Neural Architectures for Reviewer Expertise Modelling

Authors

Abstract

The increased popularity of machine learning and related disciplines is pressuring corresponding conference to scale up. Large-scale conferences are challenging for our peer-review systems which must assign thousands of expert reviewers to thousands of submitted papers (e.g. AAAI 2021 received over 9,000 submissions). As a response in the last decade, conferences have partly automated the reviewer assignment process. For example, TPMS the Toronto Paper Matching System was created to predict reviewer expertise. Other systems such as OpenReview have similar functionalities. These tools prepare content-based profiles of reviewers (researchers) and submitted papers, to model the suitability of reviewer-submission pairs.

Motivated by the challenges faced by conferences, we decompose expertise modelling into two tasks. First, we obtain single paper representations by adapting recent language models. We explore different contrastive self-supervised fine-tuning approaches with the aim of obtaining representations that capture expertise and disregard other attributes of a paper. We then use an attention mechanism to combine the set of papers authored by a reviewer conditioned on a submission. These are submission-specific reviewer representations. Using two reviewer-paper suitability datasets from recent conferences we show that better individual paper representations lead to better performance on the downstream task of predicting reviewer suitability.

Introduction

The number of submissions to machine learning and artificial intelligence (AI) conferences is increasing yearly fueled by the increased popularity of the fields. The scale of current conference is challenging for our scientific peer-review process. The process relies on every submission being reviewed by an expert ideally nominated by an impartial program committee. However, short reviewing timelines, large number of submissions and new evolving subfields, make manual assessment of reviewer suitabilities impossible.

Over the last decade, the scaling up challenge has been addressed, in part, through a semi-automated process using expertise-modelling tools such as TPMS, the Toronto Paper Matching System (Charlin and Zemel 2013) and more recently OpenReview.[1] These systems analyze the content of submitted papers as well as the content of reviewer papers to predict reviewer-submission suitability.

The task of expertise modelling therefore relies on extracting expertise in collections of scientific papers. This is a type of language modelling task. One in which representations should capture elements such as topics and techniques discussed, and might discard others attributes such as a style.

Motivated by the challenges faced by recent conferences, we focus on obtaining expertise from scientific papers through unsupervised language models. While self-assessed reviewer expertise, such as *bids*, could be used as labels, expertise models are often used prior bidding in many conference processes. Therefore, it is important to obtain representations from unlabelled data alone.

We propose several augmentation strategies for fine-tuning large-scale transformers with contrastive self-supervised techniques. We focus on augmentations that can preserve expertise and discard less useful attributes for the task (such as writing style). Then, we investigate how to combine the expertise of reviewers obtained from a collection of their papers to predict reviewer-submission suitabilities.

Through an empirical study we validate some of our models against both simple representations such as Bag-of-Words and topic proportions from Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003) and more recent large-scale language model representations.

Related work

Our approaches using contrastive self-supervised learning are similar to earlier approaches based on siamese Networks (Bromley et al. 1993), and triplet loss (Hoffer and Ailon 2015) based distance metric learning approaches. Face verification (Chopra, Hadsell, and LeCun 2005) and FaceNet (Schroff, Kalenichenko, and Philbin 2015), have used similar siamese-based networks in the past to do distance-based learning.

Oord, Li, and Vinyals (2018) intended to extract useful representations from an encoder trained on general data, and then fine tune using contrastive objective for specific

[1] https://openreview.net and https://github.com/openreview/openreview-expertise

data sets. For example for their NLP task, they used a similar approach to our next sentence prediction augmentation task. They embedded a sentence to a 2400-dimension vector z, followed by a GRU, to predict 3 future sentences using a contrastive loss. Concurrent to our work, (Gao, Yao, and Chen 2021) describe an approach SimCSE, which takes an input sentence and predicts itself in a contrastive objective, with standard dropout used as a minimal form of augmentation. They also suggest a more supervised approach to contrastive learning when they had access to datasets with labelled positive and negative pairs. (Liu et al. 2021), propose Contrastive Self-supervised learning for sequential recommendations (CoSeRec). They showed two augmentation operators, which leveraged item correlations to create high quality augmentations for contrastive learning in dynamic recommendations.

