Grenze International Journal of Engineering and Technology, Jan Issue

Auto Survey of Research Papers

Darsh Mehta[1], Shivangi Kochrekar[2], Neha Kale[3] and Sunil Ghane[4]
[1-4]Department of Computer Engineering, Sardar Patel Institute of Technology Mumbai, India
Email: darsh.mehta@spit.ac.in, shivangi.kochrekar@spit.ac.in, neha.kale@spit.ac.in, sunil_ghane@spit.ac.in

***Abstract*—Researching takes a lot of time and effort, especially for novices. Our system proposes a way to make research easier by first showing various trends in that particular field, shortlisting the papers based on multiple filters, and then providing a coherent summary of those shortlisted papers. We have used Elastic Search and NLP for finding trends in form of bigrams and trigrams. The current research does not provide rational summaries. We aim to use modern summarizing techniques such as Summarizer from BERTSUM, and Long Encoder Decoder(LED) to provide coherent summaries as well as use SciSpacy library which helps to identify scientific terms easily. In the end, the user will not only get the top papers in the field but also a gist of each of those papers without breaking a sweat, thus saving time and effort.**

***Index Terms*— Abstractive Summarization, Dask, Extractive Summarization, Long Encoder Decoder, Summarizer.**

I. INTRODUCTION

A researcher's job is not easy. Continuously searching, reading, and writing research papers is an essential part of the research. It is especially difficult for individuals venturing into research with little or no experience with past publications who can be referred to as novice researchers. The sheer amount of existing research can be overwhelming for novice researchers. Oftentimes, writing a paper to explain one's work takes up more time than the actual research.

Being new to research, the latest innovations are not known to most people. Furthermore, the leading researchers and their work are also out of reach. Trying to find papers relevant to one's research is a very exhausting task. Even after putting in long hours to understand extensive and long research papers, their meaning might evade researchers.

Novices generally need to spend a large portion of time learning the skills required to build anything noteworthy. Furthermore, practical implementation consumes a lot of time. In this scenario, carving out time for profound research and writing a good quality research paper becomes next to impossible.

It can be difficult to deal with the quantity of literature that one might have accessed. The literature review is iterative. This involves managing the literature, accessing data that supports the framework of the research, identifying keywords and alternative keywords, as well as constantly looking for new sources.

A literature review must go beyond being a series of references and citations. You need to interpret the literature and be able to position it within the context of your study. This requires careful and measured interpretation and writing in which you synthesize and bring together the materials that you have read.

Library management and functioning are not satisfactory in many universities. A lot of time and energy is spent on tracing appropriate books, journals, reports, etc. Also, many of the libraries are not able to get copies of new reports and other publications on time.

Grenze ID: 01.GIJET.9.1.501
© Grenze Scientific Society, 2023

We have implemented a system that aids the research pro- cess and saves people's time and effort by first showing various trends in that particular field, shortlisting the papers based on multiple filters, and then providing a coherent summary of those shortlisted papers.
The contribution of this project is threefold: Identifying current research topics, leading researchers and their publica- tions in the user's area of interest. Shortlisting several papers based on various filters. Summarizing a set number of papers into one research paper.

