DeepSentiPeer: Harnessing Sentiment in Review Texts To Recommend Peer Review Decisions

Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
(tirthankar.pcs16,rajeev.ee15,asif,pb)@iitp.ac.in

Abstract

Automatically validating a research artefact is one of the frontiers in Artificial Intelligence (AI) that directly brings it close to competing with human intellect and intuition. Although criticized sometimes, the existing peer review system still stands as the benchmark of research validation. The present-day peer review process is not straightforward and demands profound domain knowledge, expertise, and intelligence of human reviewer(s), which is somewhat elusive with the current state of AI. However, the peer review texts, which contains rich sentiment information of the reviewer, reflecting his/her overall attitude towards the research in the paper, could be a valuable entity to predict the acceptance or rejection of the manuscript under consideration. Here in this work, we investigate the role of reviewers sentiments embedded within peer review texts to predict the peer review outcome. Our proposed deep neural architecture takes into account three channels of information: the paper, the corresponding reviews, and the review polarity to predict the overall recommendation score as well as the final decision. We achieve significant performance improvement over the baselines ($\sim$ 29% error reduction) proposed in a recently released dataset of peer reviews. An AI of this kind could assist the editors/program chairs as an additional layer of confidence in the final decision making, especially when non-responding/missing reviewers are frequent in present day peer review.

1 Introduction

The rapid increase in research article submissions across different venues is posing a significant management challenge for the journal editors and conference program chairs[1]. Among the load of works like assigning reviewers, ensuring timely receipt of reviews, slot-filling against the non-responding reviewer, taking informed decisions, communicating to the authors, etc., editors/program chairs are usually overwhelmed with many such demanding yet crucial tasks. However, the major hurdle lies in to decide the acceptance and rejection of the manuscripts based on the reviews received from the reviewers.

The quality, randomness, bias, inconsistencies in peer reviews is well-debated across the academic community (Bornmann and Daniel, 2010). Due to the rise in article submissions and non-availability of expert reviewers, editors/program chairs are sometimes left with no other options than to assign papers to the novice, out of domain reviewers which sometimes results in more inconsistencies and poor quality reviews. To study the arbitrariness inherent in the existing peer review system, organisers of the NIPS 2014 conference assigned 10% submissions to two different sets of reviewers and observed that the two committees disagreed for more than quarter of the papers (Langford and Guzdial, 2015). Again it is quite common that a paper rejected in one venue gets the cut in another with little or almost no improvement in quality. Many are of the opinion that the existing peer review system is fragile as it only depends on the view of a selected few (Smith, 2006). Moreover, even a preliminary study into the inners of the peer review system is itself very difficult because of data confidentiality and copyright issues of the publishers. However, the silver lining is that the peer review system is evolving with the likes of OpenReviews[2], author response periods/rebuttals, increased effective communications between authors and reviewers, open access initiatives, peer review workshops, review forms with

[1] Apparently CVPR, NIPS, AAAI 2019 received over 5100, 4900, 7000 submissions respectively!

[2] https://openreview.net

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1120–1130
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

objective questionnaires, etc. gaining momentum.

The PeerRead dataset (Kang et al., 2018) is an excellent resource towards research and study on this very impactful and crucial problem. With our ongoing effort towards the development of an Artificial Intelligence (AI)-assisted peer review system, we are intrigued with: *What if there is an additional AI reviewer which predicts decisions by learning the high-level interplay between the review texts and the papers? How would the sentiment embedded within the review texts empower such decision-making?* Although editors/program chairs usually go by the majority of the reviewer recommendations, they still need to go through all the review texts corresponding to all the submissions. A good use case of this research would be: slot-filling the missing reviewer, providing an additional perspective to the editor in cases of contrasting/borderline reviews. This work in no way attempts to replace the human reviewers; instead, we are intrigued to see how an AI can act as an additional reviewer with inputs from her human counterparts and aid the decision-making in the peer review process.

We develop a deep neural architecture incorporating full paper information and review text along with the associated sentiment to predict the acceptability and recommendation score of a given research article. We perform two tasks, a classification (predicting accept/reject decision) and a regression (predicting recommendation score) one. The evaluation shows that our proposed model successfully outperforms the earlier reported results in PeerRead. We also show that the addition of review sentiment component significantly enhances the predictive capability of such a system.