Paper Representation by Augmentations

Augmentations and their choice has been determined to be very important for learning good embedding encoders (Chen et al. 2020). Augmentations should try to create transformations that force the model to learn the actual semantic information, instead of superficial changes. For example a human can easily identify that a clockwise rotated image of a dog, and anti-clockwise rotated version of that image, are semantically describing the same thing.

The original SimCLR paper(Chen et al. 2020) was working with image data. The authors cropped or rotated images, to get different augmented data views. We drew inspiration from these efforts, and realised that expertise should also be identified independent of style. Different reviewers, based on their different backgrounds and origins could have different approaches to describing the same topics. We would like to separate style from content of our textual inputs. These augmentation methods should try to preserve the semantic meaning of the data while removing other style-based elements. To achieve this, we explore different augmentation approaches described below.

Retrieving top words from Bag of Words Figure 1 shows a general approach taken by us to implement different augmentation-based approaches. TF-IDF is limited to picking up key words. But we imagined that we could leverage this limitation to pick out words denoting topics. For the Bag of words augmentation task, we first calculated the TF-IDF (Term frequency-inverse document frequency) scores of all the tokens in the corpus of submission papers. TF-IDF scores are generally used to pick up the most salient words from a given text, based on their frequency in the text corpus. We can imagine that key words or rare words contain the most identifying semantic information for text. A model trained to match the BERT embeddings of the original text, with BERT representation of the top k words from TF-IDF scores, would in effect learn to focus on keywords and prepare a summary representation. We call this approach, the BOW augmentation task for our self-supervised framework. Figure 1 shows a top k ($k = 30$) BOW task. We tried with different k values, and $k = 30$ was effective for us, based on observing the loss values during fine-tuning, and accuracy on the validation set. We can postulate that thirty keywords maybe capture enough information about abstracts, where abstracts are usually about 250 to 350 words long.

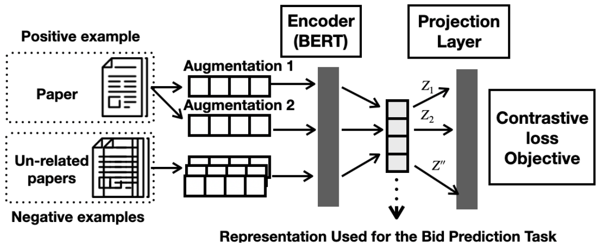

Figure 1: Our augmentation framework for self-supervised learning. The encoder (BERT model) learns to match the representation coming from two inputs, when they are basically the same text semantically.

Next sentence prediction This augmentation task is of course inspired by the pre-training tasks already being used for BERT pre-training (Devlin et al. 2018). We believe that if we train our self-supervised model to match the BERT representation of the first sentence, with the BERT representation of the immediately following sentence, it would force the encoder to learn the salient aspects of research publications. Usually a next sentence prediction task, implies predicting if a given sentence follows the immediate previous sentence. Due to the nature of scientific papers, all sentences in an abstract could be having close meanings, and a sentence not following a sentence immediately could still be closely related. It would be a problem for a Contrastive learning-based framework which needs negative samples. Thus at implementation time, to ensure that the self-supervised framework did not overfit, we picked one pair of positive samples as consecutive sentence pairs, and then random sentences picked from other different papers as negative pairs. The Next sentence augmentation task is very useful if we are short on data, since it makes the best use of the available data. For each paragraph, it can be split into multiple pairs of training samples.

LDA augmentation task: We have a trained model for LDA representations, using all the text of all submitted papers. This LDA model used unigrams, and hence removes any stylistic info. Any embedding prepared by this LDA model would capture the proportion of conference topics in a given text. We believe that this LDA embedding is a representation in low dimensional topic space and if we trained the model to match BERT's representation of the original text, with its LDA embedding, it would force the model to learn the most important summary information. Also since an LDA model would be aware of all the topics/keywords being talked about across all the other submission papers in the conference, it could be a good way to bring a factor resembling a global parameter factor. BERT representation is squeezed to $[\text{dim24}]$ to match the LDA topic vector dimension size of 24. We implemented this augmentation task, by preparing a 24 dimensional LDA vector embedding of the text, and then matching this with a squeezed 24 dimensional (from 768 to 24 dims) BERT representation of the original text.

Matching the representation of Abstract with the Introduction: We do this augmentation task with the belief that the introduction sections of a research paper, are at many times, the expanded version of what the research paper would have highlighted in its abstract section. Hence if we try to match the encoder representations between the introduction and abstract, we might learn a 'summarizer' type of encoder, which would be useful for our model.

