

MORTY: Structured Summarization for Targeted Information Extraction from Scholarly Articles

Mohamad Yaser Jaradeh¹[0000-0001-8777-2780], Markus Stocker²[0000-0001-5492-3212], and Sören Auer²[0000-0002-0698-2864]

¹ L3S Research Center, Leibniz University Hannover, Germany
jaradeh@l3s.de

² Leibniz Information Centre for Science and Technology
{auer,markus.stocker}@tib.eu

Abstract. Information extraction from scholarly articles is a challenging task due to the sizable document length and implicit information hidden in text, figures, and citations. Scholarly information extraction has various applications in exploration, archival, and curation services for digital libraries and knowledge management systems. We present MORTY, an information extraction technique that creates structured summaries of text from scholarly articles. Our approach condenses the article’s full-text to property-value pairs as a segmented text snippet called structured summary. We also present a sizable scholarly dataset combining structured summaries retrieved from a scholarly knowledge graph and corresponding publicly available scientific articles, which we openly publish as a resource for the research community. Our results show that structured summarization is a suitable approach for targeted information extraction that complements other commonly used methods such as question answering and named entity recognition.

Keywords: Information Extraction · Scholarly Knowledge · Summarization · Natural Language Processing · Literature Review Completion.

1 Introduction

By their very nature, scholarly articles tend to be dense with information and knowledge [20]. The task of information extraction (IE) has been widely researched by the community in a variety of contexts [3, 22, 13], including the scholarly domain [35, 19]. However, information extraction from scholarly articles continues to suffer from low accuracy. Reasons include ambiguity of scholarly text, information representation in scholarly articles, and lack of training datasets [29].

Other than retrospective information extraction, initiatives such as the ORKG [9], Hi-Knowledge [11], and Coda [31] collect structured scholarly information by engaging researchers in the knowledge curation process. In ORKG, information is collected by experts that extract and structure the essential information from

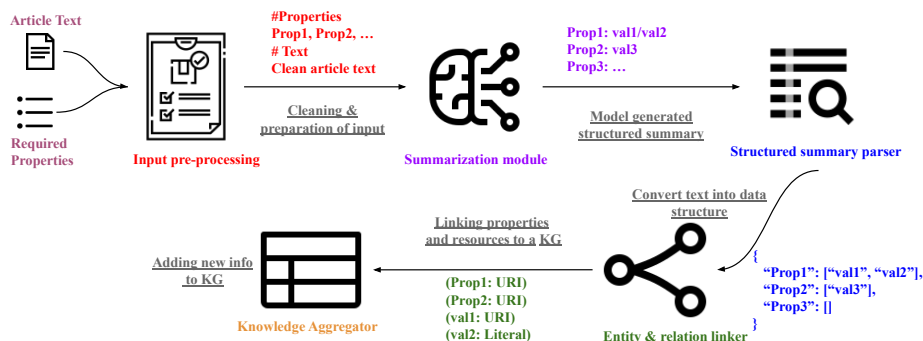


Fig. 1. Bird’s eye view on the complete workflow of employing structured summarization in the context of information extraction from scholarly documents and articles (MORTY).

articles. However, experts might not use the exact wording from the original article or might put forward a novel segment of text that did not exist before in the original text.

Information extraction techniques [28] could play a supporting role through automated extraction, suggestions to experts or autonomously adding extracted information to a data source (e.g. database or knowledge graph). However, blindly extracting information (i.e. factual extractions) is not suitable for scholarly data due to the large amount of information condensed into little text. Blind extraction refers to Open Information Extraction [8] that relies on propositions and facts as well as common entities and relations between them. For scholarly articles, a more targeted approach is required, whereby a system is able to extract a set of predefined properties and their corresponding values while ignoring others.

We propose MORTY, a method that leverages summarization tasks conducted by deep-learning language models to create structured summaries that can be parsed into extracted information, stored in a knowledge graph. We present, evaluate, and discuss MORTY. Furthermore, we highlight the research problems, possible solutions, limitations of the approach, and review open questions and future prospects.

The core contributions of this article are: First, a dataset of paper full texts with a list of property-value pairs of human-expert annotated information. Second, an approach for information extraction from scholarly articles using structured summarization³.

