Generating Summaries for Scientific Paper Review

Anonymous ACL submission

Abstract

The review process is essential to ensure the quality of publications. Recently, the increase of submissions for top venues in machine learning and NLP has caused a problem of excessive burden on reviewers and has often caused concerns regarding how this may not only overload reviewers, but also may affect the quality of the reviews. An automatic system for assisting with the reviewing process could be a solution for ameliorating the problem. In this paper, we explore automatic review summary generation for scientific papers. We posit that neural language models have the potential to be valuable candidates for this task. In order to test this hypothesis, we release a new dataset of scientific papers and their reviews, collected from papers published in the NeurIPS conference from 2013 to 2020. We evaluate state of the art neural summarization models, present initial results on the feasibility of automatic review summary generation, and propose directions for the future.

1 Introduction

Reviewing is at the center of the scientific publication process, and the quality of publications is dependent on it. In many scientific fields, including natural language processing and machine learning, submissions for publication are reviewed using a peer review system. Recently, these fields are seeing increasing volumes of submissions each year, especially in high reputation venues. This has created an issue of over-burdening of reviewers, which is not only a problem for the quality of life of scientists, but also consequently affects the quality of the reviews. With ever increasing volume of new results in these fields, submissions for publication are expected to multiply still, and the problem is only expected to deepen, which is raising concerns in the scientific community (Rogers and Augenstein, 2020).

One avenue for ameliorating this problem is relying on artificial intelligence to assist with the process, in order to remove some of the burden from the human reviewers. A possibility would be to generate reviews or article summaries automatically, in order to speed up the human's understanding of the paper, or to assist with parts of the review writing, e.g., a few sentences summary.

Text generation has seen impressive improvements in recent years, being one of the most active fields in NLP, with the highest leaps in performance of newly published models. Models such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020) have shown impressive results for text generation, as well as for other tasks, acting as language models which can generalize for a wide range of tasks in NLP with only little fine-tuning.

Text summarization is a problem of text generation. Depending on the approach, summarization can be extractive (Zheng and Lapata, 2019) or abstractive (See et al., 2017; Nallapati et al., 2016). Extractive summarization is performed by selecting key sentences from the original text, while abstractive summarization tackles the more difficult problem of generating novel text that summarizes a given input—the problem we are interested in and explore in this paper. As for text generation in general, state-of-the-art models for summarization are generally neural and transformer-based such as PEGASUS (Zhang et al., 2020) and Prophet (Qi et al., 2020). These models have been used for text summarization for different domains, including news (Desai et al., 2020) and scientific texts. For scientific text summarization, Zhang et al. (2020) have obtained best results in existing literature, based on evaluation on a dataset of articles published on arXiv and PubMed using papers' abstracts as ground truth.

Scientific texts pose specific problems for summarization, given their particular structure and way of organizing information. This is why the problem of scientific text summarization has been approached separately from general summarization systems. The problem of scientific text summarization has been approached before (Yasunaga et al., 2019; Altmami and Menai, 2020; Ju et al., 2020; Cohan and Goharian, 2017; Qazvinian et al., 2010). Top conferences in NLP have organized workshops on scholarly document processing, including shared tasks specifically focused on scientific document summarization (Chandrasekaran et al., 2019). Most approaches for scientific text summarization use an extractive (Saggion and Lapalme, 2000; Saggion, 2011; Yang et al., 2016; Slamet et al., 2018; Agrawal et al., 2019; Hoang and Kan, 2010) or citation-based approach (Cohan and Goharian, 2017; Qazvinian et al., 2010; Ronzano and Saggion, 2016), with a few exceptions attempting abstractive summarization on scientific texts (Lloret et al., 2013). Notably, Ju et al. (2020) use a combined extractive and abstractive approach based on BERT. Sun and Zhuge (2018) propose an approach based on semantic link networks for summarizing scientific texts. A recently published survey (Altmami and Menai, 2020) contains a more exhaustive overview of previous attempts at summarizing scientific papers.

Given the excellent results of recent text generation models, it is promising to consider new applications in fields where they have not been leveraged in practice before. We propose that one such task is *scientific review summary generation*. We evaluate in this paper the feasibility of automatically generating review summaries for scientific papers. We use state-of-the-art models for text summarization, and apply them to our problem. We release a dataset of articles and reviews from NeurIPS, which we use to assess the performance of automatic summarization models for the problem of review summary generation.

