Exploiting Labeled and Unlabeled Data via Transformer Fine-tuning for Peer-Review Score Prediction

Panitan Muangkammuen, Fumiyo Fukumoto, Jiyi Li, Yoshimi Suzuki
Integrated Graduate School of Medicine, Engineering,
and Agricultural Sciences, Faculty of Engineering
University of Yamanashi, Kofu, Japan
{g21dts04,fukumoto,jyli,ysuzuki}@yamanashi.ac.jp

Abstract

Automatic peer-review aspect score prediction (PASP) of academic papers can be a helpful assistant tool for both reviewers and authors. Most existing works on PASP utilize supervised learning techniques. However, the limited number of peer-review data deteriorates the performance of PASP. This paper presents a novel semi-supervised learning (SSL) method that incorporates the Transformer fine-tuning into the Γ-model, a variant of the Ladder network, to leverage contextual features from unlabeled data. Backpropagation simultaneously minimizes the sum of supervised and unsupervised cost functions, it can be easily trained in an end-to-end fashion. The proposed method is evaluated on the PeerRead benchmark. The experimental results demonstrate that our model outperforms the supervised and naive semi-supervised learning baselines. Our source codes are available online[1].

1 Introduction

Over the past few years, the number of submissions for AI-related international conferences and journals has increased substantially, making the review process more challenging. Automatic peer-review aspect score prediction (PASP) scores academic papers on a numeric range of different qualities along with aspects such as "clarity" and "originality". It can be a helpful assistant tool for both reviewers and authors. PeerRead is the first publicly available dataset of scientific peer reviews for research purposes (Kang et al., 2018). It can be used in various ways, such as paper acceptance classification (Ghosal et al., 2019; Maillette de Buy Wenniger et al., 2020; Fytas et al., 2021) and review aspect score prediction (Li et al., 2020; Wang et al., 2020). Alternatively, the dataset is modified for citation recommendation (Jeong et al., 2019) and citation count prediction (van Dongen et al., 2020).

Much of the previous work on PASP is based on supervised learning (Kang et al., 2018; Li et al., 2020). However, the dataset with annotated aspect scores is relatively very small, which deteriorates overall performance. To mitigate the drawback and improve the performance of PASP, we propose a semi-supervised learning (SSL) method that can leverage contextual features from the larger unannotated dataset. SSL has been widely utilized in many NLP tasks, such as classification (Miyato et al., 2016; Li et al., 2021), sequence labeling (Yasunaga et al., 2018; Chen et al., 2020), and parsing (Zhang and Goldwasser, 2020; Lim et al., 2020). It has shown to be effective for learning models by leveraging a large amount of unlabeled data to compensate for the lack of labeled data. SSL is also beneficial for PASP because an enormous body of publications is available online, and unlabeled data, i.e., scholarly papers, can often be obtained with minimal effort. Recently, transformer-based pre-training language models (LM) such as BERT (Devlin et al., 2019) and its variants have been very successful as many NLP tasks which utilize these LM attained unprecedented performances.

In this paper, we combine the strengths of both techniques and propose a Transformer-based Γ-model (Γ-Trans) that incorporates a pre-trained transformer into the Γ-model (Rasmus et al., 2015), a variant of ladder network (Valpola, 2014; Rasmus et al., 2015), SSL autoencoder. The unsupervised part of Γ-Trans utilizes a denoising autoencoder to help focus on relevant features derived from supervised learning. The contributions of our work can be summarized as follows:

- We propose Γ-Trans for PASP that incorporates a pre-trained transformer into SSL by fine-tuning the model using labeled and unlabeled data simultaneously.
- The experimental results show that Γ-Trans outperforms the supervised learning baselines