2 Related Work

Information Extraction. Several information extraction methods have been proposed by the community, each with their own advantages and disadvantages.

³ Code & data (with stats): <https://github.com/YaserJaradeh/MORTY>

Ji et al. [12] proposed an end-to-end system that uses a multi-task model to perform sentence classification and information extraction on legal documents. TRIE [42] uses end-to-end system to jointly perform document reading and information extraction on everyday documents such as invoices, tickets, and resumes. Chua and Duffy [4] proposes a method for finding the suitable grammar set for the parsing and the extraction of information. Specifically for scholarly context, various systems has been created to extract and retrieve information from publications and scholarly articles. exBERT [10] uses triple classification to perform knowledge graph completion. Dasigi et al. [6] proposed a method to retrieve information from papers to answer natural language questions. FNG-IE [32] is an improved graph-based approach for the extraction of keywords from scholarly big-data. Furthermore, Liu et al. [15] presents the TableSeer system that is capable of metadata extraction from tables of scholarly nature.

Language Models. With regards to automated text summarization, various language models relying on attention mechanisms [34] displayed state-of-the-art results superseding human performance. BERT [7] (scholarly counterpart SciBERT [1]) are some of the most commonly used transformer models capable to automatically summarize text. Similarly, RoBERTa [16] is an optimized approach to represent language and is capable of producing summarizations of text. BART [14] is a sequence-to-sequence model trained as a denoising auto-encoder, which improves on the pre-training phase. Zhang et al. [41] presents the PEGASUS model trained for abstractive summary generation of text. These and other models are usually built to handle “short” input sequences, e.g. 512-1K tokens. Other attempts address the issue of processing longer inputs. BigBird [40] and Longformer [2] present models that are capable of handling a much larger input, e.g. 4K-16K tokens. Other generational models aren’t created specifically for one task; instead, they are capable of performing multiple tasks depending on the input text. For instance, GPT2 [23] supports unconditional text generation. Raffel et al. [24] describe T5, a model that can perform summarization, translation, and question answering based on keywords in the input text. Some of these language models have been either pre-trained on scholarly data such as PubMed⁴ and arXiv [5] datasets, or have been fine-tuned on such data for empirical evaluation in their original publication. Our contribution in this article leverages the capabilities of automatic summarization for the objective of information extraction form scholarly documents.

3 MORTY

Scholarly text is ambiguous and information dense. We illustrate the problem by taking a look at the abstract of this article. If we want to extract a single piece of information (i.e. a property) such as the “research problem” addressed by the article, it is necessary to comprehend the text and look even behind the textual representation. In this example, the research problem is “information extraction from scholarly articles”. The method of looking up certain properties

⁴ https://www.nlm.nih.gov/databases/download/pubmed_medline.html

such as “research problem” in the text, proves insufficient because the phrase may not exist as is or is spelled differently. This can be extended by looking up synonyms for the property or by finding verbs that represent the same intent (e.g., addresses, tackles, etc.). Other times, regardless of how the property is represented, the value itself is implicit or not represented as expected, which requires more abstractive answers than extractive ones [33].

We argue that these cases barely scratch the surface of the problem. Certain properties could require values placed throughout the text, combined together, and even morphed into dissimilar wording. Others, cannot be found in the text, but are included in figures, tables, or even in citations [26, 37]. Furthermore, some properties could be of annotation-nature, i.e. the property and the value are not in the original text, but tacit knowledge of an expert annotating an article.

MORTY leverages the capabilities of deep learning language models to comprehend the semantics of scholarly text and perform targeted information extraction via text summarization. Scholarly articles typically follow a certain structure. IMRaD [30] refers to Introduction, Methods, Results, Discussion. The concept has been applied to abstracts for a high-level overview of the four essential aspects of the work. Structured abstracts [18] follows the IMRaD principles by including the same points in the abstract. This motivated us to incorporate structure into automatic textual summaries, which can be easily parsed for the sake of information extraction (a.k.a. structured summary).