Using augmentations for Contrastive loss objective Using the different augmentation tasks described earlier, we aimed to train the encoder, along with a projection layer on top (refer to Figure 1). The projection layer on top of the BERT transformer encoder squeezes the high dimensional (768 size), to a lower-dimensional vector representation (24 size) upon which we can apply a cosine similarity loss.

SimCLR loss is a form of *Contrastive loss*. SimCLR (Chen et al. 2020) loss is defined as the normalized temperature-scaled cross-entropy loss, or 'NT-Xent' (see Equation 1) . We have used this loss objective in our fine tuning regime of BERT transformer:

$$L(z_i, z_j) = -\log \frac{\exp(\frac{z_i z_j}{\tau})}{\sum_{k=1}^{2N} 1_{k \neq 1} \exp(\frac{z_i z_k}{\tau})}. \tag{1}$$

From the Equation1, we can observe that the numerator captures the cosine similarity values, z_i, z_j for a pair of samples i and j. The denominator is contrasting this with cosine similarity values between negative samples z_i, z_k, where k is any negative sample/example for our sample i. For a batch size of N, we end up with $2N - 1$ terms in the denominator, as for every sample $i \in N$ we have produced an augmented sample j. Here τ is a temperature value for the softmax function. We found that the value of $\tau = 0.5$ works well for our model runs. There is some discussion in self-supervised literature about the optimal temperature values, which could be useful for selecting hard negative examples during training, but we did not investigate in that direction.

Fine-tuning setup

To adapt our transformer models to the new evolving terminologies in AI/ML or computer science, we can use a transformer model trained on a very large generic language corpus and then fine-tune it for our specific conference submission papers corpus. In fact, NeurIPS 2019 had 6,810 submission papers. There were also 3,961 reviewers, each with their corpus of published papers, sometimes containing as many as 100 papers each.

Training a BERT model from scratch is a computationally expensive and time taking process. Hence we do not attempt to pre-train but only fine-tune the submission papers corpus for a conference. We envision this is how a program chair could use it when deploying this for an actual conference.

For our case, based on different experiments, the [CLS] token approach worked best among other feature extraction approaches. We chose to pass gradients through the parameters of the last two layers of the BERT model during our BERT fine-tuning phase. Ethayarajh (2019) state that representations of higher layers of BERT, store more context specific information.The same results have been observed by many others ((van Aken et al. 2019), (Rogers, Kovaleva, and Rumshisky 2020)), as usually BERT based for downstream tasks only try to fine tune or add to the final layers of BERT. We chose the last 2 layers to be fine-tuned since empirical experiments have shown that the final layers of BERT are often responsible for the context level meanings, while the initial layers focus more on word meanings. We believe this way, the BERT transformer would fine-tune to learn how to best summarize a sequence, using [CLS] token. In our use case, these text sequences are from research paper abstracts or introduction sections (up to 512 tokens).

We chose to use the BERT base model instead of BERT large which has more parameters, to have quicker runs, and shorter inference time. We load a pre-trained model from hugging face (Wolf et al. 2020) and attempt to fine tune it.[2] The authors reported using 3–5 epochs for most of their fine-tuning tasks (Devlin et al. 2018). Though larger models would have more model capacity and more expressive representations (Devlin et al. 2018).

From embeddings to score/bid prediction

Once our encoder is ready, we can use it to prepare paper representations for each of the publications individually. The individual paper representations for a reviewer are combined, to predict their bids against a submission paper. The flow of steps is as follows:

- Once the BERT model has trained for sufficient epochs, we can extract the encoder. We mention sufficient steps, as when sufficient data samples are present during the fine-tuning, the training loss showed a continuous downward trend. But one disadvantage of the self-supervised framework is that the performance can be evaluated when we report the final downstream task performance. Keeping this point in mind, we trained at two settings, epochs=50 and epochs=200 for self-supervised fine-tuning of BERT.
- The fine-tuned encoder can then be used to infer papers, both reviewer publications, and submission publications. These BERT embeddings are referred to as the paper representations.
- After the paper representations for all papers of a reviewer, we learn a function to combine the reviewer's corpus of research work. This information is then used along with the submission paper representation, for the bid prediction task.

