MetaGen: An academic Meta-review Generation system

Chaitanya Bhatia
chaitanya.bhatia.cse15@iitbhu.ac.in
Indian Institute of Technology, (BHU) Varanasi

Tribikram Pradhan
tpradhan.cse17@iitbhu.ac.in
Indian Institute of Technology, (BHU) Varanasi

Sukomal Pal
spal.cse@iitbhu.ac.in
Indian Institute of Technology, (BHU) Varanasi

ABSTRACT

Peer reviews form an essential part of scientific communications. Research papers and proposals are reviewed by several peers before they are finally accepted or rejected. The procedure followed requires experts to review the research work. Then the area/program chair/ editor writes a meta-review summarizing the review comments and taking a call based on the reviewers' decisions. In this paper, we present MetaGen, a novel meta-review generation system which takes the peer reviews as input and produces an assistive meta-review. This meta-review generation can help the area/program chair writing a meta-review and taking the final decision on the paper/proposal. Thus it can also help to speed up the review process for conference/journals where a large number of submissions need to be handled within a stipulated time. Our approach first generates an extractive draft and then uses fine-tuned UniLM (Unified Langauge Model) for predicting the acceptance decision and making the final meta-review in an abstractive manner. To the best of our knowledge, this is the first work in the direction of meta-review generation. Evaluation based on ROUGE score shows promising results and comparison with few state-of-the-art summarizers demonstrates the effectiveness of the system.

KEYWORDS

Random Walk with Restart; Text summarization; Language Model; Acceptance decision prediction; Meta-review generation

ACM Reference Format:
Chaitanya Bhatia, Tribikram Pradhan, and Sukomal Pal. 2020. MetaGen: An academic Meta-review Generation system. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401190

1 INTRODUCTION

One of the most widely used ways for evaluation of research papers is scholarly peer review (refereeing). It involves multiple experts (peers) to independently review a paper and submit their opinion on the quality and content of the paper and whether it should be accepted in the conference or not. The peers are generally expert researchers in the field of the conference. After the peers have reviewed the paper, the Area/Program Chair (AC/PC) or journal editor writes a meta-review keeping all the peer reviews in consideration. Meta-reviews generally accompany the decision regarding acceptance/rejection; thus, they focus on points which help to justify the decision. With the increasing amount of research work being done, there is a need to speed up the review process, so that more submissions can be processed within the stipulated time.

Research on the working of the peer review system has been limited due to data confidentiality and copyright issues. Over the past few years, conferences have been moving to open review system in which the reviews are posted publicly. Although most of the pros and cons are debatable, such practices lead to the availability of more data for studying the review procedure.

Fine-tuning pre-trained language models has been instrumental in achieving state-of-the-art performance for various low resource Natural Language Processing (NLP) tasks[1]. A lot of recent research work on pre-trained language models has been based on Transformer [2], which makes use of self-attention layers for learning. The release of Bidirectional Encoder Representations from Transformers (BERT)[3] has sparked much research in NLP. BERT has excels at most language understanding tasks but it does not perform well in sequence-to-sequence tasks. Dong et al. released UniLM[4] which addresses this short coming of BERT by introducing a separate training objective for sequence to sequence tasks like text summarization. While training using the sequence-to-sequence objective, the tokens in source segment can attend to each other bidirectionally, while the tokens in target segment can only attend to the tokens to their left(unidirectionally).

Original: This paper proposes and evaluates using graph convolutional networks for semi - supervised learning of probability distributions (histograms) . The paper was reviewed by three experts , all of whom gave a Weak Reject rating . The reviewers acknowledged the strengths of the paper , but also had several important concerns including quality of writing and significance of the contribution , in addition to several more specific technical questions . The authors submitted a response that addressed these concerns to some extent . However , in post - rebuttal discussions , the reviewers chose not to change their ratings , feeling that quality of writing still needed to be improved and that overall a significant revision and another round of peer review would be needed . In light of these reviews , we are not able to recommend accepting the paper , but hope the authors will find the suggestions of the reviewers helpful in preparing a revision for another venue .

Decision: Reject
Generated: This paper proposes and evaluates a graph convolutional network for semi - supervised learning of a hypergraph . The paper received three reviews from experts working in this area . The main concerns of the reviewers were the quality of writing in the paper , the significance of the contribution , and the lack of ablation studies . The authors ' rebuttal addressed some of these concerns . However , there is still a concern that the technical contribution of the paper is incremental . In particular , one reviewer commented that the writing quality could be significantly improved .

