Received: 11 September 2022 | Revised: 31 January 2023 | Accepted: 25 February 2023

DOI: 10.1002/asi.24749

RESEARCH ARTICLE

JASIST WILEY

Generating keyphrases for readers: A controllable keyphrase generation framework

Yi Jiang[1,2] | Rui Meng[3] | Yong Huang[1,2] | Wei Lu[1,2] | Jiawei Liu[1,2]

[1]School of Information Management, Wuhan University, Wuhan, Hubei, China

[2]Information Retrieval and Knowledge Mining Laboratory, Wuhan University, Wuhan, Hubei, China

[3]School of Computing and Information, University of Pittsburgh, Pittsburgh, Pennsylvania, USA

Correspondence
Wei Lu, School of Information Management, Wuhan University, Wuhan, Hubei, China.
Email: weilu@whu.edu.cn

Funding information
Key Project of the National Natural Science Foundation of China, Grant/Award Number: No.72234005

Abstract

With the wide application of keyphrases in many Information Retrieval (IR) and Natural Language Processing (NLP) tasks, automatic keyphrase prediction has been emerging. However, these statistically important phrases are contributing increasingly less to the related tasks because the end-to-end learning mechanism enables models to learn the important semantic information of the text directly. Similarly, keyphrases are of little help for readers to quickly grasp the paper's main idea because the relationship between the keyphrase and the paper is not explicit to readers. Therefore, we propose to generate keyphrases with specific functions for readers to bridge the semantic gap between them and the information producers, and verify the effectiveness of the keyphrase function for assisting users' comprehension with a user experiment. A controllable keyphrase generation framework (the CKPG) that uses the keyphrase function as a control code to generate categorized keyphrases is proposed and implemented based on Transformer, BART, and T5, respectively. For the Computer Science domain, the Macro-avgs of P@5, R@5, and F_1@5 on the Paper with Code dataset are up to 0.680, 0.535, and 0.558, respectively. Our experimental results indicate the effectiveness of the CKPG models.

1 | INTRODUCTION

As important metadata summarizing the core contents of a document, keyphrases are intended to index a document and enable readers quickly to find it and identify whether or not it is relevant to their specific needs or interests, which can also help to improve the visibility and influence of the documents (Gbur & Trumbo, 1995; Hartley & Kostoff, 2003; Turney, 2002). Meanwhile, because of their high importance and abstractness in the documents, keyphrases are also regarded as a suitable representation of topics, concepts, and knowledge, and are widely used for topic evolution study, knowledge mining, information retrieval, text summarization, and other NLP tasks (Cheng et al., 2020; Firoozeh et al., 2020; Hernandez-Castaneda et al., 2020; Hu et al., 2019; K. Lu & Kipp, 2014; W. Lu et al., 2019, 2021; Sesagiri Raamkumar et al., 2017; Sun et al., 2020; Yang et al., 2022; Yoon et al., 2018). Owing to the importance and good performance of keyphrases regarding these tasks, automatic keyphrases prediction, including keyphrase extraction and generation, based on different algorithms, has attracted extensive attention (Çano & Bojar, 2019; Hasan & Ng, 2014). However, with the development of NLP technologies, especially deep learning, the role of keyphrases is becoming increasingly less obvious in these related tasks. The neural network model can produce a semantic representation of the important information in the text using the end-to-end learning mechanism, which is far more useful and valuable for the task than the statistical importance that keyphrases can provide. The same applies to readers. Simply using phrases

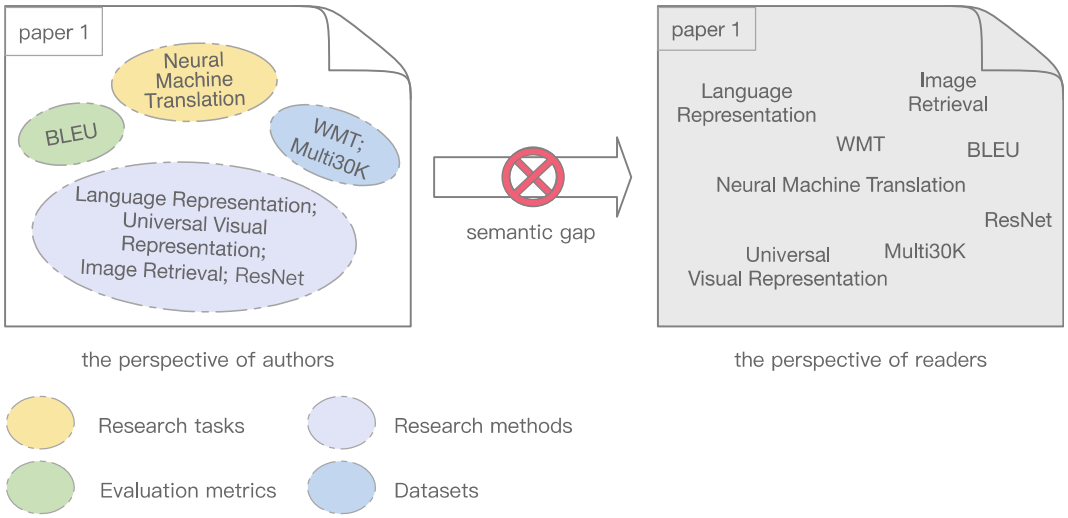

FIGURE 1 Information differences related to keyphrases between authors and readers.

that are of higher importance is insufficient to enable readers to grasp the main idea of the paper quickly, because the semantic information is more important for their comprehension. It seems that keyphrases, at present, can contribute little to either the downstream tasks or serving readers. As a result, the necessity of keyphrases and automatic keyphrase prediction, to some extent, appears debatable. Moreover, some publishers, journals, and conferences no longer require authors to provide keyphrases for their papers, such as Wiley, JASIST, ACL, and AAAI. Thus, it is time to reconsider the value of keyphrases. From the start, keyphrases were intended to serve readers first and foremost. Meanwhile, as a vital medium of scientific knowledge dissemination, scientific papers, along with their keyphrases, are produced to enable researchers to share their ideas and findings. Hence, more attention should be paid to the meaningful role that keyphrases may play in assisting readers' comprehension and, at the same time, the task of automatic keyphrase prediction should also be re-examined from the perspective of helping readers to understand papers.

One important reason why keyphrases perform unsatisfactorily in terms of assisting readers' comprehension is that, under the current keyphrase schema, keyphrases are regarded as equal, and authors do not provide any information about the relationship between the keyphrases and the paper apart from simply stating the keyphrases themselves. As shown in Figure 1, the kinds of "role" boundaries between keyphrases are explicit to the author but implicit to readers; that is, the keyphrases' information received by readers is incomplete. To some extent, there is a semantic gap between authors and readers regarding keyphrase comprehension. Authors, the most common keyphrase annotators of academic papers, are better able to comprehend each keyphrase than readers, for they understand the role of each keyphrase clearly, but whether or not readers can make sense of the key points of the paper quickly and accurately via reading the keyphrases alone largely depends on the semantics of these short phrases. Regretfully, not all keyphrases are able to convey their full message in a specific context. For the keyphrase example in Figure 1, we might guess that both "*Neural Machine Translation*" and "*Image Retrieval*" are the main research tasks, for image retrieval is indeed a common task. However, "*Image Retrieval*" is the core technique of the proposed "*Universal Visual Representation*" method. In addition, for those with little domain knowledge, making sense of what acronyms such as "*WMT*" and "*BLEU*" mean in the paper might be challenging, and they might even mistake the dataset "*WMT*" as a subtask of *M*achine *T*ranslation. Under these circumstances, reading keyphrases is clearly no longer an effective or efficient way to grasp a paper's main ideas quickly, as it is difficult to distinguish a keyphrase's specific meaning purely from the phrase itself, particularly within a short time. Clearly, the semantic gap between readers and authors is the main obstacle to readers' rapid comprehension and also a major drawback regarding the current keyphrase schema.