[1] https://github.com/panitan-m/gamma_trans

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2233–2240

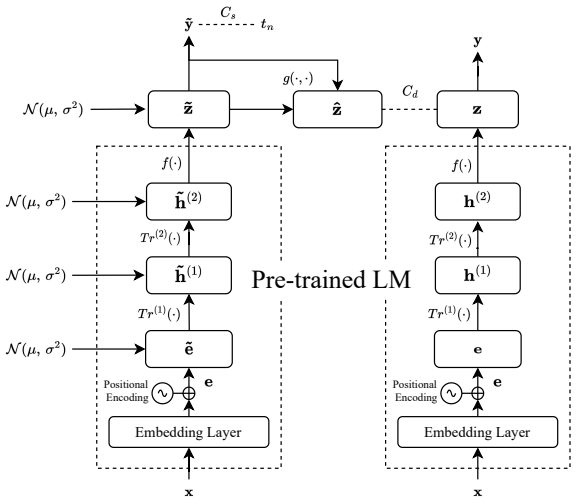

Figure 1: Γ-Trans architecture. The pre-trained transformer has two layers which are shown in a dotted frame. The model is fine-tuned by supervised cost C_s and denoising cost C_d

and naive SSL methods with a small amount of labeled training data.

- We compare several BERT variants and the size of unlabeled to examine the effectiveness of Γ-Trans for PASP.

2 Γ-Transformer

The existing works applying ladder networks to the NLP task, e.g., information extraction (Nagesh and Surdeanu, 2018) and sentiment analysis (Pan et al., 2020; Zheng et al., 2021). The latter utilizes the encoder of the ladder network (Rasmus et al., 2015) to extract the features from the pre-trained LM without fine-tuning it. By freezing the features from the LM, the model only utilizes the fully connected layers from the encoder of the network without exploiting the transformer layer of the LM. To mitigate the issue, we fine-tune the LM along with training the Γ-model as well as acquiring the sequence embedding from the pre-trained LM. The model can be plugged into any feedforward network without decoder implementation, i.e., the denoising cost is only on the top layer of the model.

Figure 1 illustrates the Γ-Trans network. Let $\mathbf{x}$ be the input and y be the output with targets t. The labeled training data of size N consists of pairs $\{\mathbf{x}(n), t(n)\}$, where $1 \leq n \leq N$. The unlabeled data of size M has only input $\mathbf{x}$ without the targets t, an $\mathbf{x}(n)$, where $N+1 \leq n \leq N+M$. As shown in Figure 1, the network consists of two forward passes, the clean path and the corrupted pass. The former is illustrated in a dotted frame on the right-hand side in Figure 1 and produces clean $\mathbf{z}$ and $\mathbf{y}$, which are given by:

$$\begin{aligned} \mathbf{z} &= f(\mathbf{h}^{(L)}) = N_B(\mathbf{W}\mathbf{h}^{(L)}) \\ \mathbf{y} &= \phi(\gamma(\mathbf{z}+\beta)) \end{aligned}$$

$$\begin{aligned} \mathbf{h}^{(0)} &= \mathbf{e} \\ \mathbf{h}^{(l)} &= Tr^{(l)}(\mathbf{h}^{(l-1)}), \end{aligned} \tag{1}$$

where $\mathbf{e}$ denotes the input embedding of $\mathbf{x}$ with positional encoding, $Tr^{(l)}$ refers to the transformer block at layer l in the L-layer pre-trained LM (e.g., BERT), and N_B indicates a batch normalization. $\mathbf{W}$ shows the weight matrix of the linear transformation f. ϕ refers to an activation function, where β and γ are trainable scaling and bias parameters, respectively.

