LDKP - A Dataset for Identifying Keyphrases from Long Scientific Documents

Debanjan Mahata[1,2,*,†], Navneet Agarwal[2,†], Dibya Gautam[2,†], Amardeep Kumar[3,†], Swapnil Parekh[4], Yaman Kumar Singla[5,2], Anish Acharya[6] and Rajiv Ratn Shah[2]

[1]Moody's Analytics, USA
[2]MIDAS Labs, IIIT-Delhi, India
[3]Instabase, India
[4]New York University, USA
[5]Adobe Media and Data Science Research (MDSR), India
[6]University of Texas at Austin, USA

Abstract

Identifying keyphrases (KPs) from text documents is a fundamental task in natural language processing and information retrieval. Vast majority of the benchmark datasets for this task are from the scientific domain containing only the document title and abstract information. This limits keyphrase extraction (KPE) and keyphrase generation (KPG) algorithms to identify keyphrases from human-written summaries that are often very short ($\approx$ 8 sentences). This presents three challenges for real-world applications: i) human-written summaries are unavailable for most documents, ii) a vast majority of the documents are long, and iii) a high percentage of KPs are *directly* found beyond the limited context of the title and the abstract. Therefore, we release two extensive corpora mapping KPs of $\approx 1.3M$ and $\approx 100K$ scientific articles with their fully extracted text and additional metadata including publication venue, year, author, field of study, and citations for facilitating research on this real-world problem. Additionally, we also benchmark and report the performances of different unsupervised as well as supervised algorithms for keyphrase extraction on long scientific documents. Our experiments show that formulating keyphrase extraction as a sequence tagging task with modern transformer language models capable of processing long text sequences such as longformer has advantages over the traditional algorithms, not only resulting in better performances in terms of F1 metrics but also in learning to extract optimal number of keyphrases from the input documents.

Keywords

keyphrase extraction, keyphrase generation, keyphrasification, automatic identification of keyphrases, long documents, longformer, language models

1. Introduction and Background

Identifying keyphrases (KPs) is a form of extreme summarization, where given an input document, the task is to find a set of **representative** phrases that can effectively summarize it [1]. Over the last decade, we have seen an exponential increase in the velocity at which unstructured text is produced on the web, with the vast majority of them untagged or poorly tagged. KPs provide an effective way to search, summarize, tag, and manage these documents. Identifying KPs have proved to be useful as preprocessing, pre-training [2], or supplementary tasks in other tasks such as search [3, 4, 5], recommendation systems [6], advertising [7], summarization [8], opinion mining [9] to name a few. This has motivated researchers to explore machine learning algorithms for automatically mapping documents to a set of keyphrases commonly referred as the *keyphrase extraction* (KPE) task [10, 6], for extractive approaches, and *keyphrase generation* (KPG) task [11, 12] for generative approaches. Recently, it was also referred as *Keyphrasification* [1].

Various algorithms have been proposed over time to solve the problem of identifying keyphrases from text documents that can primarily be categorized into supervised and unsupervised approaches [18]. Majority of these approaches take an abstract (*a summary*) of a text document as the input and produce keyphrases as output. However, in industrial applications across different domains such as advertising [19], search and indexing [20], finance [21], law [22], and many other real-world use cases, document summaries are not readily available. Moreover, most of the documents encountered in these applications are greater than 8 sentences (the average length of abstracts in KP datasets, see Table 1). We also find that a significant percentage of keyphrases (>18%) are *directly* found beyond the limited context of a document's title and abstract/summary. These constraints limit the potential

DL4SR'22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

*Debanjan Mahata participated in this work as an Adjunct Faculty at IIIT-Delhi.