Unlike keyphrases, entities have their own phrases and entity categories to which they belong, such as: research question, research method, dataset, and evaluation metric (Hou et al., 2019; Jain et al., 2020; Luan et al., 2018). They have clearer semantics, for the entity label clearly informs readers what kind of knowledge resource they are and what role they play in the paper. Several researchers point out that identifying the

semantic roles of scientific concepts can answer some questions quickly, such as "*what methods were proposed or improved for solving a particular problem?*" (Gupta & Manning, 2011; Tsai et al., 2013; W. Lu et al., 2019). Inspired by these views, we suggest labeling keyphrases with their specific function, like "Term Function" in Cheng (2015), to highlight the relevance of the core content and the keyphrase. This will make it far easier for readers to understand the main topic of a study within a short time. For example, given exact keyphrase functions, we would no longer feel confused about the role of "Image Retrieval," which would increase our comprehension of acronyms, such as the dataset "*WMT*" and the metric "*BLEU*." Even if we desire more detailed information, we could also find relevant resources in the document as soon as possible according to their role.

In this study, we propose to generate keyphrases with specific functions to help readers quickly understand the main idea of a paper. Firstly, a user experiment is conducted and the results show that keyphrases with function labels are really useful in assisting readers' comprehension. Then, we propose an end-to-end framework, the controllable keyphrase generation framework (the CKPG), which uses the keyphrase function as a control code and generates keyphrases for the specified category. To verify the effectiveness and feasibility of the CKPG, we take the Compute Science (CS) domain as an example and summarize a keyphrase function schema that divide the CS keyphrase function into five categories: Field, Task, Method, Dataset, and Metric. Further, we implement CKPG models based on Transformer (Vaswani et al., 2017), BART (Lewis et al., 2019), and T5 (Raffel et al., 2020), respectively. A dataset with similar keyphrase function categories based on papers extracted from the Paper with Code (PwC) is constructed. Experiments are conducted on this dataset and satisfactory performance is achieved. What's more, the proposed CKPG method shows its superiority when compared with the two-stage extraction method.

In the remainder of this article, we first review the related work in Section 2 and conduct a user experiment in Section 3. Then, we introduce the CKPG framework in Section 4, and describe the experiment in Section 5. Our discussion is presented in Section 6. Finally, Section 7 concludes the article.

2 | RELATED WORK

2.1 | Term function in scientific texts

Term function refers to the semantic role that a term or a phrase plays in a scientific text, revealing the specific aspect of the paper to which the term or phrase is relevant (Cheng, 2015; W. Lu et al., 2019). Different from bibliometric methods, identifying the term function by analyzing the scientific text itself can help to identify on which topic or task a paper focuses, what materials and methods are used, and other typical research questions (Augenstein et al., 2017; W. Lu et al., 2019; Tsai et al., 2013). Many methods have been proposed to recognize those important academic terms with certain functions. Kondo et al. (2011) used machine learning to extract the "Head," "Method," and "Goal" from research papers' titles based on the structure of the title. Later, Nanba et al. (2010) constructed a system which can recognize the application of elemental "Technologies" and their "Effects" in research papers and patents, providing a useful tool for researchers to grasp the outline of the technical trends in a certain field. Based on semantic extraction patterns, Gupta and Manning (2011) extracted the concepts of "Focus," "Technique," and "Domain" from the abstract to characterize a research work. Tsai et al. (2013) identified and categorized the scientific concepts as a way to understand the research literature of a scientific community in depth, and proposed an unsupervised bootstrapping algorithm to recognize the two categories of concepts, "Technique" and "Application." Cheng (2015) constructed a term function framework for an academic text, in which the term functions were classified into domain-independent and domain-specific. The domain-independent term function, including "Question" and "Method," was proposed from the perspective of the common process of scientific research, while the domain-specific term function differed from domain to domain; for example, for Computer Science, it was "Tool," "Data," or "Evaluation metrics," whereas, for Mathematics, it could be "Theorem," "Inference," or "Formula." Using this framework, two methods based on conditional random fields and machine learning to rank were established to recognize the domain-independent term functions. Jiang et al. (2021) also applied the "Research Questions" and "Research Methods" of papers to improve the performance of the keyphrase extraction models. In SemEval 2017 Task 10, mention-level keyphrase identification and classification were required and three keyphrase types ("Process," "Task," and "Material") were included (Augenstein et al., 2017). Holding the view that research was a problem-solving activity, Heffernan and Teufel (2018) regarded descriptions of problems and solutions as essential elements when describing this activity and presented an automatic classifier to identify "Problems" and "Solutions" in scientific texts. In addition, W. Lu et al. (2019) integrated term functions into the categories of "Research Topic," "Research Method," "Research Object," "Research Area," and "Data," then

revealed the patterns of author-selected keywords in scientific papers from the perspective of term function.

Clearly, then, the classification of term functions in scientific texts varies according to the research purpose and perspective. We are more supportive of defining specific keyphrase function classification frameworks for specific domains, as more aspects of research are covered by the domain-specific classification schema than the domain-independent schema, so keyphrases with domain-specific term functions can provide readers with more comprehensive and relevant information, which could help them quickly to understand the core ideas of the study.

In addition to the above studies, many researchers have focused on scientific information extraction, in which "Task," "Method," "Metric," and "Dataset" were common types for scientific entities (Hou et al., 2019; Jain et al., 2020; Kardas et al., 2020; Luan et al., 2018). Like term function, the type of entity could reflect the category of the resource in scientific texts (C. Zhang et al., 2021). The task of scientific information extraction is similar to our work, as we also need to identify the citations and their role in the paper so the categories of keyphrase functions are similar to those of scientific entities. However, there remain differences between them due to the essential distinction between entities and keyphrases. Although scientific entities are relevant to the text, they are too fine-gained and not all of them reveal the core information of the paper, so it might be time-consuming to identify a study's focus based on its entities. On the contrary, keyphrases are more condensed since they are summarized based on the core aspects of the text. What's more, keyphrase extraction is far more challenging (Augenstein et al., 2017).

2.2 | Approaches to automatic keyphrase prediction

The existing approaches to keyphrase prediction can be categorized into two groups: extraction-based and generation-based methods. The majority of the extractive methods consist of two steps. First, a set of phrases is extracted from the source text as candidate keyphrases with heuristic rules, such as specific part-of-speech (POS) patterns (Hasan & Ng, 2014). Then, all candidates are ranked according to their importance to the text through unsupervised methods, such as TF-IDF based (Salton & Buckley, 1988) and PageRank based ranking methods (Florescu & Caragea, 2017; Z. Liu et al., 2010; Mihalcea & Tarau, 2004; Rose et al., 2010), or supervised methods, such as KEA, Maui and other models (Witten et al., 2005; Hulth, 2003; Medelyan et al., 2009; K. Zhang et al., 2006). These traditional methods can only extract keyphrases that appear in the source text, and the features that they use for ranking the candidate phrases are simply statistical rather than semantic. In order to overcome these limitations, Meng et al. (2017) proposed copyRNN which is an RNN-based generative model that employs an encoder–decoder framework with a copy mechanism for keyphrase prediction. Like manual annotating, copyRNN depends on an understanding of the content and is able to predict absent keyphrases. Subsequently, the sequence-to-sequence model became popular and further optimized generative models were proposed. CorrRNN models the correlation among multiple keyphrases by incorporating coverage and review mechanisms, and effectively alleviates the duplication and coverage problems associated with the keyphrase generation task (J. Chen et al., 2018). W. Chen et al. (2019) focused on the leading role of the title in the overall document, in order to leverage the content of which sufficiently, they proposed a novel model named the Title-Guided Network (TG-Net) to generate the keyphrases for papers. Luo et al. (2020) presented SenSeNet to incorporate the meta-sentence inductive bias toward keyphrase generation. It automatically captured the logical structure of the text and estimated whether a sentence was sufficiently important for the generation task. Considering the wrong bias introduced by the predefined order in previous sequence-to-sequence models, Ye et al. (2021) introduced a new training paradigm, ONE2SET, without concatenating keyphrases into a sequence, and a novel model, SETTRANS, to predict a set of keyphrases in parallel. The methods for automatic keyphrase prediction are becoming increasingly accurate and efficient. Nowadays, most of the generative models make extensive use of the semantic information in the text, but few consider the keyphrase function, and the keyphrases that they generate are not differentiated by role. In this study, we aim to generate keyphrases with exact functions for papers and a framework using control codes is proposed for this purpose.

