Can We Automate Scientific Reviewing?

Weizhe Yuan, Pengfei Liu, Graham Neubig,
Carnegie Mellon University
{weizhey,pliu3,gneubig}@cs.cmu.edu

Abstract

The rapid development of science and technology has been accompanied by an exponential growth in peer-reviewed scientific publications. At the same time, the review of each paper is a laborious process that must be carried out by subject matter experts, and thus providing high-quality reviews of this growing number of papers is a significant challenge. In this work, we ask the question "can we automate scientific reviewing?", discussing the possibility of using state-of-the-art natural language processing (NLP) models to generate first-pass reviews. Arguably the most difficult part of this is defining what a "good" review is in the first place, so we first discuss possible evaluation measures for such reviews. We then collect a dataset of papers in the machine learning domain, annotate them with annotations of different aspects of each review, and train targeted summarization models that take in the paper and generate a review. Comprehensive experimental results show that system-generated reviews tend to touch upon more aspects of the paper than human-written reviews, but the generated text can suffer from lower factuality for all aspects except the explanation of the core ideas of the papers, which are largely factually correct. We finally summarize *eight* challenges in the pursuit of a good review generation system together with potential solutions, which, hopefully, will inspire more future research on this subject. We make all codes, dataset publicly available: https://github.com/neulab/ReviewAdvisor as well as a *ReviewAdvisor* system: http://review.nlpedia.ai/.

1 Introduction

The number of published papers is growing exponentially (Tabah, 1999; De Bellis, 2009; Bornmann and Mutz, 2015). While this may be positively viewed as indicating acceleration of scientific progress, it also poses great challenges for researchers, both in reading and synthesizing the relevant literature for one's own benefit, and for performing *peer review* of papers to vet their correctness and merit. With respect to the former, a large body of existing work explores automatic summarization of a paper or a set of papers for automatic survey generation (Jha et al., 2015a, 2013; Mohammad et al., 2009; Xing et al., 2020; Jha et al., 2015b; Yasunaga et al., 2019b; Cohan et al., 2018b). However, despite the fact that peer review is an important, but laborious part of our scientific process, automatic systems to aid in the peer review process remain relatively underexplored. Bartoli et al. (2016) investigated the feasibility of generating reviews by surface-level term replacement and sentence reordering etc, and Wang et al. (2020) (contemporaneously and independently) propose a two-stage information extraction and summarization pipeline to generate paper reviews. However, both do not extensively evaluate the quality or features of the generated review text.

In this work, we are concerned with providing at least a preliminary answer to the ambitious overarching question: *can we automate scientific reviewing?* Given the complexity of understanding and assessing the merit of scientific contributions, we do not expect an automated system to be able to match a well-qualified and meticulous human reviewer at this task any time soon. However, some degree of review automation may assist reviewers in their assessments, or provide guidance to junior reviewers who are just learning the ropes of the reviewing process. Towards this goal, we examine two concrete research questions, knowing the answers to which are prerequisites to building a functioning review assistant:

Q1: What are the desiderata of a good automatic reviewing system, and how can we quantify them for evaluation? Before developing an automatic review system, we first must quantify what constitutes a good review in the first place. The challenge of answering this question is that a review commonly involves both objective (e.g. "lack of details necessary to replicate the experimental protocol") and subjective aspects (e.g. "lack of potential impact").

Due to this subjectivity, defining a "good" review is itself somewhat subjective.

As a step in this direction, we argue that it is possible to view review generation as a task of *opinion-aware summarization*, where the summary not only tries to summarize the core idea of a paper, but also expresses opinions on specific aspects of that paper (e.g. novelty or potential impact). We then evaluate review quality from multiple perspectives, in which we claim a good review not only should make a good summary of a paper (summary accuracy) but also consist of factually correct (recommendation accuracy, summary-level factuality) and fair comments (bias) from diverse aspects (aspect coverage), together with informative evidence (informativeness).

Q2: Using state-of-the-art NLP models, to what extent can we realize these desiderata? We provide an initial answer to this question by building a dataset of reviews from machine learning domain, training state-of-the-art summarization models to generate reviews from scientific papers, and evaluating the output according to our evaluation metrics described above. Specifically, in this work, we not only collect reviews from an existing open peer review system, but also make fine-grained annotations of aspect information for each review, which provides the possibility for a richer evaluation of generated reviews. In terms of system design, we develop a scientific review generation model, which we dub *ReviewAdvisor*, with different architectural designs (§4), and comprehensively evaluate them, interpreting their relative advantages.

Lastly, we highlight our main observations in this paper:

(1) Can our designed review generation system replace human reviewers so far? No. A better position for it could be a machine-assisted system.

(2) What are review generation systems (not) good at? Although the automated review system suffers from lower factuality problem for many aspects, it (i) can precisely summarize the core idea of the input paper, which can be either used as a draft for human reviewers or help them (or general readers) quickly understand the main idea of a reviewed paper (or pre-printed papers). (ii) cover more aspects than human with coherent comments together with relevant evident sentences from the paper, which can provide a preliminary template for reviewers and help them quickly identify salient information.

(3) Will system generate biased reviews? Yes. Systems with different architectural designs will exhibit varying degrees of biases. In this work, we present methods to identify and quantify potential biases in reviews (§6.3). For example, (i) Papers of native English speakers tend to obtain higher scores on "`Clarity`" from human reviewers than non-native English ones, but the system reviewers (our proposed models) will narrow this gap.

(ii) System reviewers will be harsher than human reviewers when giving a score regarding the paper's "`Originality`" for non-native English speakers.

(iii) Both human reviewers and system reviewers favor non-anonymous papers more than anonymous papers in all aspects.

(4) What's next in pursuing a better automatic review generation system? We summarize seven challenges that can be explored for future directions in §8.2.

2 What Makes a Good Peer Review?

As concluded by Jefferson et al. (2002b): "*Until we have properly defined the objectives of peer-review, it will remain almost impossible to assess or improve its effectiveness.*" Therefore we first discuss the possible objectives of peer review.

2.1 Peer Review for Scientific Research

A research paper is commonly first reviewed by several committee members , who should assign one or several *scores* and give detailed comments. The comments, and sometimes scores, cover diverse *aspects* of the paper (e.g. "clarity," "potential impact"; detailed in §3.2.1), and these aspects are often directly mentioned in review forms of scientific conferences or journals.[1] Then a senior reviewer will often make a *final decision* (i.e., "reject" or "accept") as well as comments summarizing the decision (i.e., a *meta-review*).

After going through many review guidelines[2],

[1]For example, one example from ACL can be found at: https://acl2018.org/downloads/acl_2018_review_form.html

[2]https://icml.cc/Conferences/2020/ReviewerGuidelines, https://nips.cc/Conferences/2020/PaperInformation/ReviewerGuidelines, https://iclr.cc/Conferences/2021/ReviewerGuide

resources[3] about how to write a good review, we summarize *some* of the most frequently mentioned desiderata below:

1. **Decisiveness:** A good review should take a clear stance, selecting high-quality submissions for publication and suggesting others not be accepted (Jefferson et al., 2002a; Smith, 2006).

2. **Comprehensiveness:** A good review should be well-organized, typically starting with a brief summary of the paper's contributions, then following with opinions gauging the quality of a paper from different aspects. Many review forms explicitly require the evaluation for different aspects to encourage comprehensiveness.

3. **Justification:** A good review should provide specific reasons for its assessment, particularly whenever it states that the paper is lacking in some aspect. This justification also makes the review more constructive (another oft-cited desiderata of reviews), as these justifications provide hints about how the authors could improve problematic aspects in the paper (Xiong and Litman, 2011).

4. **Accuracy:** A review should be factually correct, with the statements contained therein not being demonstrably false.

5. **Kindness:** A good review should be kind and polite in language use.

Based on above desiderata, we make a first step towards evaluation of reviews for scientific papers and characterize a "good" review from multiple perspectives.

2.2 Multi-Perspective Evaluation

Given input paper D and meta-review R^m, our goal is to evaluate the quality of review R, which can be either manually or automatically generated. We also introduce a function $\textsc{Dec}(D) \in \{1, -1\}$ that indicates the final decision of a given paper reached by the meta-reviewer: "`accept`" or "`reject`".

[3] https://players.brightcove.net/3806881048001/rFXiCa5uY_default/index.html?videoId=4518165477001, https://soundcloud.com/nlp-highlights/77-on-writing-quality-peer-reviews-with-noah-a-smith, https://www.aclweb.org/anthology/2020.acl-tutorials.4.pdf, https://2020.emnlp.org/blog/2020-05-17-write-good-reviews

Further, $\textsc{Rec}(R) \in \{1, 0, -1\}$ represents the acceptance recommendation of a particular review: "`accept`," "`neutral`," or "`reject`. (see Appendix A.7 for details) Below, we discuss automatic evaluation metrics that can be used to approximate the desiderata of reviews described in the previous section.

2.2.1 D1: Decisiveness

First, we tackle the *decisiveness*, as well as accuracy of the review's recommendation, through **recommendation accuracy (RACC)**. Here we use the final decision regarding a paper and measure whether the acceptance implied by the review R is consistent with the actual accept/reject decision of the reviewed paper. It is calculated as:

$$\text{RAcc}(R) = \textsc{Dec}(D) \times \textsc{Rec}(R) \tag{1}$$

A higher score indicates that the review more decisively and accurately makes an acceptance recommendation.

2.2.2 D2: Comprehensiveness

A comprehensive review should touch on the quality of different aspects of the paper, which we measure using a metric dubbed **aspect coverage (ACOV)**. Specifically, given a review R, aspect coverage measures how many aspects (e.g. `clarity`) in a predefined aspect typology have been covered by R.

2.2.3 D3: Justification

As defined in §2.1, a good peer review should provide hints about how the author could improve problematic aspects. For example, when reviewers comment: "this paper lacks important references", they should also list these relevant works. To satisfy this justification desideratum, we define a metric called **informativeness (INFO)** to quantify how many negative comments[4] are accompanied by corresponding evidence.

First, let $n_a(R)$ denote the number of aspects covered in R, $n_{\text{pa}}(R)$ and $n_{\text{na}}(R)$ denote the number of aspects with positive sentiment polarity and aspects with negative sentiment polarity in R. $n_{\text{nae}}(R)$ denotes the number of aspects with negative sentiment polarity that are supported by evidence. The judgement of supporting evidence is conducted manually (details in Appendix A.7).

