A Framework for Scientific Article Summarization

Blake Allen[1], Priscilla Burity[1], Derek Topper[1]

Abstract— **In this paper, we explore techniques for improving abstractive summarization of scientific papers. Our primary focus is on augmenting and enhancing state-of-the-art transformer techniques, such as PEGASUS. We demonstrate a custom pre-processing scoring technique for reducing total sentences that the transformer is fed, via a set of sentence quality metrics. Our results show that our method of scoring demonstrates improved ROUGE-N and ROUGE-L performance across all model types. Our approach indicates that our methodology can also aid in significant reductions in computation time. We also explore a custom attention mechanism for selection of top words, an ensemble Pegasus approach and fine-tune a Pegasus transformer on novel scientific data.**

I. Introduction

LITERATURE review is a key step at the beginning of any research project, as it provides a foundation of knowledge on the topic of interest. It helps prevent duplication and give credit to other researchers. Automatically summarizing scientific articles allows researchers in their investigation by speeding up the research processes. In this analysis, we were able to generate abstractive summaries from the text of scientific articles using transformers.

Abstractive summarization includes heuristic approaches to train the system in making an attempt to understand the whole context and generate a summary based on that understanding. This is a human-like way of generating summaries, which are more effective as compared to the extractive approaches, which essentially involves extracting particular pieces of text (usually sentences) based on weights assigned to words. Here, the length of the summary can be manipulated by defining the maximum and minimum number of sentences to be included in the summary.

There are some important differences between summarizing generic texts and scientific articles, as discussed in (Teufel and Moens, 2002). The first difference is related to the text structure, as scientific articles usually starts with an introduction, followed by related works, methodology, etc. Second, scientific articles are usually much longer than generic texts. Finally, scientific articles summarization comes from the need to turn literature reviews faster, and literature reviews should go over not only individual articles of interest but also the network of citations of such articles. In this project we're particularly concerned with this third difference, and also take into account citation sentences as inputs to our summaries.

[1]UC Berkeley School of Information, e-mail: {blaked,burityp,topper}@berkeley.edu.

II. Background

There has been a plethora of previous research in the field of scientific summarization (Altmami and Menai, 2020).

There are multiple ways to approach this problem. Some researchers (Lloret et al., 2011) have used an approach that summarizes papers through an automatic abstract generation-based summarization approach. Other researchers (Qazvinian and Radev, 2008) have altered their strategy to focus on an automatic citation-based summarization approaches. Furthermore, some researchers (Lloret et al. , 2011) have employed abstractive summarization techniques, where summaries are generated as new words, while others (Mei and Zhai, 2008) have employed extractive summarization techniques where summaries are created from existing text.

However, many regularly-cited projects have been stymied by a lack of available papers to critically analyze and train models on ((Mei and Zhai, 2008), (Qazvinian and Radev, 2008)). This is one area that we found room for improvement, through utilizing the text of papers in novel ways to generate summarizations.

Important advances in summarization tasks were made in recent years, with the transformers architecture. In fact, the transformer-based models BART (Lewis et al., 2019), T5 (Raffel et al., 2019), and PEGASUS (Zhang et al., 2019) are the state-of-the-art in summarization tasks. Recent research (Goodwin, 2020) has suggested that, for abstractive multi-document summarization tasks, there is no statistically significant difference in summary quality among them.

Among those models though, PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence) was specifically designed for abstractive summarization and was chosen to be our baseline model in this project. The novelty of PEGASUS' architecture lies in its self-supervised pre-training objective. It essentially removes the dependency of data on labeled samples and makes a huge amount of unexplored, unlabelled data accessible for training.

In 2019, the Yale LILY lab has made available over 1,000 papers in the Association of Computational Linguistics (ACL) anthology network with their citation networks (e.g. citation sentences, citation counts) and their comprehensive, manual summaries. (Yasunaga et al., 2019) introduces the corpus in detail and shows how ScisummNet enables the training of LSTM models for scientific papers. This is a very interesting dataset, as 1) it provides a large num-

ber of manual summaries that are a good reference for abstractive summarization; 2) it includes citation sentences, which we deem as a fundamental part of an effective and comprehensive summarization of a scientific work.