2 Related Work

Artificial Intelligence in academic peer review is an important yet less explored territory. However, with the recent progress in AI research, the topic is gradually gaining attention from the community. Price and Flach (2017) did a thorough study of the various means of computational support to the peer review system. Mrowinski et al. (2017) explored an evolutionary algorithm to improve editorial strategies in peer review. The famous Toronto Paper Matching system (Charlin and Zemel, 2013) was developed to match paper with reviewers. Recently we (Ghosal et al., 2018b,a) investigated the impact of various features in the editorial pre-screening process. Wang and Wan (2018) explored a multi-instance learning framework for sentiment analysis from the peer review texts. We carry our current investigations on a portion of the recently released PeerRead dataset (Kang et al., 2018). Study towards automated support for peer review was otherwise not possible due to the lack of rejected paper instances and corresponding reviews. Our approach achieves significant performance improvement over the two tasks defined in Kang et al. (2018). We attribute this to the use of deep neural networks and augmentation of review sentiment information in our architecture.

3 Data Description and Analysis

The PeerRead dataset consists of papers, a set of associated peer reviews, and corresponding accept/reject decisions with aspect specific scores of papers collected from several top-tier Artificial Intelligence (AI), Natural Language Processing (NLP) and Machine Learning (ML) conferences. Table 1 shows the data we consider in our experiments. We could not consider NIPS and arXiv portions of PeerRead due to the lack of aspect scores and reviews, respectively. For more details on the dataset creation and the task, we request the readers to refer to Kang et al. (2018). We further use the submissions of ICLR 2018, corresponding reviews and aspect scores to boost our training set for the decision prediction task. One motivation of our work stems from the finding that aspect scores for certain factors like *Impact*, *Originality*, *Soundness/Correctness* which are seemingly central to the merit of the paper, often have very low correlation with the final recommendation made by the reviewers as is made evident in Kang et al. (2018). However, from the heatmap in Figure 1 we can see that the reviewer's sentiments (compound/positive) embedded within the review texts have visible correlations with the aspects like *Recommendation*, *Appropriateness* and *Overall Decision*. This also seconds our recent finding that determining the scope or appropriateness of an article to a venue is the first essential step in peer review (Ghosal et al., 2018a). Since our study aims at deciding the fate of the paper, we take predicting recommendation score and overall decision as the objectives of our investigation. Thus our proposal to augment sentiment of reviews to the deep neural architecture seems intuitive.

Venues	#Papers	#Reviews	Aspect	Acc/Rej
ICLR 2017	427	7270	Y	172/255
ACL 2017	137	275	Y	88/49
CoNLL 2016	22	39	Y	11/11
ICLR 2018	909	2741	Only Rec	336/573
Total	1495	10325	–	607/888

Table 1: Dataset Statistics

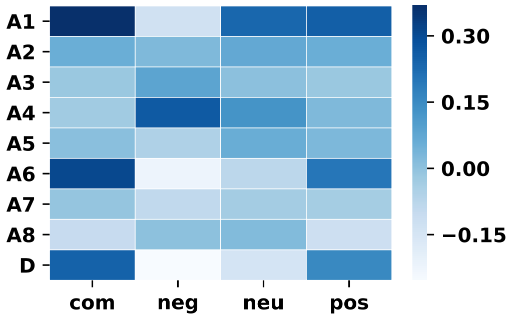


Figure 1: Pearson Correlation of Review Sentiment (:X) with different Aspect Scores (:Y) on ACL 2017 dataset. A1→Appropriateness, A2→Clarity, A3→Impact, A4→Meaningful Comparison, A5→Originality, A6→Recommendation, A7→Soundness/Correctness, A8→Substance, D→Decision. pos→Positive Sentiment Score, neg→Negative Sentiment Score, neu→Neutral Sentiment Score, com→Compound Sentiment Score. To calculate the sentiment polarity of a review text, we take the average of the sentence wise sentiment scores from Valence Aware Dictionary and sEntiment Reasoner (VADER) (Hutto and Gilbert, 2014).

4 Methodology

4.1 Pre-processing

At the very beginning, we convert the papers in PDF to .json encoded files using the Science Parse[3] library.