†These authors contributed equally.

debanjanmahata85@gmail.com (D. Mahata)

Dataset	Size (no. of docs)	Long Documents	Avg no. of sentences	Avg no. of words	Present KPs	Absent KPs
SemEval 2017 [6]	0.5K	×	7.36	176.13	42.01%	57.69%
KDD [13]	0.75K	×	8.05	188.43	45.99%	54.01%
Inspec [14]	2K	×	5.45	130.57	55.69%	44.31%
KP20K [11]	568K	×	7.42	188.47	57.4%	42.6%
OAGKx [15]	22M	×	8.87	228.50	52.7%	47.3%
NUS [16]	0.21K	✓	375.93	7644.43	67.75%	32.25%
SemEval 2010 [10]	0.24K	✓	319.32	7434.52	42.01%	57.99%
Krapivin [17]	2.3K	✓	370.48	8420.76	44.74%	52.26%
LDKP3K (S2ORC ← KP20K)	100K	✓	280.67	6027.10	76.11%	23.89%
LDKP10K (S2ORC ← OAGKx)	1.3M	✓	194.76	4384.58	63.65%	36.35%

Table 1
Characteristics of the proposed datasets compared to the existing datasets.

of currently developed KPE and KPG algorithms to only theoretical pursuits.

Many previous studies have pointed out the constraints imposed on KPE algorithms due to the short inputs and artificial nature of available datasets [23, 24, 25, 26, 27]. In particular, Cano and Bojar [25] while explaining the limitations of their proposed algorithms, note that the title and the abstract may not carry sufficient topical information about the article, even when joined together. While most datasets in the domain of KPE consist of titles and abstracts [15], there have been some attempts at providing long document KP datasets as well (Table 1). Krapivin et al. [17] released 2,000 full-length scientific papers from the computer science domain. Kim et al. [10] in a SemEval-2010 challenge released a dataset containing 244 full scientific articles along with their author and reader assigned keyphrases. Nguyen and Kan [16] released 211 full-length scientific documents with multiple annotated keyphrases. All of these datasets were released more than a decade ago and were more suitable for machine-learning models available back then. With today's deep learning paradigms like un/semi-supervised learning requiring Wikipedia sized corpora (>6M articles), it becomes imperative to update the KPE and KPG tasks with similar sized corpus.

In this work, we develop two large datasets (LDKP - Long Document Keyphrase) comprising of 100K and 1.3M documents for identifying keyphrases from full-length scientific articles along with their metadata information such as venue, year of publication, author information, inbound and outbound citations, and citation contexts, among others. We achieve this by mapping the existing KP20K [11] and OAGKx [15] corpus to the documents available in S2ORC dataset [28]. We make the dataset publicly available on Huggingface hub (Section 2.2) and also integrate the processing of these datasets with the *datasets*[1] and *transformerkp*[2] libraries. We hope that researchers working in this area would acknowledge the shortcomings of the popularly used datasets and methods in KPE and KPG and devise exciting new approaches for overcoming the challenges related to identifying keyphrases from long documents and contexts beyond summaries. This would make the models more useful in practical real-world settings. We think that *LDKP* can also complement recent efforts towards creating suitable benchmarks [29] for evaluating methods being developed to understand and process long text sequences.

2. Dataset

We propose two datasets resulting from the mapping of S2ORC with KP20K and OAGKx corpus, respectively. Lo et al. [28] publicly released S2ORC as a huge corpus of 8.1M scientific documents. While it has full text and metadata (see Table 2) the corpus does not contain keyphrases. We took this as an opportunity to create a new corpus for identifying keyphrases from full-length scientific articles. Therefore, we took the KP20K and OAGKx scientific corpus for which keyphrases were already available and mapped them to their corresponding documents in S2ORC.