In this project we will compare the performance of our summarizations (as measured by ROUGE score ROUGE-1/ROUGE-2/ROUGE-L) with the performance of of summarizations in (Yasunaga et al., 2019) using a similar dataset as test set (that of the CL-Scisumm project). We'll also compare the performance of our summarizations across different approaches for pre-processing the text of papers, which will be discussed in the next section.

We evaluate system summaries against the gold summaries by ROUGE (Lin, 2004), which serves as a good metric for this work, as we aim to measure the comprehensiveness of summaries.

We also attempt a custom model which pulls data from attention layers in the pre-trained PEGASUS (google/pegasus-large) model, applies a transformation to the attention, and attempts to locate the top k words indicated by the model. This approach, which did not achieve high ROUGE scores, was an interesting dive into how attention mechanisms can be summed, transformed and further information can be extracted from the model.

III. Methods

Our implementation takes advantage of three datasets.

1 - ScisummNet: 1,000 papers in the ACL' Anthology Network Corpus with their citation networks (e.g. citation sentences, citation counts) and their comprehensive, manual summaries, introduced by (Yasunaga et al., 2019).

2 - Arxiv dataset: The Arxiv 'preprint' server allows for bulk downloads of the PDFs of the scientific papers. 600,000 papers were downloaded and then parsed from PDF encoding to plain text. The plain text corresponds to the input of the model training task, and we targeted the abstract for the summarization target of the training task. Extracting the abstract from the full-text became a hard sub-problem of the dataset building, in order to subvert this possibly error prone sub-problem, abstracts were pulled from the Arxiv website. It should be noted that these abstracts where of the time were not identical to the abstract of the paper providing sometimes more brevity than the original paper. A random sampling of 10,000 papers provided a mini-dataset to test small scale experiments on.

3 - AAN: The AAN dataset (Radev, 2013) was originally built in 2009 as the The Association of Computational Linguistics' Anthology Network Corpus. It is maintained by Yale University's Language, Information, and Learning at Yale Group and Professor Dragomir R. Radev. As of 2021, it contains over 36,000 papers, which we parsed down into the 4,200 papers that contained multiple citations from other papers, which we used for our cited text spans. We also ensured there was no overlap between AAN papers and ScisummNet papers.

IV. Experimentation

With these three datasets, we implement the following exercises:

A. Exercise 1-PEGASUS with ScisummNet data

Through this approach, we evaluated the performance of PEGASUS summarizing the bodies and citation sentences of scientific articles in the ScisummNet dataset.

In the preparation of the dataset, the first step is related to the citations. Instead of using the citations directly, we found cited text spans in the document, which are sentences in the body that have the largest similarity with the citation sentences. The idea of using cited text spans instead of the citations directly comes from the CL-Scisumm shared task, that includes cited text spans as one of the tasks. While (Yasunaga et al., 2019) measures similarity in terms of the TF-IDF, we opted by ROUGE-1 score, which runs faster and yielded more interesting outputs. We picked the 3 sentences in the body that better conveyed the information in the citations.

In the second step, we ensure that, for a given paper, only sentences with good quality are used as model inputs. The quality of each sentence is evaluated based on the share of 'undesired' tokens in the sentence. 'Undesired' tokens are defined as those which turn the reading less fluent, and include math symbols, numbers in general, tokens with no synsets, etc. If the quality standards are not reached for a given sentence, it is removed from the article. After the quality check phase, the number of sentences in the articles' body is reduced on average by 33 %.

The third step consists of an extractive phase. Here, the 25, 50, 100, 200 and 300 most important sentences will be extracted from the body, which can be used to better condition the transformer. The importance of each sentence is measured by ROUGE-1 score of each sentence vs. the rest of the document. (Subramanian et al., 2020) found that that the extractive step significantly improves the performance of summarization results.

A.1 Exercise 1.1: PEGASUS-large on ScisummNet data

We divided the 1000 papers sample into training (60 %), validation (20 %) and test (20 %) sets. In this first exercise we only apply the model 'PEGASUS-large' to the test set and evaluate the model. PEGASUS-large was selected after an exhaustive search of various transformer methods, as it produced the highest quality summarizations.

A.2 Exercise 1.2: Fine-tuned PEGASUS-large with ScisummNet data

Here we fine-tune our model with the train set and evaluate on the test set. This allowed us to alter

PEGASUS, in order to improve resulting ROUGE scores.