4.2 *DeepSentiPeer Architecture*

Figure 2 illustrates the overall architecture we employ in our investigation. The left segment is for the decision prediction while the right segment predicts the overall recommendation score.

4.2.1 Document Encoding

We extract full-text sentences from each research article and represent each sentence $s_i \in \mathbb{R}^d$ using the Transformer variant of the Universal Sentence Encoder (USE) (Cer et al., 2018), d is the dimension of the sentence semantic vector which is 512. A paper is then represented as,

$$\mathbf{P} = \mathbf{s}_1 \oplus \mathbf{s}_2 \oplus ... \oplus \mathbf{s}_{n_1}, \mathbf{P} \in \mathbb{R}^{\,n_1 \times d}$$

$\oplus$ being the concatenation operator, n_1 is the maximum number of sentences in a paper text in the entire dataset (padding is done wherever necessary). Similarly, we do this for each of the reviews and create a review representation as

$$\mathbf{R} = \mathbf{s}_1 \oplus \mathbf{s}_2 \oplus ... \oplus \mathbf{s}_{n_2}, \mathbf{R} \in \mathbb{R}^{\,n_2 \times d}$$

n_2 being the maximum number of sentences in the reviews.

4.2.2 Sentiment Encoding

The sentiment encoding of the review is done using VADER Sentiment Analyzer. For a sentence s_i, VADER gives a vector $\mathbf{S}_i$, $\mathbf{S}_i \in \mathbb{R}^4$. The review is then encoded (padded where necessary) for sentiment as

$$\mathbf{r}_{senti} = \mathbf{S}_1 \oplus \mathbf{S}_2 \oplus ... \oplus \mathbf{S}_{n_2}, \mathbf{r}_{senti} \in \mathbb{R}^{n_2 \times 4}.$$

4.2.3 Feature Extraction with Convolutional Neural Network

We make use of a Convolutional Neural Network (CNN) to extract features from both the paper and review representations. CNN has shown great success in solving the NLP problems in recent years. The convolution operation works by sliding a filter $\mathbf{W}_{f_k} \in \mathbb{R}^{l \times d}$ to a window of length l, the output of such h^{th} window is given as,

$$\mathbf{f}^k_h = g(\mathbf{W}_{f_k} \cdot \mathbf{X}_{h-l+1:h} + b_k)$$

$\mathbf{X}_{h-l+1:h}$ means the l sentences within the h^{th} window in Paper $\mathbf{P}$. b_k is the bias for the k^{th} filter, $g()$ is the non-linear function. The feature map $\mathbf{f}^k$ for the k^{th} filter is then obtained by applying this

