For example, Meng et al. (2017) proposed CopyRNN, which is an early study with attention and copy mechanism. Chen et al. (2018) took correlation among multiple keyphrases into consideration to eliminate duplicate keyphrases. To further enhance keyphrase generation, other studies tried to utilize extra information: Ye and Wang (2018) proposed to assign synthetic keyphrases to unlabeled documents and then use them to enlarge the training data; Chen et al. (2019a) retrieved similar documents from the training data for the input document and encoded their keyphrases as external knowledge, while Chen et al. (2019b) leveraged title information for this task. To increase the diversity of keyphrases, a reinforcement learning approach is introduced by Chan et al. (2019) to encourage their model to generate the correct number of keyphrases with an adaptive reward. Although existing models are capable of predicting both present and absent keyphrases, there is still potential to facilitate keyphrase generation with unlabeled data such as relevant documents. In doing so, CDKGEN offers a more effective and efficient solution.

7 Conclusion

In this paper, we proposed CDKGEN, a keyphrase generator based on the Transformer with cross-document attention and the copy mechanism, and compared it to several baselines on different benchmark datasets. The main contributions are as follows. First, we proposed cross-document attention to learn from relevant documents to enhance keyphrase generation. Second, we designed CD-KGEN to integrate the proposed cross-document attention with the Transformer and the copy mechanism. CDKGEN achieved the state-of-the-art performance on five widely used benchmark datasets, which demonstrates its strong capability to generate highly accurate and diversified keyphrases.

References

- Gábor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170.
- Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174.
- Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase Generation with Correlation Constraints. In *Proceedings of the 2018* Conference on Empirical Methods in Natural Language Processing, pages 4057–4066.
- Wang Chen, Hou Pong Chan, Piji Li, Lidong Bing, and Irwin King. 2019a. An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2846–2856.
- Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R Lyu. 2019b. Title-Guided Encoding for Keyphrase Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6268–6275.
- Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. *arXiv* preprint arXiv:1911.00720.
- Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting Key Terms from Noisy and Multitheme Documents. In *Proceedings of the 18th international conference on World wide web*, pages 661–670.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings* of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1631–1640.
- Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. 2019. Multi-Scale Self-Attention for Text Classification. *arXiv* preprint *arXiv*:1912.00544.
- Khaled M Hammouda, Diego N Matute, and Mohamed S Kamel. 2005. Corephrase: Keyphrase Extraction for Document Clustering. In *International workshop on machine learning and data mining in pattern recognition*, pages 265–274.
- Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. 2019. Efficient Adaptation of Pretrained Transformers for Abstractive Summarization. *arXiv preprint arXiv:1906.00138*.

- Anette Hulth. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.
- Anette Hulth and Beáta B Megyesi. 2006. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 537–544.
- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A Conditional Transformer Language Model for Controllable Generation. arXiv preprint arXiv:1909.05858.
- Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. 2019. Sample Efficient Text Summarization Using a Single Pre-Trained Transformer. *arXiv* preprint arXiv:1905.08836.
- Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.
- Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large Dataset for Keyphrases Extraction. Technical report.
- Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. 2016. Unsupervised Keyphrase Extraction: Introducing New Kinds of Words to Keyphrases. In *Australasian Joint Conference on Artificial Intelligence*, pages 665–671.
- Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised Approaches for Automatic Keyword Extraction using Meeting Transcripts. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 620–628.
- Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.
- Zhiyuan Liu, Xinxiong Chen, Yabin Zheng, and Maosong Sun. 2011. Automatic Keyphrase Extraction by Bridging Vocabulary Gap. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 135–144.
- Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651.

- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.
- Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1318–1327.
- Olena Medelyan, Ian H Witten, and David Milne. 2008. Topic Indexing with Wikipedia. In *Proceedings of the AAAI WikiAI workshop*, volume 1, pages 19–24.
- Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing Order into Text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase Extraction in Scientific Publications. In *International conference on Asian digital libraries*, pages 317–326.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv preprint arXiv:1705.04304*.
- Vahed Qazvinian, Dragomir Radev, and Arzucan Özgür. 2010. Citation Summarization Through Keyphrase Extraction. In *Proceedings of the 23rd international conference on computational linguistics* (COLING 2010), pages 895–903.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3973–3983.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. DivGraphPointer: A Graph Pointer Network for Extracting Diverse Keyphrases. In *Proceedings of the 42nd International ACM SI-GIR Conference on Research and Development in Information Retrieval*, page 755–764.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All

- You Need. In Advances in neural information processing systems, pages 5998–6008.
- Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence Volume 2*, page 855–860.
- Minmei Wang, Bo Zhao, and Yihua Huang. 2016. Ptr: Phrase-based Topical Ranking for Automatic Keyphrase Extraction in Scientific Publications. In *International Conference on Neural Information Processing*, pages 120–128.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354.
- Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical Automated Keyphrase Extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pages 129–152.
- Yi-fang Brook Wu, Quanzhi Li, Razvan Stefan Bot, and Xin Chen. 2005. Domain-specific Keyphrase Extraction. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 283–284.
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050.
- Hai Ye and Lu Wang. 2018. Semi-Supervised Learning for Neural Keyphrase Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4142–4153.
- Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2018. One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. *arXiv preprint arXiv:1810.05241*.
- Hongming Zhang, Yan Song, and Yangqiu Song. 2019a. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 872–881.
- Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019b. Knowledge-aware Pronoun Coreference Resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 867–876.

- Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 836–845.
- Yingyi Zhang, Jing Li, Yan Song, and Chengzhi Zhang. 2018. Encoding Conversation Context for Neural Keyphrase Extraction from Microblog Posts. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1676–1686.