PeerSum: A Peer Review Dataset for Abstractive Multi-document Summarization

Anonymous ACL submission

Abstract

We present PEERSUM, a new MDS dataset using peer reviews of scientific publications. Our dataset differs from the existing MDS datasets in that our summaries (i.e., the meta-reviews) are highly abstractive and they are real summaries of the source documents (i.e., the reviews) and it also features *disagreements* among source documents. We found that current state-of-the-art MDS models struggle to generate high-quality summaries for PEERSUM, offering new research opportunities.

1 Introduction

Abstractive multi-document summarization (MDS) aims to generate a description that summarizes the salient information in a cluster of topically related documents (a.k.a. the source documents). It resembles how humans summarize/synthesize information and has many applications such as summarizing news, emails, medical and legal documents (Ma et al., 2020). It is a challenging task as it requires aggregating information and paraphrasing words/sentences from the source documents.

There are only a handful of large scale MDS datasets, e.g., WCEP (Ghalandari et al., 2020), Multi-News (Fabbri et al., 2019), Multi-XScience (Lu et al., 2020), and WikiSum (Liu et al., 2018), and most of them have the following limitations: (1) the summaries are not *real summaries* of the source documents; (2) the summaries are highly extractive; and (3) the source documents are loosely connected and do not feature complex relations such as conflicts. In contrast, the reviews in PEERSUM have disagreements and the meta-reviews need to address this conflict, making the summarization task challenging.

In this paper, we develop and release PEERSUM, a novel MDS dataset using peer reviews of scientific publications. In PEERSUM, each review of a paper forms a source document, and the meta-review constitutes the ground truth summary.[1] To the best of our knowledge, PEERSUM is the first MDS dataset in the peer review domain. Although PeerRead (Kang et al., 2018) is also a peer review dataset, it does not utilise meta-reviews and thus is not built for summarization. PEERSUM contains 10,862 summaries currently, and we envisage its size to grow continuously with more reviews published over time. The dataset and code are available at: ANONYMISED.

Features	ICLR	NeurIPS
#papers	7,543	3,319
#reviews/cluster	3.35	3.50
#comments/cluster	0.36	–
#responses/cluster	5.90	–
Avg. confidence	3.74	–
Avg. rating	5.20	–

Table 1: PEERSUM statistics.

PEERSUM has several distinctive features over existing MDS datasets:

- The summaries (meta-reviews) are a real summary of the source documents (reviews).
- The summaries are highly abstractive and source documents are relevant to the summary.
- Source documents are topically closely connected but they also feature disagreements, which makes PEERSUM a challenging dataset.
- PEERSUM has additional information such as discussions and acceptance outcome and two scores associated with each review: *rating* (which reflects a reviewer's overall sentiment towards the paper) and *confidence* (which reflects a reviewer's confidence in their judgement of the paper and the degree of influence in shaping the meta-review).
- The average rating (aggregated over reviews) and acceptance outcome can be used as an al-

[1] Henceforth we use {review, source document} and {meta-review, summary} interchangeably.

Metric	PEERSUM-R	PEERSUM-RC	PEERSUM-ALL	WikiSum	Multi-News	WCEP	Multi-XScience
domain	Peer review	Peer review	Peer review	Wikipedia	News	News	Academic
#Clusters	10,862	10,862	10,862	1,655,709	56,216	10,200	40,528
#Doc/Cluster	3.40	3.65	7.75	40	2.75	63.55	4.42
#Sentence/Doc	24.25	23.70	22.33	2.92	36.98	19.47	7.08
#Token/Doc	267.90	261.72	246.28	34.18	444.74	269.98	107.73
#Sentence/Summary	6.81	6.81	6.81	4.99	11.18	1.42	4.84
#Token/Summary	79.45	79.45	79.45	75.62	139.15	19.95	71.76

Table 2: Statistics of PEERSUM and other MDS datasets.

ternative evaluation for assessing the quality of generated summaries.