Reviewer representations by Attention

In addition to having paper/document level representations, we also need reviewer-level representations derived from the set of papers they have authored. It is important to be able to combine the different research areas worked on by a researcher. It would be wrong to assume that a researcher would have worked only on a single topic. Some topic areas of researchers might be more represented in their corpus, than others in their publication history. A reviewer could

[2] https://huggingface.co/

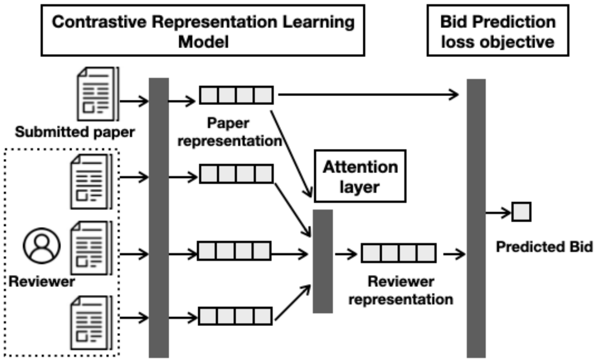

Figure 2: The pipeline for bid prediction task using the attention network. Each reviewer paper is attended to conditioned on the submission paper, to give a predicted bid.

have written several papers during the course of their academic life. Several times, the work that was written originally may not be of current interest to this reviewer.

The important difference we have to consider is that the model for getting reviewer representation, which is a combination of many reviewer papers, and the representation of a submission paper, which is based on a single paper, would be different.

Naive approach: A naive approach to combine different paper representations, would be to do an average of all the individual representations. We believe this is fundamentally problematic as:

- A most basic approach can be an average of all representations. If the different papers of a reviewer are focused on different areas, an average representation is going to lose all uniqueness information and bring them closer to an average value. To experimentally prove our intuitions, we have maintained a Simple Network model in all result tables as a baseline. This simple network is based on a mean of the reviewer paper representations, and is a rough approximation of the current TPMS.
- An averaging approach of reviewer papers is independent of submissions, and stays constant. An average representation could miss out on capturing topics of an individual paper closer to a submission paper, if the other reviewer interest areas are very different. We will introduce a method to conditionally aggregate information from a reviewer's papers based on a submission paper.

Attention Network We opted to model our combination based on an attention network. Initially, we were also tempted by the idea that an attention-based approach could be human interpretable, as the attention scores for each of the reviewer papers could be observed and interpreted.

To condition reviewer papers on submitted papers, we prepare context information. Looking at a fixed submitted paper, p_i, attend softly to one of the reviewer papers, r_j.

We propose the Attention Network as an attention scoring-based model, which learns to give a score to each of the reviewer's papers, conditioned on its relevance to the bidder submission paper. That is if a reviewer, r_j is to be judged for suitability for a submission paper p_i, the model computes $P(paper_{(k,r_j)} \mid p_i)$, where k is some paper in the corpus of reviewer r_j. Figure 2 shows our pipeline. Each reviewer paper is attended to by the attention layer, conditioned on a submission paper, to predict a bid score. To implement attention we opted for the general attention framework as highlighted by Vaswani et al. (2017). We can visualize this as a series of matrix manipulations where the model learns to weight different Key (K) values for a given Query (Q). Each of these values is attained using linear weight matrices, W_k, W_Q, W_v.

We experimented with different forms of this equation for our use case. We observed that the value V (obtained from $W_v *$ reviewer paper) in the original general attention did not improve performance. The modified equation which we used for our case:

$$Z = \text{Attention}(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}}) \qquad (2)$$

We experimentally verified that softmax was an essential component for the model, though changing temperature within the softmax to increase or decrease attention focus did not bring large performance changes. Since attention operates on fixed sized sequences, the dimension of the attention matrix is padded to the largest number of papers written by a reviewer so that the model training in a batch is simplified. Thus a reviewer representation R_k in our formulation, is always weighted with Z of dimension M , where M is the padding value ($Z \in \mathcal{R}^M$):

$$R_k^{\text{weighted}} = Z * R_k \qquad (3)$$

where $R_k^{\text{weighted}} \in \mathcal{R}^P$, and $\mathcal{R}^P$ is the dimension of each of the paper representations p_i.

We weigh the different reviewer papers by using the attention scores. Z gives the context weights for each reviewer paper, conditioned on the submission paper which this reviewer is bidding at the moment. Effectively Z is probabilistically choosing papers of the reviewer from all of their corpus, to decide whether they are an expert in the submission paper area or not.