II. Literature Survey

Sefid and Giles [1] propose a BERT-based document encoder. Their encoder model stacks multiple transformer layers on sentence vectors to model the inter-sentence relations in the document. This will help build sentence representations. Their SciBERTSUM model, an extension of BERTSUM can generate sentence embeddings for sentences in a document containing multiple sections. To represent inter-sentence relations, it applies a linear sparse attention mechanism between sentences.
Ernst et. al. [2] extended clustering-based summarization to apply at the more fine-grained propositional level. First, they extracted all propositions from the input set of documents and filtered out non-salient propositions with a salience-detection model. Then, all salient propositions are clustered into groups based on their semantic similarity. The largest clusters, i.e., those containing more redundant information across the documents, are selected to participate in the summary. Finally, each cluster is fused to form a sentence for the abstractive summary.
Du and Huo [3] introduce a novel model for news sum- marization based on multi feature, fuzzy logic, and genetic algorithm. More precise summaries may be generated by identifying the important aspects of the news text and fusing the fuzzy logic system with the genetic algorithm to give the text features more optimal weights. Their model's performance was discovered to be noticeably better than that of other approaches.
Sotudeh et. al. [4] produce long summaries of scientific papers using three main approaches: Experimenting with var- ious versions of pretrained transformer encoders fine-tuned for the summarization task as an extractive method, a novel multitasking learning model which jointly incorporates the dis- course structure of documents into the extractive summarizer and a two-stage method where an abstractive summarizer is attached to the extractive summarizer to generate abstractive summaries.
Zhang et. al. [5] propose a transformer sentence encoder for document encoding called HIBERT which is short for Hi- erachical Bidirectional Encoder Representations from Trans- formers and a technique to pretrain it uses unlabeled data. They first use unlabeled data to train the hierarchical encoder of the extractive model. Then the model initialized from the pre-trained encoder learns to classify sentences.
Madhuri and Kumar [6] propose a statistical approach to execute extractive text summarization on a document. Input is given in the form of a text file. Preprocessing of the file is done by firstly tokenizing the files and then stop words are removed from the document. After that, each token is assigned a figure of speech like a noun, pronoun, adjective, and so on. Then each token is assigned a weight based on the number of times it occurs in a document. Individual terms are ranked based on the weighted frequency which is calculated as a fraction of the frequency of an individual term by the frequency of the term with maximum frequency. The summary of the document is formed by using the sentences having the highest weight frequency.
Erera et al. [7] describe a novel technique for delivering summaries for Computer Science papers. They determined the most valuable scenarios for scientific document discovery, and exploration and developed a model that retrieves and sum- marizes scientific documents to satisfy a given information need, either via a query or by selecting categorized values such as scientific datasets, tasks, and more. Their system ingested 270,000 documents. The summary length of each section was set at 10 phrases for the section-based summary. The section- agnostic summary was determined to be of the same length as the section-based summary. 24 papers and 48 summaries were examined in total.
Shirwandkar and Kulkarni [8] propose a unique approach to summarizing text documents by using a combination of Fuzzy Logic and Restricted Boltzmann Machine (RBM). The RBM method had an F1 score of 0.79 while the proposed methodology had a score of 0.84.
Chen et. al. [9] propose the essence vector (EV) model which is a novel unsupervised paragraph embedding method. It aims not only to distill the paragraph's most representative information but also to exclude the general background infor- mation. Secondly, keeping in mind the increasing relevance of spoken content processing, the denoising essence vector model (D-EV) is also proposed. The D-EV model inherits the benefits of the EV model and infers a more robust rendition of a given spoken passage against flawed speech

recognition. Finally, a novel summarization system, that considers both the relevance and the redundancy information simultaneously, is also put forward.

According to the approach of Sethi et. al. [10], nouns play a significant part in the lexical chain, but adjectives should also be taken into consideration. The number of papers that it was tested on is not specified, nor are the findings.

Yasunaga et al. [11] propose a neural multi-document summarization (MDS) system with sentence relation graphs. A Graph Convolutional Network (GCN) with sentence em-beddings obtained from Recurrent Neural Networks as input node features are used on the relation graphs. Using multiple layer-wise propagations, the GCN generates high-level hidden sentence features for salience estimation. Then, while avoiding redundancy, a greedy heuristic to extract important sentences is employed. Three types of sentence relation graphs are considered in their DUC 2004 experiments. Furthermore, the advantage of combining sentence relations in graphs with the representation power of deep neural networks is demonstrated.

Ramanujam and Kaliappan [12] propose a new method for multidocument text summarizing that combines a timestamp technique with a Naive Bayesian Classification approach. The timestamp gives the summary an orderly appearance, resulting in a summary that appears to be coherent. It extracts the most relevant data from abundant documents. A scoring strategy is used to calculate the score for each word to calculate the word frequency. In terms of readability and comprehensibility, the higher the linguistic quality, the better. To demonstrate its efficacy, a comparison with the existing MEAD algorithm is done. It is used to investigate the outcomes of the MEAD algorithm's timestamp procedure.

Saleh et. al. [13] propose a genetic algorithm-based extrac- tive multi-document text summarization model (GA). First, it is treated as a discrete optimization problem, and a specific fitness function is created to give better results with the model. Then, to characterize the adopted GA, representation in binary form is proposed, along with a heuristic mutation and a local repair operator. Experiments are conducted on ten topics from the DUC2002 datasets (Document Understanding Conference)(d061j through d070f).