2 Dataset

We build a dataset of articles and associated reviews by scraping NeurIPS's conference website,[1] and collecting all articles published in NeurIPS between 2013 and 2020, along with their reviews. To obtain the full text of the papers, we downloaded the PDFs from the website and extracted the text using Grobid.[2] Reviews were extracted directly from the HTML content of the web pages, and, where needed, heuristics were used in order to exclude the texts of the author's responses. Each article can have several reviews. Table 1 summarizes statistics about the dataset.

[1] https://papers.neurips.cc

[2] https://github.com/kermitt2/grobid

Articles	5,950
Reviews	18,926
Avg review len (words)	399
Avg review len (sentences)	21
Avg abstract len (words)	159
Avg abstract len (sentences)	7

Table 1: Dataset statistics.

3 Summarization Experiments

Reviews of scientific articles are usually comprised of a short summary, followed by the comments comprising the reviewer's evaluation of the article, mentioning its strengths and its weaknesses. The initial summary of the paper is usually a short objective description of its contents, so in theory it could be inferred solely based on the article's content. Based on this premise, we formulate the problem of automatic review generation as a text summarization problem.

Pre-processing. We aim to separate the two different parts of each review: the initial part containing a short summary of the paper, from the following comments and evaluation of the paper. A manual inspection of extracted reviews in our dataset for papers up to 2019 shows that many reviews include replies to author responses from the rebuttal phase of the review, and these can be found either in the beginning or end of the review, without a consistent pattern, sometimes separated from the main review by ASCII separators (strings of "-"/"="/" "). We then rely on heuristics in order to correctly extract the summary part of the review, by searching the review text for keywords such as "rebuttal" or "response": if these are found at the beginning of the review, we then look for ASCII separator characters, and consider the original review to begin after the separator; otherwise, we assume the summary is found at the beginning of the review. For papers from NeurIPS 2020, the different sections of the review are clearly marked (*summary*, *strengths*, *weaknesses*, *clarity* and *correctness*), so this pre-processing step was not needed. After this step, we split the obtained text into sentences and select the first k sentences as the summary. Our motivation in doing so was driven by several works on extreme classification (Narayan et al., 2019, 2018) aimed at generating short, one-sentence news sum-

	R-1	R-2	R-L	BERTScore
vs. arXiv abstracts (Zhang et al., 2020)	.447	.173	.258	-
vs. abstract (NeurIPS)	.236	.046	.151	.793
vs. review summaries (individual whole)	.169	.023	.117	.789
vs. review summaries (concatenated whole)	.206	.033	.127	.784

Table 2: Performance of pretrained model

mary to answer the question: "What is the article about?".

Model. Language modeling in NLP has recently seen great advancements, and is one of the most active areas of research in NLP, with new results being published every few months. The best performing models are based on neural architectures, among which transformers play an important role. Text summarization in particular is a type of text generation problem; the current state of the art in text generation is PEGASUS (Zhang et al., 2020), which is a transformers-based model trained to generate summaries by masking important sentences in a source text. PEGASUS obtained state-of-the-art results in text summarization across 12 different datasets in different domains, including scientific texts.

We experiment with using PEGASUS in order to generate summaries of scientific articles in our dataset, and assess its performance compared to the collected reviews.

Model pre-trained on abstracts. We first experiment with a pre-trained version of PEGASUS for scientific text summarization, which was trained to generate abstracts of scientific texts based on a dataset of arXiv articles (Cohan et al., 2018). In order to ensure no overlap between the test set used for evaluation in our experiments and the articles in the arXiv database used in pre-training of the model, we select as our test set only the articles in our dataset published in 2020 (the arXiv dataset was published in 2018) - we use 1000 of these articles as our test set and keep the rest of 898 as a validation set. The 2020 reviews are also the highest-quality of our dataset, since the summary section of the review is clearly marked and used as is for evaluation (as opposed to extracted based on heuristics).

Model fine-tuned on reviews. Second, we attempt to generate paper summaries which best approximate a review. For this purpose, we fine-tune the pre-trained model used in the previous experiment on our own data, using as targets the reviews in our dataset. As a training set, we use the articles and reviews in our dataset published before 2020. While our dataset is smaller than the arXiv dataset used for the pre-trained model, it is expected to be similar to the original training data. For each article, one review is selected at random and used as ground truth for training the summarization model. The training set contains 4,052 papers and their reviews.