Equation 4 describes the components of attention network. For a reviewer paper r_j belonging to a reviewer R_k, attention (attn) learns a probabilistic score z_k, conditioned against a submission paper p_i.

We observed through our ablation studies that the add, diff, and multi components were necessary to be able to achieve a good performance. The network is defined as:

$$\begin{aligned} z_k &= \text{attn}(r_j, p_i) \qquad \forall r_j \in R_k, \\ \text{add} &= p_i + (z_k * R_k), \\ \text{diff} &= p_i - (z_k * R_k), \\ \text{multi} &= p_i * (z_k * R_k), \end{aligned} \qquad (4)$$

Equation 5 shows how these different components 'add', 'diff' and 'multi' are combined using linear weights and

$\tanh$ non-linearity. At the end we use a output with linear weight *combo*, scaled with a scale factor for regression.

$$
\begin{aligned}
\text{combo} &= \tanh(W_{\text{add}} * \text{add}) + \tanh(W_{\text{diff}} * \text{diff}) \\
&\quad + \tanh(W_{\text{multi}} * \text{multi}), \\
\text{out} &= \text{scaling} * W_{combo}(\sigma(\text{combo}))
\end{aligned}
\tag{5}
$$

Through ablation studies, we observed that the tanh non-linearity is essential for modeling. We keep the scale factor at the end as 3, since the bid values are between $(0, 3)$ for both the data sets.

Our experiments show that this model gave the best performance amongst all possible models when used with LDA embeddings. We postulate that the low 25 dimensional embeddings of LDA captured a lot of useful semantic information, and it was much more suited to the relatively small-sized data available to us.

Simple linear network A simple linear network uses the mean of reviewer paper representations. Equation 6 shows our formulation, where a reviewer R_k, could have M papers (or padded till M number of papers) r_j ($R_k \in \mathcal{R}^M$). This is our baseline comparison model for reviewer expertise. It can also be seen as a rough approximation of the current TPMS model for reviewer representation.

Averaging nullifies the uniqueness of individual representations, and this model would be expected to perform poorer than an attention-based model.

$$
\begin{aligned}
\text{add} &= p_i + \frac{1}{M}\sum_j (r_j), \qquad\qquad r_j \in R_k, \\
\text{diff} &= p_i - \frac{1}{M}\sum_j (r_j), \\
\text{combo} &= \tanh(\text{add}) + \tanh(\text{diff}), \\
\text{out} &= \text{scaling} * W_{combo}(\sigma(combo))
\end{aligned}
\tag{6}
$$

Loss objective for the bid prediction task We model the bid prediction problem as a regression. In this work of ours, we often refer to this bid prediction task as ***the downstream task*** since we have kept the paper representation learning and prediction problems separate. The bid prediction task tries to anticipate the bid, which a reviewer would give on a submission paper, based on their self-attested expertise and interest.

The model predicts a continuous value within the range of bid values, using a scale factor. We use a mean squared error loss (Equation 7) as the objective:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{7}$$

where y_i is the actual bid value, and $\hat{y_i}$ its prediction.

Evaluation

In this section, we evaluate the effectiveness of our methods on papers collected from two peer-reviewed conferences, namely *NeurIPS 2019* and *PLDI 2021*. We first explain our experimental setup and then we present the results.

Experimental setup

NeurIPS 2019 *NeurIPS* (Neural Information Processing Systems) is a scientific conference on artificial intelligence and machine learning.https://nips.cc/ It is considered one of the most prestigious in its field, and is at the forefront of AI/ML research. We had access to the reviewer corpus, submission papers and reviewer bids for the NeurIPS 2019 conference.

For the NeurIPS 2019 conference, each of the reviewers can bid on any submitted paper based on their interests and expertise. The bid values can be either 0, 1, 2, or 3, with the interpretations: 0: Not willing to review, 1: Not willing, unless absolutely unavoidable, 2: Willing to review, 3: Eager to review. We treat these bid values as self-attested expertise elicitation, than interest alone. The dataset has $3,961$ reviewers, $6,810$ submissions and $788,371$ bids on papers by reviewers.