Figure 1 depicts a high-level view of the MORTY approach to information extraction on scholarly articles comprising several workflow phases. It starts with pre-processing of the article text (i.e., the conversion from traditional PDF into text as well as cleaning and removing some needless segments of the text). A summarization model is then capable of rendering a large text snippet into a much shorter structured summarization that contains pairs of properties and their corresponding values. Later stages take care of parsing the produced summary via finding pre-defined syntactical patterns in the produced text. Then interlinking extracted values to knowledge graph entities via exact lookup functionalities. Lastly the newly extracted and aligned data gets added to a destination knowledge graph. The fundamental component of the approach is the summarization module due to the fact that all other components of MORTY are self-consistent. This article tackles the following research question (**RQ**): *How can we leverage structured summaries for the task of scholarly information extraction?*

4 Evaluation

Since the main component of MORTY is the structured summary generation, **we focus our evaluation on that component solely**. Other components of the approach are deterministic in behavior and can be disregarded for the sake of this evaluation. We created a dataset using the ORKG infrastructure,

and empirically evaluated the feasibility of the summarization task with various models and approaches based on this dataset.

4.1 Dataset Collection

We require a source for human-curated annotations of scholarly articles. ORKG is a knowledge graph that contains this sort of information. Hence, we leverage the ORKG to create a dataset of scholarly articles’ texts with a set of property-value pairs. First, we took a snapshot of the ORKG data⁵ and we filtered on papers that are open access or have pre-prints on arXiv. This ensures restriction-free access to the PDF files of articles. Second, we parse the PDF files using GROBID [17] into text. Furthermore, we employ a heuristic to clean the text. The heuristic involves the following steps: i) Remove a set of pre-defined sections (such as abstract, related work, background, acknowledgments, and references); ii) Remove all URLs from the text, as well as all Unicode characters; iii) Remove tables, figures, footnotes, and citation texts. Lastly, we collect all annotations from the ORKG excluding some properties that contain values of URIs and other structural properties⁶. Afterwards that data was collected in a format that the summarization model is trained on and can process. We split the data in 80-10-10 training-validation-testing split in favor of the testing set.

Table 1. Overview of models used in the evaluation, categorized per task. With the number of parameters, the max input size they can handle, and what dataset they are fine-tuned on beforehand to our training.

Model	# of Params	Input Size	Finetuned on
Summarization			
ProphetNet -large [38]	391M	2K	CNN
BART -large [14]	460M	4K	CNN
GPT2 -large [23]	774M	2K	-
Pegasus -large [41]	568M	2K	Pubmed
BigBird -large [40]	576M	4K	Pubmed
T5 -large [24]	770M	4K	-
Longformer -large [2]	459M	8K	Pubmed
Question Answering			
BERT -large [7]	335M	1K	SQuAD2
Longformer -large [2]	459M	8K	SQuAD2
Named Entity Recognition			
BERT -large [7]	335M	1K	CoNLL
RoBERTa -large [16]	355M	2K	CoNLL

⁵ Data snapshot was taken on 02.02.2022.

⁶ Properties that are used solely for information organization and have no semantic value.

4.2 Baselines

Throughout the evaluation, multiple baselines were investigated (see Table 1). Various language models were used that are capable and pre-trained on summarization tasks. The maximum input size for each model varies depending on its architecture. In our created dataset, the average entry contained around 5K tokens, with a maximum around 9K and a minimum around 1.5K. Some of the models we used (e.g. Pegasus) are capable of abstractive summarization, i.e. can create summaries with words that don't exist in the original input text. This is important when annotated properties and values are not present in the text, but are formulated differently.

Furthermore, the feasibility of the task is evaluated using two other categories of NLP tasks. Extractive question answering (similarly to [39]) language models are leveraged to try to extract values for certain questions. The questions are formulated as follows: "what is the {property-label}?" . This type of baselines is inherently flawed because some of the properties and values from the datasets are not as is in the input text. Another method for evaluation is to perform named entity recognition by recognizing the individual values as entities of interest and then classifying them into one of the classes (properties).

4.3 Evaluation Results and Discussion

The evaluation took place on a machine with 2 GPUs RTX A6000 each with a 48GB vRAM. Training scripts were adapted from the fine-tuning scripts of each of model's code repositories with the help of the Transformers [36] library. The training used a batch size = 2 and epochs = 20 with early stopping enabled.