This is the first time in the keyphrase community that such a large number of full-length documents with comprehensive metadata information have been made publicly available for academic use. Here, we want to acknowledge another concurrent work [30] that looks at the task of keyphrase generation from a newly constructed corpus

[1] https://github.com/huggingface/datasets
[2] transformerkp - is a transformer based deep learning library for training and evaluating keyphrase extraction and generation algorithms, https://github.com/Deep-Learning-for-Keyphrase/transformerkp

```
Tokenized Document:  [['Correlating', 'the', 'effects', 'of', 'flow', 'and',
                       'telepresence', 'in', 'virtual', 'worlds:', 'Enhancing',
                       'our', 'understanding', 'of', 'user', 'behavior', 'in',
                       'game-based', 'learning'], ...]

Document BIO Tags:  [['O', 'O', 'O', 'O', 'B', 'O', 'B', 'O', 'B', 'I', 'O', 'O',
                      'O', 'O', 'O', 'O', 'O', 'O', 'O'], ...]
```

Figure 1: B-I-O tagged tokens from a random sample in the LDKP dataset where, 'B' - start of a keyphrase span, 'I' - inside keyphrase span, 'O' - outside keyphrase span.

Paper details	Paper Identifier	Citations and References
Paper ID	ArXiv ID	Outbound Citations
Title	ACL ID	Inbound Citations
Authors	PMC ID	Bibliography
Year	PubMed ID	References
Venue	MAG ID	
Journal	DOI	
Field of Study	S2 URL	

Table 2
Information available in the metadata of each scientific paper in LDKP corpus.

of long documents - FULLTEXTKP. However, they do not make the corpus publicly available and the corpus is significantly smaller than ours containing only $\approx 142K$ documents.

We release two datasets LDKP3K and LDKP10K corresponding to KP20K and OAGKx, respectively. The first corpus consists of $\approx$ **100K** long documents with keyphrases obtained by mapping KP20K to S2ORC. The KP20K corpus mainly contains title, abstract and keyphrases for computer science research articles from online digital libraries like ACM Digital Library, ScienceDirect, and Wiley. Using S2ORC documents, we increase the average length of the documents in KP20K from 7.42 sentences to 280.67 sentences. This also increased the percentage of present keyphrases in the input text by 18.7%.

The second corpus corresponding to OAGKx consists of **1.3M** full scientific articles from various domains with their corresponding keyphrases collected from academic graph [31, 32]. The resulting corpus contains 194.7 sentences (up from 8.87 sentences) on an average with 10.95% increase in present keyphrases. An increase in percentage of present keyphrases in both the corpus when expanded to full length articles clearly indicates the occurrence of a significant chunk of the keyphrases beyond the abstract. Since both datasets consist of a large number of documents, we present three versions of each dataset with the training data split into *small*, *medium* and *large* sizes, as given in Table 3. This was done in order to provide an opportunity to the researchers and practitioners with scarcity of computing resources to evaluate the performance of their methods on a smaller dataset.

2.1. Dataset Preparation

In the absence of any unique identifier shared across datasets, we used paper title to map documents in S2ORC with KP20K/OAGKx. This had its own set of challenges. For example, some papers in KP20K and OAGKx had unigram titles like "Editorial" or "Preface". Multiple papers can be found with the same title. We ignored all the papers with unigram and bigram titles. We resolved the title conflicts through manual verification. We also found out that some of the keyphrases in OAGKx and KP20K datasets were parsed incorrectly. Keyphrases that contain delimiters such as *comma* (which is also used as a separator for keyphrase list) have been broken down into two or more keyphrases, *e.g.*, the keyphrase '*2,4-dichlorophenoxyacetic acid*' has been broken down into ['2', '4- dichlorophenoxyacetic acid']. In some cases, the publication year, page number, DOI, *e.g.*, *1999:14:555-558*, were inaccurately added to the list of keyphrases. To solve this, we filtered out all the keyphrases that did not have any alphabetical characters in them.

Next, in order to facilitate the usage of particular sections in KPE algorithms, we standardized the section names across all the papers. The section names varied across different papers in the S2ORC dataset. For example, some papers have a section named *"Introduction"* while others have it as *"1.Introduction", "I. Introduction", "I Introduction" etc*. To deal with this problem, we replaced the unique section names with a common generic section name, like *"introduction"*, across all the papers. We did this for common sections which includes *introduction*, *related work*, *conclusion*, *methodology*, *results and analysis*.