[4] We only consider whether the reviewer has provided enough evidence on their negative opinion since we find that reviews only lack evidence on the positive comments.

One straightforward method for calculating the justification of negative comments would be $\frac{n_{\text{nae}}(R)}{n_a(R)}$. However, since aspects with positive sentiment polarity usually are not accompanied by evidence, and high-quality papers will get positive comments much of the time, we regularize this measure by using Eq. 2 to avoid 0 informativeness for high-quality papers.

$$\text{Info}(R) = \frac{n_{\text{pa}}(R)}{n_a(R)} \times \text{DEC}(D) + \frac{n_{\text{nae}}(R)}{n_a(R)} \quad (2)$$

One the one hand, we require that reviews should contain as many negative aspects with justifications as possible to help with paper improvement. And on the other hand, we take into account the quality of the paper and relax this desideratum for high-quality papers since they don't need that many revisions. If a review contains no aspect in it, then we regard it as -1 informativeness.

2.3 D4: Accuracy

We use three measures to evaluate the accuracy of assessments. First, we use **summary accuracy (SACC)** to measure how well a review summarizes contributions of a paper. It takes value of 0, 0.5, or 1, which evaluate the summary part of the review as incorrect/absent, partially correct, and correct. The correctness judgement is performed manually, with details listed in Appendix A.7.

INFO implicitly requires that negative aspects should be supported with evidence, ignoring the quality of those evidences. However, to truly help to improve the quality of a paper, the evidence for negative aspects should be reasonable in order to be constructive. Here we propose **aspect-level factuality (AFAC)**, the percentage of the supporting statements $n_{\text{nae}}(R)$ that are judged as valid support by human annotators. If $n_{\text{nae}}(R)$ is 0, we set its AFAC as 1. The details of evaluating "validity" are also described in Appendix A.7.

In addition, we propose another metric **aspect recall (AREC)**, which explicitly takes the meta review R^m into account. Because the meta review is an authoritative summary of all the reviews for a paper, it provides an approximation of which aspects, and with which sentiment polarity, should be covered in a review. Aspect recall counts how many aspects in meta review R^m are covered by general review R, with higher aspect recall indicating better agreement with the meta review. Notably, this metric potentially biases towards high scores for reviews that were considered in the writing of the meta-review. Therefore, higher aspect recall is not the only goal, which should be taken together with other evaluation metrics

2.3.1 D5: Kindness

We leave the capturing of kindness to future work due to the difficulty of accurately measuring it computationally.

2.3.2 D6: Human-review alike

This desideratum is for generated reviews only. We use **semantic equivalence** metrics to measure how close generated reviews and reference reviews are in meaning. The intuition is that we want generated reviews to read like from human reviewers. Here, we investigate two specific metrics: ROUGE (Lin and Hovy, 2003) and BERTScore (Zhang et al., 2019). The former measures the surface-level word match while the latter measures the distance in embedding space.

3 Dataset

This section introduces how we construct a review dataset with more fine-grained metadata, which can be used for system training and the multiple perspective evaluation of reviews.

3.1 Data Collection

The advent of the Open Peer Review system[5] makes it possible to access review data for analysis or model training/testing. One previous work (Kang et al., 2018) attempts to collect reviews from several prestigious publication venues including the Conference of the Association of Computational Linguistics (ACL) and the International Conference on Learning Representations (ICLR). However, there are not enough reviews accumulated in OpenReview at that time [6] and other private reviews only account for a few hundred. Therefore we decided to collect our own dataset AutoReview.

We crawled ICLR papers from 2017-2020 through OpenReview [7] and NIPS papers from 2016-2019 through NIPS Proceedings [8]. For each paper's review, we keep metadata information as much as possible. Specifically, for each paper, we include following metadata information that we can obtain from the review web page:

[5] https://openreview.net/

[6] During that time, there are no reviews of ICLR from 2018 to 2020 nor reviews of NeurIPS from 2018 to 2019.

[7] https://openreview.net

[8] http://papers.nips.cc

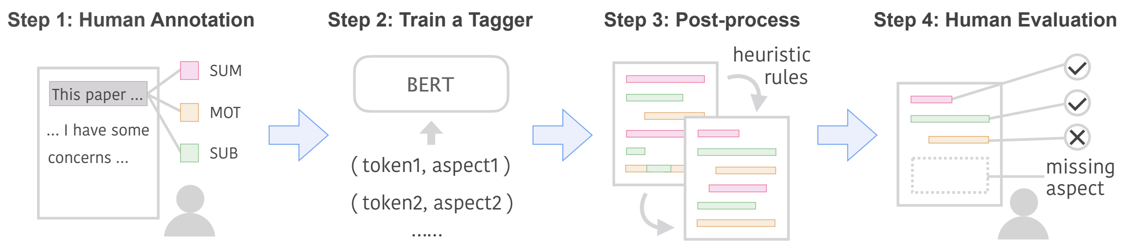


Figure 1: Data annotation pipeline.

- *Reference reviews*, which are written by a committee member.
- *Meta reviews*, which are commonly written by an area chair (senior committee member).
- *Decision*, which denotes a paper's final "reject" or "accept" decision.
- *Other* information like full content of a paper, url, title, author and etc.

we used Allenai Science-parse[9] to parse the pdf of each paper. The basic statistics of our AutoReview dataset is shown in Tab. 1.

	ICLR	NIPS	Both
Accept	1859	3685	5544
Reject	3333	0[1]	3333
Total	5192	3685	8877
Avg. Full Text Length	7398	5916	6782
Avg. Review Length	445	411	430
# of Reviews	15728	12391	28119
# of Reviews per Paper	3.03	3.36	3.17

[1] Note that NIPS only provide reviews for accepted papers to the public.

Table 1: Basic statistics of AutoReview dataset.

3.2 Aspect-enhanced Review Dataset

Although reviews exhibit internal structure, for example, as shown in Fig. 3 (*reviewer view*) , reviews commonly start with a paper summary, followed by different aspects of opinions, together with evidence. In practice, this useful information cannot be obtained directly. Considering that fine-grained information (e.g., aspect) plays an essential role in review evaluation, we conduct aspect annotation of those reviews. To this end, we first (i) structuralize each review by introducing an aspect typology, and then (ii) perform human annotation.

3.2.1 Review Structuralization

We define a typology that contains 8 aspects following ACL review guidance with small modifications, which are *Summary* (SUM), *Motivation* (MOT), *Originality* (ORI), *Soundness* (SOU), *Substance* (SUB), *Replicability* (REP), *Meaningful Comparison* (CMP) and *Clarity* (CLA). The detailed elaborations of each aspect can be found in Appendix A.1. Inside the parentheses are what we will refer to each aspect for brevity.

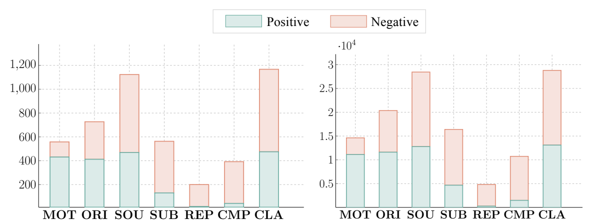
3.2.2 Aspect Annotation

Overall, the data annotation involves four steps that are shown in Fig. 1.

Step 1: Human Annotation To manually annotate aspects in reviews, we first set up a data annotation platform using Doccano [10]. Although we have pre-defined 8 aspects in §3.2.1, we decide to include sentiment polarity[11] for each aspect (except summary), so there are 15 tags in total for annotators to choose from.

To ensure the quality of our dataset, we asked 6 students from ML/NLP background to annotate the dataset. We asked them to tag appropriate text span that indicates a specific aspect without details. For example, "The results are new [Positive Originality] and important to this field [Positive Motivation]".

We got 1000 human annotated reviews in total. The aspect statistics in this partial dataset are shown in Fig. 2-(a).



(a) Human-labeled dataset. (b) Automatic-labeled dataset.

Figure 2: (a)-(b) represent distributions over seven aspects obtained by human and BERT-based tagger respectively. Red bins represent positive sentiment while green ones suggest negative sentiment.

[9] https://github.com/allenai/science-parse

[10] https://github.com/doccano/doccano

[11] For example, for motivation, we set two tags, one is positive motivation, the other is negative motivation.

Step 2: Train an Aspect Tagger Since there are over 20000 reviews in our dataset, using human labor to annotate them all is unrealistic. Therefore, we consider using the partial annotated data to train an aspect tagger to do the rest annotation job. The basic architecture of our aspect tagger contains a pre-trained model BERT (Devlin et al., 2019) and a multi-layer perceptron. The training details can be found in Appendix A.2.

Step 3: Post-process However, after inspecting the automatically labeled dataset, we found that there appears to be some common problems such as interleaving different aspects and inappropriate boundaries etc. To address those problems, we used seven heuristic rules to refine the prediction results and they were executed sequentially. The detailed heuristics can be found in Appendix A.3. An example of our model prediction after applying heuristic rules is shown in Appendix A.4. Fig. 2-(b) shows the distribution of all reviews over different aspects.

Step 4: Human Evaluation To evaluate the data quality of reviews' aspects, we conduct human evaluation. Specifically, we consider all aspect tags (including positive ones and negative ones) and measure both aspect precision and aspect recall. We randomly chose 300 samples from our automatically annotated dataset and assigned each sample to three different annotators to judge the annotation quality. As before, these annotators are all from ML/NLP background.

The detailed calculation for aspect precision and aspect recall can be found in Appendix A.5. Under these criterion, we achieved 92.75% aspect precision and 85.19% aspect recall which means that our data quality is reasonably well.