A.3 Exercise 1.3: Improving PEGASUS-large with 'Body Quality' metric on ScisummNet data

We then pre-process our data in order to ensure that our model is being trained on the best tokens possible. Body quality, is a custom scoring metric to remove math symbols, equations, and things that should not be featured in a summary. It includes the SDF optimizer quality check criteria. The top N sorted quality sentences are returned by the algorithm.

We then evaluate ROUGE 1, ROUGE 2 and ROUGE L recall metrics to see if body quality could improve recall of the PEGASUS-Large transformer on the ScisummNet data.

B. Exercise 2: Custom model

Analyzing the attention mechanisms of the pre-trained PEGASUS model led to some interesting seeming insights. The goal was to see if there could be useful information extracted from a summation and transformation of the encoder's attention mechanisms. This attention deemed "shallow attentionïs pulled from the first 2 layers of the encoder attentions and shown below in Figures III.2.1 and III.2.2.

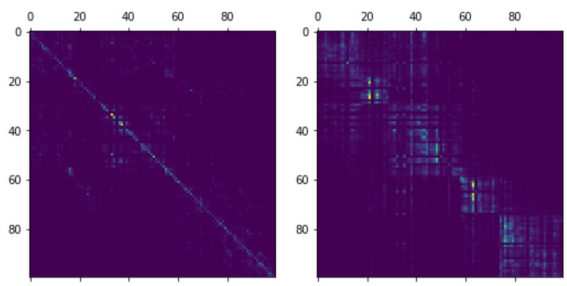

Fig. 1: Figures III.2.1 and III.2.2 show typical self attention visualizations.

We can see that some words have a much larger attention value than others and greater global attention. Applying a summation and a matrix transformation we attempt to visualize the bands of the most important words based on the total amount of attention. This simple mathematical transformation can be described as: $C = \sum_{j=0}^{n} \sum_{i=0}^{i} X_{ij}^{t} X_{ij}$, where n is the depth to which we will sum the shallow attention, we chose a value to 2; summing the first two attention layers of the pre-trained model.

As depicted in Figures III.2.3, III.2.4 and III.2.5, by using self attention mechanisms, our model is able to deduce that there are tokens with higher degree of attention activations across the entire set of attention mechanisms. The framework for this approach is further demonstrated in Figure III.2.6.

C. Exercise 3: Ensemble Approach

We also explore the effectiveness of an ensemble approach in analyzing larger parts of the scientific text. The default PEGASUS model can only input word tokens at a time, while most scientific articles

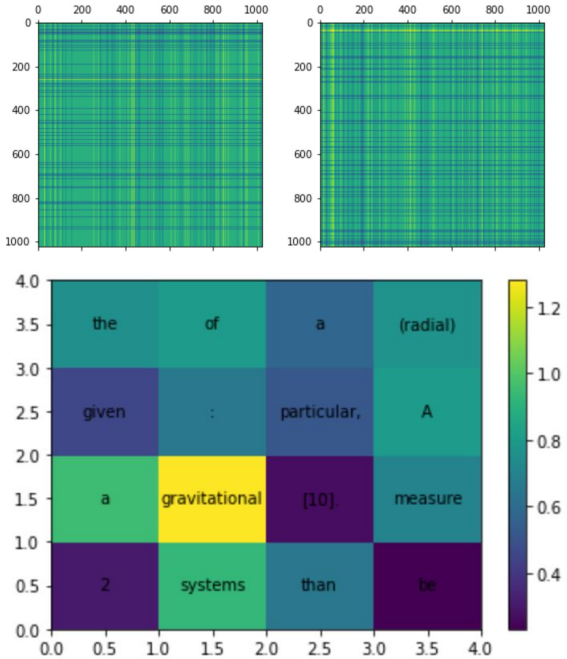

Fig. 2: Figures III.2.3 and III.2.4 show the results of our attention transformations. The self attention mechanisms seem to indicate that some words have higher attention activations across the entire set of attention mechanisms. Figure III.2.5 hones in on one example shows how the word 'gravitational' is picked by the model, a highly important word for a paper about gravitational dynamics.