Bid value	Count	Percent share of Bids data
0	658,875	83.73%
1	28,164	3.57%
2	62,490	7.94%
3	37,320	4.74 %

Table 1: Original bid counts for NeurIPS 2019 dataset

Table 1 shows the distribution of bids. We noticed that bids are skewed towards 0. On further investigation, we found that there were a large number of reviewers giving 0 bid values on more than 300 papers at a time.

Pre-processing Interestingly we observed that 0 is the most common bid value used by reviewers. We observed that 98 reviewers bid 0 on more than $5,000$ submission papers. Other than this set of reviewers, we hypothesise that maybe most reviewers were very particular about not getting assigned papers they were not interested in, and 0 bid was a way to ensure that.

When we first trained our downstream model for the bid prediction task, on the original NeurIPS 2019 dataset, we could not achieve any success. Attempts to model bid prediction failed on this original dataset. Previous work (Charlin, Zemel, and Boutilier 2012) had highlighted the same by pointing out that bids are inherently noisy as (1) it is difficult for reviewers to offer reasonable assessments of all but a small fraction of the papers, given the large numbers involved and (2) reviewers usually have access to only title and abstract, which is very less information to make a judgment. We also found that not all reviewers have the same understanding of what bid values 0, 1, 2, or 3 mean. Hence leading to even more noise.

We then decided to use only area chairs (AC) reviewers bids data. For 332 AC reviewers, we had 67k data points (bids on papers by reviewers). We postulate that ACs being senior members behave more responsibly in the bidding process. If they notice something in their area of expertise, they would bid highly. If they are aware that a submission paper is out of their area of expertise they do not bid on it. Hence this makes the data less noisy, and easier to learn, if a model

Encoder model with **Simple Network**	NeurIPS 2019		PLDI 2021	
	Validation loss	Test loss	Validation loss	Test loss
Naive bidding	0.94	0.94	0.87	0.87
Vanilla BERT	0.51 (0.84, 0.84)	0.53 (0.84, 0.84)	0.73	0.68 ±0.044
SciBERT	0.83	0.83 ±0.02	0.70	0.69 ±0.05
LDA augmentation BERT	0.42	**0.44**	0.70	0.69 ±0.02
Next Sentence augmentation BERT	0.48	0.49	0.73	0.70 ±0.02
Bag of Words augmentation BERT	0.61	0.61	0.67	**0.67** ±0.03
Intro and abstract match augmentation BERT	0.60	0.60	0.68	0.67 ±0.04

Table 2: Results on NeurIPS 2019 and PLDI 2021 datasets with Simple Network. Encoder fine-tuned using SimCLR framework.

Encoder model with **Attention Net**	NeurIPS 2019		PLDI 2021	
	Validation loss	Test loss	Validation loss	Test loss
Naive bidding	0.94	0.94	0.87	0.87
Vanilla BERT	0.19	**0.19** ±0.00	0.45	**0.46** ±0.04
SciBERT	0.17	**0.17** ±0.01	0.50	0.51 ±0.05
LDA augmentation BERT	0.23	0.23 ±0.01	0.48	**0.47** ±0.03
Next Sentence augmentation BERT	0.20	0.21 ±0.02	0.60	0.59 ±0.04
Next Sentence augmentation 200 epochs BERT	0.18	**0.18** ±0.00	0.56	0.56 ±0.04
Bag of Words augmentation BERT	0.21	0.21 ±0.00	0.68	0.64 ±0.05
Intro and abstract match augmentation BERT	0.21	0.21	0.72	0.67 ±0.07

Table 3: Results on NeurIPS 2019 and PLDI 2021 datasets with attention network. Encoder fine-tuned using SimCLR framework. The best reported values are in bold. Notice that all values are much better than those achieved by the simple Net.

is trying to predict bids based on the assumption that people bid only on areas that they know about. For our BERT transformer fine-tuning, we were able to use the $6,810$ submission papers.

PLDI 2021 *PLDI* is a conference for programming languages and programming systems research.[3] The dataset that we have access to is for PLDI 2021. For this conference we have 70 reviewers and 1000 submission papers. Though we had access to reviewers corpus for only 67 reviewers.

We explored the dataset and saw that the distribution of bids is between -100 to 100. For this conference, -100 implies the least possible interest in a submission paper by a reviewer, and 100 reflects the highest possible interest. We observed that data was skewed heavily towards the negative side, implying people bid proactively to avoid getting assigned papers they are not interested in (which is similar to NeurIPs 2019 dataset).