4 Searching for Effective Systems

4.1 Task Formulation

Based on the internal structure of a scientific review, we conceptualize review generation as an *opinion-aware scientific paper summarization* task. Although scientific paper summarization is not new, most current works summarize a paper (i) either from an "*author view*" that only use content written by the author to form a summary (Cohan et al., 2018a; Xiao and Carenini, 2019; Erera et al., 2019; Cohan et al., 2018a; Cachola et al., 2020), (ii) or from a "*reader review*" that argues a paper's summary should take into account the view of those

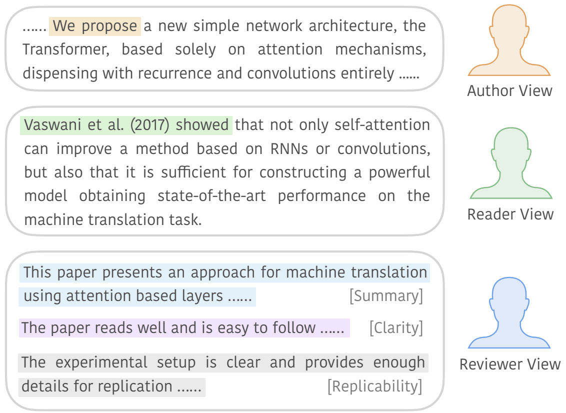


Figure 3: Summarization from three different views for the paper "Attention Is All You Need" (Vaswani et al., 2017).

in the research community (Qazvinian and Radev, 2008; Cohan and Goharian, 2017; Yasunaga et al., 2019a).

In this work, we extend the view of scientific paper summarization from "author", "reader" to "reviewer". and claim that a good summary of a scientific paper can not only reflect the core idea but also contains critical comments made by domain experts. The advantages lie in: (i) *reviewers*: it can relieve them from the burden of reviewing process (i) *readers*: help them quickly grasp the main idea of the paper and let them know what domain experts' comments on this paper are. The three views of scientific paper summarization is shown in Fig. 3.

4.2 System Design

Despite the fact that our dataset contains fewer training samples compared with other benchmark summarization datasets, the superior few-shot learning ability of recent contextualized pre-trained models (Radford et al., 2019; Brown et al., 2020) enables us to train a review generation system. We use BART (Lewis et al., 2019) as our pre-trained model.

However, even if we can take the advantage of those pre-trained models, how to deal with lengthy text in the context of using a pre-trained model (BART, for example, has a length limit of 1024) remains challenging. After multiple trials, we found one type of effective method and detail below. We put our other explorations in Appendix. A.8.

4.2.1 Two-stage Systems for Long Documents

Instead of regarding text generation as a holistic process, we decompose it into two steps, and consider using a *extract-then-generate* paradigm. Specifically, we first perform content selection, extracting salient text pieces from source documents (papers), then generate summaries based on these extracted texts.

To search for an effective way for content selection that really matters for constructing a solid *ReviewAdvisor* system, we operationalize the first extraction step in diverse ways. One thing to notice is that the extraction methods we use here mainly focus on heuristics. We leave more complicated selection methods for future work.

For comparison reason, we also construct oracle for each paper which is the extraction that achieves highest average ROUGE scores with respect to reference review. We do so by using a greedy method described in Nallapati et al. (2017). Note that for each paper with multiple reviews, we construct multiple oracles for that paper. We assume that oracle extractions can reflect where reviewers pay more attention to when they are writing reviews. The selected sentence position distribution in oracles is shown in Fig. 4.

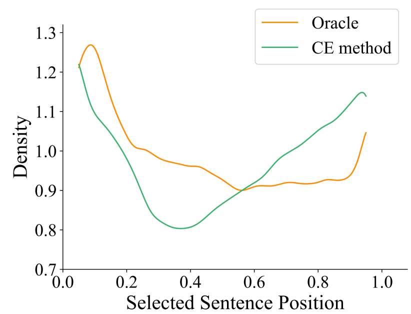


Figure 4: Selected sentence position distribution. We use the relative position of each sentence with regard to the whole article, thus taking values from 0 to 1.

Section-based Extraction Scientific papers are highly structured. As a convention, a scientific paper usually describes problem background, related work comparison, as well as its own contributions in the introduction part. Our section-based extractor simply extracts the introduction of a paper[12].

Cross-entropy (CE) Method Extraction Here we select salient sentences from the full text range. The way we do so is through a two-step selection process, which are

1. Select sentences that contain certain keywords. Those selected sentences form a set $\mathcal{S}$.
2. Select a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that sentences in $\mathcal{S}'$ cover diverse content and satisfy certain length constraint.

In the second step, we use Cross-entropy method introduced in Feigenblat et al. (2017). The details of this two-step process can be found in Appendix A.9. The selected sentence position distribution using this method is shown in Fig. 4. We can see that the extractor tends to select sentences from the beginning of a paper as well as the ending part of a paper just as the oracle extractor does. This makes sense because the beginning part is the introduction part which talks about the essence of the whole paper and the ending part mostly contains the analysis of experimental results and conclusions etc.

Hybrid Extraction We combine the abstract of a paper and its CE extraction to form a hybrid extraction.

4.2.2 Aspect-aware Summarization

Typically in the *extract-then-generate* paradigm, we can just use the extractions directly and build a sequence to sequence model to generate text. Here, in order to generate reviews with more diverse aspects and be able to interpret the generated reviews through the lens of their internal structure, we make a step towards a generation framework involving *extract-then-generate-and-predict*. Specifically, we use our annotated aspects (§3.2) as additional information, and design an auxiliary task that aims to predict aspects of generated texts (reviews). Fig. 5 illustrates the general idea of this.

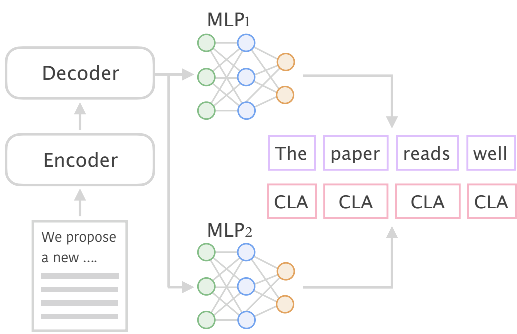


Figure 5: Aspect-aware summarization.

The loss of this model is shown in Eq. 3

$$\mathcal{L} = \mathcal{L}_{\text{seq2seq}} + \alpha \mathcal{L}_{\text{seqlab}} \tag{3}$$

[12]If a paper doesn't contain introduction part (due to pdf parsing problem), we use the abstract part of that paper instead.

where $\mathcal{L}_{\text{seq2seq}}$ denotes sequence to sequence loss which is the negative log likelihood of the correct next tokens, and $\mathcal{L}_{\text{seqlab}}$ denotes sequence labeling loss which is the negative log likelihood of the correct labels of next tokens. α is a tunable parameter, and we found $\alpha = 0.1$ works well.

5 Experiment

Here we consider three extraction strategies in §4.2.1 as well as two generation frameworks, one is the vanilla sequence to sequence model, the other is jointly sequence to sequence and sequence labeling.

5.1 Settings

Dataset We use full text (without Appendix) as source document[13]. And we filtered papers with full text fewer than 100 words since they don't contain enough information for models to learn. For reviews, we only use 100-1000 word reviews for training, which account for 92.57% of all the reviews. Reasons are: (i) Review that has more than 1000 words is noisy which often contains lots of updates. (ii) Review that has fewer than 100 words often comes from reviewers with less confidence.

This results in 8742 unique papers and 25986 paper-review pairs in total, the split of our dataset is shown in Tab. 2.

	Train	Validation	Test
Unique papers	6993	874	875
Paper-review pairs	20757	2571	2658

Table 2: Split of dataset.

Model The pretrained sequence to sequence model we used is BART. For all models, we initialized the model weights using the checkpoint: "bart-large-cnn"[14]. For *extract-then-generate-and-predict* framework, we add another multilayer perceptron on top of BART decoder, and initialize it with 0.0 mean and 0.02 standard deviation. We used Adam optimizer with $4e^{-5}$ learning rate and a linear learning rate scheduler which increases the learning rate linearly from 0 to the learning rate in the first 10% steps (warmup period) and then decreases the learning rate linearly to 0. We fine-tuned our models on the whole dataset for 5 epochs. We set a checkpoint at the end of every epoch and finally took the one with the lowest validation loss to do generation task.

During generation, we used beam search decoding with beam size 4. We set minimum length of 100 and maximum length of 1024. Length penalty of 2.0 and trigram blocking (Paulus et al., 2017) were used as well.

[13]If a paper has more than 250 sentences, we truncate it and take the first 250 sentences when we do the extraction step.

[14]We also tried XSUM checkpoint, however that results in much shorter reviews, and sentences in it tend to be succinct.

5.2 Results

Based on the evaluation metrics we defined in §2.2, we conduct both automatic evaluation and human evaluation to characterize both reference reviews and generated reviews. The aspect information in each review is got using aspect tagger we trained in §3.2.

Automatic Evaluation Automatic evaluation metrics include *Aspect coverage*, *Aspect Recall* and *Semantic Equivalence*. Notably, for each source input, there are multiple reference reviews. When aggregating ROUGE and BERTScore[15], we take the maximum instead of average. The results are shown in Table 3.

Human Evaluation Metrics that require human labor include *Recommendation accuracy*, *Informativeness*, *Aspect-level factuality* and *Summary accuracy*. We select 31 papers (21 accepted, 10 rejected) from ML/NLP/CV/RL domains and those papers are not in the training set. Details regarding human judgment are shown in Appendix A.7. The evaluation results are shown in Table 3.

Overall, we have the following observations:

(i) Using extractions from the full text range to generate reviews will result in higher aspect coverage and aspect recall compared to only using the introduction part. This is reasonable since models can receive more diverse information from the full text.

(ii) By looking at ROUGE and BERTScore, we can see that using extractive + abstractive methods can result in reviews that are much closer in meaning to reference reviews compared to pure extractive methods.

(iii) In terms of *aspect coverage*, *aspect recall* and *informativeness*, reference reviews fall behind most of our systems. This means that even reviews from the reviewers may also fall short on our defined criterion.

[15]We have used our own custom baseline to rescale BERTScore, details can be found in Appendix A.7.

