Keyphrase Prediction from Video Transcripts: New Dataset and Directions

Amir Pouran Ben Veyseh[1], **Quan Hung Tran**[2], **Seunghyun Yoon**[2], **Varun Manjunatha**[2], **Hanieh Deilamsalehy**[2], **Rajiv Jain**[2], **Trung Bui**[2], **Walter W. Chang**[2], **Franck Dernoncourt**[2], **and Thien Huu Nguyen**[1]

[1]Department of Computer Science, University of Oregon, OR, USA
[2]Adobe Research, USA
{apouranb, thien}@cs.uoregon.edu
{franck.dernoncourt}@adobe.com

Abstract

Keyphrase Prediction (KP) is an established NLP task, aiming to yield representative phrases to summarize the main content of a given document. Despite major progress in recent years, existing works on KP have mainly focused on formal texts such as scientific papers or weblogs. The challenges of KP in informal-text domains are not yet fully studied. To this end, this work studies new challenges of KP in transcripts of videos, an understudied domain for KP that involves informal texts and non-cohesive presentation styles. A bottleneck for KP research in this domain involves the lack of high-quality and large-scale annotated data that hinders the development of advanced KP models. To address this issue, we introduce a large-scale manually-annotated KP dataset in the domain of live-stream video transcripts obtained by automatic speech recognition tools. Concretely, transcripts of 500+ hours of videos streamed on the `behance.net` platform are manually labeled with important keyphrases. Our analysis of the dataset reveals the challenging nature of KP in transcripts. Moreover, for the first time in KP, we demonstrate the idea of improving KP for long documents (i.e., transcripts) by feeding models with paragraph-level keyphrases, i.e., hierarchical extraction. To foster future research, we will publicly release the dataset and code.

1 Introduction

Keyphrases are one or multiple consecutive words that could represent the main ideas in a document. Keyphrases are commonly categorized as Present or Absent. A present keyphrase explicitly appears in the document, while an absent keyphrase does not exist in the document. Keyphrases can serve as concise summary for a document, hence benefiting various NLP applications Information Retrieval (Hersh, 2021) and Text Summarization (Adhikari et al., 2020). Due to their usefulness, in the more than two decades, KP has been studied in many research works (Turney, 2000; Wu et al., 2005; Jiang et al., 2009; Hasan and Ng, 2014; Mahata et al., 2018; Chen et al., 2020; Ye et al., 2021).

Whereas traditionally feature engineering has been used for KP (Turney, 2000; Sheeba and Vivekanandan, 2014), recently deep learning is proved to be more efficient for this task (Ye et al., 2021; Ahmad et al., 2021). However, one limitation in the current works is that they are mainly limited to the formal text such as scientific papers (Meng et al., 2017) and web-logs (Xiong et al., 2019). As such, the challenges in other domains are still unresolved. Among others, video transcript is one of the less-explored domains that could significantly benefit from KP. For instance, it could be used for video summarization and retrieval or benefit people who are deaf and hard of hearing (DHH) (Kafle et al., 2019). On the other hand, KP for transcripts that are automatically obtained are more challenging than the formal written documents as these transcripts involve noisy text, incomplete/repeated sentences and phrases, informal vocabulary, and noncohesive information flow. Although there have been a few related attempts to evaluate feature engineering methods on meeting transcripts (Sheeba and Vivekanandan, 2014, 2012), the available resources, with a handful of transcripts and keyphrases, are not useful to train/evaluate the recent advanced deep models.