Figure 1: Example of original and generated meta-review

Ensuring the correctness of acceptance decision is crucial for a conference/journal. Rejection of a good paper discourages future research work and is not suitable for the research community. A good

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401190

meta-review must stress on the points which led to the decision so that the authors can improve their work.

In this work, we propose **MetaGen**, a novel academic meta-review generation system which takes the peer reviews as input and generates the final meta-review. To the best of our knowledge, this is the first attempt at solving this problem. Our approach has three steps: (i) making an extractive draft, (ii) using fine-tuned UniLM[4] to predict the acceptance decision of a paper and (iii) generating final meta-review of the paper using UniLM. The fine-tuning and evaluation are done using data from previous years' ICLR conference available on OpenReview website[1]. We also compare MetaGen with some of the existing text summarization methods.

To summarize our contributions in this paper:

- We propose MetaGen, a system for generating an assistive meta-review and predicting acceptance of a paper from given academic peer reviews.
- We compare our system with some of the existing text summarization approaches on ICLR dataset.

2 ARCHITECTURE OF METAGEN

MetaGen can be divided into three steps:

- Draft Generation
- Acceptance Decision Prediction
- Review Generation

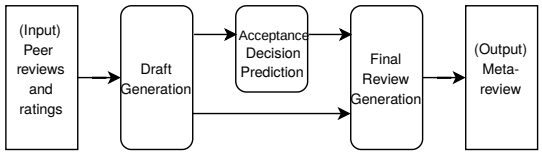

Figure 2: Architecture of MetaGen

2.1 Draft Generation

This step generates a compact representation containing the important points in the review, which will serve as input for decision prediction and final review generation. This step uses extractive text summarization techniques to generate the draft review. The main steps involved in draft generation are:

2.1.1 Input Pre-processing. We process the input reviews and split them into sentences. We use coreference analysis to identify consecutive sentences which refer to the same entity using different words/names. Such sentences are then combined into one. We use the neuralcoref package of SpaCy[2] to perform coreference analysis. We also combine sentences based on the occurrence of sentence connectors (e.g. Also, Hence etc.). This helps to make sure that the points are coherent and self-contained as much as possible.

We then calculate the term frequency-inverse sentence frequency (tf-isf) for all the terms. tf-isf can be interpreted as term frequency-inverse document frequency (tf-idf) at sentence level. The tf-isf values are used for generating sentence vectors using Bag of Words model. We also cluster the sentences(vectors) to make the draft review less redundant. We select the longest sentence as the representative of the cluster. With each review sentence, we also attach the tag [REV_i] (where i denotes reviewer number) denoting which reviewer wrote that comment. This is done to track reviewers and their respective reviews. For cluster representatives, we add tags for all the reviewers whose sentences are a part of that cluster.

2.1.2 Random Walk with Restart (RWR). In this step, we generate the sentence graph for RWR[5]. The weights in the graph are decided based on the cosine similarity of sentence vectors. Additional weight is added to edges pointing to sentences which have meta-review specific terms or contain updates to reviews post the rebuttal. Such sentences are identified based on the occurrence of specific words. This is done to bias the random walk to include more useful sentences. The meta-review specific terms were decided based on frequency analysis of the training dataset. The final edge weight is given by:

$$Weight(S_i, S_j) = cos(S_i, S_j) + \alpha.count(SpecificTerms(S_j)) \quad (1)$$

where S_i and S_j are the start and end nodes of the edge respectively. α is a tuning parameter $\in (0, 1)$. After the graph is generated RWR is performed, and the top-k sentences are selected. In our experiments, we set the value of k to 16. We also add the aspect scores given by the reviewers to the draft review. The aspect scores from the reviewers are preceded by the tokens [SCR_(i)] with i denoting reviewer number.

Generating a draft review helps to represent important points in short, which reduces the memory requirement and the time taken for training. The maximum average ROUGE-1 recall, for the dataset, possible by concatenating all the peer reviews for a meta-review is equal to 0.52. The draft summary generated by our method has an average ROUGE-1 recall of 0.43. The size of the draft summary was selected to maximize the input given without causing memory overflow on the test GPU.

2.2 Acceptance Decision Prediction

The draft review is used to predict the decision regarding acceptance of the paper. For acceptance decision prediction and review generation, we use UniLM[4]. We fine-tune UniLM to predict one of the labels {accept|reject}. For training, the actual acceptance decisions of the papers from ICLR dataset are used. The predicted label/decision is added to the draft review for the next step to provide extra context for generating meta-review.