Xu et. al. [14] used paragraphs as units, and model sentences in paragraphs are considered. A hierarchical model is trained to learn structure among sentences and is used to select important and informative sentences to generate a survey. The model works well but the paragraph length is set to 5 for the model to work out best. Invalid sentences are also added to paragraphs with lengths less than 5.

Portenoy et. al [15] proposed a framework to use the reference lists from existing review papers as labeled data to train supervised classifiers. It uses both citation and text-based features on 654 review papers. They assumed that every article in a review paper's bibliography is a relevant article to be included in a field's survey; in practice, an article can be cited for many different reasons. There are different types of review articles, but their method ignores potential nuances between them.

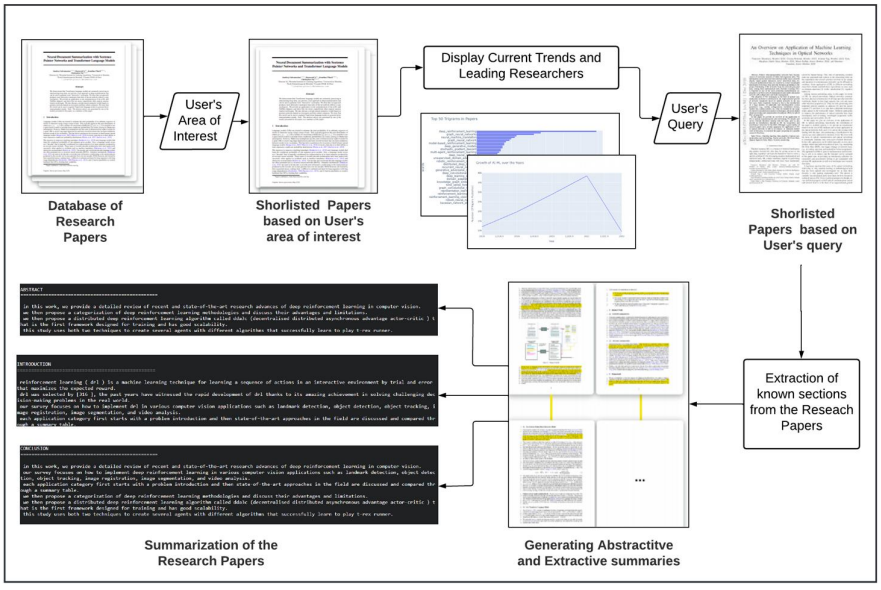

Fig. 1. System Diagram for Research Paper Summarization

We observed that almost most of the current work results in models which do not provide coherent summaries or cannot be used on long documents and multiple research papers. The system we proposed will generate an automated summary of all the shortlisted papers and even section-wise summaries of each shortlisted paper the working of which is explained inthe next section.

III. Proposed Methodology

In this section, we describe our proposed system, the salient features, and the procedure we employed to create the system. From our study of the related work and existing technologies, we identified some limitations as mentioned in the Literature Survey. Then we devised a feasible solution that is better equipped at solving this problem.

The steps showing the working of our system are repre- sented diagrammatically in Fig. 1 and are described as follows:

A. Assembling Data

For nearly 30 years, ArXiv has served the public and research communities by providing open access to scholarly articles, from the vast branches of physics to the many subdisciplines of computer science to everything in between, including math, statistics, electrical engineering, quantitative biology, and economics. This rich corpus of information offers significant, but sometimes overwhelming depth. In these times of unique global challenges, efficient extraction of insights from data is essential. To help make the arXiv more accessible, a free, open pipeline on Kaggle to the machine-readable arXiv dataset: a repository of 1.7 million articles, with relevant features such as article titles, authors, categories, abstracts, full-text PDFs, and more. The content of a single JSON file is shown in Fig. 2

B. Data Analysis and Visualization

In the first half of the project, we help novice researcher to explore as much as they want. All the user needs to do is select a particular category, and apply filters like date, authors and so on and the user can easily see the trends in various categories which will help them to make informed decisions as to which area they want to explore in-depth. Since more than 1.7M articles are present in the metadata, we needed an efficient method of loading the data into our database and for that purpose, we used the Dask library in python. Dask is used for parallel computing in python and with the Dask Bag, we can load our data with a small memory footprint using Python iterators. There are over 100+ categories to choose from and every time loading the entire data might not be space and time efficient. So, we take the field of interest of the user at the very start so that we can only load those particular documents that have the same category, for example, if the user wants to learn about