To address such limitations, we propose a large manually-labeled dataset for the domain of video transcripts. Specifically, we collect 500+ live-stream videos from the Behance platform. The videos are automatically transcribed by Microsoft Automatic Speech Recognition (ASR) tool. Since the video transcripts might be lengthy, summarizing the entire transcript into a few keyphrases might be challenging. Moreover, such keyphrases might not be helpful for partial retrieval where a part of the transcript is requested. As such, we annotate the collected transcripts in two levels: (1) Para-

Proceedings of the 29th International Conference on Computational Linguistics, pages 7147–7155
October 12–17, 2022.

graphs: A paragraph, consisting of multiple consecutive sentences, is a chunk of a transcript that provides a single point. Annotators first identify the paragraphs in a transcript. Next, the representative keyphrases for every paragraph are annotated; (2) Chapter: In addition to the paragraphs, we ask the annotators to provide a few keyphrases that could summarize multiple consecutive paragraphs that convey a single topic (e.g., how to make a special edit on an image). We call these units "*Chapter*", which are comparable to documents in other KP datasets. Annotators will first find the boundaries for chapters, then provide the keyphrases for each chapter.

We conduct extensive analysis on both levels of the KP task on the prepared dataset. Our analysis shows that KP in transcripts is a challenging task and more research is required. More importantly, for the first time for KP, we show that extracting keyphrases of long documents in a hierarchical order could result in better performance on document level KP. Specifically, our analysis shows that obtaining paragraph-level keyphrases and providing them to chapter-level KP systems could significantly boost the performance. The provided dataset and analysis could bring forth opportunities for more research on transcripts for KP.

2 Data Annotation

Data Collection: This work aims to annotate KP data for the domain of ASR text. To this end, we employ live-stream videos released on the social media platform `Behance.net`. The videos are streamed by artists and designers to share/discuss their creative projects. As such, verbal content from the speakers (in English) is important for video understanding. While the videos have initial subjects, their content is unplanned, hence the streamer might cut sentences, discuss multiple topics, and employ informal phrases. The videos have an average length of 48 minutes. To obtain the verbal content of the streamed videos, we employ the Microsoft ASR tool. In total, 361 videos with a total length of more than 500 hours are transcribed. A transcript, on average, contains 7,219 words.

Annotation: As presented in the introduction, the lengthy nature of transcripts motivates us to annotate keyphrases at two levels. First, at the paragraph level, we define a paragraph in a transcript to have the same role as paragraphs in formal written documents. Concretely, a paragraph is defined as a chunk of text that conveys a particular point or idea. A transcript consists of multiple disjoint paragraphs. Since the ASR text does not provide paragraph information, we manually annotate the collected transcripts with paragraphs. Afterward, for each paragraph of the transcript, the important keyphrases are selected. To this end, a keyphrase for a paragraph should have the following features: (a) Concisely summarize the main idea in the paragraph; (b) Be related to the main subject of the video; (c) Explicitly appear in the paragraph; (d) Does not appear in the previous or next paragraphs; (e) Form a proper English noun/verb phrase. The paragraphs that are entirely off-topic do not have any keyphrases. Second, at the chapter level, we provide keyphrases for chapters in the transcripts. A chapter consists of multiple paragraphs to represent a single topic. For instance, in a photo editing video, the discussion on how to change the background can form a chapter. A keyphrase of a chapter should observe the following criteria: (a) Concisely summarize the main topics in the chapter; (b) May not explicitly appear in the chapter; (c) Does not overlap with the paragraph keyphrases or other chapter level keyphrases; (d) Form a proper English noun/verb phrase. Note that paragraphs and chapters might have multiple keyphrases that are sorted based on their importance.

To annotate data for each level, we hire 10 annotators from the `upwork.com` platform which is a website for hiring freelancers with different expertise. Since the collected videos are related to photo editing software, e.g., Photoshop, we require the annotators to have experience both in data annotation and in using major photo editing tools. We train the annotators for KP at each level. To prevent chapter-level keyphrases to be biased toward paragraph-level keyphrases, we split annotator pool for paragraph and chapter level annotation (five for each). The transcripts are distributed evenly to the five annotators at each level for annotation. As such, a transcript is annotated entirely by a paragraph annotator and a chapter annotator (including boundary annotation). Chapter annotation is done after and uses outputs from paragraph annotation.