2.3 Review Generation

This step generates the final meta-review using the acceptance decision and draft review. We fine-tune UniLM for sequence-to-sequence task to make it learn to predict the meta-reviews of the papers. We filter out those papers from the dataset whose original meta-reviews had less than 50 tokens so that the model does not generate very short meta-reviews. From the original meta-reviews, for training, we also filter out the sentences containing the decision regarding acceptance/rejection. Since the number of tokens that can be predicted in the output is limited, this helps to increase the coverage of data from which the system can learn.

The fine-tuning for review generation and acceptance decision prediction is done separately. The decision in training data corresponds to the actual acceptance decision. While generating the meta-review for evaluation, the predicted decision is used. The

[1] www.openreview.net
[2] https://spacy.io/universe/project/neuralcoref/

meta-review generated in this step can be used directly or with minimal changes as seen fit by the AC/PC/editor.

3 EXPERIMENTS

In this section, we give the implementation details, describe the dataset and analyze our results.

3.1 Implementation Details

In the draft review generation process, the restart probability for RWR was kept equal to 0.15 and α was set equal to 0.2. For UniLM, we use the official implementation provided by the authors [3]. We fine-tune UniLM separately for decision prediction and final review generation for 80 epochs with a batch size of 16. We select the model with the lowest loss on validation dataset for comparison. The training was done on a Volta V100 GPU with 16GB memory. Gradient accumulation was done on each step, to avoid memory overflow, making the effective batch size equal to 1. The beam size was set to 5 for decoding. The maximum size of the input sequence was kept equal to 568 tokens while the maximum target length was kept equal to 120 tokens. The size was selected to account for the output tokens wasted on sentence separator. After removal of separator tokens, the average length of the generated meta-reviews was around 100 tokens while the longest meta-review generated was 114 tokens long. Our source code is available on github[4].

3.2 Dataset Description

We conduct cross-year experiments on data from ICLR 2017-2020. The data for ICLR 2018, 2019 and 2020 was scraped from OpenReview website. While the ICLR 2017 data was used as provided in PeerRead dataset[6]. The dataset is a collection of json files with each file having the meta-review, acceptance decision, peer reviews, abstract and the aspect scores given by each reviewer. For training and evaluation, all those papers were removed from the dataset which were withdrawn by the authors or whose meta-reviews were shorter than 50 tokens.

Year	No. of Papers	Avg. length	Acc. rate
2017	365	340	0.38
2018	583	85	0.36
2019	1124	123	0.36
2020	1761	164	0.25

Table 1: Details of ICLR dataset (post-filtering)

The details of the datasets after filtering are given in Table 1. We conduct cross-year experiments by keeping 365 samples randomly from a year at a time for testing and using the remaining for training. 200 samples were selected randomly irrespective of the year and were used for validation.

3.3 Evaluation metrics

For evaluation of the acceptance decision we use accuracy, while for the final meta-review we use ROUGE-1, ROUGE-2 and ROUGE-L scores [7]. We report the F1 score for all the ROUGE metrics.

[3] https://github.com/microsoft/unilm
[4] https://github.com/cb1711/MetaGen

3.4 Experimental results

3.4.1 ***Acceptance prediction.*** Here, we present the accuracy of our approach in predicting decision on ICLR different years' data. We also compare the accuracy achieved without using the scores provided by the reviewers. We compare MetaGen with the following approaches on ICLR2017 data from PeerRead dataset.

(1) **DeepSentiPeer:** DeepSentiPeer[8] uses a sentiment aware deep neural architecture to predict the acceptance decision and aspect score for the paper. We show comparison with the variants using review+sentiment and paper+review+sentiment as input.
(2) **Baseline (Kang et al.)[6]:** It uses hand-engineered features for binary classification of paper.

Method	Accuracy
Baseline	55.26
DeepSentiPeer(Review+Sentiment)	69.79
DeepSentiPeer(Paper+Review+Sentiment)	71.05
MetaGen(Without Aspect Score)	**76.31**
MetaGen	**76.31**

Table 2: Decision prediction accuracy comparison

Table 2 shows the comparison of MetaGen with the approaches mentioned above. All the models were trained using data from ICLR2017 and ICLR2018 and tested on the test split of ICLR2017 data provided in PeerRead dataset. MetaGen outperforms the other approaches for acceptance prediction. Here, due to the small amount of training data available, the model cannot properly learn the significance of the aspect score in decision prediction and thus, we observe the same accuracy as without aspect score.

Year	With Aspect score	Without Aspect score
2017	93.42	76.16
2018	83.56	72.60
2019	89.04	73.69
2020	85.75	76.16

Table 3: Accuracy of decision prediction in cross year experiments

Table 3 shows the accuracy of our approach in cross year experiments. The data is split as described in Section 3.2. Here, the impact of using aspect score for predicting the decision is much clearly visible. Across all the datasets, there is an increase of around 10% in accuracy when using reviewer ratings.