First, we evaluate the performance of various language models for structured summarization. Table 2 shows the results of the Rouge F1 metric (following [41]) for all considered language models. Second, we evaluate the feasibility of the task using techniques other than summarization, namely extractive question answering and using named entity recognition. Though the tasks of summarizations, named entity recognition, and question answering are not directly comparable; we include this analysis to show the validity of summarization as a candidate for targeted IE tasks compared to other approaches. Table 3 describes the performance of the two different approaches using two models for each case. Each model has different maximum input size and for the QA task the models were previously tuned on the SQuAD2 [25] and the CoNLL [27] datasets for the NER task. For the QA metrics, the reported number are computed @1, meaning only candidate results at the first place.

The results show that the task is viable using summarization and that structured summaries are able to extract the required information out of the scholarly articles. When considering the normal summarization task, i.e. summarizing text into a coherent shorter text snippet, the top model [21] at the time of writing this article are performing with 51.05, 23.26, 46.47 for Rouge 1-2-L respectively⁷.

⁷ <https://paperswithcode.com/sota/text-summarization-on-pubmed-1>

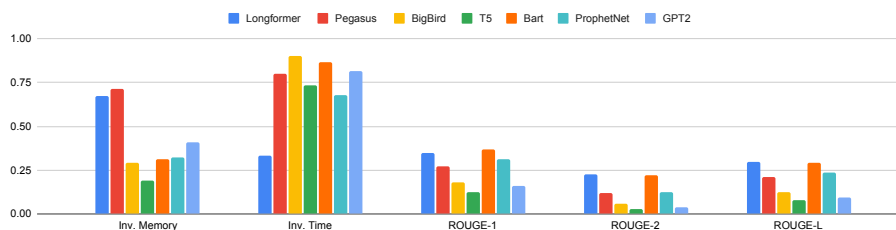


Fig. 2. Summarization metrics overview of used models including the inverse time needed for training, and inverse memory consumption. Time and memory values are normalized and inversed. Higher values are better.

Table 2. Rouge F1 scores for 1-gram, 2-grams, and longest-gram variations of the summarization models. Top best results are indicated in bold, second best in italic.

	Rouge-1	Rouge-2	Rouge-L
ProphetNet	31.1	12.5	23.7
BART	36.7	<i>22.0</i>	<i>29.4</i>
GPT2	16.1	3.6	9.3
Pegasus	27.1	11.7	21.2
BigBird	17.9	5.9	12.5
T5	12.2	2.8	7.9
Longformer	<i>34.7</i>	22.4	29.6

Table 3. Precision, recall, f1-score results of other baseline models on the question answering (QA) and named entity recognition (NER) tasks.

	Precision	Recall	F1-score
Question Answering			
BERT	20.8	18.1	19.1
Longformer	23.7	22.8	23.2
Named Entity Recognition			
BERT	17.2	17.0	17.0
RoBERTa	19.7	19.5	19.6

This kind of summarization is far easier than structured summarization since the aim is merely coherent text creation, not structured summary of text fragments. Examining Table 2 and Table 1, we note that input size affects the performance of the model. The summarization model requires the processing of the complete input article text to extract values from it, and if the model can not handle the full article then it will suffer in performance metrics. We note that, BART and Longformer summarization models performed best across all metrics.

ProphetNet and Pegasus performed well compared to other models, but they were not able to beat Longformer in part due to limited maximum input size. GPT2 model suffered due to its nature as a generative model. It was not able to generate the structured summary rather generating more coherent text. Surprisingly, although big models with a 4K max input size BigBird and T5 did not perform comparatively to top models in the list.

Figure 2 depicts an overview of the Rouge metrics of the summarization models as well as time and space requirement of each. Though Longformer is the best performing model on average, it requires more time to train compared to BART. On the other hand, BART requires almost twice the memory compared to Longformer.

Table 4. Examples: Expected vs. model predicted values.