The clean path shares the mappings $Tr^{(l)}$ and f with the corrupted path. The corrupted $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{y}}$ are produced by adding Gaussian noise $\mathbf{n}$ in the corrupted path (left-hand side of Figure 1):

$$\begin{aligned} \tilde{\mathbf{z}} &= f(\tilde{\mathbf{h}}^{(L)}) + \mathbf{n} \\ \tilde{\mathbf{y}} &= \phi(\gamma(\tilde{\mathbf{z}}+\beta)) \\ \tilde{\mathbf{h}}^{(0)} &= \tilde{\mathbf{e}} + \mathbf{n} \\ \tilde{\mathbf{h}}^{(l)} &= Tr^{(l)}(\tilde{\mathbf{h}}^{(l-1)}) + \mathbf{n}. \end{aligned} \tag{2}$$

A supervised cost C_s is the average negative log-probability of the noisy output $\tilde{\mathbf{y}}$ matching the target t_n given the input $\mathbf{x}_n$:

$$C_s = -\frac{1}{N}\sum_{n=1}^{N} \log P(\tilde{\mathbf{y}} = t_n | \mathbf{x}_n), \tag{3}$$

Aspect	#Pos (Neg)
Clarity (Clr)	40
Originality (Ori)	59
Impact (Imp)	22
Meaningful Comparison (Com)	52
Soundness/Correctness (Cor)	54
Substance (Sub)	66
Overall Recommendation (Ova)	60

Table 1: Statistics of the ACL Dataset. #Pos (Neg) refers to the equal number of papers for each class.

Metric	Aspect	Supervised Learning Methods				Semi-supervised Learning Methods			
		BERT	**PR**	**RR**	**Muti**	**VAT**	**Γ-model**	**Ladder**	**Γ-Trans**
Acc.	Clr	0.613	0.600	0.541	0.713	0.613	0.675	0.688	**0.763**
	Ori	0.508	0.593	0.659	0.525	0.593	0.559	0.610	**0.661**
	Imp	0.591	0.705	0.606	0.708	0.750	0.750	**0.841**	0.818
	Com	0.538	0.625	0.621	0.673	0.615	0.577	0.673	**0.692**
	Cor	0.546	0.639	0.529	0.509	0.556	0.648	0.657	**0.713**
	Sub	0.538	0.644	-	0.585	**0.667**	0.644	0.621	**0.667**
	Ova	0.575	0.625	0.535	**0.698**	0.658	0.558	0.683	0.683
	Avg.	0.559	0.633	0.582	0.630	0.636	0.630	0.682	**0.714**
F1	Clr	0.632	0.658	0.671	0.706	0.674	0.739	0.721	**0.790**
	Ori	0.605	0.652	0.651	0.675	0.529	0.606	0.642	**0.743**
	Imp	0.627	0.740	0.717	0.746	0.718	0.797	**0.864**	0.848
	Com	0.597	0.672	0.626	0.655	0.623	0.674	0.723	**0.729**
	Cor	0.606	0.655	0.588	0.615	0.529	0.687	0.698	**0.763**
	Sub	0.601	0.696	-	0.627	0.639	0.701	0.718	**0.747**
	Ova	0.608	0.693	0.520	0.682	0.637	0.663	0.732	**0.749**
	Avg.	0.611	0.681	0.629	0.672	0.621	0.695	0.728	**0.767**

Table 2: Experimental results. Best result is in bold and 2nd best is underlined.

where N denotes the number of labeled data. Given the corrupted $\tilde{\mathbf{z}}$ and prior information $\tilde{\mathbf{y}}$, the denoising function g reconstructs the denoised $\hat{\mathbf{z}}$:

$$\begin{aligned} \hat{\mathbf{z}} &= g(\tilde{\mathbf{z}}, \mathbf{u}) \\ \mathbf{u} &= N_B(\tilde{\mathbf{y}}), \end{aligned} \tag{4}$$

where g is identical to the one of Rasmus et al.'s (2015) consisting of its own learnable parameters. The unsupervised denoising cost function is given by:

$$C_d = \frac{1}{N+M} \sum_{n=1}^{N+M} \frac{\lambda}{d} \|\mathbf{z}_n - N_B(\hat{\mathbf{z}}_n)\|, \tag{5}$$

where M indicates the number of unlabeled data, λ is a coefficient for unsupervised cost, and d refers to the width of the output layer. The final cost C is given by:

$$C = C_s + C_d$$

3 Experiments

3.1 Experimental Settings

We performed the experiments on the ACL dataset with the score of review aspects that are included in the PeerRead Dataset (Kang et al., 2018). We used the mean score of multiple reviews and classified them ranging from 1 to 5 into two classes: ≥ 4 (Positive) and < 4 (Negative). We balanced the data, i.e., the same size of two classes, by randomly downsampling the majority class. Table 1 shows the statistics of the dataset. Although the PeerRead dataset contains both paper and review texts, we only used the papers to predict the aspect scores. We utilized the first 512 tokens of the paper according to the maximum length of the most common pre-trained LM, BERT (Devlin et al., 2019). For the unlabeled data, we also used the ACL papers obtained from ScisummNet Corpus [2] (Yasunaga et al., 2019), which provides 1,000 papers in the ACL anthology.

We used 5-fold cross-validation to evaluate all systems with an 80/20 split for the train and test sets. We selected the best model based on the performance of the test set. The final result is calculated from the average of the five folds. As the evaluation metric, we used accuracy and F1-score.

3.2 Baselines and Implementation Details

We compare Γ-Trans with supervised learning and semi-supervised learning baselines.

Supervised Learning

- **BERT-base** (Devlin et al., 2019) - A pre-trained LM. We fine-tuned the model on the PASP task.

[2] https://cs.stanford.edu/~myasu/projects/scisumm_net/

Model	Ladder	Γ-Trans
BERT	0.732	0.749
RoBERTa	0.694	0.712
SciBERT	**0.744**	**0.774**
Longformer	0.686	0.756

Table 3: F1 on *Overall recommendation* score prediction. Comparison between Ladder and Γ-Trans on different transformer-based pre-trained LMs.

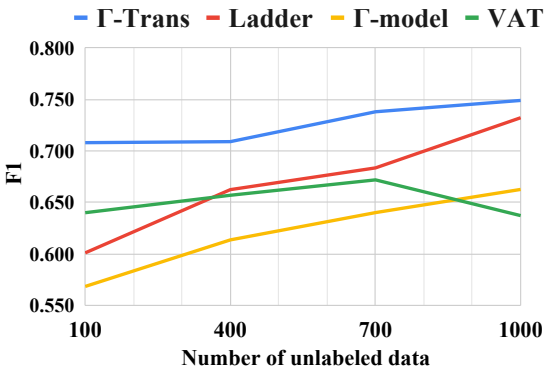

Figure 2: F1 score against the number of unlabeled data on *Overall recommendation* score prediction.

- **PeerRead (PR)** - Similar to Kang et al.'s (2018), we implemented a GRU (Gated Recurrent Unit) model (Cho et al., 2014) using GloVe [3] embeddings (Pennington et al., 2014) as input word representations without tuning.
- **ReviewRobot (RR)** (Wang et al., 2020) - This method extracts evidence by comparing the knowledge graph of the target paper and a large collection of background papers and uses the evidence to predict scores.
- **Multi-task** (Li et al., 2020) - A multi-task approach that automatically selects shared network structures and other review aspects as auxiliary resources. The model is based on CNN text classification model.

Semi-Supervised Learning

- **VAT** (Miyato et al., 2016) - This method exploits information from unlabeled data by applying perturbations to the word embeddings in a neural network.
- **Γ-model** (Rasmus et al., 2015) - It is a variant of ladder networks in which a denoising cost is only on the top layer and means that most of the decoder can be omitted.
- **Ladder** - A deep denoising autoencoder with skip connections and reconstruction targets in the intermediate layers (Rasmus et al., 2015).

The Γ-model and Ladder employ a ladder network on top of frozen BERT-base representations. Each baseline and the implementation details are shown in Appendix A Implementation details.