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

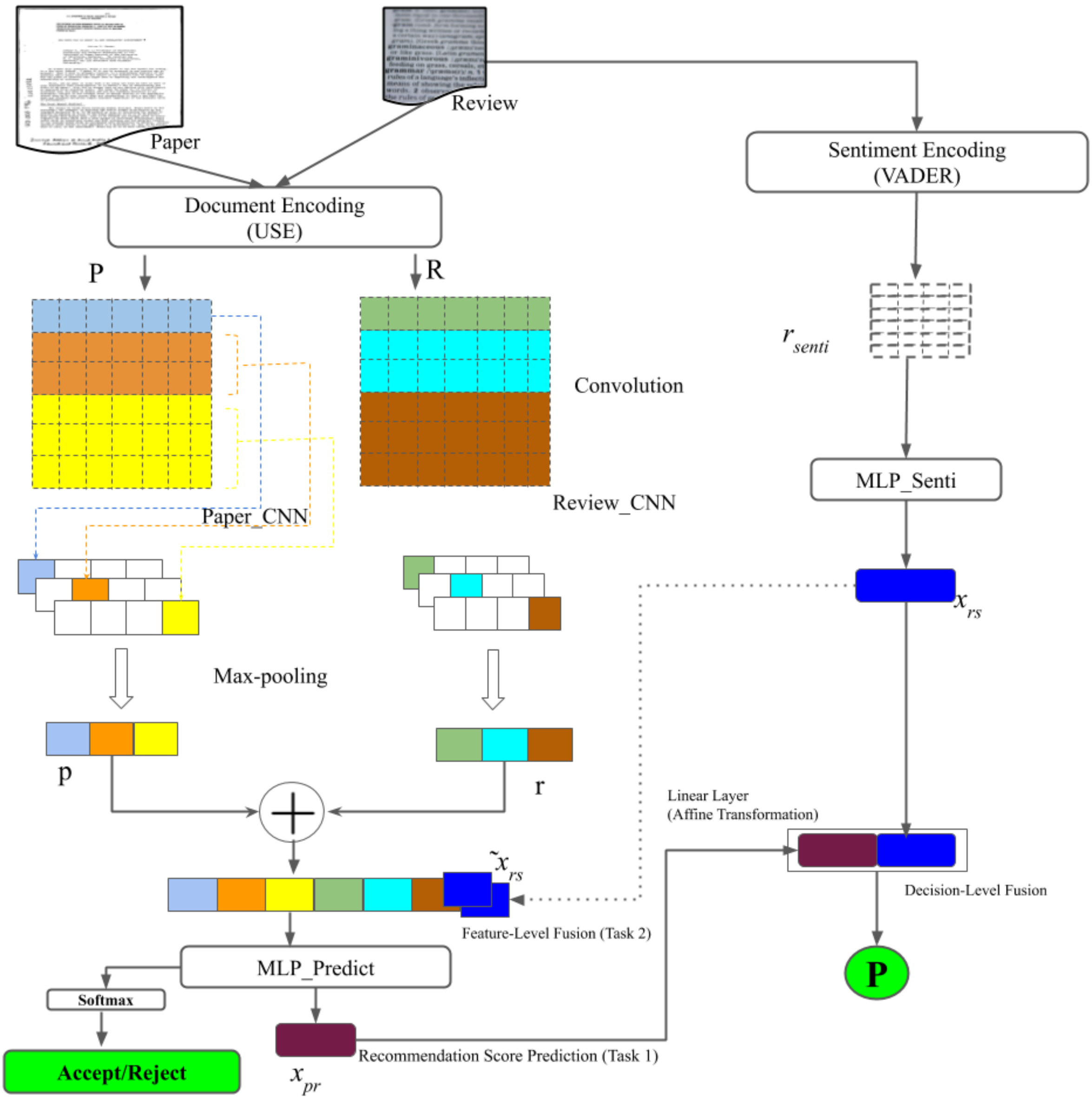


Figure 2: *DeepSentiPeer*: A Sentiment Aware Deep Neural Architecture to Predict Reviewer Recommendation Score. Decision-Level Fusion and Feature-Level Fusion of Sentiment are shown for Task 1 and Task 2, respectively.

filter to each possible window of sentences in the **P** as

$$\mathbf{f}^k = [\mathbf{f}_1^k, \mathbf{f}_2^k, ..., \mathbf{f}_h^k, ..., \mathbf{f}_{n_1-l+1}^k], \mathbf{f}^k \in \mathbb{R}^{n_1-l+1}.$$

We then apply a max-pooling operation to this filter map to get the most significant feature, $\hat{\mathbf{f}^k}$ as $\hat{\mathbf{f}^k} = max(\mathbf{f}^k)$. For a paper **P**, the final output of this convolution filter is then given as

$$\mathbf{p} = [\hat{\mathbf{f}^1}, \hat{\mathbf{f}^2}, ..., \hat{\mathbf{f}^k}, ..., \hat{\mathbf{f}^F}], \mathbf{p} \in \mathbb{R}^F,$$

F is the total number of filters used. In the same way, we can get ***r*** as the output of the convolution operator for the Review **R**.

We call the outputs ***p*** and ***r*** as the high-level representation feature vector of the paper and the review, respectively. We then concatenate these feature vectors (Feature-Level Fusion). The reason we extract features from both is to simulate the editorial workflow, wherein ideally, the editor/chair would look at both into the paper and the corresponding reviews to arrive at a judgement.

