Topic-Centric Unsupervised Multi-Document Summarization of Scientific and News Articles

Amanuel Alambo
Computer Science and Engineering
Wright State University
Dayton, OH
alambo.2@wright.edu

Cori Lohstroh
ONEIL Center
Wright State University
Dayton, OH
lohstroh.2@wright.edu

Erik Madaus
ONEIL Center
Wright State University
Dayton, OH
madaus.2@wright.edu

Swati Padhee
Computer Science and Engineering
Wright State University
Dayton, OH
padhee.2@wright.edu

Brandy Foster
ONEIL Center
Wright State University
Dayton, OH
brandy.foster@wright.edu

Tanvi Banerjee
Computer Science and Engineering
Wright State University
Dayton, OH
tanvi.banerjee@wright.edu

Krishnaprasad Thirunarayan
Computer Science and Engineering
Wright State University
Dayton, OH
t.k.prasad@wright.edu

Michael Raymer
Computer Science and Engineering
Wright State University
Dayton, OH
michael.raymer@wright.edu

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378403

***Abstract*—Recent advances in natural language processing have enabled automation of a wide range of tasks, including machine translation, named entity recognition, and sentiment analysis. Automated summarization of documents, or groups of documents, however, has remained elusive, with many efforts limited to extraction of keywords, key phrases, or key sentences. Accurate abstractive summarization has yet to be achieved due to the inherent difficulty of the problem, and limited availability of training data. In this paper, we propose a topic-centric unsupervised multi-document summarization framework to generate extractive and abstractive summaries for groups of scientific articles across 20 Fields of Study (FoS) in Microsoft Academic Graph (MAG) and news articles from DUC-2004 Task 2. The proposed algorithm generates an abstractive summary by developing salient language unit selection and text generation techniques. Our approach matches the state-of-the-art when evaluated on automated extractive evaluation metrics and performs better for abstractive summarization on five human evaluation metrics (entailment, coherence, conciseness, readability, and grammar). We achieve a kappa score of 0.68 between two co-author linguists who evaluated our results. We plan to publicly share MAG-20, a human-validated gold standard dataset of topic-clustered research articles and their summaries to promote research in abstractive summarization.**

***Index Terms*—Abstraction, Language Units, Multi-document Summarization, Text Generation, Hierarchical Clustering**

This effort was sponsored in whole or in part by the Air Force Research Laboratory, USAF, under Memorandum of Understanding/Partnership Intermediary Agreement No FA8650-18-3-9325. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

I. Introduction

With the large number of articles published in the research and media community, there is an increasing demand to produce summaries that are coherent, concise, informative, and grammatical. Summarization comes in two forms: *extractive* and *abstractive*. Extractive summarization [1]–[3] is focused on extracting significant sentences [4] from source documents and has been well studied. Abstractive summarization aims at ways of fusing or paraphrasing sentences in source documents to form abstractive sentences. Due to the challenges of capturing abstractive concepts shared among sentences across source documents, and synthesizing an informative summary, there has been limited progress in abstractive summarization. While there are recent advances in unsupervised multi-document abstractive summarization [5], [6], [7], they are usually limited to forming summaries by copying words in source documents and re-arranging the words to form new sentences. These approaches identify salient phrases from sentences in source documents and fuse them to form abstractive summaries. Thus, they do not perform abstraction of sentences.

Our framework consists of two phases: an *extractive phase* and an *abstractive phase*. In the extractive phase, we follow a three-fold approach. First, we identify the core article and peripheral articles for each set of related articles. Second, we instantiate clusters using the language units of a core article and perform centroid based clustering to place language units from peripheral articles into the clusters initialized by the language units in a core article. Third, we fuse language units in a cluster using an enhanced Multi-Sentence Compression

(MSC) [8], [9] technique with a novel algorithm to maximize topical coverage and relevance of a path to the language units in the cluster. In the abstractive summarization phase 1) we employ text generation to generate abstractive language units (ALUs); and 2) we use MSC to fuse the generated ALUs into an abstractive summary. Unlike DUC-2004, where articles are topically clustered, scientific articles in MAG-20 [10] do not come topically-grouped. Therefore, for MAG-20, we use topical hierarchical agglomerative clustering (HAC) to cluster articles.