Annotation Agreement: Following prior work (Xiong et al., 2019), we assess the task difficulty of KP over video transcripts by evaluating the agreements of annotators at different levels. For each annotation level, we ask all the five annotators to independently annotate a sample of 5% of the tran-

Cut-off	Paragraph		Chapter	
	Exact	Partial	Exact	Partial
Keyphrases@1	60.21%	62.93%	58.92%	60.16%
Keyphrases@2	45.18%	58.09%	41.14%	54.19%
Keyphrases@3	37.12%	49.18%	35.21%	49.12%

Table 1: Average of pair-wise agreements among annotators at different cutoffs for paragraph and chapter level KP.

Statistics	Paragraph	Chapter
Number of samples	19,597	2,742
Number of keyphrases	34,392	12,155
Avg. keyphrase per sample	1.75	4.35
Avg. length of keyphrase	1.36	1.69
Avg. sample length	133.21	1047.70

Table 2: Statistics of the proposed dataset. The number of keyphrases represents the total number of annotated keyphrases for each level. The length of a keyphrase or sample is expressed in terms of the number of words.

scripts. Afterward, we compute the agreements of the five annotators at cutoffs @1, @2, and @3 with same rank comparison, using Exact Match (a keyphrase position is counted if the keyphrase is exactly the same from the annotators), and Partial Match (a keyphrase position is counted if the keyphrases from the annotators share at least one word). Table 1 shows the average of pair-wise agreements between annotators (i.e., comparing each pair of annotators). This table shows that KP in transcripts is a challenging task for both chapters and paragraphs. We attribute the challenges in this domain to the disconnected information flow in spontaneous talking compared to formal written documents that follow a clear information flow. Moreover, KP at the chapter level imposes more challenges as the agreement between judges drops from paragraph to chapter level. Finally, we show the statistics of the dataset in Table 2. A sample annotation is also presented in Appendix A.

Model	Paragraph		Chapter	
	F1@3	F1@M	F1@3	F1@M
One2Set	35.12	38.72	25.16	28.33
SEG-NET	34.19	38.92	24.42	29.37
BART	35.74	39.09	26.71	30.98
T5	36.09	39.12	25.78	30.18
GPT-2	37.90	41.27	27.91	32.27

Table 3: Performance of the models on the test sets for paragraph and chapter level keyphrase prediction.

Model	Keyphrases		Sentences+Keyphrases	
	F1@3	F1@M	F1@3	F1@M
BART	29.89	31.99	30.91	33.51
T5	28.71	31.72	29.85	32.80
GPT-2	30.08	33.28	33.69	35.72

Table 4: Performance of models on the chapter level test set. "Keyphrases": models use paragraph keyphrases as input; "Sentences+Keyphrases": models employs both paragraph keyphrases and hosting sentences.

3 Experiments

We randomly split the 361 transcripts into train/development/test sets with the ratio 80/10/10, respectively. The paragraphs and chapters of the transcripts in each split are then employed for our experiments in this section. Specifically, we first assess the challenges of KP at each level. Next, we empirically study how paragraph-level information can be helpful for chapter-level KP.

Baselines: We evaluate the performance of the following baselines on the proposed dataset: (1) **Generative Language Models**: The content of a paragraph or chapter are prompted to a generative language model (LM) to produce the keyphrases. Specifically, the language models are trained in an auto-regressive manner on sequence $S = [w_1, \ldots, w_n, [SEP], kp_1, \ldots, kp_m]$, where w_i is the i-th word in the input paragraph or chapter and kp_i is the i-th keyphrase. We employ GPT-2 (Radford et al., 2019), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) as three different versions of this baseline[1]. Note that for the chapters, since the transformer-based LMs impose a length limit, we truncate the input to the length of the maximum size of the LMs; (2) **SEG-NET** (Ahmad et al., 2021): In this baseline, salient sentences in the input text are first selected, then keyphrases are predicted by a generative model consuming the selected salient sentences. To select important sentences, a binary classifier is trained to distinguish sentences that contain a present keyphrase or partially overlap with an absent keyphrase; and (3) **One2Set** (Ye et al., 2021): The prediction of keyphrases is modeled as a set prediction task. Instead of imposing an order on the output of a transformer-based decoder, the model predicts keyphrases in parallel. We evaluate the models based on the macro-averaged F1@3 and F1@M. In the former, the predictions are truncated/padded at cutoff 3 while in the latter all model