4.2.4 Multi-layer Perceptron

We employ a Multi-Layer Perceptron (*MLP_Predict*) to take the joint paper+review representations $\mathbf{x}_{pr}$ as input to get the final

	Task 1 →	**Aspect Score Prediction (RMSE)**		
	Test Datasets →	**ICLR ‡**	**ACL †**	**CoNLL †**
	Approaches ↓	**2017**	**2017**	**2016**
Baselines	Majority Baseline	1.6940	2.7968	2.9133
	Mean Baseline	1.6095	2.4900	2.6086
Comparing Systems	Only Paper (Kang et al., 2018)	1.6462	2.7278	3.0591
	Only Review (Kang et al., 2018)	1.6955	2.7062	2.7072
	Paper+Review (Kang et al., 2018)	1.6496	2.5011	2.9734
Proposed Architecture *DeepSentiPeer*	Only Review	1.5812	2.7191	2.6537
	Review+Sentiment	**1.4521**	2.6845	**2.5524**
	Paper+Review+Sentiment	**1.1679**	**2.3790**	**2.5399**

Table 2: Results on Aspect Score Prediction Task. Training is done with only ICLR 2017 papers/reviews, † → Cross-Domain: Training on ICLR and testing upon entire data of ACL/CoNLL available in PeerRead dataset, ‡ → Test set is kept the same as (Kang et al., 2018), RMSE→Root Mean Squared Error. CNN variant as in (Kang et al., 2018) is used as the comparing system.

representation as

$$\mathbf{x}_{pr} = f_{MLP\_Predict}(\theta_{predict}; [\mathbf{p}, \mathbf{r}]),$$

where $\theta_{predict}$ represents the parameters of the *MLP_Predict*. We also extract features from the review sentiment representation $\mathbf{x}_{rs}$ via another MLP (*MLP_Senti*).

$$\mathbf{x}_{rs} = f_{MLP\_Senti}(\theta_{senti}; \mathbf{r}_{senti}),$$

θ_{senti} being the parameters of *MLP_Senti*. Finally, we fuse the extracted review sentiment feature and joint paper+review representation together to generate the overall recommendation score (Decision-Level Fusion) using the affine transformation as

$$prediction = (\mathbf{W}_d \cdot [\mathbf{x}_{pr}, \mathbf{x}_{rs}] + b_d).$$

We minimize the Mean Square Error (MSE) between the actual and predicted recommendation score. The motivation here is to augment the human judgement (review+embedded sentiment) regarding the quality of a paper in decision making. The long-term objective is to have the AI learn the notion of good and bad papers from the human perception reflected in peer reviews in correspondence with paper full-text.

4.2.5 Accept/Reject Decisions

Instead of training the deep network on overall recommendation scores, we train the network with the final decisions of the papers in a classification setting. The entire setup is same but we concatenate all the reviews of a particular paper together to get the review representation. And rather than doing decision-level fusion, we perform feature-level fusion where the decision is given as

$$\mathbf{x}_{prs} = f_{MLP\_Predict}(\theta; [\mathbf{p}, \widetilde{\mathbf{r}}, \widetilde{\mathbf{x}}_{rs}])$$

$$\mathbf{c} = Softmax(\mathbf{W}_c \cdot \mathbf{x}_{prs} + b_c),$$

where $\mathbf{c}$ is the output classification distribution across accept or reject classes. $\widetilde{\mathbf{r}}$ is the high-level representation of review text after concatenating all reviews corresponding to a paper and $\widetilde{\mathbf{x}}_{rs}$is the output of *MLP_Senti* on the concatenated review text. We minimize Cross-Entropy Loss between predicted $\mathbf{c}$ and actual decisions.

4.3 Experimental Setup

As we mention earlier, we undertake two tasks:

Task 1: *Predicting the overall recommendation score* (Regression) and

Task 2: *Predicting the Accept/Reject Decision* (Classification).

To compare with Kang et al. (2018), we keep the experimental setup (train vs test ratio) identical and re-implement their codes to generate the comparing figures. However, Kang et al. (2018) performed Task 2 on ICLR 2017 dataset with handcrafted features, and Task 1 in a deep learning setting. Since our approach is a deep neural network based, we crawl additional paper+reviews from ICLR 2018 to boost the training set.