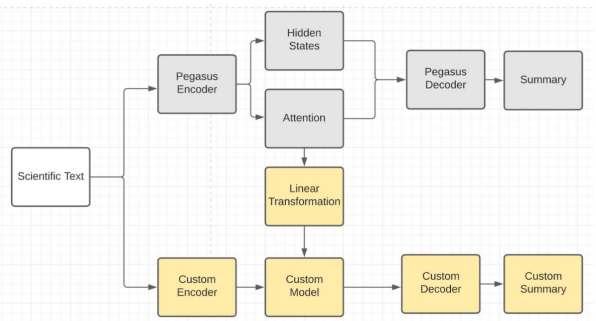

Fig. 3: Figure III.2.6 demonstrates the pipeline for feeding transformed attention into our custom model.

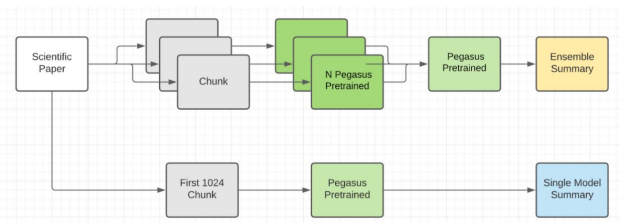

Fig. 4: Figure III.3.1 demonstrates the ensemble PEGASUS pre-trained approach.

are much larger in length. We ran PEGASUS through all chunks of the input data, concatenated the output and ran a final pass through the PEGASUS pre-trained model. The framework for this approach is depicted in Figure III.3.1.

D. Exercise 4: Transfer Learning

We also explore the effectiveness of an transfer learning approach in training our fine-tuned PEGASUS model on the ScisummNet data and then applying it to the pre-processed AAN dataset. This allowed us to see the effect of analyzing a larger da-

taset of papers with our model, in order to see the results of our model in a wider context.

V. Results

From these approaches, we were able to achieve results that made marginal improvements on the baseline results from previous approaches to this problem ((Yasunaga et al, 2019), (Altmami and Menai, 2020)). For the below custom scoring table (Figure IV.1), we were able to achieve our results by running multiple models on the ScisummNet data. For each of those examples, we report the ROUGE-1/ROUGE-2/ROUGE-L scores. This refers to the tokenization level, where we compare the longest subsequence of unigrams, bigrams and N-grams. We apply this methodology to PEGASUS and these numbers represent the ROUGE score values for each of the selected inputs plus the cited text spans.

The effect of this improvement is elucidated in Figures IV.2 and IV.3, where the achieved performance is depicted based on each model's inputs. The processed, carefully-selected, high-quality sentences obtained better ROUGE scores than the unprocessed body sentence baseline.

ScisummNet Model Input	T5	Pegasus	Pegasus Fine-Tuned
Abstract	63.5 / 50.34 / 52.01	71.89 / 53.25 / 64.02	73.48 / 56.74 / 66.66
Body	23.25 / 6.44 / 19.09	39.38 / 16.48 / 24.03	40.27 / 17.72 / 25.42
Body w/ quality check	42.86 / 22.41 / 32.14	46.86 / 15.95 / 28.12	46.88 / 16.04 / 28.1
Body w/ quality check limited 25 sent.	37.74 / 19.32 / 28.42	41.57 / 9.22 / 22.83	41.7 / 9.35 / 22.75
Body w/ quality check limited 50 sent.	38.06 / 18.64 / 28.39	42.02 / 9.67 / 23.85	41.53 / 9.55 / 23.32
Body w/ quality check limited 100 sent.	40.73 / 20.46 / 30.29	44.08 / 11.91 / 25.28	43.56 / 11.91 / 24.86
Body w/ quality check limited 200 sent.	40.48 / 20.02 / 29.88	46.12 / 14.72 / 27.73	45.94 / 14.66 / 27.39
Body w/ quality check limited 300 sent.	26.97 / 9.6 / 18.28	46.89 / 16.06 / 28.27	46.8 / 16.18 / 28.12

Fig. 5: Figure IV.1: The results of our model, from Exercise 1 (ROUGE-1/ROUGE-2/ROUGE-L). All of the model inputs shown include cited texts spans.

In regards to the custom model approach, these particular transformations were not effective in translating to learn-able parameters but further investigation and other transformations may still be warranted.