Method		ACOV	AREC	RACC	INFO	AFAC	SACC	R-1	R-2	R-L	BS
HUMAN		46.23	58.80	41.83	69.75	81.05	94.35	–	–	–	–
EXTRACTIVE											
INTRO		–	–	–	–	–	–	38.62	8.84	25.11	29.22
CE		–	–	–	–	–	–	38.56	7.81	25.94	29.11
ABSCE		–	–	–	–	–	–	37.55	8.53	25.85	31.99
EXTRACTIVE+ABSTRACTIVE											
	Aspect										
INTRO	×	45.63	55.52	9.68	81.99	40.93	87.10	41.39	11.53	38.52	42.29
	√	46.09	58.24	6.45	73.27	40.80	87.10	41.31	11.41	38.38	42.33
CE	×	54.13	60.73	-6.45	84.48	44.22	87.10	42.37	11.72	39.86	41.78
	√	55.06	61.62	-19.35	81.59	37.37	77.42	42.27	11.62	39.73	41.71
ABSCE	×	48.89	58.31	-6.45	79.26	38.19	96.77	43.11	12.24	40.18	42.90
	√	49.90	57.56	-22.58	76.58	42.02	93.55	42.99	12.19	40.12	42.63

Table 3: Results of the baseline models as well as different aspect-enhanced models under diverse automated evaluation metrics. "BS" represents BERTScore. Red color denotes the lowest value and blue color denotes the highest value in each column.

(iv) Our systems can achieve much higher informativeness than reference reviews. Although we observe that both generated reviews and reference reviews give informative evidence for negative aspects almost always, reference reviews tend to say more positive aspects even when the paper is rejected at last, which cause great penalization. In our definition, we encourage reviews for rejected papers to contain much more negative aspects with evidence to help their revisions.

(v) When using metrics that require higher level understanding of the source paper like recommendation accuracy and aspect-level factuality, our systems have much lower performance. This means our systems cannot distinguish high-quality papers from low-quality papers well and the evidence for negative aspects cannot be trusted most of the time.

(vi) Current systems can correctly summarize the contributions of papers most of the time as shown by Summary accuracy. This is what we can hope for by now, and it's obvious that there is a long way to go towards reliable automatic review generation.

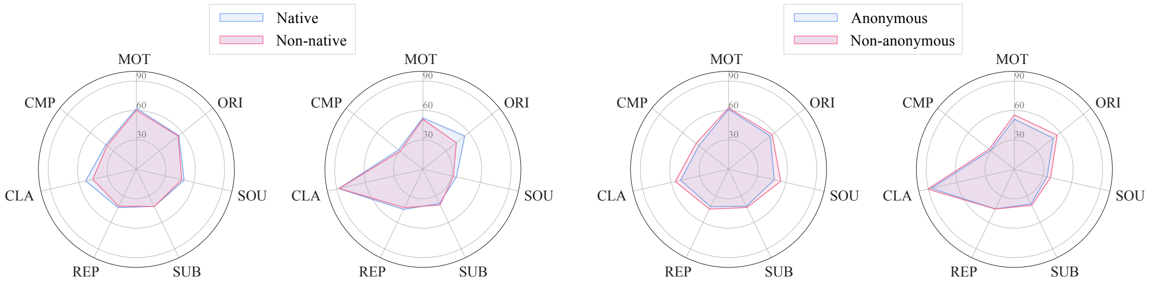
(vii) Our fine-grained evaluation metrics enable us to compare different systems and interpret their relative merits. For example, our systems can achieve higher informativeness than reference reviews while suffering from much lower aspect-level factuality. This means that if we want our systems to match the performance of real reviewers, we should focus on improving the factuality of our systems instead of letting it give more revision advice (which are not factually correct most of the time). Another example is that models using aspect information achieves consistently higher aspect coverage while suffering from lower recommendation accuracy. This means that for aspect-enhanced models, it is much urgent to improve its ability to distinguish papers of different quality instead of letting it generate more aspects.

We conduct detailed case studies in Appendix A.10.

6 Bias Analysis

6.1 Bias in NLP

Biases in text are prevalent while challenging to detect (Manzoor and Shah, 2020; Stelmakh et al., 2019). For example, in natural language processing, researchers are trying to identify societal biases (e.g, gender) in data and learning systems on different tasks (Bolukbasi et al., 2016; Zhao et al., 2018; Stanovsky et al., 2019). Previous works on biases analysis in scientific peer review usually focus on disparities in numerical feedback instead of text. Manzoor and Shah (2020) recently uncover latent bias in text. In this work, besides designing a model to generate reviews, we also perform bias analysis, in which we propose a method to identify and quantify biases both in human-labeled and system-generated data in a more fine-grained fashion.



(a) Reference reviews. (b) Generated reviews. (c) Reference reviews. (d) Generated reviews.

Figure 6: Spider chart of aspect scores with respect to different groups

6.2 Measuring Bias in Reviews

To characterize potential biases existing in reviews, we (i) first define *aspect score*, which calculates the percentage of positive occurrences [16] with respect to each review's aspects. The polarity of each aspect is obtained based on our learned tagger in §3.2.2; (ii) then we aim to observe if different groups G_i (e.g., paper is anonymous during reviewing or not) of reviews R would exhibit *disparity* $\delta(R, \mathbf{G})$ in different aspects. The calculation of disparity can be visualized in Fig. 7.

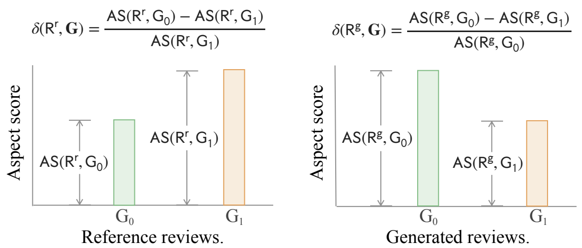


Figure 7: *Aspect score* $\text{AS}(R, G_i)$ and *disparity* $\delta(R, \mathbf{G})$ in reference reviews (R^r) and generated reviews (R^g). $\mathbf{G} = [G_0, G_1]$ denotes different groups.

Based on above two definitions, we characterize bias in two ways respectively:
(1) **spider chart**, which directly visualizes aspect scores of different groups of reviews w.r.t each aspect.
(2) **disparity difference**, which represents the difference between disparities in generated reviews R^g and reference reviews R^r and can be formally calculated as:

$$\Delta(R^g, R^r, \mathbf{G}) = \delta(R^g, \mathbf{G}) - \delta(R^r, \mathbf{G}) \quad (4)$$

where $\mathbf{G} = [G_0, G_1]$ denotes different groups based on a given partition criterion. Positive value means generated reviews favor group G_0 more compared to reference reviews, and vice versa.

In this work, we group reviews from two perspectives. The basic statistics are shown in Tab. 4.

	Native	Non-native	Anonym.	Non-anonym.
Total	651	224	613	217
Acc.%	66.51%	50.00%	57.59%	78.34%

Table 4: Test set statistics based on nativeness and anonymity.

Nativeness We categorize all papers in test set into "native" (G_0) and "non-native" (G_1) based on whether there is at least one native speaker in the author list as well as whether the institution is in an English-speaking country.

Anonymity We categorize all papers in test set into "anonymous" (G_0) and "non-anonymous" (G_1) based on whether the paper has been released pre-printly before half month after the conference submission deadline[17].

6.3 Analysis

6.3.1 Spider chart

Taking our model with introduction extraction as an example, the aspect scores of generated reviews and reference reviews are shown in Fig. 6.

Regarding *nativeness*, `native` papers receive higher score in most aspects in both reference reviews and generated reviews. Regarding *anonymity*, `non-anonymous` papers receive higher score in all aspects in both reference reviews and generated reviews. Those information reveal the favor for certain groups. Our observations are:

Nativeness Regarding human reviews:

[16] If an aspect doesn't appear in a review, then we count the score for that aspect 50 (stands for neutral). Details see Appendix A.6.

[17] We discard papers from ICLR 2017 since the reviewing process was single blind.

	MOT	ORI	SOU	SUB	REP	CMP	CLA	Total
Nativeness	-0.72	+18.71	+3.84	-3.66	+0.73	-13.32	+2.40	43.39
Anonymity	-5.69	-4.43	+2.76	-0.64	+5.65	+5.80	+3.02	28.00

Table 5: Disparity differences regarding nativeness and anonymity. Total is the sum of absolute value of disparity difference.

(1) By looking at Fig. 6-(a), there is a significant gap in `Clarity`, which is reasonable since non-native authors usually have trouble conveying their ideas. (2) Scores of the two groups are much closer in other aspects.

Regarding system-generated reviews:

(1) By looking at Fig. 6-(b), Auto-review system narrows the disparity in `Clarity` but amplify it in `Originality`. (2) Scores of the two groups are much closer in other aspects.

Anonymity Regarding human reviews: we find gaps are non-negligible in `Soundness`, `Clarity` and `Meaningful Comparison`.

Regarding system-generated reviews: we observe that gaps are non-negligible in `Motivation`, `Originality`, `Soundness`. The largest gap is not as large as that in human reviews.

6.3.2 Disparaty difference

However, through spider chart, gaps between different groups are relatively small and hard to discern. Besides, those gaps can only show the absolute favor for a certain group in different aspects. We are also interested in whether generated reviews are more in favor of a certain group **compared to reference reviews**. To do this, we calculate disparity differences and list them in Tab. 14.

Occurrences of positive numbers and negative numbers indicate that compared to reference reviews, generated reviews favor different groups in different aspects. This essentially means that overall our model does not particularly favor a certain group compared to reference reviews.

However, the total aspect bias regarding nativeness is much larger than anonymity. This, in some sense suggests that our model is more sensitive to nativeness compared to anonymity. We list the bias analysis for other models in Appendix A.11.

7 Related Work

Scientific Review Generation One previous work (Bartoli et al., 2016) investigated the feasibility of generating fake reviews by surface-level term replacement and sentence reordering etc. However, apart from that, there is very little work in this area, except one contemporaneous and independent work by Wang et al. (2020), who propose a two-stage information extraction and summarization pipeline to generate paper reviews. Their evaluation focuses mainly on the accuracy of information extraction, and the evaluation of the generated summaries is somewhat precursory, assessing only a single criterion "constructiveness and validity" manually over 50 papers. Our paper (1) proposes a wide variety of diagnostic criteria on review quality, (2) uses a very different summarization methodology, and (3) evaluates the generated results extensively.

Peer Review Peer review is an essential component of the research cycle and is adopted by most journals and conferences to identify important and relevant research. However, at the same time it is easy to identify many issues: expensiveness, slowness, existence of inconsistency (Langford and Guzdial, 2015) and bias (Tomkins et al., 2017), etc.