[1]For T5 and BART, the task is formulated as seq2seq and $[SEP]$ is used to separate the input and output sequences.

Model	Paragraph						Chapter					
	P@3	R@3	F1@3	P@M	R@M	F1@M	P@3	R@3	F1@3	P@M	R@M	F1@M
One2Set	39.54	31.58	35.12	40.45	37.13	38.72	24.80	25.53	25.16	30.04	26.80	28.33
SEG-NET	40.80	29.42	34.19	41.27	36.82	38.92	23.94	24.91	24.42	31.12	27.80	29.37
BART	37.98	33.74	35.74	36.51	42.06	39.09	21.43	35.44	26.71	28.19	34.38	30.98
T5	35.42	36.78	36.09	38.19	40.09	39.12	26.77	24.86	25.78	29.38	31.02	30.18
GPT-2	39.12	36.75	37.90	40.72	41.83	41.27	25.49	30.83	27.91	30.59	34.14	32.27

Table 5: Performance of the models on the test sets for paragraph and chapter level keyphrase prediction.

Model	With Paragraph Keyphrases						With Paragraph Keyphrases and Sentences					
	P@3	R@3	F1@3	P@M	R@M	F1@M	P@3	R@3	F1@3	P@M	R@M	F1@M
BART	31.30	28.60	29.89	30.83	33.24	31.99	27.13	35.91	30.91	32.00	35.16	33.51
T5	24.46	34.74	28.71	29.64	34.11	31.72	26.92	33.49	29.85	35.71	30.32	32.80
GPT-2	32.86	27.73	30.08	31.72	35.00	33.28	34.07	33.31	33.69	31.10	41.95	35.72

Table 6: Performance of the generative models on the chapter level test set.

predictions are employed. Finally, we fine-tune the hyper-parameters for the models on development data.

Results: Table 3 shows the performance of the baselines on the paragraph and chapter level test sets There are several observations from the table. First, models employing a pre-trained language model, i.e., BART, T5, and GPT-2, outperform the baselines that train the transformers from scratch, i.e., One2Set and SEG-NET. We will thus focus on the generative models BART, T5, and GPT-2 in the next experiments. Second, the models have better performance on the paragraph level than the chapter level. This is expected as the models are required to encode larger context at the chapter level. Also, as the models employ transformers with input length restriction, they cannot encode the entire chapter. Our next experiments will explore an approach to handle long documents for KP. Finally, the performance of KP models is still far from being perfect in our dataset, e.g., the F1@M of One2Set on the NUC dataset (Nguyen and Kan, 2007) is 13% better than those on our dataset at chapter level (Ye et al., 2021), thus further demonstrating the modeling challenges of KP in video transcripts and presenting room for further research.

To provide detailed performance of the models, we report the precision and recall at cutoffs 3 and M. Specifically, for P@3 and R@3, the model predictions are truncated to the first three predictions. Following prior work (Ye et al., 2021), for cases that the model predicts less than three keyphrases, the prediction is padded with random keyphrases to have three keyphrases. For P@M and R@M, all model predictions are employed to evaluate the performance. The model performance is presented in Tables 5 and 6.