In order to make the dataset useful for training a sequence tagging model we also provide token level tags in *B-I-O* format as previously done in [33]. We marked all the words in the document belonging to the keyphrases as 'B' or 'I' depending on whether they are the first word of the keyphrase or otherwise. Every other word, which were not a part of a keyphrase were tagged as 'O'. The

Dataset		LDKP3K (no. of docs)	Size (no. of docs)
Train	Small	20,000	20,000
	Medium	50,000	50,000
	Large	90,019	1,296,613
Test		3,413	10,000
Validation		3,339	10,000

Table 3
LDKP datasets with their train, validation and test dataset distributions.

ground truth keyphrases associated with the documents were identified by searching for the same string pattern in the document's text. The text is tokenized using a whitespace tokenizer and a mapping between each token and it's corresponding tag is provided as shown in Figure 1.

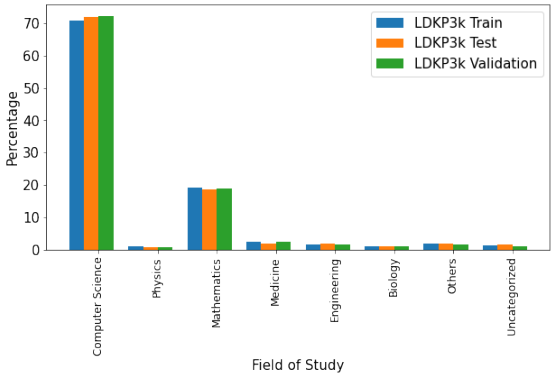

Figure 2: Distribution of field of studies for train, test and validation split of LDKP3k dataset.

The proposed dataset LDKP3k and LDKP10k are further divided into train, test and validation splits as shown in Table-3. For LDKP3k, these splits are based on the original KP20K dataset. For LDKP10k, we resorted to random sampling method to create these splits since OAGKx, the keyphrase dataset corresponding to LDKP10k, wasn't originally divided into train, test and validation splits. Figures 2 and 3 show the distribution of papers in terms of field of study across all the splits of the LDKP3k and LDKP10k datasets, respectively.

2.2. Dataset Usage

We make all the datasets publicly available on Huggingface hub and enable programmatic access to the data using the *datasets* library. For example, Figure 4 shows a sample code for downloading the LDKP3K dataset with the 'small' training data split. Similarly, other configurations like 'medium' and 'large' can also be downloaded, each having different sizes of the training data but the same validation and test dataset. Figure 4 also shows how each split of the dataset can be accessed.

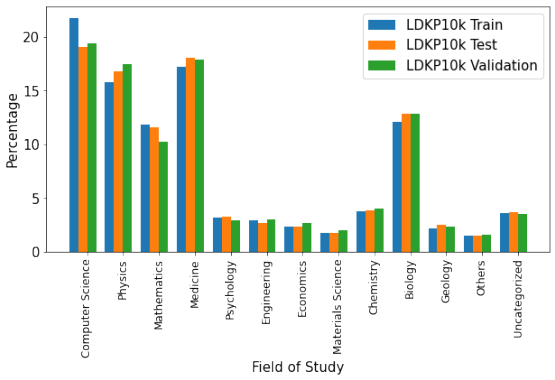

Figure 3: Distribution of field of studies for train, test and validation split of LDKP10k dataset.

```python
from datasets import load_dataset

# get dataset
dataset = load_dataset("midas/ldkp3k", "small")

# get train split
train_data = dataset["train"]

# get validation split
validation_data = data["validation"]

# get test split
test_data = data["test"]
```

Figure 4: Sample code for downloading the 'small' split of the LDKP3K dataset.

Please refer to the Huggingface hub pages for LDKP3k and LDKP10k for detailed information about downloading and using the dataset.