Some efforts have been put into analyzing the peer review process including automating review assignment (Jin et al., 2017; Nguyen et al., 2018; Anjum et al., 2019; Jecmen et al., 2020), examining bias problems (Tomkins et al., 2017; Stelmakh et al., 2019), examining consistency problem (Langford and Guzdial, 2015) as well as performing sentiment analysis on reviews (Wang and Wan, 2018; Chakraborty et al., 2020). Several decision classification methods have been explored to help make accept or reject decision given a paper. Those methods are either based on textual (Kang et al., 2018; Jen et al., 2018; Qiao et al., 2018) or visual (Von Bearnensquash, 2010; Huang, 2018) information. However, they don't actually help the reviewers since reviewers still need to write reviews for both accepted papers and rejected papers. In contrast, our work starts in the interests of reviewers and try to help automate review writing.

8 Discussion

We first summarize what we have achieved in this work and how the current *ReviewAdvisor* system can render help. Then we discuss challenges and potential directions for the automatic review generation task, which, hopefully, encourages more future researchers to explore this task, and in the right direction.

8.1 Machine-assisted Review System

Instead of replacing a human reviewer, a better position for *ReviewAdvisor* is to regard it as a machine-assisted review system. Although there is still a large room for improvement, current *ReviewAdvisor* can assist reviewers from the following aspects:

(1) Based on the evaluation of §5.2, *Summary accuracy* of our systems is quite high, suggesting that it can be either used for reviewers to finish the description of *Summary*, or help general readers to quickly understand the core idea of recently pre-printed papers (e.g., papers from arXiv[18]).

(2) Based on evaluation of §5.2, reviews generated by *ReviewAdvisor* can cover more aspects and generate more informative reviews. Although the associated opinions may suffer from the factuality problem, it still can be useful since it can provide a preliminary template for reviewers, especially enabling junior non-native English reviewers to know (i) what a review generally should include (ii) how to phrase it for each aspect, and then develop their own reviews based on the templates. Additionally, for each aspect (e.g., *Clarity*), our system can provide relevant evident sentences from the paper, helping reviewers quickly identify salient information when reviewing the paper (Detailed example in our Appendix A.10.1).

8.2 Challenges and Promising Directions

8.2.1 Model

(1) *Long Document Modeling*: The average length of one scientific paper is commonly larger than 5,000 words, far beyond the input text's length that mainstream neural sequence models (e.g., LSTM, Transformer) or pre-trained models (e.g., BERT, BART) normally use. This work (in §4.2.1) bypasses the difficulty by using a two-stage system, but there will be more other strategies to be explored.

(2) *Pre-trained Model for Scientific Domain*: Although previous works, as exemplified by (Beltagy et al., 2019) have pre-trained BERT on scientific domain, we observe that those models with transformer decoder perform much worse than BART on sequence generation tasks in terms of fluency and coherence, which calls for general sequence to sequence models pre-trained on scientific domain for higher-quality review generation.

(3) *Structure Information*: Review generation systems could get a deeper understanding of a given research paper if structure information can be provided. To this end, outputs from scientific paper-based information extraction tasks (Hou et al., 2019; Jain et al., 2020) can be utilized to guide review generation.

(4) *External Knowledge*: Besides the paper itself, review systems can also rely on external knowledge, such as a citation graph constructed based on more scientific papers or a knowledge graph connecting concepts across different papers.

8.2.2 Dataset

(5) *More Open, Fine-grained Review Data*: In this work, we annotate fine-grained information (aspect) of each review manually. However, this information can be relatively easily obtained from the peer review system. How to access this information appropriately would be an important and valuable step in the future.

(6) *More Accurate and powerful Scientific Paper Parser*: Existing parsing tools (e.g. science-parse, grobid) for scientific papers are commonly designed for certain specific paper templates, and also still struggle at extracting fine-grained information, such as the content of tables and figures.

8.2.3 Evaluation

(7) *Fairness and Bias in Generated Text*: In this work, we make a step towards identifying and quantifying two types of biases existing in human and system-generated reviews. Future works can explore more along this direction based on our dataset that contains fine-grained aspect annotation.

(8) *Reliability*: In addition to generating a review, a reliable system should also provide a level of confidence w.r.t the current comment. Moreover, whether review scores are calibrated is another valuable question.

Acknowledgment

This work cannot be accomplished without the help of many researchers. We would like to thank people for their generous support, especially,

[18] https://arxiv.org/

Volunteer to help us with the human evaluation: Gábor Berend, Zhouhan Lin, William W. Cohen, Pengcheng Yin, Tiange Luo, Yuki M. Asano, Junjie Yan, Tuomas Haarnoja, Dandan Guo, Jie Fu, Lei Chen, Jinlan Fu, Jiapeng Wu, Wenshan Wang, Ziyi Dou, Yixin Liu, Junxian He, Bahetiyaer Bare, Saizheng Zhang, Jiateng Xie, Spyros Gidaris.

References

Omer Anjum, Hongyu Gong, Suma Bhat, Wen-Mei Hwu, and JinJun Xiong. 2019. PaRe: A paper-reviewer matching approach using a common topic space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 518–528, Hong Kong, China. Association for Computational Linguistics.

Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. 2016. Your paper has been accepted, rejected, or whatever: Automatic generation of scientific paper reviews. In *International Conference on Availability, Reliability, and Security*, pages 19–28. Springer.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Lutz Bornmann and Rüdiger Mutz. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66(11):2215–2222.

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*.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S. Weld. 2020. Tldr: Extreme summarization of scientific documents. *ArXiv*, abs/2004.15011.

Souvic Chakraborty, Pawan Goyal, and Animesh Mukherjee. 2020. Aspect-based sentiment analysis of scientific reviews. *arXiv preprint arXiv:2006.03257*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, W. Chang, and Nazli Goharian. 2018a. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018b. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2017. Scientific article summarization using citation-context and article's discourse structure. *arXiv preprint arXiv:1704.06619*.

Nicola De Bellis. 2009. *Bibliometrics and citation analysis: from the science citation index to cybermetrics*. scarecrow press.

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.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, O. Nakash, O. Boni, Haggai Roitman, Doron Cohen, B. Weiner, Y. Mass, Or Rivlin, G. Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, F. Bonin, and D. Konopnicki. 2019. A summarization system for scientific documents. In *EMNLP/IJCNLP*.

Guy Feigenblat, Haggai Roitman, Odellia Boni, and David Konopnicki. 2017. Unsupervised query-focused multi-document summarization using the cross entropy method. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 961–964, New York, NY, USA. Association for Computing Machinery.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213, Florence, Italy. Association for Computational Linguistics.

Jia-Bin Huang. 2018. Deep paper gestalt. *arXiv preprint arXiv:1812.08775*.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Steven Jecmen, Hanrui Zhang, Ryan Liu, Nihar B Shah, Vincent Conitzer, and Fei Fang. 2020. Mitigating manipulation in peer review via randomized reviewer assignments. *arXiv preprint arXiv:2006.16437*.

Tom Jefferson, Philip Alderson, Elizabeth Wager, and Frank Davidoff. 2002a. Effects of editorial peer review: a systematic review. *Jama*, 287(21):2784–2786.

Tom Jefferson, Elizabeth Wager, and Frank Davidoff. 2002b. Measuring the quality of editorial peer review. *Jama*, 287(21):2786–2790.

William Jen, Shichang Zhang, and Muyun Chen. 2018. Predicting conference paper acceptance.

Rahul Jha, Amjad Abu-Jbara, and Dragomir Radev. 2013. A system for summarizing scientific topics starting from keywords. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 572–577, Sofia, Bulgaria. Association for Computational Linguistics.

Rahul Jha, Reed Coke, and Dragomir R. Radev. 2015a. Surveyor: A system for generating coherent survey articles for scientific topics. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2167–2173. AAAI Press.

Rahul Jha, Catherine Finegan-Dollak, Ben King, Reed Coke, and Dragomir Radev. 2015b. Content models for survey generation: A factoid-based evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 441–450, Beijing, China. Association for Computational Linguistics.

Jian Jin, Qian Geng, Qian Zhao, and Lixue Zhang. 2017. Integrating the trend of research interest for reviewer assignment. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1233–1241.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New Orleans, USA.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Langford and Mark Guzdial. 2015. The arbitrariness of reviews, and advice for school administrators.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Emaad Manzoor and Nihar B. Shah. 2020. Uncovering latent biases in text: Method and application to peer review.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *ArXiv*, abs/1611.04230.

Jennifer Nguyen, Germán Sánchez-Hernández, Núria Agell, Xari Rovira, and Cecilio Angulo. 2018. A decision support tool using order weighted averaging for conference review assignment. *Pattern Recognition Letters*, 105:114–120.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, page 689–696, USA. Association for Computational Linguistics.

Feng Qiao, Lizhen Xu, and Xiaowei Han. 2018. Modularized and attention-based recurrent convolutional neural network for automatic academic paper aspect scoring. In *International Conference on Web Information Systems and Applications*, pages 68–76. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and T. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. *ArXiv*, abs/1911.05507.

Reuven Y Rubinstein and Dirk P Kroese. 2013. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media.

R. Smith. 2006. Peer review: A flawed process at the heart of science and journals. *Journal of the Royal Society of Medicine*, 99:178 – 182.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. 2019. On testing for biases in peer review. In *Advances in Neural Information Processing Systems*, pages 5286–5296.

Albert N Tabah. 1999. Literature dynamics: Studies on growth, diffusion, and epidemics. *Annual review of information science and technology (ARIST)*, 34:249–86.

Andrew Tomkins, Min Zhang, and William D Heavlin. 2017. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48):12708–12713.

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

Carven Von Bearnensquash. 2010. Paper gestalt. *Secret Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Ke Wang and Xiaojun Wan. 2018. Sentiment analysis of peer review texts for scholarly papers. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 175–184.

Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. 2020. Reviewrobot: Explainable paper review generation based on knowledge synthesis. In *Proceedings of INLG*.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *ArXiv*, abs/1909.08089.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190, Online. Association for Computational Linguistics.