2 Related Work

There are several MDS datasets: WCEP (Ghalandari et al., 2020), Multi-News (Fabbri et al., 2019), Multi-XScience (Lu et al., 2020), and Wikisum (Liu et al., 2018). WCEP and Multi-News are from the news domain; the summaries are human-written and the source documents are news articles (WCEP source documents are potentially noisy since it includes similar articles retrieved from the web). Multi-XScience uses the related work of a publication as the summary and its cited publications as source documents. Wikisum follows a similar idea and uses a Wikipedia article as the summary and its cited web articles as source documents. The summaries of Multi-XScience and Wikisum are not genuine summaries as the source documents may not always be relevant.

3 PEERSUM

We scrape review data for two top-tier international conferences: ICLR[2] and NeurIPS[3]. For ICLR, there are three types of source documents: (1) reviews (written by assigned reviewers); (2) comments ("unofficial reviews" written by the public); and (3) responses (either by the publication authors or reviewers).[4] We present an illustration how these document types are extracted for ICLR in the Appendix. Note also that we have an acceptance outcome associated with the meta-review, and two scores for each review: rating and confidence, which denote a reviewer's sentiment and confidence of their judgement, respectively.

For ICLR, each research paper forms a cluster, where we have reviews (with scores), comments and responses as the source documents and the meta-review (with an acceptance outcome) as the ground truth summary. For NeurIPS, we only have reviews and meta-reviews and no scores. In total, we have 10,862 clusters for ICLR 2017–2021 and NeurIPS 2019–2020; Table 1 presents some summary statistics. To understand the impact of different types of source documents for producing the summary, we construct three variants of PEERSUM: (1) PEERSUM-R, which contains only the reviews; (2) PEERSUM-RC, which includes both reviews and comments; and (3) PEERSUM-ALL, which contains everything: reviews, comments and responses. We release scripts to crawl and process more reviews for both ICLR and NeurIPS as they become available, allowing the PEERSUM to grow over time.

4 Comparison with Other MDS Datasets

We compare PEERSUM with four existing MDS datasets: WikiSum (Liu et al., 2018), Multi-News (Fabbri et al., 2019), WCEP (Ghalandari et al., 2020), and Multi-XScience (Lu et al., 2020), in terms of summary abstractiveness, document relevance, and cross-document relationships. Specifically, we use the WCEP-100 version of WCEP and the WikiSum dataset collected by Liu and Lapata (2019a). We perform tokenization, lemmatization, and stopwords removal using Spacy[5] for all these datasets, and summarize their statistics in Table 2. Note that relevance (Section 4.2) and cross-document analysis (Section 4.3) are done using tokenized data without lemmatization and removing stopwords.

4.1 Abstractiveness of Summaries

Table 3 reports the percentage of unigrams, bigrams, and trigrams in the summaries which are not

[2] https://openreview.net/

[3] https://proceedings.neurips.cc/

[4] These responses are either discussions following a review or comment, or they form a new thread themselves (e.g., when authors gave a general response). In other words, responses have a thread-like structure akin to discussion forums.

[5] https://spacy.io/

Dataset	Unigram	Bigram	Trigram	Relevance
PeerSum-R	35.34	80.29	90.92	0.429
PeerSum-RC	35.10	80.17	90.87	0.424
PeerSum-All	28.65	77.67	90.31	0.415
WikiSum	22.75	63.55	79.34	0.265
Multi-News	23.49	66.10	82.01	0.398
WCEP	5.25	37.62	65.27	0.246
Multi-XScience	44.09	86.54	96.40	0.343

Table 3: Percentage of novel unigrams, bigrams, and trigrams (%) in summary and relevance scores.

Datasets	Rouge-L
PeerSum-R	30.50
PeerSum-RC	29.28
PeerSum-All	27.41
WikiSum	13.87
Multi-News	26.54
WCEP	16.79
Multi-XScience	18.96

Table 4: Similarity between source documents.