Motivated by the intuition that comprehending long documents requires understanding their smaller segments, we postulate that chapter-level KP models should appropriately capture paragraph information. In particular, we argue that paragraph-level keyphrases should be extracted first to provide summarization for paragraphs to improve chapter-level KP models afterward (hierarchical extraction). As such, we explore two methods to study this intuition: (1) Instead of truncating chapters, the input to the chapter KP systems will be the keyphrases of the paragraphs in the chapters. During training, we concatenate the golden keyphrases of all paragraphs in a chapter, i.e., $S = [kp_1, [SEP], kp_2, [SEP], \ldots, kp_m[SEP]]$. The models then predict chapter-level keyphrases using S. At inference time, we use the pre-trained paragraph-level KP model, which is based on the same model for the chapter KP system, to form the sequence S; (2) Since the keyphrases might not fully cover context of paragraphs, we further concatenate the keyphrases and their host sentences in the paragraphs to form the sequence S. Formally, the input to the chapter level KP system is $S = [S_1, [SEP_S], kp_1, [SEP_p], S_2, \ldots, S_m, [SEP_S], kp_m, [SEP]$, where S_i is the sentence in the chapter that contains the keyphrase kp_i. Using the generative model baselines, the results for the two methods are presented in Table 4. Comparing the paragraph keyphrase-augmented models with their vanilla counterparts in Table 3, it is evident that providing paragraph keyphrases significantly improves the performance of all models. We attribute this to better representations that the models with paragraph-level information

can obtain for chapters. Moreover, comparing the mere use of keyphrases with the augmentation of both keyphrases and sentences, the latter produces higher performance for chapter models. Overall, such results corroborates our intuition about the benefits of paragraph-level keyphrases for chapter-level KP, thus suggesting a potential direction of hierarchical modeling of long documents for KP.

4 Related Works

Keyphrase Prediction (KP) has been studied extensively in the past (Barker and Cornacchia, 2000; Turney, 2000; Hulth, 2003; Wan and Xiao, 2008; Hasan and Ng, 2014; Ye et al., 2021). Prior works can be categorized into extraction-based and generation-based solutions. In the former, keyphrases are extracted from input text, using either rule-based methods (Medelyan et al., 2009) or deep learning models (Sun et al., 2020) via sequence labeling (Gollapalli et al., 2017). In the generation-based models, deep generative models are employed to encode input documents and generates keyphrases (Chen et al., 2018; Zhao and Zhang, 2019; Ahmad et al., 2021). However, existing works on KP are mostly trained and evaluated on formal text. To this end, our work introduces a large-scale hierarchical KP dataset for video transcripts with informal and non-cohesive texts. We also note some related attempts to evaluate KP systems on meeting transcripts (Sheeba and Vivekanandan, 2014, 2012); however, the small size of these datasets hinders their relevance to deep learning era.

5 Conclusion

We present a novel hierarchical KP dataset over live-stream video transcripts. The dataset contains transcripts of 361 videos that are annotated at both paragraph and chapter levels. Our experiments show that KP in video transcripts is challenging and hierarchical extraction is helpful for KP in long documents. In the future, we will include more tasks in our dataset for video transcripts.

Ethical Considerations

In this work we present a dataset on the transcripts of a publicly accessible video-streaming platform `behance.net`. Complying with the discussion presented by Benton et al. (2017), research with human subjects information is exempted from the required full Institutional Review Board (IRB) review if the data is already available from public sources or if the identity of the subjects cannot be recovered. However, to protect the identity of the streamers and any other people whose information are shared in the video transcript, we impose extra consideration on the presented dataset. First, in this dataset, we exclude the usernames or any other identity-related information of the streamers in the transcripts to prevent disclosing their identity. Moreover, the proposed dataset only provides textual data (at paragraph and sentence levels), hence the other content of the videos (e.g., images, audios) are not revealed to protect human identity. Finally, to reduce the risk of disclosing the information of the people mentioned in the transcripts, in the final version of the dataset, we exclude the transcripts that explicitly or implicitly refer to the identify of the target people.