Wenting Xiong and Diane Litman. 2011. Automatically predicting peer-review helpfulness. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 502–507, Portland, Oregon, USA. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, A. R. Fabbri, Irene Li, D. Friedman, and Dragomir R. Radev. 2019a. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019b. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7386–7393. AAAI Press.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv*, pages arXiv–1904.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

A Appendices

A.1 Aspect definition

We define eight aspects for reviews, which are: *Summary*, *Motivation*, *Originality*, *Soundness/Correctness*, *Substance*, *Replicability*, *Meaningful comparison* and *Clarity*.

Some examples for different aspects are listed below.

Summary

- summary of the paper

Motivation

- motivation quite reasonable
- addresses an important problem
- work fundamental, impactful
- motivation not convincing
- no practical use

Originality

- idea novel
- little technical contribution
- lack of novelty

Soundness/Correctness

- theory sound
- proposed approach is promising
- design decisions thoughtful, sensible and well-justified
- unclear whether assumptions are warranted
- no enough justification to demonstrate improvements / claims
- results not good
- experimental results not convincing
- some reasoning not convincing

Substance

- experiment sufficient
- sufficient analysis
- experiment not sufficient
- no detailed and insightful ablation studies
- result analysis not enough
- lack extended discussion of proposed method

Replicability

- results can be reproduced
- technical ambiguities

Meaningful comparison

- discussion of related work is thorough
- does a great job comparing related work
- comparison not fair
- comparison not thorough

Clarity

- well-written and easy to follow
- unpolished writing

A.2 Training the aspect tagger

We formalize the annotation process as a sequence labeling problem where the input is a sequence contains n words $S = w_1, \cdots, w_n$, and the target is a sequence of tags one for each word $T = t_1, \cdots, t_n$. We aim to find a mapping f such that $T = f(S)$ can convey reasonable aspect information in the input sequence.

We first segment each review into sentences and consider each sentence as an individual training example [19].

For a tokenized sequence contains n tokens $(w_1, w_2, \cdots, w_n)$, we use BERT to get a contextualized representation for each token $(\boldsymbol{e_1}, \boldsymbol{e_2}, \cdots, \boldsymbol{e_n})$, where $\boldsymbol{e_i}$ represents the vector for ith token.

Then those contextualized representations can be used as features for token classification:

$$p_i = \mathrm{softmax}(\mathbf{W}\boldsymbol{e_i} + \mathbf{b})$$

where $\mathbf{W}$ and $\mathbf{b}$ are tunable parameters of the multilayer perceptron. p_i is a vector that represents the probability of token i being assigned to different aspects.

We use the negative log likelihood of the correct labels as training loss:

$$\mathcal{L} = -\sum_{t \in \mathcal{T}} \log p_{tj}$$

where j is the label of token t, and $\mathcal{T}$ denotes all the tokens.

We used 900 annotated reviews for training and 100 for validation which is equivalent to using 16543 training data and 1700 validation data since we consider sentence as the basic individual training sample. The initial BERT checkpoint we used is bert-large-cased. We used Adam optimizer (Kingma and Ba, 2014) with a learning rate of $5e^{-5}$ to finetune our model. We trained for 5 epochs and saved the model that achieved lowest loss on validation set as our aspect tagger.

A.3 Heuristics for refining prediction results

The seven heuristic rules used for refining the prediction results are listed below.

1. If there are no other tags (they are tagged as O which stands for Outside) between two summary tags, then replace all tags between them with summary tag.

[19] We also tried using larger context such as paragraph, but found out the results less satisfying since the model identified fewer aspects.

2. If there are multiple text spans tagged as summary, keep the first one and discard others.

3. If the punctuation is separately tagged and is different from its neighbor, we replace its tag to O.

4. If two same tags are separated by a single other tag, then replace this tag with its right neighbor's tag.

5. If there exists a single token with a tag and its neighbors are O, then replace this tag to O.

6. For a non-summary non-O tag span, if its neighbors are O and the start/end of this span is not special symbol (for example, punctuations or other symbols that have 1 length), then we expand from its start/end until we meet other non-O tag or special symbol.

7. If the summary span doesn't end with a period, then we truncate or extend it at most five words to make it ends with a period.

A.4 An example of automatically annotated reviews

Using our trained aspect tagger and applying seven heuristic rules in Appendix A.3 to refine the prediction results, an example of our final annotated review is shown in Tab. 6.

Although here we do not add separate polarity tags to avoid visual burden, the polarity of each aspect the model predicts is correct.

A.5 Calculation of aspect precision and aspect recall

For measuring aspect precision, we asked three annotators to decide whether each aspect span the model predicted is appropriate. They were asked to delete a tagged span if they regarded it as inappropriate. We denote all spans the model predicts as $\mathcal{M}$, and the filtered spans from annotators as $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$. We denote $n_{\mathcal{S}}$ as the total number of text spans in $\mathcal{S}$.

Here we define correct spans as

$$\mathcal{C} = \{l | l \in \mathcal{F}_1, l \in \mathcal{F}_2, l \in \mathcal{F}_3\}$$

The aspect precision is calculated using Formula 5.

$$\text{Precision} = \frac{n_{\mathcal{C}}}{n_{\mathcal{M}}} \tag{5}$$

For measuring aspect recall, we asked three annotators to label aspect spans that they identified while the model ignored. We denote the additional labeled spans from one annotator as $\mathcal{A}$ where $\mathcal{A} = \{a_1, a_2, \cdots, a_{n_{\mathcal{A}}}\}$, a_i represents a text span. We denote the additional labeled spans from other two annotators as $\mathcal{B}$ and $\mathcal{C}$.

We define common ignored spans for every two annotators as below. $|\cdot|$ denotes the number of tokens in a span and $\cap$ takes the intersect span between two spans.

$$\mathcal{I}_1 = \{a_i \cap b_j | \frac{|a_i \cap b_j|}{\min\{|a_i|, |b_j|\}} > 0.5\}$$
$$\mathcal{I}_2 = \{b_i \cap c_j | \frac{|b_i \cap c_j|}{\min\{|b_i|, |c_j|\}} > 0.5\}$$
$$\mathcal{I}_3 = \{a_i \cap c_j | \frac{|a_i \cap c_j|}{\min\{|a_i|, |c_j|\}} > 0.5\}$$

We also define common ignored spans for three annotators as below.

$$\mathcal{I} = \{a_i \cap b_j \cap c_k | \frac{|a_i \cap b_j \cap c_k|}{\min\{|a_i|, |b_j|, |c_k|\}} > 0.3\}$$

where a_i, b_j, c_k are text spans from $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ respectively. We assume all the spans the model predicts are correct. Then we can calculate total number of spans using Formula 6.

$$\begin{aligned} n = & n_{\mathcal{M}} + n_{\mathcal{A}} + n_{\mathcal{B}} + n_{\mathcal{C}} - n_{\mathcal{I}_1} - n_{\mathcal{I}_2} - \\ & n_{\mathcal{I}_3} + n_{\mathcal{I}} \end{aligned} \tag{6}$$

The aspect recall is calculated using Formula 7.

$$\text{Recall} = \frac{n_{\mathcal{M}}}{n} \tag{7}$$

A.6 Calculation of aspect score

For accepted (rejected) papers, we calculate the average aspect score for each aspect.

The aspect score of a review is calculated as follows.

- If an aspect doesn't appear in a review, then we count the score for this aspect 50 (stands for neutral).

- If an aspect appears in a review, we denote its occurrences as $\mathcal{O} = \{o_1, o_2, \cdots, o_n\}$ where n is the total number of occurrences. And we denote the positive occurrences of this aspect as $\mathcal{O}_p = \{o_{p_1}, o_{p_2}, \cdots, o_{p_n}\}$ where p_n is the total number of positive occurrences. The aspect score is calculated using Formula 8.

$$\text{Aspect Score} = \frac{p_n}{n} \times 100 \tag{8}$$

summary	originality	clarity	meaningful comparison	motivation	substance

This paper studies the graph embedding problem by using encoder-decoder method . The experimental study on real network data sets show the features extracted by the proposed model is good forclassification . Strong points of this paper: 1. The idea of using the methods from natural language processing to graph mining is quite interesting . 2. The organization of the paper is clear Weak points of this paper: 1. Comparisons with state-of-art-methods (Graph Kernels) is missing . 2. The problem is not well motivated, are there any application of this . What is the difference from the graph kernel methods ? The comparison with graph kernel is missing . 3. Need more experiment to demonstrate the power of their feature extraction methods . (Clustering, Search, Prediction etc.) 4. Presentation of the paper is weak . There are lots of typos and unclear statements.

Table 6: An example of automatically labeled reviews.

A.7 Details for evaluation metrics

1. In §2.2, the REC function we define corresponds to the recommendation sentiment of a review, with $\{-1, 0, 1\}$ representing negative, neutral and positive.

   To decide the sentiment of a reference review, we use the rating information from reviewers. If the rating corresponds to marginal accept or marginal reject, then we regard it as neutral, if the rating is above marginal accept, then we regard it as positive, otherwise, we regard it as negative.

   To decide the sentiment of a generated review, two annotators from ML/NLP background were asked to judge the sentiment polarity of a review. If they agreed with each other, then we use the agreed sentiment, if they disagreed with each other, then we label the sentiment of that review as neutral. The Cohen kappa of two annotators is 0.5778.

2. The judgement of evidence for negative aspects is conducted by human. One annotator from ML background was asked to judge whether each identified negative aspect has accompanied evidence regardless of the quality of the evidence. So long as there is a reason, we count that as an evidence.

3. The judgement of summary accuracy and valid support for negative aspects is performed by one of the first authors of the reviewed paper. Summary and each negative aspect in the review should be scored 0, 0.5 or 1 which represents agree, partially agree and disagree. We gave the authors the following instruction:

   We have created a Google doc for your paper, and you can score the summary as well as each aspect with its corresponding comments inside the red brackets. ”1” denotes agree, ”0.5” denotes partially agree, ”0” denotes disagree. You only need to assign a score based on your judgment. For summary, agree means that you think it's factually correct. For each aspect, agree means that you think the strength/weakness the reviewer points out is reasonable or constructive.

4. When calculating BERTScore, we use our own custom baseline (P: 71.99, R: 67.17, F: 73.03) to make the score more readable, which is got through randomly pairing 20000 reference reviews.