Paper 1	The setting of the main experiment is not valid.
	They evaluated their method with baselines in the aspect of the number of parameters and training speed and performance for various memory sizes and reported plentiful ablation study results too.
Paper 2	Introduction section is not well-written.
	This paper is well written and looks correct.

Table 5: Examples of disagreements.

found in the corresponding source documents. Intuitively, higher proportion of novel n-grams make a summary more abstractive (Fabbri et al., 2019; Ghalandari et al., 2020) and hence more difficult to generate. We see that all PeerSum variants' summaries are highly abstractive (although it is lower than that of Multi-XScience).

4.2 Relevance Between Summaries and Source Documents

Most MDS datasets' summaries are not real summaries of the source documents (e.g., WCEP, Multi-XScience and Wikisum). To assess whether the source documents are relevant to producing the summaries, we use SUPERT (Gao et al., 2020), which uses an unsupervised method to extract salient sentences from source documents and measures the relevance between the ground truth summary and the extracted summary sentences using BERT: a high relevance implies that the ground truth summary is of high quality and encapsulates key points from the documents.

Relevance scores are presented in Table 3 (4th column). We see that PeerSum datasets have the highest relevance scores, indicating that they are high quality summaries for the source documents. Taking both the abstractiveness results (Section 4.1) and these relevance scores together, this means that PeerSum summaries are highly abstractive and source documents are relevant to the summaries. In contrast, although Multi-XScience's summaries are very abstractive, they have low relevance to the source documents since the related work of a publication is not a real summary of the cited publications.

4.3 Cross-document Relationships

To understand the quality of the source documents, particularly on how topically related they are with each other, we calculate ROUGE-L between source documents of a cluster and then take the mean over all clusters for an MDS dataset. We present the results in Table 4. We see that PeerSum datasets have the highest inter-document similarity, indicating that they are topically closely related to each other. Unsurprisingly, WikiSum, WCEP and Multi-XScience have the lowest similarity scores because their source documents are either cited references or retrieved web articles.

Even though reviews of PeerSum have high similarity with each other, it also features *disagreements*. Approximately 9% ICLR publications have at least a pair of reviews that have a rating difference ≥ 5 (noting review ratings range from 1 (reject) to 10 (seminal paper)), indicating disagreements that the meta-reviews need to resolve, which makes the summarization task challenging. We present several examples of conflicting reviews in Table 5.

5 Experimental Study

We next test a suite of state-of-the-art single-document and multi-document abstractive summarization models on PeerSum and other MDS datasets, to understand if PeerSum constitutes an interesting summarization challenge. To incorporate comments and responses for PeerSum-RC and PeerSum-All, we simply treat them as additional source documents.

For single-document summarization models, we use **BertSum** (Liu and Lapata, 2019b) and **PEGASUS** (Zhang et al., 2020a), and we merge all documents into a single large document as input. For

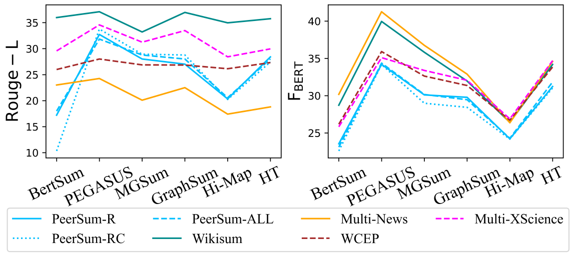


Figure 1: Summarization performance of summarization models over MDS datasets.

	BS	PGS	MG	GS	HM	HT
PEERSUM-R	1.04	0.76	0.99	0.90	1.13	0.79
PEERSUM-RC	1.03	0.79	1.02	0.91	1.15	0.81
PEERSUM-ALL	0.99	0.75	1.01	0.88	1.18	0.75
PEERSUM-R	77.5	87.6	84.4	84.9	75.2	86.3
PEERSUM-RC	77.1	87.2	84.3	84.6	75.1	86.1
PEERSUM-ALL	77.7	87.6	85.1	84.7	75.1	86.3

Table 6: Average rating regression performance (first 3 rows; lower = better); accept/reject classification performance (last 3 rows; higher = better). BS=BertSum, PGS=PEGASUS, MG=MGSum, GS=GraphSum, HM=Hi-MAP.

multi-document summarization models, we use: **MGSum** (Jin et al., 2020), **GraphSum** (Li et al., 2020), **Hi-MAP** (Fabbri et al., 2019), and **HT** (Liu and Lapata, 2019a). We run these models using the default recommended hyper-parameter settings.