3.4.2 ***Meta-review generation.*** On fine-tuning UniLM learns to generate sentences as though the AC had written them. This factor can be controlled by modifying the training data slightly to make the model learn that it is generating the meta-review as the system and not the actual AC.

Since no work has been done in this direction, our system can't be compared directly with any meta-review generation methods. However given the similarity of the tasks we compare our approach with the following text summarization methods:

(1) **LexRank:** LexRank[9] is an extractive baseline which uses PageRank on sentence graph to select the sentences. We use

the implementation provided by Sumy[5] package in Python for experimentation.

(2) **KLSum:** KLSum [10] is an extractive baseline which finds the summary which has minimum Kullback-Liebler divergence with the overall given document to be summarized. We use the implementation provided by Sumy package in Python for experimentation.

(3) **Pointer-Generator Network:** Pointer-Generator Network [11] is an abstractive approach which adds a pointing mechanism to enable the network to copy out-of-vocabulary words directly from the source. We use the implementation[6] provided by authors for comparison and keep the maximum and minimum output lengths as 120 and 115 tokens respectively. The model is trained and tested on the same set of papers as MetaGen. We first train the model till convergence without coverage mechanism and then train it for some steps with coverage mechanism (for reducing redundancy).

For comparison, we truncate the output of each of the above approaches to 115 tokens. The peer reviews are concatenated and then given as input to LexRank, KLSum and Pointer-Generator Network.

Method	**2017**	**2018**	**2019**	**2020**
LexRank	0.29	0.31	0.31	0.33
KLSum	0.25	0.28	0.29	0.29
Pointer Generator	0.17	0.21	0.22	0.23
MetaGen	0.29	0.33	0.34	0.36

Table 4: Comparison of ROUGE-1 scores

Method	**2017**	**2018**	**2019**	**2020**
LexRank	0.07	0.05	0.06	0.07
KLSum	0.07	0.05	0.05	0.06
Pointer Generator	0.03	0.04	0.05	0.05
MetaGen	0.08	0.07	0.08	0.10

Table 5: Comparison of ROUGE-2 scores

Method	**2017**	**2018**	**2019**	**2020**
LexRank	0.24	0.23	0.27	0.24
KLSum	0.22	0.21	0.26	0.23
Pointer Generator	0.15	0.16	0.20	0.18
MetaGen	0.25	0.25	0.31	0.28

Table 6: Comparison of ROUGE-L scores

Tables 4, 5 and 6 show the comparison of ROUGE-1, 2 and L scores respectively. The average ROUGE scores for all the approaches are calculated on the same set of randomly chosen papers for each year.

3.5 Discussion

MetaGen outperforms the text summarization approaches in the task of meta-review generation. The lower results on 2017 data can be attributed to the higher average length of meta-reviews in 2017, due to which the recall is lower. In contrast with the extractive methods, the generated meta-reviews seem to be written by a human and are more coherent. Compared to pointer generator network, less sentences are exactly copied and repeated. Since the number of training samples is small and pointer generator network does not rely on pretrained language model, its performance suffers greatly. Despite training with coverage mechanism, in most of the cases, pointer generator network repeated sentences. As a result, pointer generator network gets the lowest ROUGE score out of the compared methods. Despite being an extractive method, LexRank performs well in most of the cases. However, the generated meta-reviews are much less coherent. In most cases, MetaGen is able to identify if the concerns have been addressed properly or not.

[5]https://pypi.org/project/sumy/

[6]https://github.com/abisee/pointer-generator

4 CONCLUSION AND FUTURE WORK

In this paper, we presented MetaGen: an academic meta-review generation system which generates meta-review using a three-step process. Our approach relies on language model fine-tuning to generate an abstractive meta-review and predict the acceptance decision. We also compare it with some existing text summarization techniques on ICLR dataset. Our work can make the review process faster and thus encourage more submissions.

MetaGen relies on the assumption that the reviewers update their reviews once the authors address their comments. However, this assumption may not always be true. A possible future direction is including authors' comments as an input to find if the reviewers' concerns have been resolved. The efficiency of the approach will also increase with more data and improvements in language models. Work can also be done to model meta-review as a set of answers to questions and structuring the meta-review in that way.

REFERENCES

[1] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *CoRR*, abs/1905.03197, 2019.

[5] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[6] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and NLP applications. *CoRR*, abs/1804.09635, 2018.

[7] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[8] Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, and Pushpak Bhattacharyya. DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1120–1130, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[10] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[11] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368, 2017.