A.8 Adjust BART for long document

The first attempts we made are to directly adjust BART for long text, either through expanding its position encoding or to segment the input text and deal with those segments individually.

Below are three ways we have attempted.

Arc-I: Position Encoding Expanded BART Since the original BART model is pretrained on 512 sequence length and fintuned on 1024 sequence length [20]. We followed this approach and tried copying the first 1024 position encodings periodically for longer sequence and finetuned the model on our own dataset.

Arc-II: Independently-windowed BART In this architecture, we simply chunked the documents into multiple windows with 1024 window size, and then use BART to encode them separately. The

[20] https://github.com/pytorch/fairseq/issues/1413

final output of the encoder side is the concatenation of those window outputs. The decoder can then generate texts as normal while attending to the whole input representations.

Arc-III: Dependently-windowed BART In **Arc-II**, we ignore the interdependence between each chunk which may lead to incoherence in generated texts. Here, to model the inter-window dependencies, we followed the approach introduced in Rae et al. (2020). We kept a compressive memory of the past and used this memory to compute the representation of new window. The final output of the encoder side is the concatenation of those window outputs as in **Arc-II**.

However, none of these adjustments can generate satisfying fluent and coherent texts according to our experiments. Common problems include interchanges between first and third person narration (They... Our model...), contradiction between consecutive sentences, more descriptive texts and fewer opinions, etc.

A.9 CE extraction details

The basic sentence statistics of our dataset is listed in Table 7.

	ICLR	NIPS	Both
Avg. Sentence Num.	216	198	209

Table 7: Sentence statistics AutoReview dataset.

We use two steps to extract salient sentences from a source document, which are

1. Keywords filtering
2. Cross-entropy method

A.9.1 Keywords filtering

For keywords filtering, we have predefined 48 keywords and in the first stage, we select sentences containing those keywords as well as their inflections.

The 48 keywords are shown in Tab. 9. After applying keywords filtering, the statistics of selected sentences are shown in Tab. 8.

	ICLR	NIPS	Both
Avg. Sentence Num.	97	85	92

Table 8: Statistics of selected sentences.

A.9.2 Cross Entropy Method

Following Feigenblat et al. (2017)'s approach in unsupervised summaization. We formalize the sentence extraction problem as a combinatorial optimization problem.

We define the performance function R as below.

$$R(S) = -\sum_{w \in S} p_S(w) \log p_S(w)$$

$$p_S(w) = \frac{\text{Count}(w)}{\text{Len}(S)}$$

Where S represents the concatenation of selected sentences, $\text{Len}(S)$ represents the number of words in S, $\text{Count}(w)$ represents the number of times w appears in S.

The intuition behind this performance function is that we want to select sentences that can cover more diverse words. Note that when calculating $R(S)$, we do preprocessing steps (i.e. lowercasing, removing punctuation, removing stop words etc.).

For each paper containing n sentences, we aim to find a binary vector $p = (p_1, \cdots, p_n)$ in which p_i indicates whether the ith sentence is selected such that the conbination of selected sentences achieves highest performance score and also contains fewer than 30 sentences.

We did this by using Cross Entropy Method (Rubinstein and Kroese, 2013). The algorithm is shown below.

1. For each paper containing n sentences, we first assume that each sentence is equally likely to be selected. We start with $p_0 = (1/2, 1/2, ..., 1/2)$. Let $t := 1$.

2. Draw a sample $X_1, \cdots, X_N$ of Bernoulli vectors with success probability vector p_{t-1}. For each vector, concatenate the sentences selected and get N sequences $S_1, \cdots, S_N$. Calculate the performance scores $R(S_i)$ for all i, and order them from smallest to biggest, $R_{(1)} \leq R_{(2)} \leq \cdots \leq R_{(N)}$. Let γ_t be $(1-\rho)$ sample quantile of the performances: $\gamma_t = R_{(\lceil (1-\rho)N \rceil)}$.

3. Use the same sample to calculate $\hat{p}_t = (\hat{p}_{t,1}, \cdots, \hat{p}_{t,n})$ via

$$\hat{p}_{t,j} = \frac{\sum_{i=1}^{N} I_{\{R(S_i) \geq \gamma_t\}} I_{\{X_{ij}=1\}}}{\sum_{i=1}^{N} I_{\{R(S_i) \geq \gamma_t\}}}$$

where $I_{\{c\}}$ takes the value 1 if c is satisfied, otherwise 0.

KEYWORDS							
find	prove	examine	address	suggest	baseline	optimize	outperform
show	design	explore	analyze	achieve	maximize	efficient	generalize
imply	reduce	propose	explain	perform	minimize	effective	understand
study	metric	observe	benefit	improve	increase	introduce	investigate
bound	better	present	compare	dataset	decrease	interpret	demonstrate
apply	result	develop	measure	evaluate	discover	experiment	state-of-the-art

Table 9: Predefined keywords.

4. Perform a smoothed update.

$$p_t = \alpha\hat{p_t} + (1-\alpha)p_{t-1}$$

5. If the value of γ_t hasn't changed for 3 iterations, then stop. Otherwise, set $t := t + 1$ and return to step 2.

The elements in p_t will converge to either very close to 0 or very close to 1. And we can sample from the converged p_t to get our extraction.

We chose $N = 1000$, $\rho = 0.05$ and $\alpha = 0.7$ when we ran this algorithm. If we happen to select more than 30 sentences in a sample, we drop this sample. Note that we slightly decrease the initial probability when there are more than 90 sentences after filtering to ensure enough sample number in the first few iterations.

A.10 Detailed result analysis and case study

In §5.2, we have shown that our aspect-enhanced model performs better in terms of aspect coverage and aspect recall. Here we take our aspect-enhanced model using CE extraction to do a more fine-grained analysis to diagnose what our system can do and what it cannot.

A.10.1 Case study

In the aspect-enhanced model, we have added external sequence labeling loss during training. And it would be interesting to see the results of the model generating a token and predict its label simultaneously. Tab. 10 shows an example review when we do so.

We can see that the model can not only generate fluent text but also be aware of what aspect it's going to generate including the right polarity of that aspect. Although the generated aspects are often small segments and there are some minor alignment issues, but the model is clearly aspect-aware.

It would also be interesting to trace back to the evidence when the model generates a specific aspect. To do that we inspect where the model attends when it generates a specific aspect by looking at the attention values with respect to the source input. [21].

And interestingly, we found that the model attends to the reasonable place when it generates a specific aspect. Fig. 8 presents the attention heatmap of several segment texts , the bottom of the figure shows aspects the model generates.

There are some common patterns we found when we examined the attention values between the source input and output. For example, when the model generates summary, it will attend to sentences that contain strong indicators like "we propose" or "we introduce"; when it generates originality, it will attend to previous work part as well as places describing contributions of this work; when it generates substance, it will attend to experiment settings and number of experiments conducted; when it generates meaningful comparison, it will attend to places contains "et al.".

Although we cannot say that the model makes reasonable judgments based on those places it attends to. At least, when it generates a specific aspect, it is attending to right places.

A.10.2 Detailed results analysis

To have a deeper look at what the model generates, we select five examples for each aspect the model mentions and list them in Tab. 12.

Although those comments seem like from real reviewers, there are common issues, which we list below.

1. In §6.3.1, we observe that generated reviews achieve much higher scores in the *Clarity* aspect for model with introduction extraction. Same phenomenon occurs in other models as well. After careful inspection, we found that "The paper is well-written and easy to follow"

[21] The way we aggregate attention values is to take the maximum, no matter is to aggregate tokens to a word or to aggregate different attention heads or to aggregate words to an aspect.

summary	clarity+[1]	substance+	soundness-[1]	substance-

This paper studies the problem of transfer learning from a single pre-trained network onto a new task . The authors propose a method of combining the outputs of multiple pre-trainable classifiers by training on their raw predictions and then fine-tuning on the target samples . The proposed method is based on the use of (non-linear) maximal correlation analysis that originated with Hirschfeld [9] to this problem . The paper is wellwritten and easy to follow . The experimental results show that the proposed method outperforms the state-of-the-art methods on the CIFAR-100 , Stanford Dogs , and Tiny ImageNet datasets . However , it is not clear to me how the performance of the method is affected by the number of target training samples . It would be better if the authors can provide some theoretical analysis on the effect of the size of the target dataset .

[1] + denotes positive sentiment. - denotes negative sentiment

Table 10: Generate tokens with its aspects jointly

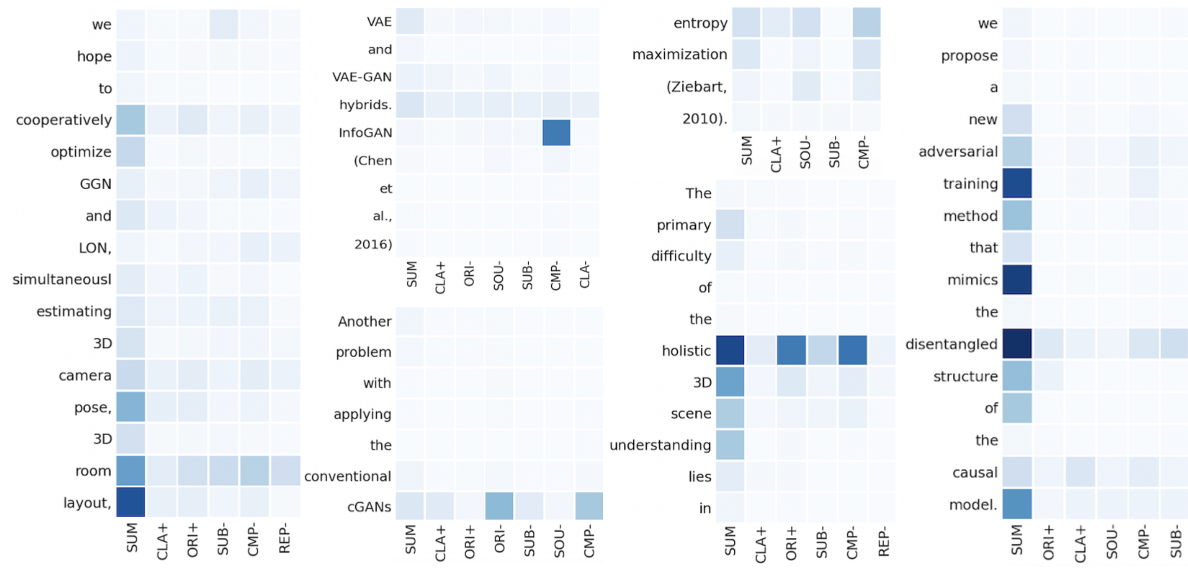


Figure 8: Attention heatmap between source document and generated reviews. $+$ denotes positive sentiment and $-$ denotes negative sentiment.

appears in more than 90% generated reviews. This is due to the fact that in the training data, this exact sentence appears in more than 10% papers. This accounts for the high number of occurrences of *Positive Clarity*.