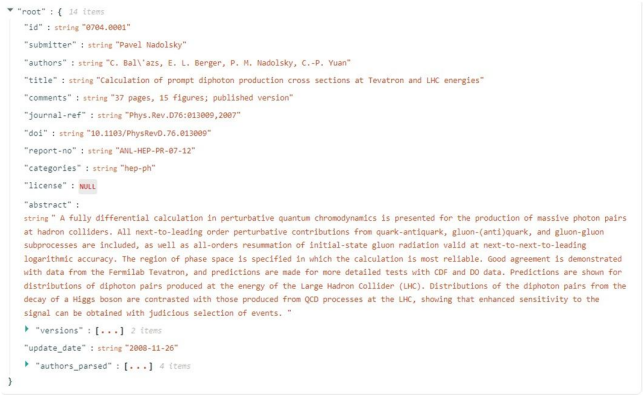

Fig. 2. Structure of Data in a JSON file

Machine Learning and Statistics, the user will select only those two categories. Next, we try to extract the year of the paper from the version attribute using a pandas function called to datetime(). The user has the option to see the trends after a particular year only as it is quite possible that considering papers of 10 years ago might not give a clear idea about the scope of the topic at present. Then we do some preprocessing, dropping some irrelevant columns and checking for any null values in the data. The first graph we display shows a simple

trend in the user's input category over the years as shown in Fig. 3. This gives the user a rough idea to the user if the studies on that particular topic have beenon a constant rise or have been stagnant over a period of time.
Next, we show the top authors in that category togive the user an option of following the author's work and try to gather additional resources provided by the author to gaina better understanding of the topic. The results are shown in Fig. 4.
Now, the most important graph shows the most researched topics in a particular field. For example, in recent years the study of Artificial Intelligence has been on a constant rise, but it is a very broad field, so a novice might not know which topicis being most researched. So we have tried to create bigrams and trigrams of the most researched topics based on the work of popular authors in that particular field and by studying the frequency of those topics in recent publications as shown in Fig. 5 and Fig. 6. This will definitely help the user to zero in on a particular topic to study in-depth.

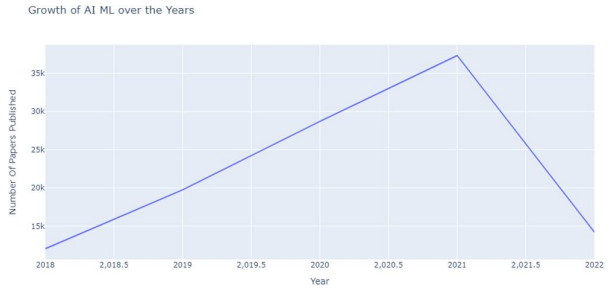

Fig. 3. Current trend for the category chosen by the user

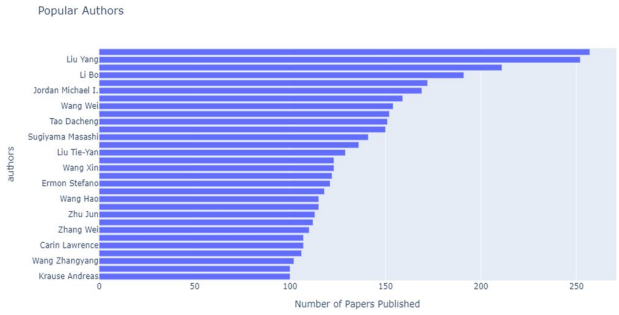

Fig. 4. Top Authors for the category chosen by the user

C. Generating an Automated Survey

After the user decides to explore a particular topic, there willbe a plethora of existing work already which might overwhelmthe user. Hence our aim is to create a system that will shortlist the most influential papers in that field and provide a summary of those papers so that the user does not need to go through all the papers hence saving time and energy. As input from the user, we take the query or the topic the user wantsto know about, the maximum number of papers that the user wants to be shortlisted, and if the user wants to give relevance to the author.
The shortlisting of papers takes place as follows:

- We use the search engine of Arxiv itself which usesElastic Search to find the shortlisted papers.
- In case too many papers exist for the particular query, we shortlist the number of papers equal to the maximum number of papers using the Relevance criterion defined by Arxiv itself.
- In case, the user wants to give relevance to the authors, the user needs to specify that in the input itself stating weight_ authors=True. In such a scenario we use the Google Scholarly API, to find the most influential authorsin that domain. The shortlisting of the author takes place on the expertise of the author in that domain and the h1 score which is calculated by the number of citations the author has received.
- Combining that with the search results from Arxiv's sear ch engine, we shortlist the papers.

Next, we define the standardized headings such as Abstract, Introduction, Conclusion, and References and anything other than that is assumed to be included in the Proposed Methodol-ogy. We now extract each of these sections from the shortlistedpapers to summarize them to give a gist to it to the user.

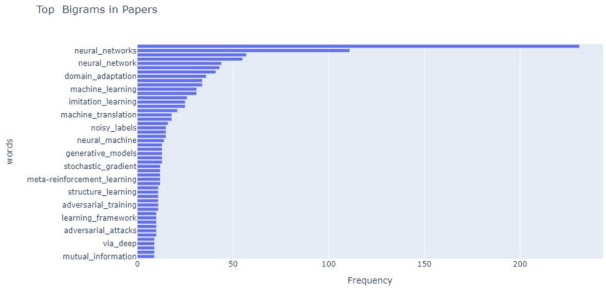

Fig. 5. Top Bigrams for the category chosen by the user

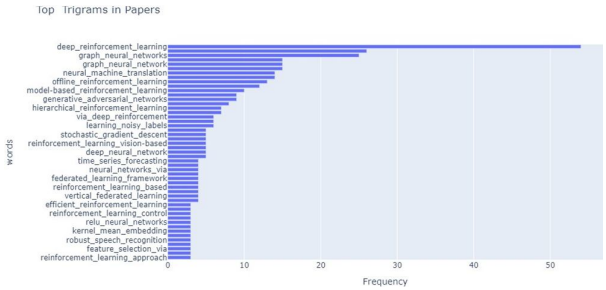

Fig. 6 Top Trigrams for the category chosen by the user

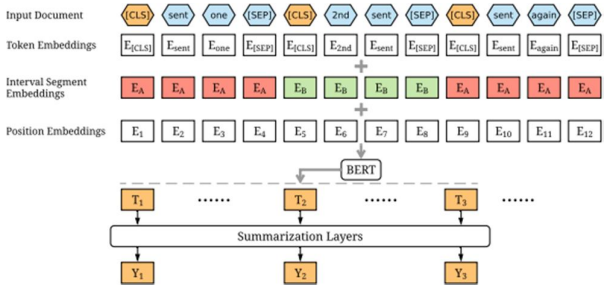

Fig. 7 Architechture of BERTSUM Model

After extracting various sections from the paper, we now preprocess the text extracted. For preprocessing, we follow the standard pipeline of cleaning the text like converting them into lowercase, removing stop words, punctuations, and any other special characters, and then tokenizing them to pass it furtherfor summarizing it. For summarizing we have used a combination of Abstractiveand Extractive Summaries to get better results. An extractive summary is the one in which we take the sentences as it is from the corpus and concatenate them to form a summary. Onthe other hand, Abstractive Summary more or less rewrites thewhole document by building internal semantic representation and then generates a summary using novel sentences using NLP techniques. For Extractive Summaries, we used the Sum-marizer model, an inbuilt module provided by BERTSUM.

```
found 10 papers
Searching Author: Vidhiwar Singh Rathour
Searching Author: Masahiko Ueda
Searching Author: Ngan Le
Searching Author: Khoa Luu
Searching Author: Marios Savvides
Searching Author: Xiao-Jun Zeng
Searching Author: Kashu Yamazaki
Searching Author: Abulikemu Abuduweili
Searching Author: Changjian Li
Searching Author: Kaiyue Wu
Searching Author: Changliu Liu
downloading pdfs..
downloading below selected papers:
['2001.09608', '2108.11510', '2202.05135', '2108.03258', '2203.12114']
```