Additionally, in regards to the ensemble model approach, the PEGASUS pre-train outperformed the ensemble approach. We found that this seems to occur because most scientific articles contain their abstract in the first part of the text. Further analysis of the paper proved unfruitful for the task of generating a better summary. A decrease in the ROUGE scores was demonstrated, as shown in Figures IV.4 and IV.5.

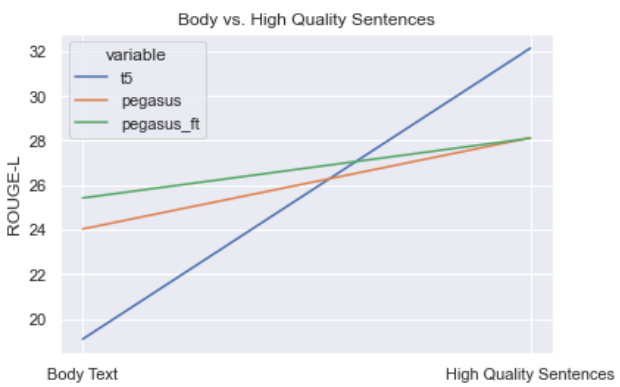

Fig. 6: In Figure IV.2, We can see that we were able to achieve notably better performance on the processed, high quality sentences relative to the unprocessed body sentences. This recall performance increased across all models.

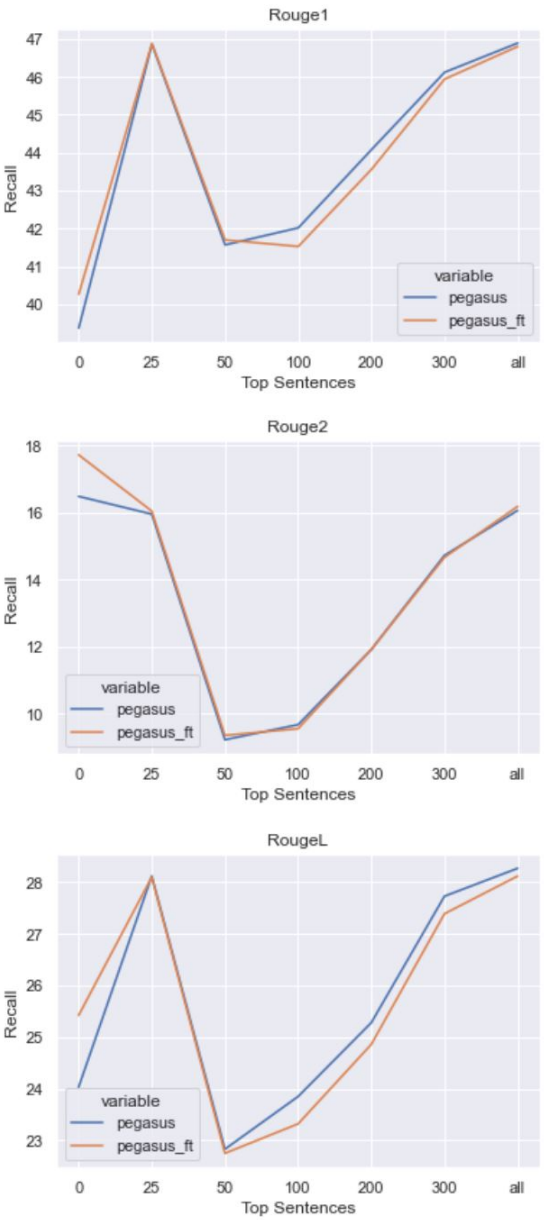

Fig. 7: Figure IV.3: We see a sharp increase in recall at around 25 sentences demonstrating our approach speeds up computation time while increasing net recall. After this point the advantages seem to decrease until eventually adding more data and computation will likely win out.