3 | USER EXPERIMENT FOR KEYPHRASE FUNCTION

To examine the effectiveness of the keyphrase function, we conduct a user experiment to compare the efficiency and accuracy of readers' comprehension according to the original keyphrases (*phrases only*) and the labeled keyphrases (*phrases with function labels*). Two doctors (group A) and two masters students (group B) from the Information Science domain were invited to participate in our experiment. Specifically, we gave the original keyphrases to our participants and asked them to describe the main idea of the paper in one sentence, like TLDR (Cachola et al., 2020), based on their comprehension of the keyphrases. The time spent on each paper was recorded. What's more, every participant was required to give a score of 1–5 to assess how certain they felt about

the topic of the paper. We replaced the original keyphrases with the labeled keyphrases and repeated the experiment. Then, the participants evaluated the similarity by giving a score of 1–5 for the two sentences (TLDR_kp, TLDT_kpf) which they wrote and the TLDR, title, and abstract of the paper, respectively. It should be noted that the experiment based on labeled keyphrases was conducted the following day in order to reduce the bias of experiment time produced by the second reading. Twenty Computer Science papers were selected for this experiment. The results are presented in Tables 1 and 2, and some cases of the experiment are shown in Table 3.

As shown in Table 1, both the participants with some domain knowledge (A1 and B1) and without domain knowledge (A2 and B2) spent less time writing the summary based on keyphrases and their functions (***Time_kpf***) compared with that based on the keyphrases alone (***Time_kp***). All of the ***Certainty_kpf*** scores were higher than the ***Certainty_kp*** scores, which means that all of participants felt far more convinced that what they gleaned from the labeled keyphrases was closer to the major theme of the paper. What's more, from the evaluation results in Table 2, we can see that the sentences based on the labeled keyphrases had highly similar scores to the gold TLDR (***Sim_TLDR_kpf***), title (***Sim_Title_kpf***), and the abstract (***Sim_Abs_kpf***), respectively. That is to say, with the keyphrase function, users can obtain more accurate semantic information from the keyphrases, as also indicated by the examples in Table 3. All of the above results illustrate and confirm the effectiveness of the keyphrase function in helping readers to grasp the main idea of a paper more quickly and accurately, and also indicate the necessity of keyphrase function annotation.

4 | CONTROLLABLE KEYPHRASE GENERATION FRAMEWORK

4.1 | Task definition

Different from the traditional keyphrase prediction which only focuses on the phrases themselves, controllable keyphrase generation aims to generate keyphrases with specific functions. In other words, the predicted results include not only keyphrases, but also the information of keyphrase functions, with which the semantic relationship between keyphrases and core contents of an academic paper could to be expressed explicitly. Specifically, suppose the function categories set is $C = (c_1, c_2, ..., c_q, ..., c_K)$, where K denotes the number of keyphrase functions, for an academic paper $A^{(i)}$, given the source text $X^{(i)} = x_1, x_2, ..., x_{L_i}$ where L_i denotes the length of the word sequence of $X^{(i)}$ and the target keyphrase set $KP^{(i)} = \left(KP^{(i)}_{c_1}, KP^{(i)}_{c_2}, ..., KP^{(i)}_{c_q} ..., KP^{(i)}_{c_K}\right)$ where $KP^{(i)}_{c_q}$ indicates all keyphrases with the function c_q and is defined as the word sequence $y^{(i)}_1, y^{(i)}_2, <sep>, y^{(i)}_3, y^{(i)}_4, y^{(i)}_5, <sep>, ..., y^{(i)}_{l_{(c_q)}}$, where $l_{(c_q)}$ is the length of the sequence and $<sep>$ is used to split each keyphrase. With a labeled dataset $\mathcal{D} = \left\{X^{(i)}, KP^{(i)}_{c_q}\right\}_{i=1}^{N}$, the loss function of our model is as follows:

$$\mathcal{L}(\theta) = -\sum_{1}^{N} \log p\left(KP^{(i)}_{c_q} | X^{(i)}; \theta\right), \tag{1}$$

where θ contains all model parameters.

TABLE 1 Average time used and certainty degree for each paper.

Participant	Time_kp (s)	Time_kpf (s)	Certainty_kp	Certainty_kpf
A1	276	**259.3**	3.3	**4.6**
A2	350	**306.3**	1.6	**3.8**
B1	119.9	**87.5**	2.4	**3.4**
B2	173.3	**139.4**	2.9	**3.9**

Note: The better figures are shown in bold.

TABLE 2 Evaluation results for the similarity between TLDRs written by the readers and the gold TLDR, title, and abstract of the paper.

Participant	Sim_TLDR_kp	Sim_TLDR_kpf	Sim_Title_kp	Sim_Title_kpf	Sim_Abs_kp	Sim_Abs_kpf
A1	1.8	**2.7**	2.0	**3.2**	1.7	**3.3**
A2	1.5	**3.4**	1.4	**3.3**	1.1	**3.7**
B1	2.1	**3.3**	1.8	**3.3**	2.1	**3.0**
B2	2.8	**3.5**	2.8	**4.0**	2.8	**3.5**

Note: The better figures are shown in bold.

TABLE 3 Examples of TLDR_kp and TLDR_kpf written by different participants.

Participant	Content
A1	***Keyphrases:*** Text to Speech（task）, Speech Synthesis（field）, Non-Autoregressive Generation（method）, One-to-Many Mapping（task）, LJSpeech（dataset）, MOS（metric）, FastSpeech2（method） ***TLDR_kp:*** We proposed a method based on non-autoregressive generation which can boost the accuracy of speech synthesis by one-to-many mapping and we applied it to the task of LJSpeech, MOS and FastSpeech2. ***TLDR_kpf:*** We proposed a method called FastSpeech2 which is based on non-autoregressive generation for speech synthesis, and used the metric of MOS to measure the effect of this method on text to speech and one to many mapping tasks. ***gold TLDR:*** We proposed a non-autoregressive TTS model named FastSpeech 2 to better solve the one-to-many mapping problem in TTS and surpass autoregressive models in voice quality.
A2	***Keyphrases:*** Graph Neural Network（field）, Oversmoothing（task）, Normalization（method）, PairNorm（method） ***TLDR_kp:*** We use Oversmoothing, Normalization, and PairNorm methods to optimize the training effect of Graph Neural Networks. ***TLDR_kpf:*** To alleviate the oversmoothing problem in graph neural networks, we propose a new regularization method named pairnorm. ***gold TLDR:*** We proposed a normalization layer for GNN models to solve the oversmoothing problem.
B1	***Keyphrases:*** Speech Recognition（field）, Streaming ASR（task）, Dual-mode ASR（method）, LibriSpeech（dataset）, MultiDomain（dataset）, Latency（metric） ***TLDR_kp:*** We proposed a method called LibriSpeech, which combined streaming ASR and Dual-mode ASR. It performs well on multidomain Latency speech recognition tasks. ***TLDR_kpf:*** Our method, Dual-mode ASR, obtained higher latency on MultiDomain and LibriSpeech for streaming ASR tasks in the field of speech recognition. ***gold TLDR:*** Dual-mode ASR unifies and improves Streaming ASR with full-context modeling, simplifying the development and deployment workflow and improving both latency and accuracy.
B2	***Keyphrases:*** Machine Comprehension（task）, Conversational Agent（task）, Natural Language Processing（field）, FlowQA（method）, CoQA（dataset）, QuAC（dataset） ***TLDR_kp:*** The paper constructed a machine conversational agent, using datasets such as FlowQA, CoQA, and QuAC and used the method of Natural Language Processing and Machine Comprehension. ***TLDR_kpf:*** The paper used a CoQA and QuAC dataset, proposed a FlowQA model to comprehend natural language automatically and built a machine conversational agent. ***gold TLDR:*** We propose the Flow mechanism and an end-to-end architecture, FlowQA, that achieves SotA on two conversational QA datasets and a sequential instruction understanding task.