Evaluation. We evaluate the models using the ROUGE metric, and compare the generated summaries both to the abstract and the reviews. We report ROUGE-1, ROUGE-2 and ROUGE-L, as well as BERTScore, using the RoBERTa-large model[3] (Zhang et al., 2019). Our setup can be evaluated on multiple labels for the same input text: in our test set, one paper can have several reviews. We evaluate our models with multiple labels: first by considering them separately as independent examples, and second by concatenating all reviews for a given input article into one single reference text, and evaluating against it.

We show examples of generated reviews using our model, along with the original reviews for the same article, in the Appendix.

Results. We report separately the results of the pre-trained and the fine-tuned model. We compare different setups, using as target texts both the abstracts and the reviews. In the case of the reviews, we consider separately as a target test the whole review or only the summary section, varying the number of extracted sentences from 1 to 5, and experiment with the two evaluation setups: concatenating the different reviews corresponding to one article, or considering them as separate test examples.

Tables 2 and 3 and show the results for all setups. The pre-trained model obtains better results when evaluated against abstracts than against reviews, across configurations and metrics. Although the pre-trained model was trained to generate abstracts, the fine-tuned model still obtains slightly better results compared to abstracts, suggesting it might

[3] `roberta-large_L17_no-idf_version=0.3.9 (hug_trans=4.2.2)`

	R-1	R-2	R-L	BERTScore
vs. abstract (NeurIPS)	.261	.034	.141	.812
vs. review summaries (indiviual whole)	.230	.031	.148	.817
vs. review summaries (concatenated whole)	.254	.046	.145	.806
vs. review summaries (concatenated 5 sents)	.273	.047	.155	.808
vs. review summaries (concatenated 4 sents)	.279	.046	.158	.810
vs. review summaries (concatenated 3 sents)	.287	.045	.164	.813
vs. review summaries (concatenated 2 sents)	.290	.042	.170	.817
vs. review summaries (concatenated 1 sent)	.246	.032	.160	.821
vs. review summaries (individual 5 sents)	.227	.030	.149	.818
vs. review summaries (individual 4 sents)	.220	.028	.147	.819
vs. review summaries (individual 3 sents)	.207	.026	.117	.819
vs. review summaries (individual 2 sents)	.176	.022	.127	.820
vs. review summaries (individual 1 sent)	,114	.053	.091	.822

Table 3: Performance of fine-tuned model on abstract and review summary

	R-1	R-2	R-L	BERT Score
vs. full review (concat)	.152	.036	.092	.803
vs. full review (individual)	.241	.040	.139	.806
vs. strenghts (concat)	.270	.039	.159	.815
vs. strengths (individual)	.200	.038	.135	.820
vs. weaknesses (concat)	.232	.028	.134	.803
vs. weaknesses (individual)	.212	.027	.134	.808

Table 4: Performance of fine-tuned model on full review and other review sections

solve a relevant domain adaptation aspect. The fine-tuned model also shows improved results for review summary generation. In terms of ROUGE scores, the optimal number of sentences of the summary extracted from the review summary seems to be 2 in the concatenated setup, while in the individual setup, the performance increases with the number of sentences. BERTScore strictly decreases with the number of sentences for both setups. Especially in the concatenated setup, using the first 1-2 sentences in the review summary as labels outperforms evaluating against the full review summary, suggesting that the generated summaries generally contain information present in the beginning of the review.

3.1 Feasibility of Generating Full Reviews

The fine-tuned model is better at generating review summaries than the pre-trained model, across setups.

The generation of a full review, including critical interpretations from the reviewers, is a much more challenging problem than generating paper summaries. In order to assess how well a summarization model can approximate a full review, including not only the summary, but also the critical comments sections, we separately evaluate our model using the full reviews as targets, as well as against the separate sections (we consider the *Strengths* and *Weaknesses* sections), as show in Table 4. We notice that the performance is generally lower than for the review summary, but still comparable. The *Strengths* section seems to have the most in common with the review summary according the better results.

4 Conclusions

We have formulated the problem of scientific text review generation, as a novel task in NLP with practical applications for the scientific community. Review generation is related to the text summarization task, but has its own specific features, which is what makes it a difficult problem to solve. We have taken the first steps towards building an automatic system for review generation; and have collected and are releasing a dataset of scientific articles and reviews which can be used for future experimentation into the topic.