The key contributions of our study are 1) an ALU generation technique using GPT-2; 2) a novel MSC-based algorithm for selection of informative paths; and 3) a gold standard dataset of topical clusters of articles from MAG-20 and their multi-doc abstractive summaries. We use abstracts of articles from MAG for this study.

II. Related Work

Different techniques have been proposed for unsupervised abstractive summarization, including sequence to sequence models [11], [12], neural models with and without attention [5], [13], [14], abstract meaning representation (AMR) [15], [16], and centroid-based summarization [6], [7]. Our approach extends the state of the art techniques in centroid-based summarization by employing language unit identification from articles and a novel text generation technique.

Abstractive summarization has received significant attention due to progress in deep representation learning [17], [18]. [5] propose MeanSum, which consists of an autoencoder and a summarization module to produce abstractive summaries. The abstractive summarization approach of [7] called ILPSumm includes identification of informative content and clustering of similar sentences from the documents to form summaries. [6] extended the technique proposed in [7] by introducing a paraphrastic fusion model, called ParaFuse, based on context-sensitive substitution of target words. While lexical substitution enables the generation of novel words, it is limited when it comes to capturing the context in the source document. [14] propose a Pointer Generator Network to summarize news articles from the CNN/Dailymail dataset.

Further, [19] propose a framework that takes an article and a topic and generates a summary specific to the topic. However, their work is supervised and relies on the availability of human-generated training corpus to train their model.

III. Data Collection

We work with two datasets to better understand and evaluate our proposed approach: 1) the DUC-2004 benchmark dataset; and 2) scientific articles from MAG.

We queried MAG for the 100 most-cited abstracts for each of the 20 FoS published in 2016 - 2020. The 20 FoS we used are: Artificial Intelligence, Artificial Neural Network, Big Data, Case-Based Reasoning, Cybernetics, Cyberwarfare, Data Mining, Data Science, Decision Support System, Electronic Warfare, Expert System, Human-Machine Interaction, Intelligent Agent, Knowledge-Based Systems, Machine Learning, Multi-Agent System, Prediction Algorithms, Predictive Analytics, Predictive Modeling, and Sensor Fusion.

IV. Proposed Method

A. Extractive Phase

Fig. 1 shows the sequence of steps we devised for MAG-20 and DUC-2004 extractive summarization. The difference in the extractive phase of these tasks is MAG-20 has topic modeling and hierarchical agglomerative clustering in its pipeline.

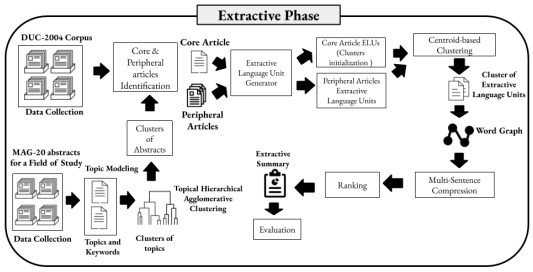

Fig. 1: Extractive Summarization. The pipeline following *Cluster of Abstracts* is for each cluster of topics.

1) Topic Modeling: For MAG-20, we determine groups of topically related abstracts for each FoS. We first build LDA [20] topic models for an FoS using number of topics in the range of 2 to 92. We then determine the optimal number of topics from an ensemble of the LDA models that maximizes the coherence score. The topics, and thus, keywords generated using the LDA model that gives the highest coherence score, are used for Topical HAC.

2) Topical Hierarchical Agglomerative Clustering: Different topics generated using an LDA model can have semantically redundant keywords. We thus cluster topics having high similarity among their keywords using HAC. We use SciBERT [21] embeddings to represent each keyword in a topic. A topic is then represented as a concatenation of the representations of its keywords. Once each topic is represented, we conduct topical HAC. To determine the number of clusters for a collection of topics, we ran HAC for several clusters ranging from 2 to the total number of topics. We use Silhouette coefficient to determine the optimal number of clusters.