5.1 Reference-based Evaluation

We measure the performance using standard ROUGE-L (Lin, 2004) which assesses the lexical overlap between generated summaries and ground truth summaries. Following Koto et al. (2020), we also measure content overlap using BERTScore (Zhang et al., 2020b).[6] As Figure 1 shows, PEERSUM and its variants appear to be the hardest datasets based on BERTScore (admittedly this trend is less strong with ROUGE-L). Interestingly, the MDS models (MGSum, GraphSum, Hi-MAP and HT) do not have an upper hand compared to the single-document counterparts (BERTSum and PEGASUS) — particularly when using BERTScore as the evaluation metric — even though they are designed to model cross-document relationships. PEGASUS appears to be the best model, and we suspect its strong performance is due to its summarization-tailored pretraining objective and large-scale pretraining data.

5.2 Alternative Evaluation

As PEERSUM contains ratings for each review, the average rating (over all reviews) reflects the overall sentiment, which is likely to capture the tone of the meta-review (summary). In addition to that we also have an accept/reject outcome with the meta-review.[7] We therefore propose an alternative evaluation strategy by first training two models: (1) a rating regression model; and (2) an outcome binary classifier; using the meta-reviews as input and the average ratings/outcomes as labels, and then use these two models to score/classify generated summaries. The closer a system's summary scores are to the average ratings or acceptance outcomes, the better the model (as it implies that the generated summaries are similar to the meta-reviews). We fine-tune BERT-base to build the regression and classification models.

Table 6 presents the mean-squared error ("MSE"; first 3 rows, lower is better) and accuracy ("ACC"; last 3 rows, higher is better) for different systems. We see that PEGASUS again yields the best performance, consistent with our previous finding (Section 5.1). Interestingly, including comments (PEERSUM-R vs. PEERSUM-RC) appears to hurt the summarization systems, suggesting that they are not helpful for producing the meta-review. This means that meta-reviewers by and large use only the official anonymous reviews when judging a paper, which is the recommended practice. That said, we do see benefits of including everything — reviews, comments and responses. A further question remains how to best incorporate them, particularly for responses which have a conversation structure.

6 Conclusion

We developed PEERSUM, an MDS dataset based on peer reviews. We analyzed PEERSUM and found that its summaries are of high quality and very abstractive. We test a suite of summarization models and found that PEERSUM constitutes a challenging dataset. PEERSUM is also unique in that it contains documents with conversation structure (the responses) and additional metadata such as review ratings (which allows for an alternative evaluation).

[6] Koto et al. (2020) use the precision and recall metrics of BERTScore to measure focus and coverage; here we combine both by computing the F1 score.

[7] There are technically 4 classes: oral, spotlight, poster and reject, but the first 3 are different presentations for accepted papers, and so we collapse them into the 'accept' class.

References

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *ACL*, pages 1074–1084.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *ACL*, pages 1347–1354.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the wikipedia current events portal. In *ACL*, pages 1302–1308.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *ACL*, pages 6244–6254.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and NLP applications. In *NAACL-HLT*, pages 1647–1661.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2020. FFCI: A framework for interpretable automatic evaluation of summarization. *CoRR*, abs/2011.13662.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *ACL*, pages 6232–6243.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *ICLR*.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *ACL*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *EMNLP*, pages 3728–3738.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *EMNLP*, pages 8068–8074.