1. LDKP3K - https://huggingface.co/datasets/midas/ldkp3k
2. LDKP10K - https://huggingface.co/datasets/midas/ldkp10k

We also enable access of the datasets using the *transformerkp* library, which abstracts away the preprocessing steps and make the data splits readily available to the user for the tasks of keyphrase extraction using sequence tagging and keyphrase generation using seq2seq methods, respectively with different transformer based language models. Details of downloading and using the datasets with *transformerkp* for the tasks of keyphrase extraction and generation could be found over here - https://deep-learning-for-keyphrase.github.io/transformerkp/how-to-guides/keyphrase-data/

Method	Krapivin		NUS		SemEval-2010		LDKP3k		LDKP10k	
	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
PositionRank	0.042	0.052	0.060	0.086	0.074	0.098	0.059	0.062	0.052	0.061
TextRank	0.036	0.047	0.071	0.090	0.085	0.117	0.082	0.094	0.068	0.074
TopicRank	0.071	0.080	0.130	0.152	0.111	0.132	0.108	0.110	0.098	0.102
SingleRank	0.001	0.003	0.005	0.008	0.009	0.010	0.016	0.025	0.011	0.014
MultipartiteRank	0.103	0.107	0.150	0.193	0.116	0.145	0.129	0.110	0.104	0.106
TopicalPageRank	0.009	0.012	0.046	0.059	0.014	0.024	0.019	0.027	0.020	0.031
SGRank	**0.140**	**0.131**	**0.195**	**0.203**	**0.177**	**0.201**	**0.138**	**0.128**	**0.136**	**0.132**

Table 4
Results on long document datasets using unsupervised graph-based models.

Method	Krapivin		NUS		SemEval-2010		LDKP3k		LDKP10k	
	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
TFIDF	0.033	0.052	0.063	0.111	0.062	0.070	0.093	0.099	0.072	0.080
KPMiner	**0.125**	**0.151**	0.169	0.212	**0.155**	**0.181**	**0.164**	**0.152**	**0.151**	**0.142**
Yake	0.105	0.107	**0.177**	**0.235**	0.088	0.129	0.140	0.132	0.114	0.114

Table 5
Results on long document datasets using unsupervised statistical models.

3. Experiments

In this section, we evaluate several popular keyphrase extraction algorithms on the proposed LDKP3K and LDKP10K datasets, along with three of the other existing smaller datasets in scientific domain comprising of full length documents - *Krapivin*, *SemEval-2010*, and *NUS*. A majority of the previous works have reported scores for *Krapivin*, *SemEval-2010*, and *NUS*, by only considering the title and abstract as the input We further report the benchmark results and also discuss the comparative advantage of different algorithms to provide future research direction.

3.1. Unsupervised Methods

There are multiple unsupervised methods for extracting keyphrases from a document. We used the following popular statistical models: TfIDf, KPMiner [34], YAKE [35] and the following graph-based algorithms: TextRank [36], PositionRank [37], SingleRank [38], TopicRank [39], MultipartiteRank [40] and SGRank [41]. All the implementations were taken from the PKE toolkit [42], except SGRank, for which we used the implementation available in the textacy[3] library. These algorithms first identify the candidate keyphrases using lexical rules followed by ranking the candidates using either a statistical approach or a graph-based approach [1]. We directly reported the performance scores of these methods on the test datasets (Table 4).

[3] https://github.com/chartbeat-labs/textacy

3.2. Supervised Methods

For supervised keyphrase extraction, we report results for two traditional models, namely - KEA [43] and WINGNUS [23], which treat keyphrase extraction as a binary classification task. A recent trend is to treat keyphrase extraction as a sequence tagging task [33, 2, 1]. Transformer based language models like BERT [44], RoBERTa [45], KBIR [2], have already shown to achieve SOTA results on the task of keyphrase extraction when only the title and abstract is taken as the input. However, all these models have a limitation of processing only 512 sub-word tokens. This led us to try Longformer [46], which can handle long sequences of text of up to 4,096 sub-word tokens. We acknowledge that there are several other recent models such as [47, 48] which could have been also tried. We are surely interested to try the others in a future work. Further, we would train larger models on the LDKP *large* corpus.