3.3 Results and Discussion

Table 2 shows the results. We can see from Table 2 that the SSL methods, Ladder and Γ-Trans, outperform all supervised learning baselines, and the results by Γ-Trans are the best among other SSL methods on average. This shows that our assumption, incorporating fine-tuning the pre-trained LM into the ladder network, helps improve the performance significantly. BERT has the worst performance and even performs worse than other supervised learning baselines that utilize a common neural network layer, GRU or CNN. It is probably because the number of supervised data alone is insufficient to tune millions of parameters of BERT.

Among the prediction of aspects, *Impact* aspect is the best score in both metrics. We investigated the distribution of each aspect score from the data and found that more than 60% of the papers whose impact score is ≥ 4 also have a score of ≥ 4 in other aspects, while other aspects are not. This indicates that the *Impact* aspect has relatively distinctive features compared with other aspects. In contrast, *Meaningful Comparison* score prediction has the worst performance.One possible reason is the limited length of the input sequence, i.e., the first 512 tokens. This data length includes abstract and introduction sections, but does not include related work section which deteriorates the performance of *Meaningful Comparison* score.

We recall that Γ-Trans fine-tunes the LM through training the ladder network. To examine how the LM affects the overall performance on PASP, we tested several pre-trained LMs. Table 3 shows the *Overall recommendation* score prediction by F1 obtained from several transformer-based pre-trained LMs with Γ-Trans and the second best method, Ladder. Our approach can generate better results in all models. We can see that SciBERT (Beltagy et al., 2019), a BERT model pre-trained on a large corpus of scientific publications, improves the performance, while RoBERTa (Liu et al., 2019)

[3]Common Crawl (840B tokens, 2.2M vocab, 300d vectors)

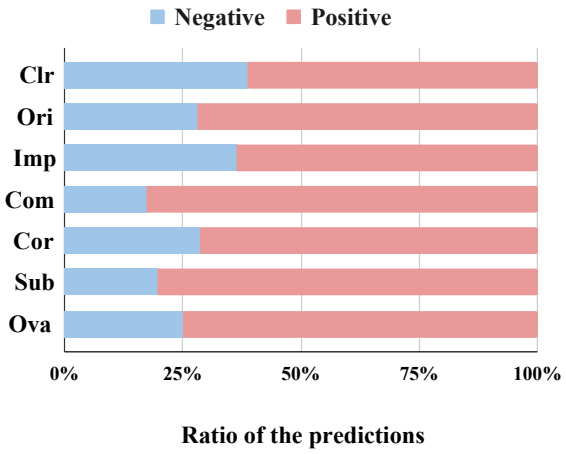

Figure 3: Ratio between the number of negative predictions and positive predictions of each aspect.

does not, compared to BERT. Table 3 also shows that Longformer[4] performs better than BERT on Γ-Trans, but not Ladder. This indicates that a longer sequence of textual information helps improve the performance of PASP. In contrast, Ladder does not work well with Longformer. One reason is that Ladder can not utilize the attention mechanism of Longformer for the different domains of ACL papers as it only employs the sequence embeddings obtained from the Longformer.

We also examined how the number of unlabeled data for training affects overall performance. Figure 2 shows the F1-score of the SSL methods against the number of unlabeled data obtained by 5-fold cross-validation. Overall, the graph shows that more unlabeled data helps improve the performance in every SSL method except VAT, whose performance drops at 1,000 unlabeled data. Γ-Trans consistently outperformed other SSL methods, and especially the result with 100 unlabeled data outperformed other methods with 700 unlabeled data.

3.4 Error Analysis

We analyzed the prediction probability on the *Overall Recommendation* aspect test data. The average probability of the selected class is 50.26% which is relatively low. The close probability of two classes indicates that the extracted features between the two classes are not much different from each other. The average probabilities of the correct and incorrect predictions are 50.30% and 50.13%, respectively, showing no significant difference.

[4]We used the first 1,000 tokens of each paper in the experiment.