We introduce a topical similarity metric (Equation-1) for measuring the similarity between a pair of topics. Each keyword in a topic is compared with all the keywords in another topic, and the sum of highest similarity scores is preserved.

$$sim(\text{Topic-i, Topic-j}) = \sum_{i \in \text{Topic-i}} maxcos(i, \text{Topic-j}) \quad (1)$$

where

$maxcos(i, \text{Topic-j}) =$ maximum of cosine similarities between term i and terms in Topic-j

Cluster	Topic IDs
0	0, 1, 6, 7, 8, 9, 14, 21
1	2, 3, 4, 5, 13, 17, 20
2	10, 11, 12, 15, 16, 18, 19

TABLE I: Topics and their cluster membership

A MAG-20 abstract is assigned to a topic that is the most dominant among all possible topics the abstract addresses. Table I shows the three clusters and their constituent topic IDs. Table II shows topical distribution among abstracts for a selected field of study. It can be seen two abstracts have the same dominant topic. These abstracts form a set of documents on which we perform multi-document summarization.

Abstract ID	Dominant Topic	Dominant Topic Contribution(%)	Topic Keywords
6	19	0.87	inspire, state, device, accelerator, small, size, high, power, ved, advantage
4	19	0.59	inspire, state, device, accelerator, small, size, high, power, ved, advantage

TABLE II: Abstracts and Dominant Topic.

3) Core and Peripheral Articles Identification: We identify the core article from a cluster of articles based on how similar an article is to other articles. Equation-2 computes the Cross-Article Similarity Score of an article. An article with the highest cumulative similarity with other articles in a cluster is chosen as the core article. The rest of the articles in the cluster are peripheral articles.

$$CAS_i = \frac{\sum_{i,j \in C} doc2vec\_sim(i,j)}{N} \tag{2}$$

where $i \neq j$
N- Number of articles in the cluster
C- The cluster of articles
$doc2vec\_sim$ - doc2vec-based cosine similarity

4) Centroid based Clustering: After core and peripheral articles are identified, we generate extractive language units (ELUs) from the core and the peripheral articles. Recent studies in centroid based summarization utilized sentences in documents as standalone ELUs to initiate clusters and to quantify semantic relatedness [6], [7]. However, this approach breaks the interdependence among sentences in a document and eventually leads to incoherent summaries. We address this issue by identifying the sentences that are interdependent using neural coreference resolution [22] and preserving them as one ELU. Once the ELUs from the core article have instantiated clusters, the ELUs from the peripheral articles are placed into a cluster based on the cosine similarity between the embedding of an ELU from the peripheral article and the embeddings of the ELUs from the core article. An ELU embedding is constructed by concatenating the embeddings of the sentences using sent-BERT [23] and performing dimensionality reduction to 300 units using T-SNE. The purpose of dimensionality reduction is to have a uniform dimension among ELUs even when they contain different number of sentences so that cosine similarity can be computed.

5) Multi-Sentence Compression: The number of clusters formed in the centroid-based clustering stage is the same as the number of ELUs in the core article. After clusters of ELUs are formed, we build word graphs [24] for each cluster. Fig. 2 shows a sample word graph constructed for a cluster consisting of the following ELUs:

ELU_1 = "Radars are required to limit emissions in adjacent bands, but traditional rectangular pulses have high out-of-band emissions."
ELU_2 = "Millimeter wave radars are popularly used in last-mile radar based defense systems."

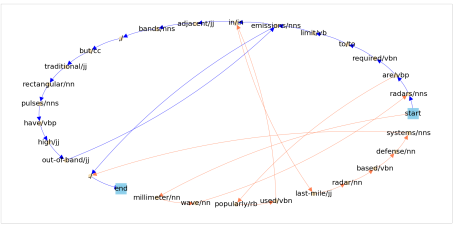

Fig. 2: Word Graph for two ELUs using NetworkX. Tokens and PoS tags of the tokens are used for a node.