3.3. Evaluation Metrics

We used $F1@5$ and $F1@10$ as our evaluation metrics [10]. Equations 1, 2 and 3 shows how $F1@k$ is calculated. Before evaluating, we lower-cased, stemmed, and removed punctuations from the ground truth as well as the predicted keyphrases, and used actual matching. Let Y denote the ground truth keyphrases and $\bar{Y} = (\bar{y_1}, \bar{y_2}, \ldots, \bar{y_m})$ denote the predicted keyphrases ordered by their quality of prediction. Then we can define the metrics as follows:

$$Precision@k = \frac{|Y \cap \bar{Y_k}|}{min\{|\bar{Y_k}|, k\}} \quad (1)$$

Method	Krapivin		NUS		SemEval-2010		LDKP3k		LDKP10k	
	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
Kea	0.041	0.063	0.069	0.134	0.077	0.090	0.109	0.118	0.087	0.096
WINGNUS	0.059	0.151	0.057	0.085	0.059	0.152	0.099	0.109	0.093	0.102
longformer-base-4096	**0.229**	**0.232**	**0.253**	**0.284**	**0.203**	**0.219**	**0.240**	**0.216**	**0.236**	**0.212**

Table 6
Results on long document datasets for supervised models.

$$Recall@k = \frac{|Y \cap \bar{Y_k}|}{|Y|} \tag{2}$$

$$F1@k = \frac{2 * Precision@k * Recall@k}{Precision@k + Recall@k} \tag{3}$$

where $\bar{Y_k}$ denotes the top k elements of the set $\bar{Y}$.

3.4. Results

Algorithm	LDKP3K	LDKP10K
SGRank	86.96	85.56
TopicRank	636.02	520.81
PositionRank	678.65	547.66
TopicalPageRank	709.51	574.50
Singlerank	773.11	624.25
TextRank	773.11	624.25
Multipartite	636.02	520.78
Yake	2475.20	1965.73
TfIDF	6472.93	4922.29
KPMiner	79.51	74.81
WINGNUS	659.47	544.91
Kea	2534.71	2032.84

Table 7
Average number of candidate keyphrases generated by the supervised and unsupervised algorithms on LDKP3K and LDKP10K datasets.

Unsupervised algorithms did not show better performance than their supervised counterparts on long documents as shown in Tables 4, 5, and 6. For the unsupervised approaches SGRank and KPMiner outperformed every other algorithm in the graph-based ranking and statistical categories respectively. One possible reason for the low performance of the other unsupervised techniques could be that during the candidate generation and ranking phases these models had to deal with more noise than what they have been tuned to. Table 7 shows the number of candidates generated by the strategies used by each of these algorithms. We can easily observe that most of the techniques resulted in generating a huge number of candidate keyphrases which might have made the downstream ranking process challenging. On the other hand, we can see that both SGRank and KPMiner had strategies which were able to significantly reduce the number of generated candidates and come up with better set of keyphrases. The other algorithms might get benefited by revisiting their pipeline and make necessary changes for processing long documents and tune their heuristics to generate better quality candidates, which are to be ranked later for identifying the keyphrases.

For the supervised approaches using Longformer in a sequence tagging setup proved to be the most promising technique as shown by the performance reported in Table 6. Treating keyphrase extraction as a sequence tagging problem also automatically learns the optimal amount of keyphrases to be predicted and helps to overcome the challenges with other strategies that has to deal with a large number of candidates as discussed above. The longformer model on an average predicted 6.25 and 6.08 number of keyphrases for the LDKP10k and LDKP3k test sets, respectively.

4. Conclusion

In this work, we identified the shortage of corpus comprising of long documents for training and evaluating keyphrase extraction and generation models. We created two very large corpus - LDKP3K and LDKP10K comprising of $\approx$ 100K and $\approx$ 1.3M documents and made it publicly available. The results of keyphrase extraction on long documents with some of the existing unsupervised and supervised models clearly depicts the challenging nature of the problem. We hope this would encourage the researchers to innovate and propose new models capable of identifying high quality keyphrases from long multi-page documents.