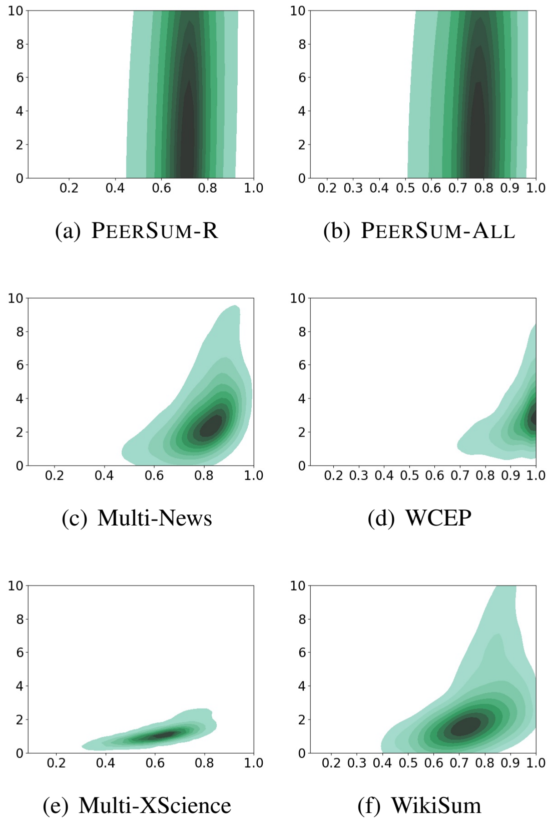
Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z. Sheng. 2020. Multi-document summarization via deep learning techniques: A survey. *CoRR*, abs/2011.04843.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *ICML*, pages 11328–11339.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *ICLR*.

A Appendix: Abstractiveness in terms of Coverage and Density

We further measure the abstractiveness of the summaries with the Coverage and Density following Multi-News. Coverage measures the proportion of a summary that comes from extracted chunks of the source documents, and Density measures the average length of these extracted chunks. A lower Coverage indicates higher abstractiveness of the summary. Following Multi-News and WCEP, we plot projections of kernel density estimations for the value of Coverage and Density of the four datasets in Fig. 2. It is obvious that in PeerSum there are fewer extracted chunks. The length of extracted chunks is a little larger because meta-reviewers usually copy sentences from reviews.