We develop an algorithm for extracting paths based on topical coverage and relevance. A path is selected using an additional criterion that a candidate path should at least span two ELUs in the cluster. Next, we generate topically informative and relevant paths from the word graph while maintaining the 100-word summary limit. Topical coverage (Equation-3) measures how well a path covers the dominant topics discussed by the articles of the ELUs. Relevance (Equation-4) measures how relevant a path is to the ELUs. The cumulative score of a path (Equation-5) is determined by a weighted sum of topical coverage and relevance. We experimented with values of α in the range of 0 to 1.

a) Topical Coverage Formulation:

$$Coverage(C_{path}, C_{topics}) = \frac{1}{|C_{path}|} \sum_{i_{C_{path}} \in C_{path}}^{|C_{path}|} \frac{1}{|C_{topics}|} * \sum_{K_c \in C_{topics}}^{|C_{topics}|} maxcos(i_{C_{path}}, K_c) \tag{3}$$

where, C_{path} - Candidate path
C_{topics} - Cluster of topics

Topical coverage is measured with respect to the cluster of topics.

b) Path Relevance Formulation:

$$Relevance(C_{path}, C_{ELU}) = \frac{\vec{v}(C_{path}) \cdot \vec{v}(C_{ELU})}{|\vec{v}(C_{path})| \cdot |\vec{v}(C_{ELU})|} \tag{4}$$

where, C_{path} - Candidate Path
C_{ELU} - Cluster of ELUs
$\vec{v}(C_{path})$ - Vectorial Representation of Candidate Path
$\vec{v}(C_{ELU})$ - Vectorial Representation of Cluster of ELUs

Path relevance is measured with respect to the ELUs.
c) Cumulative Score:

$$\begin{aligned}\text{Score}(C_{path}) = \alpha \cdot \text{Coverage}(C_{path}, C_{topics}) \\ +(1-\alpha) \cdot \text{Relevance}(C_{path}, C_{ELU})\end{aligned} \tag{5}$$

A path is selected from the word graph 1) if the path is longer than the average minimum length of a sentence in an FoS or DUC-2004 topic and smaller than the average maximum length of a sentence; 2) if the combined topical coverage and relevance for the path meets or exceeds a threshold τ of 0.5. If a path picked from the word graph is semantically similar to an already selected path by an order of threshold δ of 0.8 or more, we compare the combined topical coverage and relevance of the two paths and keep the one with a higher score and remove the other. The selection of 0.8 is based on empirical observations.

B. Abstractive Phase

Fig. 3 shows the steps we followed for abstractive summarization. The difference in the abstractive phase of MAG-20 and DUC-2004 is DUC-2004 has headline generation component.

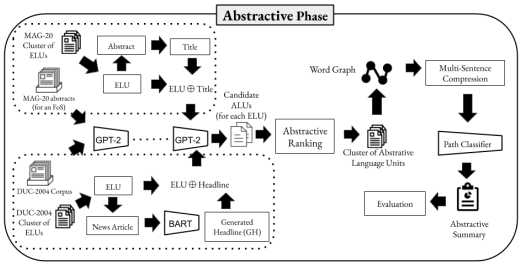

Fig. 3: Abstractive Summarization Pipeline.

1) Abstractive Language Unit (ALU) Generation: We start our abstractive phase with a pragmatic assumption that the title/headline of an article is an abstraction of the individual extractive language units (ELUs) within the same article. We propose a method to generate an ALU for an ELU using the ELU and title/headline as prompts for generating text. Combining bidirectional encodings of the title/headline with an ELU enables generating abstractive text. For ELUs consisting of two or more sentences, we encode each sentence using sentence-BERT [23], and then we concatenate these representations. Next, we perform dimensionality reduction using T-SNE to encode an ELU. For encoding a title/headline, we use sentence-BERT without dimensionality reduction. We fine-tune a GPT-2 model for an FoS (Fig. 4) and use the fine-tuned GPT-2 model to generate ALUs given a concatenation of the bidirectional encodings of the ELU and the title/headline of an article. We fine-tune a GPT-2 model such that it has

ELU	ALU
The ability to repair relationship and work together will be the key to a stable coalition.	It's a good time for a new political party that can bring stability and development.