Fig. 8. Shortlisted Papers on the topic chosen by the user

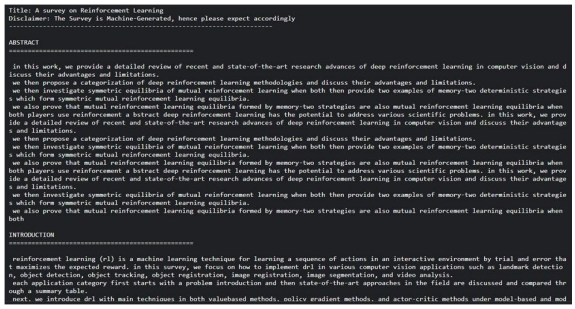

Fig. 9. Summary of all the Shortlisted Papers Section Wise – 1

BERTSUM is a summarization model which uses BERT asan encoder. The summarization model of BERT which is BERTSUM differs from the original model as a [CLS] tokenis added at the start of every sentence to separate multiple sentences and [CLS] token collects features for sentences preceding it. There is also a slight difference in embeddingas each sentence is assigned as either Ea or Eb depending onif it's an even or odd-numbered sentence. The architecture is shown in Fig. 7. The tokenizer and the model that we used is of SciBert, a BERT model trained on scientific text. It is trained on papers from the corpus of semanticsscholar.org. We have used the uncased version of it.

For Abstractive Summaries, we have used Longformer Encoder-Decoder. The drawback of traditional transformers was that it was unable to process long sentences due to their self-attention operation which increases quadratically with sequence length. Longformer has an attention mechanism that scales linearly with sequence length, which makes it easier to process longer tokens, LEDForConditionalGeneration is a BartForConditionalGeneration extension that replaces the traditional self-attention layer with Longformer's chunked self-attention layer. It works well on long-range sequence-to-sequence tasks where input exceeds 1024 tokens. We have used the official led-large-16384 checkpoint that is fine-tuned on the arXiv dataset.led-large-16384-arxiv is the official fine- tuned version of led-large-16384.

For the final summary, we combined both the summaries by first finding a similarity between the two summaries and then combining sentences having similarities above a certain threshold and passing them through the SciSpacy library so that we can extract scientific terms more accurately and present a more cogent summary of all the papers extracted. The summary generated by our system for the category of 'Machine Learning' and sub-topic 'Reinforcement Learning'is shown in Fig. 9 and Fig. 10. Fig 9 covers the summary of 'Abstract' and 'Introduction' for all the papers. A combined summary for 'Introduction' and 'Conclusion' can be seen in Fig. 10. The summary generated is cogent and concise.

Apart from that we also summarize all the papers individu- ally, so that the user doesn't have to go through all the papers and can get a gist of it on a cursory glance of the summary.

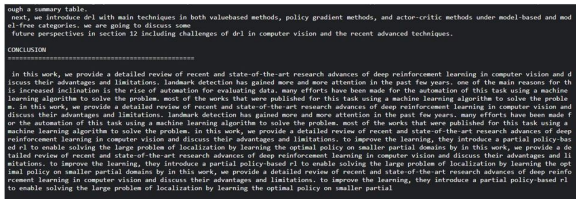

Fig. 10. Summary of all the Shortlisted Papers Section Wise – 2

Fig. 11. Paper Wise Summary of the Shortlisted Papers for first paper

IV. Results

After implementing the proposed system, we first get a list of shortlisted papers on a particular topic inputted bythe user. These papers are the most influential papers in that field. The result of this step is as shown in Fig. 8. The text from the shortlisted papers is extracted and given as inputto the trained models mentioned above for abstractive and extractive summarization and a coherent relevant summaryis obtained as an output no matter how long the inputtedpapers are. Fig. 9 and Fig. 10 show the cogent summary of all the shortlisted papers. Along with this, an individual summary for each paper is also obtained as shown in Fig. 11 and Fig. 12. The summaries generated are compared with those generated by existing software, and we see improvement inthe summaries generated by our system.

V. Conclusion

An attempt has been made to make the research process simpler for the novice researcher and at the same time make it faster and more efficient. The entire pipeline is ready, andwe have achieved more than satisfactory results with respect tothe generation of summaries of the shortlisted papers. Having such a system in place which is completely automated will definitely be useful to a larger base of users. In future we plan to make our generated summaries a bit more coherent.