For the approach with applying different data, we used the same model as our approach from Exercise 1, to train our model on ScisummNet data. The model was then set to evaluate data from the AAN dataset. The results from this approach, as shown in Figure IV.6, did not preform as well as those from Figure IV.1, which the model was trained on. This approach preformed well, relative to previous approaches to summarize scientific articles (Altmami and Menai, 2020). The AAN results notably preformed worse than the ScisummNet results, and seemed to preform better as more of the paper's sentences were included. However, these results indicate that this model can be applied to other academic papers and receive similar results.

	rouge1	rouge2	rougeL	rougeLsum
count	97.000000	97.000000	97.000000	97.000000
mean	0.379494	0.224306	0.304467	0.325009
std	0.263346	0.282148	0.255667	0.248032
min	0.049180	0.000000	0.046512	0.030120
25%	0.172131	0.014925	0.116279	0.139535
50%	0.265306	0.048193	0.177083	0.211765
75%	0.593220	0.422222	0.468750	0.535211
max	1.000000	0.985714	1.000000	0.933333

Fig. 8: Figure IV.4: Baseline PEGASUS model: Mean ROUGE scores are around thirty percent.

	rouge1	rouge2	rougeL	rougeLsum
count	97.000000	97.000000	97.000000	97.000000
mean	0.345866	0.146845	0.250356	0.289563
std	0.252653	0.229501	0.217534	0.230316
min	0.000000	0.000000	0.000000	0.000000
25%	0.156522	0.017241	0.104348	0.122951
50%	0.279070	0.050505	0.180556	0.220588
75%	0.484848	0.135135	0.322581	0.388889
max	1.000000	1.000000	1.000000	1.000000

Fig. 9: Figure IV.5: Ensemble PEGASUS model: Mean ROUGE scores are around twenty percent.

AAN Model Input	Pegasus
Body w/ quality check limited 25 sent.	29.47/5.58/16.67
Body w/ quality check limited 100 sent.	30.84/6.21/18.06
Body w/ quality check limited 300 sent.	32.09/7.33/19.67

Fig. 10: Figure IV.6 is the results from the ScisummNet-trained model being run on the AAN dataset.

VI. Conclusion

This paper explores techniques for improving abstractive summarization of scientific papers. The baseline PEGASUS and fine-tuned PEGASUS models demonstrate relatively high recall as judged by the ROUGE metric. We demonstrate that our preprocessing scoring technique for reducing total sentences improved ROUGE-N and ROUGE-L performance across all model types. We also demonstrate that this technique is not limited to merely one set of papers and show that it can be applied more broadly. Additionally, using only 25 of the top sentences can significantly reduce computation time by reducing the total amount of text the model has to encode. Further improvements could be made via more advanced scoring techniques, continued model customization and further analysis of key sentences needed for summarization in a scientific paper. Our Github repository for this project is: `https://github.com/blakedallen/summary-of-science`.

References

[1] Nouf Ibrahim Altmami and Mohamed El Bachir Menai, "Automatic summarization of scientific articles: A survey.," *Journal of King Saud University - Computer and Information Sciences*, 2020.

[2] Travis Goodwin, Max Savery, and Dina Demner-Fushman, "Flight of the PEGASUS? comparing transformers on few-shot and zero-shot multi-document abstractive summarization," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), Dec. 2020, pp. 5640–5646, International Committee on Computational Linguistics.

[3] Virapat Kieuvongngam et al., "Automatic text summarization of covid-19 medical research articles using bert and gpt-2.," 2020.

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *CoRR*, vol. abs/1910.13461, 2019.

[5] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, July 2004, pp. 74–81, Association for Computational Linguistics.

[6] Elena Lloret and Manuel Palomar María Teresa Román-Ferri, "Compendium: a text summarization system for generating abstracts of research papers," *International Conference on Application of Natural Language to Information Systems, Springer*, pp. 3–14, 2011.

[7] Qiaozhu Mei and ChengXiang Zhai, "Generating impact-based summaries for scientific literature," *Proceedings of ACL-08: HLT: 816–824*, 2008.

[8] Thang Pham et al., "Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?," 2012.

[9] Vahed Qazvinian and DragomirR Radev, "Scientific paper summarization using citation summary networks," *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 689–696, 2008.

[10] Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara, "The acl anthology network corpus," *Language Resources and Evaluation*, pp. 1–26, 2013.

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

[12] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher J. Pal, "On extractive and abstractive neural document summarization with transformer language models," *CoRR*, vol. abs/1909.03186, 2019.

[13] Simone Teufel and Marc Moens, "Summarizing scientific articles: Experiments with relevance and rhetorical status," 2002.

[14] Michihiro Yasunaga et al., "Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks," *Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33*, p. 7386–7393, 2019.

[15] Jingqing Zhang et al., "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2020.