We conclude that scientific review generation is a difficult problem, with current performance considerably below that of state-of-the-art text generation models on scientific abstracts. Nevertheless, the small improvements in performance we obtain through fine-tuning the model suggest that the problem might be approachable, and encourage us to continue to study it. We propose that more training data could be useful to obtain better results, as would a more accurate extraction of the summary section of the review. In the future, we would like to explore a more complex training strategy in order to improve performance, such as multi-task learning (to jointly train the model to generate reviews and abstracts), or conditional text generation, in order to constrain the model to generate review-like texts, while keeping the content relevant to the source article.

References

Kritika Agrawal, Aakash Mittal, and Vikram Pudi. 2019. Scalable, semi-supervised extraction of structured information from scientific literature. In *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*, pages 11–20.

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165.*

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and results: Cl-scisumm shared task 2019. *arXiv preprint arXiv:1907.09854.*

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685.*

Arman Cohan and Nazli Goharian. 2017. Contextualizing citations for scientific summarization using word embeddings and domain knowledge. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Shrey Desai, Jiacheng Xu, and Greg Durrett. 2020. Compressive summarization with plausibility and salience modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6259–6274.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435.

Jiaxin Ju, Ming Liu, Longxiang Gao, and Shirui Pan. 2020. Scisummpip: An unsupervised scientific paper summarization pipeline. *arXiv preprint arXiv:2010.09190.*

Elena Lloret, María Teresa Romá-Ferri, and Manuel Palomar. 2013. Compendium: A text summarization system for generating abstracts of research papers. *Data & Knowledge Engineering*, 88:164–175.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2019. What is this article about? extreme summarization with topic-aware convolutional neural networks. *CoRR*, abs/1907.08722.

Vahed Qazvinian, Dragomir Radev, and Arzucan Özgür. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010)*, pages 895–903.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Anna Rogers and Isabelle Augenstein. 2020. What can we do to improve peer review in nlp? *arXiv preprint arXiv:2010.03863.*

Francesco Ronzano and Horacio Saggion. 2016. An empirical assessment of citation information in scientific summarization. In *international conference on applications of natural language to information systems*, pages 318–325. Springer.

Horacio Saggion. 2011. Learning predicate insertion rules for document abstracting. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 301–312. Springer.

Horacio Saggion and Guy Lapalme. 2000. Selective analysis for automatic abstracting: Evaluating indicativeness and acceptability. In *RIAO*, pages 747–764. Citeseer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Cepi Slamet, AR Atmadja, DS Maylawati, RS Lestari, Wahyudin Darmalaksana, and Muhammad Ali

Ramdhani. 2018. Automated text summarization for indonesian article using vector space model. In *IOP Conference Series: Materials Science and Engineering*, volume 288, page 012037. IOP Publishing.

Xiaoping Sun and Hai Zhuge. 2018. Summarization of scientific paper through reinforcement ranking on semantic link network. *IEEE Access*, 6:40611–40625.

Shansong Yang, Weiming Lu, Zhanjiang Zhang, Baogang Wei, and Wenjia An. 2016. Amplifying scientific paper's abstract by leveraging data-weighted reconstruction. *Information Processing & Management*, 52(4):698–719.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Ethical Considerations

Our dataset poses no privacy issues. With regards to the task of paper review generation, it is unclear if generating reviews entirely automatically is desirable from a practical as well as ethical perspective. Instead, we approach the problem summary generation for reviews, in view of a possible computer-assisted process for review generation, which would not exclude humans. We think a computational tool for assisting with the ever-growing burden of reviewing can help the community and eventually lead to higher quality reviews, and hope our paper can encourage discussion on the topic. We leave open to discussion the question of how such a tool could best be integrated in the current review system.

A.2 Generated Summaries Examples

Generated summary	Original review
thanks to the authors for a very interesting paper. the main contribution that the authors have been able to solve this problem is the transient spike packet model, which can take into account discrete populations of neurons in a very long time invariant networks, and as the authors argue allows for a much better signal transmission and/or slower responses to fluctuations in the brain at all times. this implies that the authors argue allows for a much better model of nontriviality invariant networks, independent of population sizes and/or slower responses to fluctuations in the brain at all times. (...)	This paper uses the Drosophila2019s olfactory system (ORN to PN to LHN) as an example to demonstrate how the heterogeneity in the intrinsic parameters of single neurons could benefit for information transmission in a feedforward network, which is the main conceptual contribution of this work. Then they extend the idea of heterogeneous neurons to multiple layers of feedforward networks. (...)