Property	Expected	MORTY prediction
Preprocessing steps	Topic segmentation	Anaphora resolution
	Anaphora resolution	
	Pronoun resolution	
Data size	139 meetings	20 meetings
Summarization type	Abstractive	Abstractive
Evaluation metrics	ROUGE-2	F1
	ROUGE-SU4	
Study location	Singapore	The City of Singapore

In order to empirically judge if the summarization method is suitable for the task of information extraction, we evaluate the approach against two categories of tasks: Question Answering (QA) and Named Entity Recognition (NER). Table 3 shows the precision, recall, and f1-score metrics for two models in each category. Due to the nature of the training data and the task itself, these two categories are inherently flawed because they are extractive and not abstractive, meaning that they aim at finding values from within the text, rather than compute with novel values. Thus, these tasks are only able to retrieve parts of the values that are in the text and the rest are unattainable to them. This explains why different models in both tasks preform poorly.

Table 4 shows some examples of five properties from three different articles with the expected values and the predicted values by the summarization model (here Longformer). We observe that the model is able to extract partial values or similar values but with different wordings, as well as exact values, and completely different values. For instance, “Data size” is an annotation property, were the expected value is not in the text, rather it is a summation of other values. “Preprocessing steps” property aggregate values from multiple places in the text. The remarks made in this section answers our research question.

5 Conclusion and Future Directions

The objective of this work was to leverage structured summarization for the task of IE from scholarly articles. We evaluated various models on the summarization task, as well as compared against models performing question answering and named entity recognition. The results show that summarization is a viable and feasible approach for the IE task on scholarly articles. Based on our observations, we suggest the following open points in this domain: i) Enable longer input sizes for large language models and evaluate them. ii) Experiment with various structured summary formats and study their effect. iii) Incorporate active learning with user feedback collected from a user interface within a scholarly infrastructure. iv) Perform a user evaluation to study the efficacy of the IE task on scholarly data for users.

References

1. Beltagy, I., Lo, K., Cohan, A.: SciBERT: A pretrained language model for scientific text. 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). pp. 3615–3620. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1371>, <https://aclanthology.org/D19-1371>
2. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
3. Chang, C.H., Kaye, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering* **18**(10), 1411–1428 (2006)
4. Chua, F.C., Duffy, N.P.: Deepcpcfg: Deep learning and context free grammars for end-to-end information extraction. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 838–853. Springer International Publishing, Cham (2021)
5. Clement, C.B., Bierbaum, M., O’Keeffe, K.P., Alemi, A.A.: On the use of arxiv as a dataset (2019)
6. Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N.A., Gardner, M.: A dataset of information-seeking questions and answers anchored in research papers. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4599–4610. Association for Computational Linguistics, Online (Jun 2021). <https://doi.org/10.18653/v1/2021.naacl-main.365>, <https://aclanthology.org/2021.naacl-main.365>
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. *Communications of the ACM* **51**(12), 68–74 (2008)
9. Jaradeh, M.Y., Oelen, A., Farfar, K.E., Prinz, M., D’Souza, J., Kismihók, G., Stocker, M., Auer, S.: Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture. pp. 243–246 (2019)
10. Jaradeh, M.Y., Singh, K., Stocker, M., Auer, S.: Triple classification for scholarly knowledge graph completion. In: Proceedings of the 11th on Knowledge Capture Conference. p. 225–232. K-CAP ’21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3460210.3493582>, <https://doi.org/10.1145/3460210.3493582>
11. Jeschke, J., Enders, M., Bagni, M., Aumann, D., Jeschke, P., Zimmermann, M., Heger, T.: Hi-knowledge, version 2.0. <https://hi-knowledge.org/> (2020), (Accessed on 05/23/2022)
12. Ji, D., Tao, P., Fei, H., Ren, Y.: An end-to-end joint model for evidence information extraction from court record document. *Information Processing & Management* **57**(6), 102305 (2020). <https://doi.org/https://doi.org/10.1016/j.ipm.2020.102305>
13. Jiang, J.: Information extraction from text. In: Mining text data, pp. 11–41. Springer (2012)
14. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training