Note: Clearly mistaken keyphrases are highlighted in green.

4.2 | Keyphrase function schema

The keyphrase function schema defines the function categories of keyphrases in a certain domain, based on which keyphrases with specific functions could be predicted automatically or annotated manually to promote readers' comprehension. In this study, we take Computer Science (CS) domain as an example, and summarize a keyphrase function schema for it based on the previous research about term function and combined with our understanding of the pattern of CS papers. As shown in Table 4, there are five different functions: Field, Task, Method, Dataset, and Metric. In particular, considering the differences in the subject backgrounds of different readers and their various information needs, we prefer keyphrases that cover the existing domain knowledge which is vital to the paper, as

TABLE 4 Keyphrase function schema for the computer science domain.

Functions	Description	References
Field	A branch of the subject area. it can be regarded as a category that the paper falls into the domain.	*head* (Kondo et al., 2011), *domain* (Gupta & Manning, 2011), *research area* (Lu et al., 2019)
Task	Concrete research tasks, applications, or empirical studies related to a research project.	*application* (Tsai et al., 2013), *problem* (Cheng, 2015; Heffernan &Teufel, 2018), *task* (Augenstein et al., 2017; Luan et al., 2018; Hou et al., 2019; Jain et al., 2020; Kardas et al., 2020)
Method	Methods, models, techniques, etc., which can be novel methods proposed in the paper, common methods or cited methods used as an important aspect of the core method, or models that are largely adopted and followed by the current paper.	*technology* (Nanba et al., 2010), *technique* (Gupta & Manning, 2011; Tsai et al., 2013), *process* (Augenstein et al., 2017) *solution* (Heffernan &Teufel, 2018), *research method* (Lu et al., 2019), *method* (Kondo et al., 2011; Cheng, 2015; Luan et al., 2018; Hou et al., 2019; Jain et al., 2020; Kardas et al., 2020)
Dataset	Important dataset entities used in the study.	*material* (Augenstein et al., 2017), *data* (Cheng, 2015; Lu et al., 2019), *dataset* (Luan et al., 2018; Hou et al., 2019; Jain et al., 2020, Kardas et al., 2020)
Metric	Metric entities used to evaluate quality of a method.	*metric* (Cheng, 2015; Luan et al., 2018; Hou et al., 2019; Jain et al.; 2020, Kardas et al., 2020)

well as essential novel knowledge which can bring information gains to readers. So, this schema encourages annotators to pay attention to both existing keyphrases and the new concepts proposed in the paper.

4.3 | CKPG

We introduce CKPG (*Controllable Keyphrase Generation*), a simple yet effective method for generating keyphrases with specific functions. Based on the encoder–decoder framework, the CKPG uses the keyphrase function as a control code, whose effectiveness for the autoregressive language models has been shown in (Cachola et al., 2020; Elsahar et al., 2020; Keskar et al., 2019), and then automatically generates keyphrases of the specified category. In order to allow the parameters of the model to learn to generate different kinds of keyphrases according to the category of keyphrase function, at the generation time, each data sample $\left(X^{(i)}, KP^{(i)}\right)$ is split into K context-keyphrase pairs: $\left\{\left(X^{(i)}, KP^{(i)}_{c_1}\right), \left(X^{(i)}, KP^{(i)}_{c_2}\right), ..., \left(X^{(i)}, KP^{(i)}_{c_K}\right)\right\}$, and for each data pair $\left(X^{(i)}, KP^{(i)}_{c_q}\right)$, the function c_q is appended to the source text $X^{(i)}$. In this study, $c_q \in (<FIELD>, <TASK>, <METHOD>, <DATASET>, <METRIC>)$. During inference, we can generate each category of keyphrase and then gather them to obtain the complete annotation result of the paper. Taking the generation flows of Task and Dataset as an example, Figure 2 shows the concrete process of the CKPG:

5 | EXPERIMENT

5.1 | Corpora preparation

Keyphrase prediction studies tend to focus on the phrases themselves and do not require keyphrase functions. As a result, the existing datasets for keyphrase generation lack function information and so are irrelevant to our work. Therefore, we consider using Paper with Code (PwC),[1] a public corpus of Machine Learning papers, to carry out our experiments. Although PwC is not a formal keyphrase dataset, like KP20k, it could largely meet the requirements of our work, for it provides the main tasks, methods, datasets, codes, and evaluation results, collected from authors' submitted results of their work, manual annotations of PwC users, and public leaderboards. This corpus also contains the paper titles and abstracts, which are widely utilized as the source texts and have been observed to perform well in previous keyphrase generation tasks (Meng et al., 2017; J. Chen et al., 2018; W. Chen et al., 2019).

In this study, we extract the fields, tasks, methods, datasets, metrics, titles and abstracts from the raw corpus of PwC, provided that the paper has a full title and abstract. A total of 6,012 papers were extracted, of which 2,119 included all five categories of "keyphrases," and the remaining 3,839 contained only some of them. The average number of keyphrases per paper in the above five categories is 5.82, 2.68, 9.06, 2.59 and 2.41, respectively. Note that PwC does not contain the research fields as we define them, so we used the "main_collection" of methods as an alternative.