Acknowledgement

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IU-CRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, William L Hamilton, and Jimmy Lin. 2020. Exploring the limits of simple learners in knowledge distillation for document classification with docbert. In *Proceedings of the 5th Workshop on Representation Learning for NLP*.

Wasi Ahmad, Xiao Bai, Soomin Lee, and Kai-Wei Chang. 2021. Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *conference of the canadian society for computational studies of intelligence*.

Adrian Benton, Glen Coppersmith, and Mark Dredze. 2017. Ethical research protocols for social media health research. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 94–102, Valencia, Spain. Association for Computational Linguistics.

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*.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Sujatha Das Gollapalli, Xiao-Li Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kazi Saidul Hasan and Vincent Ng. 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 (Volume 1: Long Papers)*.

William Hersh. 2021. Information retrieval. In *Biomedical Informatics*.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Sushant Kafle, Peter Yeung, and Matt Huenerfauth. 2019. Evaluating the benefit of highlighting key words in captions for people who are deaf or hard of hearing. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2019, Pittsburgh, PA, USA, October 28-30, 2019*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880.

Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Key2Vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

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*.

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)*.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *International conference on Asian digital libraries*, pages 317–326. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. volume 1, page 9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. In *J. Mach. Learn. Res.*

JI Sheeba and K Vivekanandan. 2012. Improved keyword and keyphrase extraction from meeting transcripts. In *International Journal of Computer Applications*.

JI Sheeba and K Vivekanandan. 2014. A fuzzy logic based improved keyword extraction from meeting transcripts. In *International Journal on Computer Science and Engineering*.

Si Sun, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Jie Bao. 2020. Capturing global informativeness in open domain keyphrase extraction. In *arXiv preprint arXiv:2004.13639*.

Peter D Turney. 2000. Learning algorithms for keyphrase extraction. In *Information retrieval*.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*.

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.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. 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)*.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 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 (Volume 1: Long Papers)*.

Jing Zhao and Yuxiang Zhang. 2019. Incorporating linguistic constraints into keyphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

A Sample Annotation

To illustrate the annotated data, we present a sample annotation for a chapter in Table 7. This table shows three paragraphs of the chapter along with their keyphrases. Note that, first boundaries of the paragraphs in the transcripts are annotated. Next, for every paragraph annotators provide a few keyphrases that could summarize the main topic/points in the paragraph. Afterward, boundaries of the chapters in the transcripts, which consist of multiple paragraphs, are annotated. Finally, for every chapter, keyphrases that could best describe the main content of the chapter are provided by annotators. In the given example, the paragraph level keyphrases include "*Camera*", "*Background lights*", and "*Environment light, Rotations*" for the three paragraphs. For this chapter, the keyphrase "*Setting Environment*" is provided.

ID	Content	Paragraph Keyphrases
1	We have beautifully beautiful summer day outside with our cup of coffee. If you ever log on your arm. All right up corner you will see that currently we are located in the camera view. If you would like to adjust your 3D model I will recommend you to switch to viewport camera. In this case you will not affect your camera perspective during your model adjustment, so keep it in mind when you will be ready to come back to your camera view. Simply switch from a top corner or directly from you seen a pen or just simply click on camera.	Camera
2	Just like duck. Light are you can come. Ah, click on environment. In. Here you can adjust background lights. Opposite team environment might need background blue. You can make it more blurry or or less blurry. Also if you will switch to light you will be able to adjust.	Background lights
3	Your light you can. Uh, idiot environment light just like that. You can make it brighter or more cloudy also rotation. You can rotate your alight so keep before you will rotate your light. Keep in mind and pay close attention to your background image to your main source of light just like that.	Environment light, Rotation

Table 7: Sample annotations for keyphrases of a chapter. Annotators first find the boundaries of the paragraphs, then provide keyphrases for every paragraph. At the chapter level, annotators identify paragraphs to form chapters before assigning keyphrases for chapters. The keyphrase "*Setting Environment*" is provided for the chapter (with three paragraphs) in this example.