- for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
15. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Tableseer: Automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. p. 91–100. JCDL '07, Association for Computing Machinery, New York, NY, USA (2007). <https://doi.org/10.1145/1255175.1255193>, <https://doi.org/10.1145/1255175.1255193>
 16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
 17. Lopez, P.: Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In: International conference on theory and practice of digital libraries. pp. 473–474. Springer (2009)
 18. Nakayama, T., Hirai, N., Yamazaki, S., Naito, M.: Adoption of structured abstracts by general medical journals and format for a structured abstract. *J Med Libr Assoc* **93**(2), 237–242 (04 2005)
 19. Nasar, Z., Jaffry, S.W., Malik, M.K.: Information extraction from scientific articles: a survey. *Scientometrics* **117**(3), 1931–1990 (2018)
 20. Palmatier, R.W., Houston, M.B., Hulland, J.: Review articles: purpose, process, and structure. *Journal of the Academy of Marketing Science* **46**(1), 1–5 (Jan 2018). <https://doi.org/10.1007/s11747-017-0563-4>, <https://doi.org/10.1007/s11747-017-0563-4>
 21. Pang, B., Nijkamp, E., Kryściński, W., Savarese, S., Zhou, Y., Xiong, C.: Long document summarization with top-down and bottom-up inference. arXiv preprint arXiv:2203.07586 (2022)
 22. Piskorski, J., Yangarber, R.: Information extraction: Past, present and future. In: Multi-source, multilingual information extraction and summarization, pp. 23–49. Springer (2013)
 23. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
 24. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
 25. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822 (2018)
 26. Ray Choudhury, S., Mitra, P., Giles, C.L.: Automatic extraction of figures from scholarly documents. In: Proceedings of the 2015 ACM Symposium on Document Engineering. pp. 47–50 (2015)
 27. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
 28. Sarawagi, S.: Information extraction. Now Publishers Inc (2008)
 29. Singh, M., Barua, B., Palod, P., Garg, M., Satapathy, S., Bushi, S., Ayush, K., Rohith, K.S., Gamidi, T., Goyal, P., Mukherjee, A.: OCR++: A robust framework for information extraction from scholarly articles. *CoRR* **abs/1609.06423** (2016), <http://arxiv.org/abs/1609.06423>
 30. Sollaci, L.B., Pereira, M.G.: The introduction, methods, results, and discussion (imrad) structure: a fifty-year survey. *Journal of the Medical Library Association : JMLA* **92**(3), 364–367 (Jul 2004), [https://pubmed.ncbi.nlm.nih.gov/15243643/15243643\[pmid\]](https://pubmed.ncbi.nlm.nih.gov/15243643/15243643[pmid])

31. Spadaro, G., Tiddi, I., Columbus, S., Jin, S., Teije, A.t., Balliet, D.: The cooperation databank: Machine-readable science accelerates research synthesis (Oct 2020). <https://doi.org/10.31234/osf.io/rveh3>, psyarxiv.com/rveh3
32. Tahir, N., Asif, M., Ahmad, S., Malik, M.S.A., Aljuaid, H., Butt, M.A., Rehman, M.: Fng-ie: an improved graph-based method for keyword extraction from scholarly big-data. *PeerJ Computer Science* **7** (2021)
33. Tas, O., Kiyani, F.: A survey automatic text summarization. *PressAcademia Procedia* **5**(1), 205–213 (2007)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *CoRR* **abs/1706.03762** (2017), <http://arxiv.org/abs/1706.03762>
35. Williams, K., Wu, J., Wu, Z., Giles, C.L.: Information extraction for scholarly digital libraries. In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. pp. 287–288 (2016)
36. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
37. Xia, F., Wang, W., Bekele, T.M., Liu, H.: Big scholarly data: A survey. *IEEE Transactions on Big Data* **3**(1), 18–35 (2017). <https://doi.org/10.1109/TBDATA.2016.2641460>
38. Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., Zhou, M.: Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063* (2020)
39. Yao, X., Van Durme, B.: Information extraction over structured data: Question answering with freebase. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 956–966 (2014)
40. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems* **33**, 17283–17297 (2020)
41. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: *International Conference on Machine Learning*. pp. 11328–11339. PMLR (2020)
42. Zhang, P., Xu, Y., Cheng, Z., Pu, S., Lu, J., Qiao, L., Niu, Y., Wu, F.: TRIE: End-to-End Text Reading and Information Extraction for Document Understanding, p. 1413–1422. Association for Computing Machinery, New York, NY, USA (2020), <https://doi.org/10.1145/3394171.3413900>