(a) PeerSum-R

(b) PeerSum-All

(c) Multi-News

(d) WCEP

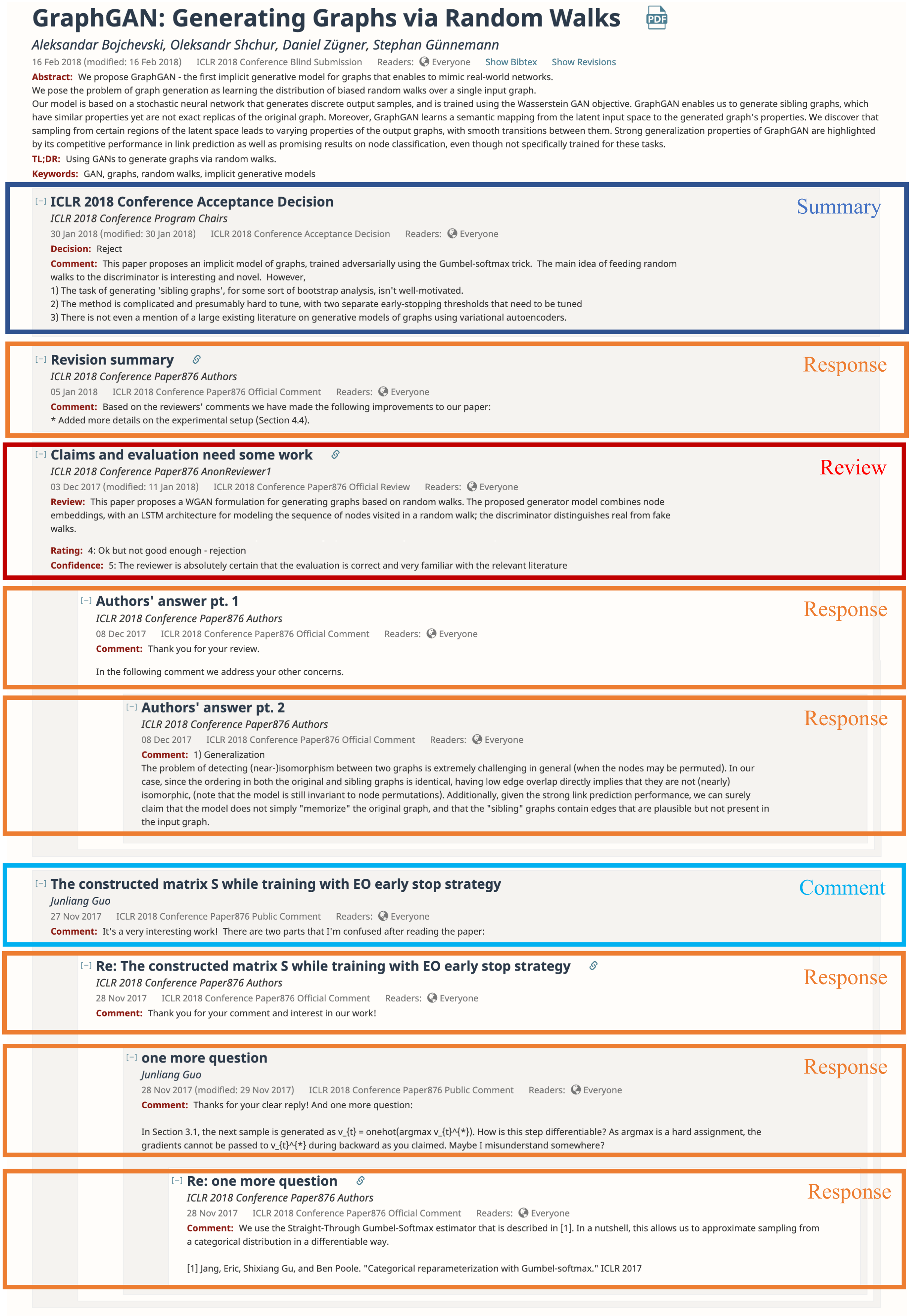
(e) Multi-XScience

(f) WikiSum

Figure 2: Projections of kernel density estimation for Coverage (x-axis) and Density (y-axis).

B Appendix: Different categories of documents in PeerSum

Figure 3 shows an example cluster for a paper in PeerSum with annotated documents. It is obvious that there are four categories of documents in PeerSum: summaries, reviews, comments, and responses. These documents are extracted from https://openreview.net/forum?id=H15RufWAW. Note that we cropped some bits to save some space.



GraphGAN: Generating Graphs via Random Walks

Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, Stephan Günnemann

16 Feb 2018 (modified: 16 Feb 2018) ICLR 2018 Conference Blind Submission Readers: Everyone Show Bibtex Show Revisions

Abstract: We propose GraphGAN - the first implicit generative model for graphs that enables to mimic real-world networks.
We pose the problem of graph generation as learning the distribution of biased random walks over a single input graph.
Our model is based on a stochastic neural network that generates discrete output samples, and is trained using the Wasserstein GAN objective. GraphGAN enables us to generate sibling graphs, which have similar properties yet are not exact replicas of the original graph. Moreover, GraphGAN learns a semantic mapping from the latent input space to the generated graph's properties. We discover that sampling from certain regions of the latent space leads to varying properties of the output graphs, with smooth transitions between them. Strong generalization properties of GraphGAN are highlighted by its competitive performance in link prediction as well as promising results on node classification, even though not specifically trained for these tasks.

TL;DR: Using GANs to generate graphs via random walks.