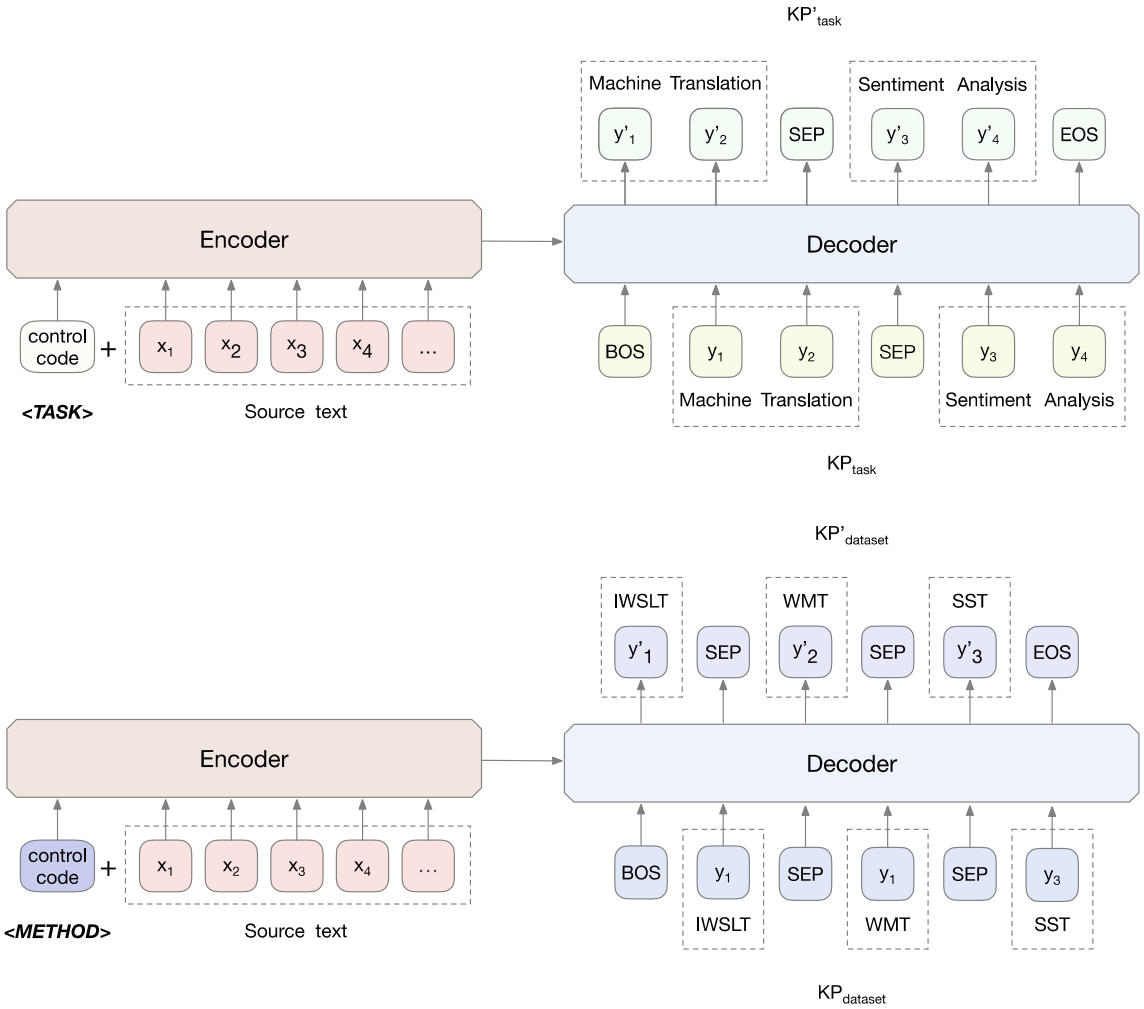

FIGURE 2 The CKPG framework (the generation process for Task and Method).

In our experiments, we first randomly select 1,000 papers with full information, half of which are used for testing and the other half for validation. The remaining 5,012 served as the training set. Then, we split each data into N tuples of $(c_q, X^{(i)}, KP_{c_q}{}^{(i)})$, where $X^{(i)}$ indicates the concatenation of the title and abstract. Finally, there are 17,235 tuples for training, 2,500 for testing, and 2,500 for validation. In addition, we applied 514,154 pairs of source text and keyphrases of the dataset KP20k (Meng et al., 2017) to pretrain our Transformer-based model.

5.2 | Implementation details

Five encoder–decoder models were trained based on Transformer (Vaswani et al., 2017), BART (Base/Large) (Lewis et al., 2019) and T5 (Base/Large) (Raffel et al., 2020) with PyTorch. 2 FairSeq (Ott et al., 2019) and OpenNMT (Klein et al., 2018) were used to implement Transformer-based models, and the other models were trained with Hugging Face. 3 We used KP20k to pretrain the Transformer-based keyword generation model, named Transformer_KP20k, and fine-tuned the five models using PwC. The truncation lengths of the source text and keyphrases sequence were 512 and 128, respectively. The dimension of the hidden layers of Transformer was set to 512. We pretrained Transformer_KP20k for 300,000 steps with a dropout rate of 0.1 and fine-tuned it on the PWC dataset for 10,000 steps. For the remaining four models, the number of fine-tuning epoch was 10. The Adam (Kingma & Ba, 2015) algorithm was used to optimize our models. The initial learning rate and batch size of T5-Large-based model were 1e-4 and 2, and those of other models were 3e-5 and 8. The warmup ratio was set to 0.1. Regarding inference, the beam size was 16 and the maximum length of the prediction keyphrases sequence was 40.

5.3 | Evaluation metric

The Precision (P), Recall (R), and F-measure (F_1) were employed to evaluate the CKPG's performance. Following the standard definition, the F-measure is computed based

on Precision and Recall, wherein Precision is the number of correctly-predicted keyphrases ($N_{Correct}$) over the number of all predicted keyphrases (N_{Pred}), and Recall is the number of correctly-predicted keyphrases over the total number of the target keyphrases (N_{Gold}). All of the above evaluation metrics are defined as Equations (2)–(4). In addition, Porter Stemmer was utilized for preprocessing when determining whether two keyphrases matched.

$$P = \frac{N_{Correct}}{N_{Pred}}, \tag{2}$$

$$R = \frac{N_{Correct}}{N_{Gold}}, \tag{3}$$

$$F_1 = \frac{2 \times P \times R}{P + R}. \tag{4}$$

5.4 | Results

P@5, R@5, and F_1@5 were used to evaluate the performance of our models. Tables 5–7 present the scores of each model on different metrics, where underlining indicates the highest score in a row, bold denotes a highest score in a column.

From the above results, we can observe that the best scores are mainly related to Task and Field, and the larger the scale of the model is, the better the performance is. Comparing these five models, the Transformer_KP20k-based model performs weaker while the T5-Large-based model performs much better both in terms of the scores on each category and in terms of the macro-averaging scores. All the macro-averaging scores reach 0.287, the best Macro-avgs of P@5, R@5, and F_1@5 are up to 0.680, 0.535, and 0.558 respectively, which are contributed by T5-Large-based model. Overall, the results indicate that our CKPG models can automatically predict the keyphrases of the specified function category.

5.5 | Case study

A prediction example is shown in Table 8, in which most of the correct keyphrases are recalled at the top 5 results. Although some results are wrong, they are very related to

TABLE 5 Experimental results of the performance of the CKPG on P@5.

	Models	Field	Task	Method	Dataset	Metric	Macro-avg
P@5	Transformer_KP20k	0.396	0.316	0.376	0.144	0.206	0.288
	T5-Base	0.518	0.725	0.510	0.307	0.264	0.465
	Bart-Base	0.739	0.749	0.637	0.501	0.545	0.634
	Bart-Large	0.750	0.707	0.628	**0.513**	**0.566**	0.633
	T5-Large	**0.771**	**0.812**	**0.740**	0.510	**0.566**	**0.680**

TABLE 6 Experimental results of the performance of the CKPG on R@5.

	Models	Field	Task	Method	Dataset	Metric	Macro-avg
R@5	Transformer_KP20k	0.470	0.632	0.313	0.346	0.502	0.453
	T5-Base	0.346	0.622	0.302	0.254	0.206	0.346
	Bart-Base	0.486	0.639	0.412	0.382	0.445	0.473
	Bart-Large	0.541	0.678	0.435	**0.437**	0.486	0.515
	T5-Large	**0.555**	**0.730**	**0.464**	0.432	**0.496**	**0.535**

TABLE 7 Experimental results of the performance of the CKPG on F@5.

	Models	Field	Task	Method	Dataset	Metric	Macro-avg
F_1@5	Transformer_KP20k	0.337	0.395	0.257	0.183	0.261	0.287
	T5-Base	0.375	0.640	0.318	0.259	0.215	0.361
	Bart-Base	0.543	0.661	0.453	0.400	0.46	0.503
	Bart-Large	0.582	0.663	0.454	**0.437**	0.495	0.526
	T5-Large	**0.599**	**0.744**	**0.514**	0.433	**0.500**	**0.558**

TABLE 8 An example of predicted results of CKPG models.

Title: On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units **Abstract:** On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units. Deep feedforward neural networks with piecewise linear activations are currently producing the state-of-the-art results in several public datasets. The combination of deep learning models and piecewise linear activation functions allows for the estimation of exponentially complex functions with the use of a large number of subnetworks specialized in the classification of similar input examples. … Also, this batch normalisation promotes the pre-conditioning of very deep learning models. We show that by introducing maxout and batch normalisation units to the network in network model results in a model that produces classification results that are better than or comparable to the current state of the art in CIFAR-10, CIFAR-100, MNIST, and SVHN datasets.	
Gold Keyphrases **Field:** Activation Functions*; **Task:** Image Classification; **Method:** Maxout*; **Dataset:** SVHN*; MNIST*; CIFAR-100*; CIFAR-10*; **Metric:** Percentage error; Percentage correct	
Predicted Keyphrases	
Field	*[Transformer-KP20k]:* Convolutions; Generative Models; Normalization; Regularization*; **Activation Functions*** *[T5-Base]:* Initialization; Convolutional Neural Networks; **Activation Functions***; Normalization; Convolutions *[BART-Base]:* **Activation Functions*** *[BART-Large]:* **Activation Functions*** *[T5-Large]:* **Activation Functions***
Task	*[Transformer-KP20k]:* **Image Classification**; Representation Learning; Regression; Fine Grained Image Classification; Named Entity Recognition *[BART-Base]:* **Image Classification** *[T5-Base]:* **Image Classification** *[BART-Large]:* **Image Classification** *[T5-Large]:* **Image Classification**
Method	*[Transformer-KP20k]:* Convolution; Batch Normalization; AutoEncoder; Sigmoid Activation; 1x1 Convolution *[BART-Base]:* **Maxout*** *[BART-Large]:* Batch Normalisation* *[T5-Base]:* Layer Normalization; Byte Pair Encoding; BPE; Softmax; Adam *[T5-Large]:* **Maxout***
Dataset	*[Transformer-KP20k]:* **CIFAR 10*, CIFAR 100*, MNIST***; STL 10, ImageNet *[BART-Base]:* **MNIST*; CIFAR-100*** *[T5-Base]:* **SVHN*; CIFAR-100***; *[BART-Large]:* **SVHN*** *[T5-Large]:* **MNIST*; CIFAR-100*; CIFAR-10***
Metric	*[Transformer-KP20k]:* Accuracy; **Percentage correct**; **Percentage error**; F1; MAP *[BART-Base]:* **Percentage error**; **Percentage correct** *[T5-Base]:* Accuracy *[T5-Large]:* **Percentage correct** *[BART-Large]:* **Percentage error**; **Percentage correct**