References

[1] R. Meng, D. Mahata, F. Boudin, From fundamentals to recent advances: A tutorial on keyphrasification, in: European Conference on Information Retrieval, Springer, 2022, pp. 582–588.

[2] M. Kulkarni, D. Mahata, R. Arora, R. Bhowmik, Learning rich representation of keyphrases from text, arXiv preprint arXiv:2112.08547 (2021).

[3] D. K. Sanyal, P. K. Bhowmick, P. P. Das, S. Chattopadhyay, T. Santosh, Enhancing access to scholarly publications with surrogate resources, Scientometrics 121 (2019) 1129–1164.

[4] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, E. Frank, Improving browsing in digital libraries with keyphrase indexes, Decision Support Systems 27 (1999) 81–104.

[5] I. Y. Song, R. B. Allen, Z. Obradovic, M. Song, Keyphrase extraction-based query expansion in digital libraries, in: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06), IEEE, 2006, pp. 202–209.

[6] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications, arXiv preprint arXiv:1704.02853 (2017).

[7] W.-t. Yih, J. Goodman, V. R. Carvalho, Finding advertising keywords on web pages, in: Proceedings of the 15th international conference on World Wide Web, 2006, pp. 213–222.

[8] V. Qazvinian, D. Radev, A. Özgür, Citation summarization through keyphrase extraction, in: Proceedings of the 23rd international conference on computational linguistics (COLING 2010), 2010, pp. 895–903.

[9] G. Berend, Opinion expression mining by exploiting keyphrase extraction (2011).

[10] S. N. Kim, O. Medelyan, M.-Y. Kan, T. Baldwin, Automatic keyphrase extraction from scientific articles, Language resources and evaluation 47 (2013) 723–742.

[11] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, Y. Chi, Deep keyphrase generation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 582–592. URL: https://www.aclweb.org/anthology/P17-1054. doi:10.18653/v1/P17-1054.

[12] A. Swaminathan, H. Zhang, D. Mahata, R. Gosangi, R. Shah, A. Stent, A preliminary exploration of gans for keyphrase generation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 8021–8030.

[13] C. Caragea, F. A. Bulgarov, A. Godea, S. D. Gollapalli, Citation-enhanced keyphrase extraction from research papers: A supervised approach, in: EMNLP, 2014.

[14] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, in: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP '03, Association for Computational Linguistics, USA, 2003, p. 216–223. URL: https://doi.org/10.3115/1119355.1119383. doi:10.3115/1119355.1119383.

[15] E. Çano, OAGKX keyword generation dataset, 2019. URL: http://hdl.handle.net/11234/1-3062, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[16] T. D. Nguyen, M.-Y. Kan, Keyphrase extraction in scientific publications, in: D. H.-L. Goh, T. H. Cao, I. T. Sølvberg, E. Rasmussen (Eds.), Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 317–326.

[17] M. Krapivin, A. Autayeu, M. Marchese, E. Blanzieri, N. Segata, Keyphrases extraction from scientific documents: Improving machine learning approaches with natural language processing, volume 6102, 2010, pp. 102–111. doi:10.1007/978-3-642-13654-2_12.

[18] E. Papagiannopoulou, G. Tsoumakas, A review of keyphrase extraction, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (2020) e1339.

[19] Z. Hussain, M. Zhang, X. Zhang, K. Ye, C. Thomas, Z. Agha, N. Ong, A. Kovashka, Automatic understanding of image and video advertisements, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1705–1715.

[20] W. Magdy, K. Darwish, Book search: indexing the valuable parts, in: Proceedings of the 2008 ACM workshop on Research advances in large digital book repositories, 2008, pp. 53–56.