Our final dataset had $5,500$ reviewer-submission bid pairs, from 67 reviewers and 320 submission papers. We can notice that this dataset is very small in comparison to the NeurIPS 2019 dataset, where we had about 10 times more data. This in turn affected our ability to fine-tune the BERT transformer which we will discuss in detail in the result section.

Pre-processing To use the same downstream network architecture as we use for the NeurIPS 2019 dataset, we performed bid normalization on the PLDI 2021 dataset to have the same range as NeurIPS 2019 dataset, i.e., between $0-3$, in two steps. In the first step, we make the data more symmetric. Upon data observation, we observed -20, -10, -5, 5, 10 and 15 as the most common bid values. -20 bid value was the most represented negative bid value. Also there were very few bid values lower than -20, implying most reviewers (except a very few) did not have the heart to rate -100 on a submission paper. We hypothesize that through an informal agreement, reviewers may be agreed to bid -20 to show their disinterest. We then transformed all values less than -20 to -20. On the positive bid side, we did not notice any clear peak value. Bid values greater than 20 were rare. Hence we restricted the 20 bid value, as the bid value reflecting the most interest. Then, in the second step, we use the $\text{Bid}_{Normalized} = 3 * \frac{(\text{Bid}+20)}{(40)}$ to convert bids in range $[-20, -20]$ to $[0, 3]$.

Evaluation setup

To judge the performance of our self-supervised models, we measure their performance on the downstream task, i.e., bid prediction. This is consistent with the self-supervised literature which tests the performance of models on their chosen downstream tasks (Chen et al. 2020). Hence all results are reported together as a combination of encoder model (BERT fine-tuned encoder using an augmentation task) combined with downstream prediction network.

Baseline methods: To judge the performance of our methods, we compare them with the following baseline methods:

[3] https://conf.researchr.org/home/pldi-2021

- **Naive Baseline**: the naive baseline is obtained when the bid prediction model always predicts the average bid value.
- **Vanilla BERT**: a pre-trained BERT transformer, with frozen parameters from the *Hugging Face library* (Wolf et al. 2020). We use it as a baseline in both setup, i.e., simple net and attention net.
- **SciBERT**: A pre-trained SciBERT transformer (Beltagy, Lo, and Cohan (2019)), with frozen parameters from the default model on Hugging Face.

Each experiment that we present in this section is a combination of:

- Choice of encoder: How the chosen encoder was fine-tuned on the self supervised framework. The chosen encoder is used for generating paper representations.
- Choice of the bid prediction model: We can use either the ***simple network***, or the ***attention network***.

The architecture used for the *simple network* is 1 hidden layer of size [768,1]. Similarly, we use 1 hidden layer of size [768,24] with attention layer for the *attention network*.

All results presented in the next section are collected in 3-fold cross-validation with Bayesian search over the hyperparameter space: learning rate, number of epochs, weight decay and batch size to find the best model. Since we cannot report results on the self-supervised fine-tuning step of BERT, we report MSE loss values on the downstream bid prediction task as the final metric.

Results on the downstream bid prediction task

In this section, we investigate the following research questions:

RQ1: To what extent does our proposed augmentation approach improve paper representation?

To answer this question we evaluate the baseline methods and our purposed self-supervised augmentation methods using the *simple network*.

Table 2 shows results on downstream tasks for both datasets. We can observe that the LDA augmentation task and the next sentence augmentation task were performing well for the NeurIPs dataset and they outperform all the baseline methods. For the PLDI 2021 dataset, the results of all augmentation approaches are close to Vanilla BERT and SciBERT, however the BOW augmentation and intro and abstract matching augmentation methods perform the best.

NeurIPS 2019 dataset has around $7,000$ submission papers available. We were able to use all of them for fine-tuning the BERT transformer model. We noticed that the size of the data for the fine-tuning task has a significant impact on the results. We were able to fine-tune the next sentence augmentation task with the BERT model till 200 epochs, with a continuous decrease in the training loss. This model makes the best use of the available data, converting $7,000$ submission papers, to $30,000$ training samples. The next sentence augmentation task is able to perform best on the Neurips dataset with MSE of 0.18.

For the NeurIPS 2019 dataset, LDA augmentation task was able to beat the Vanilla BERT. But this is not the case for the PLDI 2021 dataset. We believe that this could be related to the size of data available for fine-tuning. For the LDA augmentation task, we have $6,810$ training samples available at the fine-tuning time. We believed that since an LDA model had access to all submission papers during training, trying to match an LDA representation would bring information about the whole conference.