Keywords: GAN, graphs, random walks, implicit generative models

[−] **ICLR 2018 Conference Acceptance Decision**

ICLR 2018 Conference Program Chairs

30 Jan 2018 (modified: 30 Jan 2018) ICLR 2018 Conference Acceptance Decision Readers: Everyone

Decision: Reject

Comment: This paper proposes an implicit model of graphs, trained adversarially using the Gumbel-softmax trick. The main idea of feeding random walks to the discriminator is interesting and novel. However,
1) The task of generating 'sibling graphs', for some sort of bootstrap analysis, isn't well-motivated.
2) The method is complicated and presumably hard to tune, with two separate early-stopping thresholds that need to be tuned
3) There is not even a mention of a large existing literature on generative models of graphs using variational autoencoders.

[−] **Revision summary**

ICLR 2018 Conference Paper876 Authors

05 Jan 2018 ICLR 2018 Conference Paper876 Official Comment Readers: Everyone

Comment: Based on the reviewers' comments we have made the following improvements to our paper:
* Added more details on the experimental setup (Section 4.4).

[−] **Claims and evaluation need some work**

ICLR 2018 Conference Paper876 AnonReviewer1

03 Dec 2017 (modified: 11 Jan 2018) ICLR 2018 Conference Paper876 Official Review Readers: Everyone

Review: This paper proposes a WGAN formulation for generating graphs based on random walks. The proposed generator model combines node embeddings, with an LSTM architecture for modeling the sequence of nodes visited in a random walk; the discriminator distinguishes real from fake walks.

Rating: 4: Ok but not good enough - rejection

Confidence: 5: The reviewer is absolutely certain that the evaluation is correct and very familiar with the relevant literature

[−] **Authors' answer pt. 1**

ICLR 2018 Conference Paper876 Authors

08 Dec 2017 ICLR 2018 Conference Paper876 Official Comment Readers: Everyone

Comment: Thank you for your review.

In the following comment we address your other concerns.

[−] **Authors' answer pt. 2**

ICLR 2018 Conference Paper876 Authors

08 Dec 2017 ICLR 2018 Conference Paper876 Official Comment Readers: Everyone

Comment: 1) Generalization
The problem of detecting (near-)isomorphism between two graphs is extremely challenging in general (when the nodes may be permuted). In our case, since the ordering in both the original and sibling graphs is identical, having low edge overlap directly implies that they are not (nearly) isomorphic, (note that the model is still invariant to node permutations). Additionally, given the strong link prediction performance, we can surely claim that the model does not simply "memorize" the original graph, and that the "sibling" graphs contain edges that are plausible but not present in the input graph.

[−] **The constructed matrix S while training with EO early stop strategy**

Junliang Guo

27 Nov 2017 ICLR 2018 Conference Paper876 Public Comment Readers: Everyone

Comment: It's a very interesting work! There are two parts that I'm confused after reading the paper:

[−] **Re: The constructed matrix S while training with EO early stop strategy**

ICLR 2018 Conference Paper876 Authors

28 Nov 2017 ICLR 2018 Conference Paper876 Official Comment Readers: Everyone

Comment: Thank you for your comment and interest in our work!

[−] **one more question**

Junliang Guo

28 Nov 2017 (modified: 29 Nov 2017) ICLR 2018 Conference Paper876 Public Comment Readers: Everyone

Comment: Thanks for your clear reply! And one more question:

In Section 3.1, the next sample is generated as v_{t} = onehot(argmax v_{t}^{*}). How is this step differentiable? As argmax is a hard assignment, the gradients cannot be passed to v_{t}^{*} during backward as you claimed. Maybe I misunderstand somewhere?

[−] **Re: one more question**

ICLR 2018 Conference Paper876 Authors

28 Nov 2017 ICLR 2018 Conference Paper876 Official Comment Readers: Everyone

Comment: We use the Straight-Through Gumbel-Softmax estimator that is described in [1]. In a nutshell, this allows us to approximate sampling from a categorical distribution in a differentiable way.

[1] Jang, Eric, Shixiang Gu, and Ben Poole. "Categorical reparameterization with Gumbel-softmax." ICLR 2017

Figure 3: A document cluster for a paper in PEERSUM