[21] A. Gupta, V. Dengre, H. A. Kheruwala, M. Shah, Comprehensive review of text-mining applications in finance, Financial Innovation 6 (2020) 1–25.

[22] R. Bhargava, S. Nigwekar, Y. Sharma, Catchphrase extraction from legal documents using lstm networks., in: FIRE (Working Notes), 2017, pp. 72–73.

[23] T. D. Nguyen, M.-T. Luong, Wingnus: Keyphrase extraction utilizing document logical structure, in: Proceedings of the 5th international workshop on semantic evaluation, 2010, pp. 166–169.

[24] K. S. Hasan, V. Ng, 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), 2014, pp. 1262–1273.

[25] E. Cano, O. Bojar, Keyphrase generation: A text summarization struggle, arXiv preprint arXiv:1904.00110 (2019).

[26] Y. Gallina, F. Boudin, B. Daille, Large-scale evaluation of keyphrase extraction models, in: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, 2020, pp. 271–278.

[27] C. G. Kontoulis, E. Papagiannopoulou,

G. Tsoumakas, Keyphrase extraction from scientific articles via extractive summarization, in: Proceedings of the Second Workshop on Scholarly Document Processing, 2021, pp. 49–55.

[28] K. Lo, L. L. Wang, M. Neumann, R. Kinney, D. S. Weld, S2orc: The semantic scholar open research corpus, arXiv preprint arXiv:1911.02782 (2019).

[29] U. Shaham, E. Segal, M. Ivgi, A. Efrat, O. Yoran, A. Haviv, A. Gupta, W. Xiong, M. Geva, J. Berant, et al., Scrolls: Standardized comparison over long language sequences, arXiv preprint arXiv:2201.03533 (2022).

[30] K. Garg, J. R. Chowdhury, C. Caragea, Keyphrase generation beyond the boundaries of title and abstract, arXiv preprint arXiv:2112.06776 (2021).

[31] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, K. Wang, An overview of microsoft academic service (mas) and applications, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 243–246.

[32] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 990–998.

[33] D. Sahrawat, D. Mahata, H. Zhang, M. Kulkarni, A. Sharma, R. Gosangi, A. Stent, Y. Kumar, R. R. Shah, R. Zimmermann, Keyphrase extraction as sequence labeling using contextualized embeddings, Advances in Information Retrieval 12036 (2020) 328.

[34] S. R. El-Beltagy, A. Rafea, Kp-miner: A keyphrase extraction system for english and arabic documents, Information systems 34 (2009) 132–144.

[35] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! keyword extraction from single documents using multiple local features, Information Sciences 509 (2020) 257–289.

[36] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 404–411.

[37] C. Florescu, C. Caragea, Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1105–1115.

[38] X. Wan, J. Xiao, Collabrank: towards a collaborative approach to single-document keyphrase extraction, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), 2008, pp. 969–976.

[39] A. Bougouin, F. Boudin, B. Daille, Topicrank: Graph-based topic ranking for keyphrase extraction, in: International joint conference on natural language processing (IJCNLP), 2013, pp. 543–551.

[40] F. Boudin, Unsupervised keyphrase extraction with multipartite graphs, arXiv preprint arXiv:1803.08721 (2018).

[41] S. Danesh, T. Sumner, J. H. Martin, Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction, in: Proceedings of the fourth joint conference on lexical and computational semantics, 2015, pp. 117–126.

[42] F. Boudin, Pke: an open source python-based keyphrase extraction toolkit, in: Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations, 2016, pp. 69–73.

[43] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, Kea: Practical automated keyphrase extraction, in: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, IGI global, 2005, pp. 129–152.

[44] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.

[45] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[46] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, CoRR abs/2004.05150 (2020). URL: https://arxiv.org/abs/2004.05150. arXiv:2004.05150.

[47] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, A. Ahmed, Big bird: Transformers for longer sequences, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 17283–17297. URL: https://proceedings.neurips.cc/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

[48] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=rkgNKkHtvB.