Note: The phrases shown in bold are correct predictions and phrases marked * are present in the source text.

the article, such as "Batch Normalization" (Method) and "Fine Grained Image Classification" (Task) predicted by Transformer-KP20k-based model and "Batch Normalisation" (Method) predicted by T5-Large-based model. What's more, it can be seen from this case that, in the predicted results, keyphrases with different functions have clear boundaries regarding their semantic role, that is to say, keyphrases with function A will rarely appear in the prediction list with function B. These results show that our model can capture the important contents of the different aspects of the paper according to the categories and paraphrase them to target phrases with corresponding functions, which also verifies the effectiveness of our CKPG method.

6 | DISCUSSION

6.1 | Data distribution and results

As shown in Tables 5–7, for the five CKPG models, the ability to generate keyphrases differs between the categories. When predicting keyphrases related to Task and Field, especially the former, all these models perform far better. Therefore, to explain these differences in the results, additional statistics for each category of the keyphrase dataset were generated. Table 9 shows the number of valid and invalid papers in each category depending on whether the sample contain keyphrases with that function. The proportion and F_1@5 score of the present and absent keyphrases in the five categories are presented in Tables 10 and 11, and the vocabulary sizes of the keyphrases in the five categories, that is, the number of keyphrases after de-duplication, are listed in Table 12.

From Table 9, we find that, for the whole dataset, most of samples include keyphrases for Task, Dataset, and Metric, while less than half of the papers contain keyphrases related to Field and Method. As shown in Table 10, Task keyphrases appear most frequently in the source text, whose average proportion of present keyphrases is about 0.630. The proportion of Methods keyphrases is about 0.420, and the remaining kinds of keyphrases, especially Field, are rarely present in the title or abstract. Compared with Method, there are far more data available for Task (5961) than Method (2131), whose proportion of present keyphrases is the second largest. For F_1@5, Task can exceed Method by 0.322 at most. Then, compared with Dataset, Task has about the same amount of valid data, but Task appears in the source text far more frequently than Dataset, with respective proportions of 0.630 and 0.227. What's more, the maximum difference of F_1@5 between Task and Datasets is up to 0.381. The situation is similar with regard to Metric. Based on the above statistical results, it appears that Task has obvious advantages in terms of the volume of valid data and proportion of present keyphrases, both of which are vital factors that enable the models to learn the parameters. From Table 8, we can observe that our models are able to capture the hidden semantics of the

TABLE 9 The statistics for the valid and invalid papers for different categories of PwC.

	Field	Task	Method	Dataset	Metric
Invalid number	3,883	51	3,881	6	4
Valid number	2,129	5,961	2,131	6,006	6,008

TABLE 10 Proportion of the present keyphrases and absent keyphrases in PwC.

	Field	Task	Method	Dataset	Metric
Present	0.067	0.630	0.420	0.227	0.124
Absent	0.933	0.370	0.580	0.773	0.876

TABLE 12 Vocabulary size of the keyphrases in the five categories.

	Field	Task	Method	Dataset	Metric
Vocab size	111	1,208	962	2,876	1,331

TABLE 11 Present keyphrases and absent keyphrases' prediction performance of the CKPG.

	Models	Field	Task	Method	Dataset	Metric	Macro AVG
$present\_F_1$@5	Transformer_KP20k	0.300	0.495	0.298	0.126	0.158	0.275
	T5-Base	0.246	0.671	0.325	0.169	0.105	0.303
	Bart-Base	0.331	0.694	0.475	0.205	0.122	0.365
	Bart-Large	0.344	0.692	0.509	0.209	0.115	0.374
	T5-Large	0.286	0.744	0.501	0.207	0.133	0.374
$absent\_F_1$@5	Transformer_KP20k	0.324	0.105	0.220	0.081	0.152	0.176
	T5-Base	0.284	0.197	0.216	0.149	0.151	0.199
	Bart-Base	0.418	0.218	0.244	0.261	0.396	0.307
	Bart-Large	0.478	0.257	0.272	0.298	0.439	0.349
	T5-Large	0.483	0.290	0.303	0.298	0.438	0.362

TABLE 13 Performance of two-stage extraction and end-to-end generation.

Models		Precision	Recall	F1
Two-stage	SVM	0.735	0.230	0.329
	DecisionTree	0.742	0.230	0.330
	KNN	0.728	0.233	0.336
	RandomForest	0.750	0.234	0.336
	GBDT	0.745	0.235	0.337
	BERT	**0.972**	**0.296**	**0.437**
CKPG	Transformer_KP20k	0.288	0.453	0.287
	T5-Base	0.465	0.346	0.361
	Bart-Base	0.634	0.473	0.503
	Bart-Large	0.633	0.515	0.526
	T5-Large	**0.680**	**0.535**	**0.558**

Note: The best scores of the two strategies are shown in bold, respectively.

textual content and perform well in terms of predicting absent keyphrases. Therefore, for Field, although the two statistical results in Tables 9 and 10 have no advantage, the vocabulary size is relatively small (111), so the F_1 score is also acceptable. As for Method, Dataset, and Metric, all of which have relatively lower F_1 scores, the data for these categories are of average quality overall.

6.2 | Two-stage extraction versus end-to-end generation

In this section, we compare the performance of two-stage extraction method and the end-to-end generation method. As keyphrase extraction models could only extract keyphrases present in the text and the proportion of these present keyphrases is not very high (0.289), we assumed that the accuracy of keyphrase extraction in the first stage is 1.0 and used all the present keyphrases for keyphrase function classification. The SVM, KNN, DecisionTree, GBDT, RandomForest, and BERT classifiers were trained. For the machine learning classifiers, we constructed five features from the statistical and semantic perspectives, which were (1) char_number: the number of characters contained in the keyphrase, (2) word_number: the number of words contained in the keyphrase, (3) first_index: the ratio of the position of the keyphrase' first occurrence to the text length, (4) tf: the frequency of the keyphrase appearing in the text, and (5) kp_text_similarity: the semantic similarity of the keyphrase and the text calculated based on BERT (Devlin et al., 2019). For the deep learning classifier, we treated this task as a sentence pair classification task and fine-tuned the BERT model using keyphrase-text pairs. The results of the two-stage extraction and the CKPG results are shown in Table 13.

As shown in the above results, the CKGP models perform better overall. Although we maximize the accuracy of the keyphrase extraction, the recall of the two-stage method is much lower than the CKPG. The best F1 score of the two-stage extraction is 0.437, while that of the CKPG is up to 0.558. What's more, it should be noted that the two-stage extraction method could not guarantee a category-complete result because the candidate keyphrases might not cover all the categories. Different from the two-stage method, the CKPG framework is an end-to-end approach that is able to generate keyphrases for all categories by specifying the keyphrase function, so that category-complete prediction results can be obtained. From this aspect, the proposed CKPG framework is also superior to the two-stage extraction method.