Aspect	Neg Precision
Clarity	0.839
Originality	0.913
Impact	0.938
Meaningful Comparison	0.833
Soundness Correctness	0.870
Substance	0.923
Overall Recommendation	0.867

Table 4: Precision of negative samples

Figure 3 shows the ratio of the predictions between negative and positive. Our model tends to bias toward positive prediction in every aspect. The most biased prediction is *Meaningful Comparison*, with 84.31% on positive. One reason is that several reviewers are assigned to one paper. Assume that a sample labeled negative has a score of 3, 3, and 4. (The sample is labeled negative because the average of these scores is less than 4.) Such a sample has some positive features to trigger the model to predict it as positive. In contrast, there was no such case for positive samples.

We further investigated more on the negative predictions. Table 4 shows the precision of negative samples. Although our model predicts a positive outcome more than a negative one, the precision on the negative is very high. The highest precision is 0.938 on the *Impact* aspect and the lowest one is higher than 0.8. High precision on negative samples means a high measure of quality that indicates that our model is suitable for the first screen to filter out poor-quality works. Moreover, it is also helpful to authors for their first draft.

4 Conclusion

In this paper, we focused on the PASP task and proposed a method, Γ-Trans, that incorporates the Transformer fine-tuning technique into the Γ-model of the Ladder networks. The experimental results showed the effectiveness of our model as our model attained the best accuracy and F1 on average. Through the experiments, we found that our method helps improve the performance of all pretrained LMs including SciBERT and Longformer. Future work will include (i) extending the method for imbalanced aspect score datasets, (ii) exploiting the related information between aspects, and (iii) generating knowledgeable and explainable review comments.

Limitations

We should be able to obtain further advantages in efficacy in our pre-trained LM. We utilized the first 512 word tokens in the input paper and 768-dimensions of the hidden layer as most of the pre-trained LM restricts text length and embedding size which may lead to a lack of contextual information about aspects. Furthermore, in our experiment, fine-tuning Longformer by freezing the first ten layers on 1,000 tokens required around 50GB of GPU memory. We would improve our Γ-Trans model so that we can process papers consisting of long token sequences.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by JKA, JSPS Grant Number 21K12026.

References

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

Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8801–8811, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Panagiotis Fytas, Georgios Rizos, and Lucia Specia. 2021. What makes a scientific paper be accepted for publication? In *Proceedings of the First Workshop on Causal Inference and NLP*, pages 44–60, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Chanwoo Jeong, Sion Jang, Hyuna Shin, Eunjeong Park, and Sungchul Choi. 2019. A context-aware citation recommendation model with bert and graph convolutional networks.

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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Changchun Li, Ximing Li, and Jihong Ouyang. 2021. Semi-supervised text classification with balanced deep representation distributions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5044–5053, Online. Association for Computational Linguistics.

Jiyi Li, Ayaka Sato, Kazuya Shimura, and Fumiyo Fukumoto. 2020. Multi-task peer-review score prediction. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 121–126, Online. Association for Computational Linguistics.

KyungTae Lim, Jay Yoon Lee, Jaime Carbonell, and Thierry Poibeau. 2020. Semi-supervised learning on meta structure: Multi-task tagging and parsing in low-resource scenarios. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8344–8351.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Gideon Maillette de Buy Wenniger, Thomas van Dongen, Eleri Aedmaa, Herbert Teun Kruitbosch, Edwin A. Valentijn, and Lambert Schomaker. 2020. Structure-tags improve text classification for scholarly document quality prediction. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 158–167, Online. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification.

Ajay Nagesh and Mihai Surdeanu. 2018. Keep your bearings: Lightly-supervised information extraction with ladder networks that avoids semantic drift. 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 352–358, New Orleans, Louisiana. Association for Computational Linguistics.

Yuhao Pan, Zhiqun Chen, Yoshimi Suzuki, Fumiyo Fukumoto, and Hiromitsu Nishizaki. 2020. Sentiment analysis using semi-supervised learning with few labeled data. In *2020 International Conference on Cyberworlds (CW)*, pages 231–234.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks.