Table 5: Example generated summary, and actual review, (paper title "Lamina-specific neuronal properties promote robust, stable signal propagation in feedforward networks").

Generated summary	**Original abstract**
methods for machine- learning models play an important role in developing, and using predictive models as information on what features were important for a given output for any machine- learning model, enabling us to better understand, validate, and interpret model decisions. $< n >$ researchers have developed a number of attribution methods that provide estimates of importance of input features towards model output for specific types of models, and accurate feature estimates for any machine-learning model. $< n >$ we present a new approach to estimating feature importance for any machine- learning model using causal explanation. $< n >$ this approach can be applied to any machine- learning model since it does not require retraining or adapting the original model to demonstrate importance scores are strongly causal, provided another machine- learning objective to train a model uses test data, and another machine- learning objective to learn accurate output. $< n >$ experiments show that the uncertainty estimates provided by this methodology are significantly more accurate than existing model-agnostic methods, and that the uncertainty scores provided by this methodology are significantly more accurate than existing model-agnostic methods. $< n >$ this approach can be applied to any machine- learning model since it does not require retraining or adapting the original model to demonstrate importance scores are strongly causal	Feature importance estimates that inform users about the degree to which given inputs influence the output of a predictive model are crucial for understanding, validating, and interpreting machine-learning models. However, providing fast and accurate estimates of feature importance for high-dimensional data, and quantifying the uncertainty of such estimates remain open challenges. Here, we frame the task of providing explanations for the decisions of machine-learning models as a causal learning task, and train causal explanation (CXPlain) models that learn to estimate to what degree certain inputs cause outputs in another machine-learning model. CXPlain can, once trained, be used to explain the target model in little time, and enables the quantification of the uncertainty associated with its feature importance estimates via bootstrap ensembling. We present experiments that demonstrate that CXPlain is significantly more accurate and faster than existing model-agnostic methods for estimating feature importance. In addition, we confirm that the uncertainty estimates provided by CXPlain ensembles are strongly correlated with their ability to accurately estimate feature importance on held-out data.

Table 6: Example of generated summary and original abstract, using pretrained model (paper title "CXPlain: Causal Explanations for Model Interpretation under Uncertainty")

Generated summary	**Original review**
neural networks are extensively used today in safety critical control systems such as autonomous vehicles and airborne collision avoidance systems. $< n >$ rigorous methods to ensure correct functioning of neural network controlled systems is imperative. $< n >$ verification refers to a broad class of techniques that provide strong guarantees of correctness by exhibiting a proof of abstraction. $< n >$ verification of neural networks has attracted lot of attention in recent years. $< n >$ verification of neural networks has attracted a lot of attention in recent years. $< n >$ this paper focuses on neural networks with rectified linear unit as range of activation function and we investigate the output range problem for feed-forward neural networks with rectified linear unit as range of activation function. $< n >$ broad objective is to investigate techniques to verify neural network controlled physical systems such as autonomous vehicles. $< n >$ verification refers to a broad class of techniques that provide strong guarantees of correctness by exhibiting a proof of abstraction. $< n >$ important verification problem is that safety, wherein one seeks to ensure that the neural network controlled system never reaches an unsafe set of states. $< n >$ important computation is to compute the output of network controller given a set of input valuations. $< n >$ we focus on neural networks with rectified linear unit as range of activation function and we investigate the output range	First of all, my knowledge of formal verification of neural networks is very limited, and I apologize for the limitations this poses on my review. That said, I found this paper very interesting, well written, and from my limited understanding of the literature, this seems like a novel and highly useful tool in the toolbox for verifying neural network models. I am strongly in favor of acceptance. My main questions are the following: * It is not clear to me what increase in false positives does the method introduce by relaxing the estimate of the output of the network to a superset. * I would like to see a more formal definition of the algorithm with the m̈oving pieces(̈e.g. partitioning strategies) stated more explicitly. Then I would like to have a discussion of the considerations that go into defining these m̈oving pieces.̈ * What are the practical limitations of the method on real-world network sizes and architectures.

Table 7: Example of generated summary and original review, using pre-trained model (paper title "Abstraction based Output Range Analysis for Neural Networks")