TABLE III: ALU generated using GPT-2.

124M parameters and generates 10 candidate ALUs. While fine-tuning, we set the temperature to 0.7, number of generated samples to 10, top_k random sampling to 2 to generate more ALUs and minimize redundancy [25]. We train the GPT-2 for 10 epochs with a batch size of 10 and attain a loss of 2.16. We select an ALU that maximizes semantic similarity and minimizes syntactic similarity with the ELU used for generation. We use the normalized sum of ROUGE-1(R_1) and ROUGE-2 (R_2) for syntactic similarity. We introduce an *abstractiveness score* for an ALU, as shown in Equation-6.

We use BART [26] for headline generation for each DUC-2004 article that is later used for ALU generation along with an ELU.

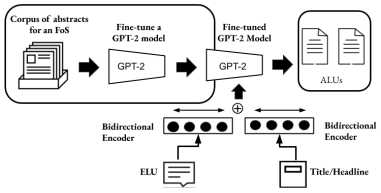

Fig. 4: ALUs generation using GPT-2.

$$\begin{aligned}\text{Score}(\text{ALU, ELU}) = \text{cossim}_{\text{xBERT}}(\text{ALU, ELU}) \\ -\frac{[R_1(\text{ALU, ELU}) + R_2(\text{ALU, ELU})}{[R_1(\text{ALU, ALU}) + R_2(\text{ALU, ALU})}\end{aligned} \tag{6}$$

where
ALU - Abstractive Language Unit
ELU - Extractive Language Unit
$\text{cossim}_{\text{xBERT}}$ - Cosine similarity on x-dimension BERT embeddings

We select an ALU that gives the highest *abstractiveness score* (Equation-6) from candidate ALUs. Table III shows a sample ELU and highest scoring ALU generated.
2) Multi-Sentence Compression: After generating ALUs for a cluster, we build a word graph and run our MSC algorithm as used in the extractive phase; i.e., the same ranking formulation and path selection algorithm is used for selecting informative paths from a word graph built, this time from a cluster of ALUs. Fig. 5 shows a cluster of ALUs and the generated fused paths that form the final abstractive summary.

V. Results and Discussion

A. Extractive Evaluation

We use ROUGE metrics for evaluating extractive summaries taking the source articles as the reference summary.

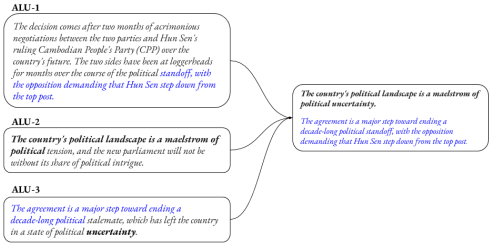

Fig. 5: Candidate ALUs and compressed ALU paths.

Model	R-1	R-2	R-L
ILPSumm	39.24	11.99	9.34
ParaFuse	**40.07**	**12.04**	**11.28**
Our Proposed Method	39.58	11.36	9.83

TABLE IV: DUC-2004 Extractive Evaluation.

DUC-2004 and MAG-20 extractive evaluation results are shown in Tables IV and V, respectively. It can be seen that our proposed method performs comparably to the baseline approaches on ROUGE-1, ROUGE-2, and ROUGE-L metrics.

B. Abstractive Evaluation

Since metrics based on lexical overlap such as ROUGE favor extractive summaries [5], we conduct abstractive summary evaluation using five human evaluation metrics we propose for this study. The metrics have been developed in consultation with two co-author Linguists. The five human evaluation metrics are: 1) Entailment; 2) Coherence; 3) Conciseness; 4) Readability; and 5) Grammar. Our co-author linguists evaluated the abstractive summaries on a scale of 1 to 5 on each of the human evaluation metrics.