6.3 | Implications

This study has the following implications. For researchers, keyphrase functions are not only the metadata that describe the different semantic relationships between the keyphrases and the paper, but their category also represents the divisions of the core aspects of the paper. So, keyphrases annotated with functions can display important details about the paper as a whole, which can provide readers with a comprehensive knowledge profile of it, enabling them quickly to form a general impression of it and understand the main contents from an overall perspective in a short time. Meanwhile, general and specific keyphrases are both adopted, so these keyphrases can be more useful in depicting the topic or theme of the paper clearly and appropriately based on a general view of the discipline and a detailed view of the paper itself. Focusing on each keyphrase, the function can provide readers with extra information besides the phrase itself, thus helping readers to understand its meaning more clearly and locate the relevant details in the text more easily. What's more, when a keyphrase is a polysemy, the function can aid comprehension and help readers to identify the keyphrase's different meanings or roles in different studies, thus enhancing the comparison and connection of knowledge. When determining whether or not to adopt a retrieved paper, the keyphrase function can guide the researcher to find the most valuable keyphrases and make efficient decisions as quickly as possible. For example, when researchers need to investigate the evaluation methods of a task, they can judge quickly whether or not the task is related to their research without spending much time on the other keyphrases, and then decide relatively quickly whether the paper is relevant or not.

In addition to those possible advantages for researchers, there are also some potential benefits and value for downstream tasks. It is easy to cluster keyphrases based on their function, and then analyze the development of a scientific community from different perspectives based on their results, such as analyzing the evolution of the most popular research problems, the improvement path of the main technologies, and the differences in the evaluation standards of different eras. As the categories of the keyphrase function of a specific domain reflect the most important and common aspects of the research area, an analysis of keyphrases clustered by function would facilitate the comprehensive development of the domain. What's more, there are many knowledge connections between different papers. For example, studies may propose different methods for the same research task, and a model can also be applied to solve various research problems in different papers. If the papers are annotated with keyphrases and corresponding functions, these relationships, as mentioned above, can be extracted and presented according to the co-occurrence of keyphrases. This makes it possible to analyze the structures of different kinds of knowledge to which one certain type of knowledge is related. In addition, like the Keyword–Citation–Keyword (KCK) network (Cheng et al., 2020), Method–Citation–Method, Task–Citation–Method, and other citation networks with the extra semantic relationships might be constructed, which also provides a fresh avenue for analyzing the knowledge structure of a discipline. In fact, keyphrases are automatically classified when annotated with their functions, which provides sound support for secondary information organization and great potential regarding more diverse information services. Faceted information retrieval could be realized based on the semantic facets of the paper; that is, the keyphrase functions. Enabling papers to be accessed and ordered in multiple semantic ways can help to diversify and personalize the retrieval process. Moreover, the relationships between the retrieval results that share the same semantic classes and intention will become clear, which will help to meet the users' information needs more accurately and comprehensively. If papers are organized according to the keyphrase functions, then citation recommendations can be implemented by category and the recommendations will become more flexible and refined. Relevant papers related to a certain aspect could be recommended according to the citation intent. That is to say, where papers related by method are required, papers with relevant methods would be recommended. Moreover, it is clearly time-consuming and labor-intensive to annotate keyphrases manually. The CKPG models implemented in this study may, to some extent, provide some practical ideas for automatic keyphrase annotation. In summary, the novel method for generating keyphrases with specific functions has a non-negligible role and value in the comprehension and application of scientific knowledge.

7 | CONCLUSION

In this article, we identify and analyze the main drawback related to the current keyphrases; that is, the implicit expression of their semantic role, which prevents readers from quickly understanding the core ideas by reading them. To address this issue, we propose to generate keyphrases from the perspective of assisting readers' comprehension, aiming to provide readers with semantically complete keyphrases as far as possible and make the keyphrases an efficient tool for the readers. Before the formal study, we conducted a user experiment, the results of which show that the keyphrases with specific functions do help readers comprehend the paper. Then the CKPG, a novel keyphrase generation framework based on the keyphrase function schema, was proposed. Moreover, we implemented five sequence-to-sequence models based on Transformer, BART, and T5 respectively, and verified the effectiveness of our CKPG method on the PwC dataset, which is reprocessed in this study and contains keyphrases related to Field, Task, Method, Dataset, and Metric.

There are some limitations to this study. First, the raw PwC data do not cover every category of keyphrases as we required, such as Field, and the diversity of keyphrases in papers is limited, since there are many generic phrases, which differs somewhat from the ideal keyphrase dataset. Therefore, it is necessary to construct high-quality datasets, such as datasets that have been manually annotated, or collected from future works. Second, this study focuses on the keyphrase functions in the Computer Science domain only and the categories may not comprehensive enough. Keyphrase categories of other domains also need to be explored and the schema summarized for CS domain should be further consummated as well. Finally, the CKPG method concentrates more on the data and less on improving the model itself, so building more efficient and robust keyphrase generation models is an important goal of our future work.

AUTHOR CONTRIBUTIONS

Yi Jiang: Conceptualization, Methodology, Formal analysis, Investigation, Data curation, Writing—original draft. **Rui Meng**: Conceptualization, Methodology, Software, Writing—review & editing. **Yong Huang**: Conceptualization, Investigation, Writing—review & editing. **Wei Lu**: Conceptualization, Supervision, Investigation,

Writing—review & editing. **Jiawei Liu**: Investigation, Writing—review & editing.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

ENDNOTES

[1] https://paperswithcode.com/about

[2] https://pytorch.org

[3] https://huggingface.co

REFERENCES

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). SemEval 2017 task 10: ScienceIE—Extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation* (pp. 546–555). Association for Computational Linguistics. https://aclanthology.org/S17-2091

Cachola, I., Lo, K., Cohan, A., & Weld, D. S. (2020). TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 4766–4777). Association for Computational Linguistics. https://aclanthology.org/2020.findings-emnlp.428

Çano, E., & Bojar, O. (2019). Keyphrase generation: A multi-aspect survey. In *25th Conference of Open Innovations Association (FRUCT)* (pp. 85–94). IEEE. https://doi.org/10.23919/FRUCT48121.2019.8981519

Chen, J., Zhang, X., Wu, Y., Yan, Z., & Li, Z. (2018). Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4057–4066). Association for Computational Linguistics. https://aclanthology.org/D18-1439

Chen, W., Gao, Y., Zhang, J., King, I., & Lyu, M. R. (2019). Title-guided encoding for Keyphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, No. 1, pp. 6268–6275). AAAI Press.

Cheng, Q. (2015). *Term function recognition of academic text*. Wuhan University https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CDFDLAST2017&filename=1016016013.nh

Cheng, Q., Wang, J., Lu, W., Huang, Y., & Bu, Y. (2020). Keyword–citation–keyword network: A new perspective of discipline knowledge structure analysis. *Scientometrics*, *124*(3), 1923–1943. https://doi.org/10.1007/s11192-020-03576-5

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186). Association for Computational Linguistics. https://aclanthology.org/N19-1423

Elsahar, H., Coavoux, M., Gallé, M., & Rozen, J. (2020). Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 1646–1662). Association for Computational Linguistics. https://aclanthology.org/2021.eacl-main.141

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. (2020). Keyword extraction: Issues and methods. *Natural Language Engineering*, *26*(3), 259–291. https://doi.org/10.1017/S1351324919000457

Florescu, C., & Caragea, C. (2017). PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1105–1115). Association for Computational Linguistics. https://aclanthology.org/P17-1102

Gbur, E. E., & Trumbo, B. E. (1995). Key words and phrases—The key to scholarly visibility and efficiency in an information explosion. *The American Statistician*, *49*(1), 29–33. https://doi.org/10.1080/00031305.1995.10476108

Gupta, S., & Manning, C. D. (2011). Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing* (pp. 1–9). Association for Computational Linguistics. https://aclanthology.org/I11-1001

Hartley, J., & Kostoff, R. N. (2003). How useful are 'key words' in scientific journals? *Journal of Information Science*, *29*(5), 433–438. https://doi.org/10.1177/01655515030295008