The swap order augmentation task was not very fruitful for fine-tuning. We can hypothesis that BERT transformers derive their power from their efficient use of word order and contextual word embeddings. Hence a swap order which effectively destroys the original sentence semantics cannot be a useful model to train an encoder for our case.

RQ2: What is the effect of using the attention network on performance?

We address this question by evaluating the baseline methods and our purposed self-supervised augmentation methods with our proposed *attention network*. Our proposed *attention network* provide better representations of reviewers given the submission paper.

Table 3 shows results on downstream tasks using the *attention Network*. Among all the augmentation task approaches for BERT fine-tuning, we observe that the *attention network* consistently outperforms the *simple network* for both datasets. This reinforces our belief that combining the reviewer representations from different papers is important, and cannot be done by just using an average representation, like how we did for a *simple network*. Using 200 epochs in the next sentence augmentation that we use for self-supervised fine-tuning on the BERT transformer, we were able to beat the Vanilla BERT model by a small margin for the NeurIPs dataset.

We notice clearly that the *attention network* is better over the *simple network* in this case over all the possible BERT fine-tuning approaches. If we compare these results with Table 2, we can see that *attention network* has consistently achieved much lower MSE values. Though Vanilla BERT has remained the best representation among all the encoders using the *attention network*, suggesting that fine-tuning transformers on a very small dataset such as PLDI 2021, are not sufficient but we can boost the performance by improving the reviewers representation significantly.

Finally, based on our observation of performance differences between NeurIPS 2019 and PLDI 2021, we can see that the next sentence augmentation approach is a safe choice for paper representation. It makes the best use of available data, and is suitable for smaller conferences (or other text corpus). In the presence of larger datasets, other approaches such as an LDA augmentation task could be more useful. This could be an important decision factor when choosing a fine-tuning approach for future conferences.

Conclusion

We showed during the course of our work on choice of paper representations and encoder that a better paper representation can improve the suitability prediction of reviewers on a submission paper. We were able to fine-tune a BERT transformer model to our corpus of conference submission papers, with a self-supervised framework.

This self-supervised framework removes our dependence on bids as signals for fine-tuning the BERT encoder. We also showed improving reviewers representation to better model their expertise given their resume is an important problem area to solve, and in this paper we proposed an attention network conditioned on the new submission papers, to build an expertise model.

Our results showed that the attention network outperforms the simple network-based model. We explored several approaches for augmentations in the self-supervised framework and were able to outperform the Vanilla BERT with attention network, by our next sentence augmentation BERT for NeurIPS 2019 dataset and LDA augmentation BERT for PLDI 2021 dataset. While the next sentence augmentation could be suitable for making the best use of available data for fine-tuning and our LDA augmentation model can leverage contextual information about the whole conference.

In this work, we use BERT as the transformer of choice. Though we are aware that BERT has its limitations when it comes to modeling longer text sequences. It would be interesting to see the performance of transformers more suitable for longer text sequences such as Reformers (Kitaev, Kaiser, and Levskaya 2020) or Big Bird (Zaheer et al. 2020). Another path to to explore in the future is to build explainable peer-review systems. Currently how the attention network chooses its scores is not very explainable. Finding out the limitations and advantages of assigning papers given reviewers expertise would be a good way to control for issues about fairness in paper-reviewer matching systems.

References

Beltagy, I.; Lo, K.; and Cohan, A. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3: 993–1022.

Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Säckinger, E.; and Shah, R. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04): 669–688.

Charlin, L.; and Zemel, R. 2013. The Toronto paper matching system: an automated paper-reviewer assignment system.

Charlin, L.; Zemel, R. S.; and Boutilier, C. 2012. A framework for optimizing paper matching. *arXiv preprint arXiv:1202.3706*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, 539–546. IEEE.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ethayarajh, K. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821*.

Hoffer, E.; and Ailon, N. 2015. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, 84–92. Springer.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Rogers, A.; Kovaleva, O.; and Rumshisky, A. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8: 842–866.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

van Aken, B.; Winter, B.; Löser, A.; and Gers, F. A. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1823–1832.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big Bird: Transformers for Longer Sequences. In *NeurIPS*.