Our co-author linguists independently reviewed the DUC-2004 and MAG-20 results generated using our approach, ILPSumm, and ParaFuse. When determining the rating for each criterion, they used the source articles to validate the summary. Then, they used their own compiled summaries to compare to the resulting abstractive summary. The closer the abstractive summary was to the details in their notes, the higher the Entailment. The human evaluators judged Coherence by sentence structure and whether the sentences showed logical progression. When examining Conciseness, they looked for areas of the abstractive summary that were repeated. They also noted whether a sentence carried the logical progression of the paragraph. For Readability, they did not take grammar or spacing into consideration; they looked for sentence fragments, word order, and took note of instances of missing subjects or verbs. When rating Grammar, they gave the abstractive summary a lower rating for comma splices or extra spacing than if there were fragments or inappropriate punctuation.

Model	R-1	R-2	R-L
ILPSumm	43.37	16.72	11.26
ParaFuse	46.78	**18.93**	**12.47**
Our Proposed Method	**47.43**	17.28	10.58

TABLE V: MAG-20 Extractive Evaluation.

In addition to the human evaluation metrics, we also use copy rate [6] for evaluating abstractive summaries. Copy Rate assesses the rate of novel word generation. As shown in Table VI, our framework achieves the lowest copy rate indicating that we are able to generate more novel words.

Task	Model	Copy Rate
DUC-2004	ILPSumm	0.99
	ParaFuse	0.76
	Our Approach	**0.68**
MAG-20	ILPSumm	0.96
	ParaFuse	0.88
	Our Approach	**0.72**

TABLE VI: Copy Rate Evaluation.

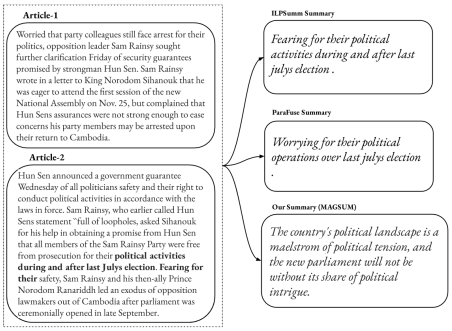

Fig. 6: Comparison of abstractive summaries.

Human Evaluator	Model	Entailment	Coherence	Conciseness	Readability	Grammar
Evaluator-I	ILPSumm	0.60	0.26	0.22	0.20	0.20
	ParaFuse	0.62	0.47	0.55	0.46	0.53
	Our Approach	**0.66**	**0.52**	**0.63**	**0.50**	**0.60**
Evaluator-II	ILPSumm	0.50	0.38	0.34	0.34	0.40
	ParaFuse	0.64	0.51	0.50	0.45	0.51
	Our Approach	**0.66**	**0.54**	**0.55**	**0.48**	**0.57**

TABLE VII: DUC-2004 Abstractive Summarization Results.

Human Evaluator	Model	Entailment	Coherence	Conciseness	Readability	Grammar
Evaluator-I	ILPSumm	**0.89**	0.63	0.71	0.53	0.38
	ParaFuse	0.82	0.64	**0.79**	0.61	0.56
	Our Approach	0.85	**0.70**	0.77	**0.65**	**0.59**
Evaluator-II	ILPSumm	**0.84**	0.71	0.70	0.65	0.47
	ParaFuse	0.83	**0.79**	0.76	0.68	0.60
	Our Approach	0.80	0.77	**0.81**	**0.70**	**0.67**

TABLE VIII: MAG-20 Abstractive Summarization Results.

Experimental results show that our proposed approach performs significantly better in human abstractive evaluation metrics and copy rate. This is mainly due to the ALU generation using a fine-tuned GPT-2 model and minimizing the syntactic similarity (Equation-6) of generated ALUs.

1) DUC-2004 Abstractive Evaluation: For DUC-2004, our proposed approach consistently performs better than ILPSumm or ParaFuse on the 5 human evaluation criteria. ILPSumm and ParaFuse show better results in entailment. In contrast, our approach generally performs comparably across the 5 criteria. Thus, we can clearly infer generating summaries that are entailed by source articles is easier than generating summaries that are coherent, concise, readable, and grammatical. This is because if summaries have words copied from the source articles, it is highly likely that they are entailed by the source articles. Since the baseline approaches (ILPSumm, and ParaFuse) have higher copy rate, they do well in entailment. However, with our approach, having a low copy rate and

generating summaries that are entailed by the sources articles is difficult; yet, our proposed approach still has the best entailment score for task DUC-2004.