2. Generated reviews struggle to ask questions which is an important component in peer reviewing. In the reference reviews, average number of questions per review ask is 2.04, while it is only 0.32 in generated reviews.

3. In generated reviews, supportive evidences for negative aspects are often invalid. For example, when it generates "The author should compare with [1, 2, 3] and [4]", it rarely provides the reference papers beyond those random numbers. Even if it provides the related papers to be compared with, those links mostly point to papers that are not relevant to the current paper at all. Another example is that it often suggests a paper to conduct experiment in more datasets like ImageNet even if the paper is a NLP paper.

All those issues above again indicate that the model is just imitating style, and it has no ability to do inference at the moment.

However, there are also strengths in generated reviews. After carefully inspecting the generated reviews, we found that their structures are very similar and typically follow the routine in Tab. 11.

This is better compared to some reviews from

STRUCTURE
Overview of the paper [summary] Some advantages [Pros] However I have some concerns: [Cons] 1. concern1 ... 2. concern2 ... 3. concern3 ... .

Table 11: Structure of generated reviews.

reviewers where there are just comments concatenated together with no explicit structure. And as has been shown in §5.2, the model can conclude contributions of a paper correctly most of the time.

A.11 Bias analysis for all models

Here, following the method we proposed in §6.2, we list the bias analysis for all models below.

B Supplemental Material

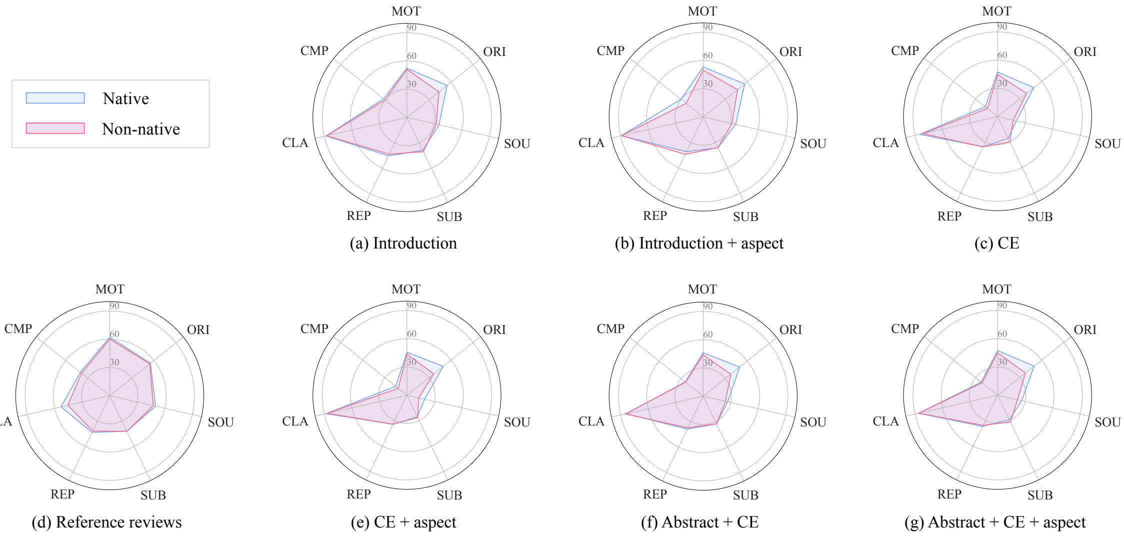


Figure 9: Spider chart of aspect scores for all models with regard to nativeness.

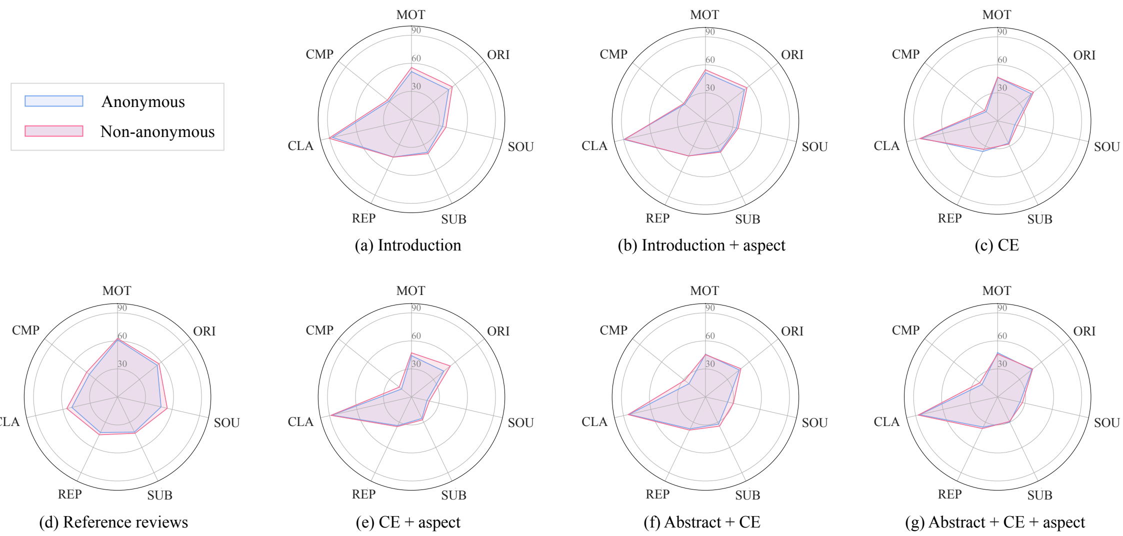


Figure 10: Spider chart of aspect scores for all models with regard to anonymity.

Motivation
1. The motivation of using the conditional prior is unclear. 2. I think this paper will be of interest to the NIPS community. 3. The idea of continual learning is interesting and the method is well motivated. 4. Overall, I think this paper is a good contribution to the field of adversarial robustness. 5. It is hard to understand the motivation of the paper and the motivation behind the proposed methods.
Originality
1. This paper presents a novel approach to cross-lingual language model learning. 2. The novelty of the paper is limited . The idea of using low rank matrices is not new. 3. The proposed method seems to be very similar to the method of Dong et al. (2018). 4. The idea of using neural networks to learn edit representations is interesting and novel . 5. The proposed method seems to be a simple extension of the batched-E-step method proposed by Shazeer et al.
Soundness
1. This assumption is not true in practice . 2. The experimental results are not very convincing . 3. But the authors do not provide any theoretical justification for this claim. 4. The theoretical results are sound and the experimental results are convincing. 5. The paper does not provide any insights on the reasons for the success of the supervised methods.
Substance
1. The experiments are well-conducted. 2. The ablation study in Section A.1.1 is not sufficient. 3. It would be better to show the performance on a larger dataset. 4. The authors should show the performance on more difficult problems. 5. The experiments are extensive and show the effectiveness of the proposed method.
Replicability
1. It is not clear how the network is trained. 2. The authors should provide more details about the experiments. 3. The authors should provide more details about the hyperparameters. 4. The authors should provide more details about the training procedure. 5. It would be better if the authors can provide more details about the hyperparameters of LST.
Meaningful Comparison
1. The author should compare with [1 , 2 , 3] and [4] . 2. The authors should compare the proposed method with existing methods . 3. It would be more convincing if the authors can compare with other methods such as AdaGrad. 4. authors should compare the performance with the state-of-the-art methods in real-world applications . 5. I also think the paper should compare the performance of intrinsic fear with the other methods proposed in [1 , 2 , 3 , 4 , 5].
Clarity
1. There are some typos in the paper. 2. The paper is well-written and easy to follow. 3. It is not clear to me how to interpret the results in Table 1. 4. It would be better if the authors can provide a more detailed explanation of the difference. 5. The paper is not well organized . It is hard to follow the description of the proposed method.

Table 12: Examples for different aspect mention from generated reviews.

	MOT	ORI	SOU	SUB	REP	CMP	CLA	Total
INTRO	-0.72	+18.71	+3.84	-3.66	+0.73	-13.32	+2.40	43.39
INTRO+ASPECT	+3.12	+15.75	+6.14	+0.66	-10.61	-13.50	+19.05	68.84
CE	+2.56	+18.33	+11.16	-13.41	-3.71	-9.94	+13.49	72.58
CE+ASPECT	+1.13	+24.77	+28.78	-2.92	-3.18	-12.02	+18.36	91.18
ABSTRACT+CE	+1.77	+23.01	+3.79	+0.44	+0.37	-15.18	-2.13	46.69
ABSTRACT+CE+ASPECT	+1.72	+22.23	+12.94	-8.30	-0.38	-13.40	+0.89	59.86

Table 13: Disparity differences regarding nativeness. Total is the sum of absolute value of disparity difference.

	MOT	ORI	SOU	SUB	REP	CMP	CLA	Total
INTRO	-5.69	-4.43	+2.76	-0.64	+5.65	+5.80	+3.02	28.00
INTRO + ASPECT	-3.53	-1.65	+7.85	+0.01	+5.93	+11.02	+4.20	34.20
CE	+1.89	-1.18	+0.05	-0.44	+13.09	+8.00	-2.56	27.21
CE+ASPECT	-4.20	-12.32	-0.52	-2.57	+2.70	+8.75	-10.31	41.37
ABSTRACT+CE	+3.18	-0.05	-7.96	-3.73	+2.25	+8.69	-12.02	37.88
ABSTRACT+CE+ASPECT	+5.45	+2.49	+2.80	+5.69	+1.33	+8.03	-3.79	29.59

Table 14: Disparity differences regarding anonymity. Total is the sum of absolute value of disparity difference.