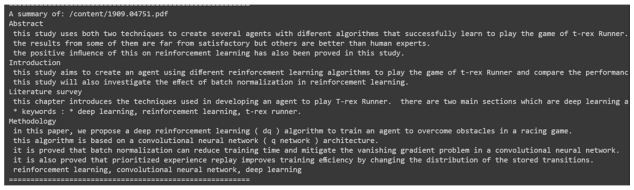

Fig. 12. Paper Wise Summary of the Shortlisted Papers for second paper

References

[1] Sefid, Athar & Giles, C. (2022). SciBERTSUM: Extractive Summariza-tion for Scientific Documents.

[2] Ori Ernst, &Avi Caciularu, &Ori Shapira, &Ramakanth Pasunuru, &Mohit Bansal, &Jacob Goldberger, &Ido Dagan(2021). A Proposition- Level Clustering Approach for Multi-Document Summarization https://doi.org/10.48550/arXiv.2112.08770

[3] Du, Y., & Huo, H. (2020). News Text Summarization Based on Multi-feature and Fuzzy Logic. IEEE Access, 1–1. doi:10.1109/access.2020.3007763

[4] Sotudeh Gharebagh, S., Cohan, A., Goharian, N.: GUIR @ Long- Summ 2020: Learning to generate long summaries from scientific documents. In: Proceedings of the First Workshop on Scholarly Doc- ument Processing. pp. 356–361. Association for Computational Lin- guistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.sdp-1.41, https://www.aclweb. org/anthology/2020.sdp-1.41

[5] Zhang, X., Wei, F., Zhou, M.: Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. arXiv preprint arXiv:1905.06566 (2019)

[6] Madhuri, J. N., & Ganesh Kumar, R. (2019). Extractive Text Summarization Using Sentence Ranking. 2019 International Conference on Data Science and Communication (IconDSC). doi:10.1109/icondsc.2019.8817040

[7] Erera, Shai & Shmueli-Scheuer, Michal & Feigenblat, Guy & Nakash, Ora & Boni, Odellia & Roitman, Haggai & Cohen, Doron & Weiner, Bar & Mass, Yosi & Rivlin, Or & Lev, Guy & Jerbi, Achiya & Herzig, Jonathan & Hou, Yufang & Jochim, Charles & Gleize, Martin & Bonin, Francesca & Konopnicki, David. (2019). A Summarization System for Scientific Documents.

[8] Shirwandkar, N. S., & Kulkarni, S. (2018). Extractive Text Summa- rization Using Deep Learning. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2018.8697465

[9] Chen, K.-Y., Liu, S.-H., Chen, B., & Wang, H.-M. (2018). An Informa- tion Distillation Framework for Extractive Summarization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(1), 161–170. doi:10.1109/taslp.2017.2764545

[10] Sethi, P., Sonawane, S., Khanwalker, S., & Keskar, R. B. (2017). Automatic text summarization of news articles. 2017 Interna- tional Conference on Big Data, IoT and Data Science (BID). doi:10.1109/bid.2017.8336568

[11] Yasunaga, Michihiro & Zhang, Rui & Meelu, Kshitijh & Pareek, Ayush & Srinivasan, Krishnan & Radev, Dragomir. (2017). Graph-based Neural Multi-Document Summarization. 452-462. 10.18653/v1/K17-1045.

[12] Nedunchelian Ramanujam, Manivannan Kaliappan, ”An Automatic Multidocument Text Summarization Approach Based on Na¨ıve Bayesian Classifier Using Timestamp Strategy”, The Scientific World Journal, vol. 2016, Article ID

1784827, 10 pages, 2016. https://doi.org/10.1155/2016/1784827
[13] Saleh, Haruna & Kadhim, Nasreen & Attea, Bara'a. (2015). A Genetic Based Optimization Model for Extractive Multi-Document Text Sum- marization. 56. 1489-1498.
[14] Xu H., Wang Z., Zhang Y., Weng X., Wang Z., & Zhou G, 'Document structure model for survey generation using neural network', Frontiersof Computer Science, 2021.
[15] Portenoy, Jason and Jevin D. West. "Supervised Learning for AutomatedLiterature Review." BIRNDL@SIGIR, 2019.