2) MAG-20 Abstractive Evaluation: For MAG-20, our approach performs better than the baseline approaches in coherence, conciseness, readability, and grammar across two of our human evaluators, while marginally losing to the baselines according to one of our evaluators. As for entailment, ILPSumm performs the best which is attributed to the high copy rate by ILPSumm. Even though our approach generates significantly more novel words than ILPSumm or ParaFuse, we lose to the best entailment score by only 4%. Further, ILPSumm, ParaFuse, and our proposed approach perform generally better on MAG-20 than on DUC-2004. We surmise this is due to the headline generation task for DUC-2004, while we use author-provided titles for MAG-20.

VI. Conclusion and Future Work

We proposed an unsupervised multi-document abstractive summarization framework that, when given a set of documents from MAG, automatically clusters the documents and then generates summaries for each cluster. Our framework consists of extractive and abstractive phases. In the extractive phase, we use coreference resolution to extract groups of inter-dependent sentences from source articles and centroid-based clustering followed by an enhanced multi-sentence compression algorithm to generate topically informative and relevant summaries. In the abstractive phase, we use text generation technique to generate abstractive language units that are synthesized into an abstractive summary. The number of summaries in our proposed method is adaptively determined based on the semantic analysis of the topics discussed in the documents. We introduce MAG-20, a dataset of topically-clustered groups of scientific articles across 20 Fields of Study and their abstractive summaries. Results show that our proposed approach performs better than state-of-the-art centroid-based summarization techniques on 5 human evaluation metrics and copy rate. In the future, we plan to use additional knowledge and metadata such as citation relationships among scientific articles for document summarization.

VII. Acknowledgment

The authors are deeply grateful to Daniel Foose for helping with developing scripts for efficient data collection from MAG.

References

[1] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "A joint sentence scoring and selection framework for neural extractive document summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 671–681, 2020.

[2] Y. Liu, "Fine-tune bert for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.

[3] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," *arXiv preprint arXiv:1802.08636*, 2018.

[4] K. Thirunarayan, T. Immaneni, and M. V. Shaik, "Selecting labels for news document clusters," in *International Conference on Application of Natural Language to Information Systems*. Springer, 2007, pp. 119–130.

[5] E. Chu and P. Liu, "Meansum: a neural model for unsupervised multi-document abstractive summarization," in *International Conference on Machine Learning*, 2019, pp. 1223–1232.

[6] M. T. Nayeem, T. A. Fuad, and Y. Chali, "Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1191–1204.

[7] S. Banerjee, P. Mitra, and K. Sugiyama, "Multi-document abstractive summarization using ilp based multi-sentence compression," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[8] K. Filippova, "Multi-sentence compression: Finding shortest paths in word graphs," in *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, 2010, pp. 322–330.

[9] Y. Zhao, X. Shen, W. Bi, and A. Aizawa, "Unsupervised rewriter for multi-sentence compression," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2235–2240.

[10] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and 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.

[11] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural abstractive text summarization with sequence-to-sequence models," *arXiv preprint arXiv:1812.02303*, 2018.

[12] C. Khatri, G. Singh, and N. Parikh, "Abstractive and extractive text summarization using document context vector and recurrent neural networks," *arXiv preprint arXiv:1807.08000*, 2018.

[13] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[14] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[15] K. Liao, L. Lebanoff, and F. Liu, "Abstract meaning representation for multi-document summarization," *arXiv preprint arXiv:1806.05655*, 2018.

[16] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," *arXiv preprint arXiv:1805.10399*, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] K. Krishna and B. V. Srinivasan, "Generating topic-oriented summaries using neural attention," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1697–1705.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[21] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[22] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," *arXiv preprint arXiv:1707.07045*, 2017.

[23] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[24] F. Boudin and E. Morin, "Keyphrase extraction for n-best reranking in multi-sentence compression," 2013.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[26] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.