Harri Valpola. 2014. From neural pca to deep unsupervised learning.

Thomas van Dongen, Gideon Maillette de Buy Wenniger, and Lambert Schomaker. 2020. SChuBERT: Scholarly document chunks with BERT-encoding boost citation count prediction. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 148–157, Online. Association for Computational Linguistics.

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 the 13th International Conference on Natural Language Generation*, pages 384–397, Dublin, Ireland. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 976–986, New Orleans, Louisiana. Association for Computational Linguistics.

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

Xiao Zhang and Dan Goldwasser. 2020. Semi-supervised parsing with a variational autoencoding parser. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 40–47, Online. Association for Computational Linguistics.

Hang Zheng, Jianhui Zhang, Yoshimi Suzuki, Fumiyo Fukumoto, and Hiromitsu Nishizaki. 2021. Semi-supervised learning for aspect-based sentiment analysis. In *2021 International Conference on Cyberworlds (CW)*, pages 209–212.

A Implementation details

A.1 Fine-tuning BERT

We used Huggingface's Transformers package to fine-tune BERT. We fine-tuned the model with learning rate = 1e-6 until convergence with a batch size of 8, maximal sequence length of 512. Optimization was done using Adam with warm-up = 0.1 and weight decay of 0.01.

A.2 PeerRead model

We used a simple MLP with a single hidden layer of 100 neurons with the last recurrent state of a single GRU layer of 100 units. We trained the MLP until convergence, using Adam optimizer, a learning rate of 1e-4 with a batch size of 8 and an L2 penalty of 1.

A.3 VAT

A.3.1 Recurrent LM Pre-training

We used a unidirectional single-layer LSTM with 1,024 hidden units. The dimension of word embedding was 256. For the optimization, we used the Adam optimizer with a batch size of 32, an initial learning rate of 0.001, and a 0.9999 learning rate decay factor. We trained for 50 epochs. We applied gradient clipping with norm set to 5.0. We used dropout on the word embedding layer and an output layer with a 0.5 dropout rate.

A.3.2 Model Training

We added a hidden layer between the softmax layer for the target and the final output of the LSTM. The dimension is set to 30. For optimization, we also used the Adam optimizer, with a 0.001 initial learning rate and 0.9998 exponential decay. Batch sizes are set to 32 and 96 for calculating the loss of virtual adversarial training. We trained for 30 epochs. applied gradient clipping with the norm as 5.0.

A.4 Multi-task

We modified the model from performing a regression task to a classification task by changing the

output layer. We used CNN with 64 filters and filter width of 2. We used fastText as initial word embeddings. The hidden dimension was 1024. We trained the model using Adam optimizer with learning rate 0.001 and batch size of 8. We trained all of the candidate multi-task models for one and two auxiliary tasks to find the best one.

A.5 Γ-model and Ladder

We used the layer sizes of the ladder network to be 768-100-500-250-250-250-2, according to the BERT's representation dimension and the number of output classes. We set the denoising cost multipliers λ to [1000, 10, 0.1, 0.1, 0.1, 0.1, 0.1] from the input layer to the output layer for the Ladder, and [0, 0, 0, 0, 0, 0, 1] for the Γ-model. The std of the Gaussian corruption noise $\mathbf{n}$ is set to 0.3. We trained the model with a learning rate of 3e-3 until convergence with a batch size of 8 for each labeled and unlabeled data, 16 in total. Optimization was done using Adam with weight decay of 0.01.

A.6 Γ-Trans

We used Huggingface's Transformers package to fine-tune transformer-based pre-trained LMs. The denoising cost multipliers λ is set to 1. We set the std of the Gaussian corruption noise $\mathbf{n}$ to 0.3 in both Γ-model and Ladder. For optimization, we used the Adam optimizer, with a 1e-4 initial learning rate, 0.01 weight decay, and 0.1 warm-up. Batch size is set to 8 for both labeled and unlabeled data, 16 in total.