Hasan, K. S., & Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 1262–1273). Association for Computational Linguistics. https://aclanthology.org/P14-1119

Heffernan, K., & Teufel, S. (2018). Identifying problems and solutions in scientific text. *Scientometrics*, *116*(2), 1367–1382. https://doi.org/10.1007/s11192-018-2718-6

Hernandez-Castaneda, A., Garcia-Hernandez, R. A., Ledeneva, Y., & Millan-Hernandez, C. E. (2020). Extractive automatic text summarization based on lexical-semantic keywords. *IEEE Access*, *8*, 49896–49907. https://doi.org/10.1109/ACCESS.2020.2980226

Hou, Y., Jochim, C., Gleize, M., Bonin, F., & Ganguly, D. (2019). Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5203–5213). Association for Computational Linguistics. https://aclanthology.org/P19-1513

Hu, K., Luo, Q., Qi, K., Yang, S., Mao, J., Fu, X., Zheng, J., Wu, H., Guo, Y., & Zhu, Q. (2019). Understanding the topic evolution of scientific literatures like an evolving city: Using Google Word2Vec model and spatial autocorrelation analysis. *Information Processing & Management*, *56*(4), 1185–1203. https://doi.org/10.1016/j.ipm.2019.02.014

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (pp. 216–223). Association for Computational Linguistics. https://doi.org/10.3115/1119355.1119383

Jain, S., van Zuylen, M., Hajishirzi, H., & Beltagy, I. (2020). SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7506–7516). Association for Computational Linguistics. https://aclanthology.org/2020.acl-main.670

Jiang, Y., Huang, Y., Xia, Y., Li, P., & Lu, W. (2021). Recognition of lexical functions in academic texts: Application in automatic keyword extraction. *Journal of the China Society for Scientific and Technical Information*, *40*(2), 152–162.

Kardas, M., Czapla, P., Stenetorp, P., Ruder, S., Riedel, S., Taylor, R., & Stojnic, R. (2020). AxCell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 8580–8594). Association for Computational Linguistics. https://aclanthology.org/2020.emnlp-main.692.pdf

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). *CTRL: A conditional transformer language model for controllable generation*. arXiv preprint arXiv:1909.05858. http://arxiv.org/abs/1909.05858

Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980. http://arxiv.org/abs/1412.6980

Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., & Rush, A. M. (2018). OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas* (Volume 1: Research track). Association for Machine Translation in the Americas. https://aclanthology.org/W18-1817

Kondo, T., Nanba, H., Takezawa, T., & Okumura, M. (2011). Technical trend analysis by analyzing research papers' titles. In Z. Vetulani (Ed.), *Human language technology. Challenges for computer science and linguistics* (Vol. 6562, pp. 512–521). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20095-3_47

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880). Association for Computational Linguistics. https://aclanthology.org/2020.acl-main.703

Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (pp. 366–376). Association for Computational Linguistics. https://aclanthology.org/D10-1036

Lu, K., & Kipp, M. E. I. (2014). Understanding the retrieval effectiveness of collaborative tags and author keywords in different retrieval environments: An experimental study on medical collections. *Journal of the Association for Information Science and Technology*, *65*(3), 483–500. https://doi.org/10.1002/asi.22985

Lu, W., Huang, S., Yang, J., Bu, Y., Cheng, Q., & Huang, Y. (2021). Detecting research topic trends by author-defined keyword frequency. *Information Processing & Management*, *58*(4), 102594. https://doi.org/10.1016/j.ipm.2021.102594

Lu, W., Li, X., Liu, Z., & Cheng, Q. (2019). How do author-selected keywords function semantically in scientific manuscripts? *Knowledge Organization*, *46*(6), 403–418. https://doi.org/10.5771/0943-7444-2019-6-402

Luan, Y., He, L., Ostendorf, M., & Hajishirzi, H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3219–3232). Association for Computational Linguistics. http://arxiv.org/abs/1808.09602; https://aclanthology.org/D18-1360

Luo, Y., Li, Z., Wang, B., Xing, X., Zhang, Q., & Huang, X. (2020). *SenSeNet: Neural keyphrase generation with document structure*. arXiv preprint arXiv:2012.06754. https://arxiv.org/pdf/2012.06754

Medelyan, O., Frank, E., & Witten, I. H. (2009). Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 1318–1327). Association for Computational Linguistics. https://doi.org/10.3115/1699648.1699678

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 582–592). Association for Computational Linguistics. https://aclanthology.org/P17-1054

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (pp. 404–411). Association for Computational Linguistics. https://aclanthology.org/W04-3252

Nanba, H., Kondo, T., & Takezawa, T. (2010). Automatic creation of a technical trend map from research papers and patents. In *Proceedings of the 3rd International Workshop on Patent Information Retrieval—PaIR'10* (p. 11). Association for Computing Machinery. https://doi.org/10.1145/1871888.1871891

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (pp. 48–53). Association for Computational Linguistics. https://aclanthology.org/N19-4009

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In M. W. Berry & J. Kogan (Eds.), *Text mining* (pp. 1–20). John Wiley & Sons. https://doi.org/10.1002/9780470689646.ch1

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Sesagiri Raamkumar, A., Foo, S., & Pang, N. (2017). Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems. *Information Processing & Management*, *53*(3), 577–594. https://doi.org/10.1016/j.ipm.2016.12.006

Sun, J., Hu, S., Nie, X., & Walker, J. (2020). Efficient ranked multi-keyword retrieval with privacy protection for multiple data owners in cloud computing. *IEEE Systems Journal*, *14*(2), 1728–1739. https://doi.org/10.1109/JSYST.2019.2933346

Tsai, C.-T., Kundu, G., & Roth, D. (2013). Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management—CIKM'13* (pp. 1733–1738). Association for Computing Machinery. https://doi.org/10.1145/2505515.2505613

Turney, P. D. (2002). *Learning algorithms for keyphrase extraction*. arXiv preprint arXiv:Cs/0212020. http://arxiv.org/abs/cs/0212020

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference*

on Neural Information Processing Systems (pp. 6000–6010). Curran Associates Inc.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (2005). KEA: Practical automated keyphrase extraction. In Y.-L. Theng & S. Foo (Eds.), *Design and usability of digital libraries* (pp. 129–152). IGI Global. https://doi.org/10.4018/978-1-59140-441-5.ch008

Yang, J., Lu, W., Hu, J., & Huang, S. (2022). A novel emerging topic detection method: A knowledge ecology perspective. *Information Processing & Management*, *59*(2), 102843. https://doi.org/10.1016/j.ipm.2021.102843

Ye, J., Gui, T., Luo, Y., Xu, Y., & Zhang, Q. (2021). One2Set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (pp. 4598–4608). Association for Computational Linguistics. https://aclanthology.org/2021.acl-long.354

Yoon, Y. S., Zo, H., Choi, M., Lee, D., & Lee, H. (2018). Exploring the dynamic knowledge structure of studies on the internet of things: Keyword analysis. *ETRI Journal*, *40*(6), 745–758. https://doi.org/10.4218/etrij.2018-0059

Zhang, C., Mayr, P., Lu, W., & Zhang, Y. (2021). Extraction and evaluation of knowledge entities from scientific documents. *Journal of Data and Information Science*, *6*(3), 1–5. https://doi.org/10.2478/jdis-2021-0025

Zhang, K., Xu, H., Tang, J., & Li, J. (2006). Keyword extraction using support vector machine. In J. X. Yu, M. Kitsuregawa, & H. V. Leong (Eds.), *Advances in web-age information management* (Vol. 4016, pp. 85–96). Springer Berlin Heidelberg. https://doi.org/10.1007/11775300_8

How to cite this article: Jiang, Y., Meng, R., Huang, Y., Lu, W., & Liu, J. (2023). Generating keyphrases for readers: A controllable keyphrase generation framework. *Journal of the Association for Information Science and Technology*, *74*(7), 759–774. https://doi.org/10.1002/asi.24749