For Task 1, n_1 is 666 and n_2 is 98 while for Task 2, n_1 is 1494 and n_2 is 525. We employ a grid search for hyperparameter optimization. For Task 1, F is 256, l is 5. ReLU is the non-linear function $g()$, learning rate is 0.007. We train the model with SGD optimizer, set momentum as 0.9

	Task 2 →	**Accept/Reject (Accuracy)**		
	Test Datasets →	**ICLR ‡**	**ACL †**	**CoNLL †**
	Approaches ↓	**2017**	**2017**	**2016**
Baseline	Majority Baseline	60.52	33.33	39.94
Comparing System	Only Paper (Kang et al., 2018)	55.26*	35.93	41.23
Proposed Architecture ***DeepSentiPeer***	Only Review	65.35	57.12	62.91
	Review+Sentiment	**69.79**	**59.31**	**62.22**
	Paper+Review+Sentiment	**71.05**	**64.76**	**67.71**

Table 3: Results on Accept/Reject Classification Tasks. Training is done with ICLR 2017+ICLR 2018 papers/reviews, † → Cross-Domain: Training on ICLR and testing upon the entire data of ACL/CoNLL, ‡Test Set is kept the same as (Kang et al., 2018), RMSE→Root Mean Squared Error, $*$ →65.79% if only trained with ICLR 2017, Comparing System (Kang et al., 2018) is feature-based and considers only paper, and not the reviews.

and batch size as 32. We keep dropout at 0.5. We use the same number of filters with the same kernel size for both paper and review. In Task 2, for Paper_CNN F is 128, l is 7 and for Review_CNN F is 64 and l is 5. Again we train the model with Adam Optimizer, keep the batch size as 64 and use 0.7 as the dropout rate to prevent overfitting. We intentionally keep our CNN/MLP shallow due to less training data. We make our codes [4] available for further explorations.

5 Results and Analysis

Table 2 and Table 3 show our results for both the tasks. We propose a simple but effective architecture in this work since our primary intent is to establish that a sentiment-aware deep architecture would better suit these two problems. For **Task 1**, we can see that our review sentiment augmented approach outperforms the baselines and the comparing systems by a wide margin ($\sim$ 29% reduction in error) on the ICLR 2017 dataset. With only using review+sentiment information, we are still able to outperform Kang et al. (2018) by a margin of 11% in terms of RMSE. A further relative error reduction of 19% with the addition of paper features strongly suggests that only review is not sufficient for the final recommendation. A joint model of the paper content and review text (the human touch) augmented with the underlying sentiment would efficiently guide the prediction.

For **Task 2**, we observe that the handcrafted feature-based system by Kang et al. (2018) performs inferior compared to the baselines. This is because the features were very naive and did not address the complexity involved in such a task. We perform better with a relative improvement of 28% in terms of accuracy, and also our system is end-to-end trained. Presumably, to some extent, our deep neural network learned to distinguish between the probable accept versus probable reject by extracting useful information from the paper and review data.

5.1 Cross-Domain Experiments

With the additional (but less) data of ACL 2017 and CoNLL 2016 in PeerRead, we perform the cross-domain experiments. We do training with the ICLR data (core Machine Learning papers) and take the test set from the NLP conferences (ACL/CoNLL). NLP nowadays is mostly machine learning (ML) centric, where we find several applications and extensive usage of ML algorithms to address different NLP problems. Here we observe a relative error reduction of 4.8% and 14.5% over the comparing system for ACL 2017 and CoNLL 2016, respectively (Table 2). For the decision prediction task, the comparing system performs even worse, and we outperform them by a considerable margin of 28% (ACL 2017) and 26% (CoNLL 2017), respectively (Table 3). The reason is that the work reported in Kang et al. (2018) relies on elementary handcrafted features extracted only from the paper; does not consider the review features whereas we include the review features along with the sentiment information in our deep neural architecture. However, we also find that our approach with only Review+Sentiment performs inferior to the Paper+Review method in Kang et al. (2018) for ACL 2017. This again seconds that inclusion of paper is vital in recommendation decisions. Only paper is enough for a human reviewer, but with the current state of AI, an AI

[4] https://github.com/aritzzz/DeepSentiPeer

reviewer would need the supervision of her human counterparts to arrive at a recommendation. So our system is suited to cases where the editor needs an additional judgment regarding a submission (such as dealing with missing/non-responding reviewers, an added layer of confidence with an AI which is aware of the past acceptances/rejections of a specific venue).

6 Analysis: Effect of Sentiment on Reviewer's Recommendation

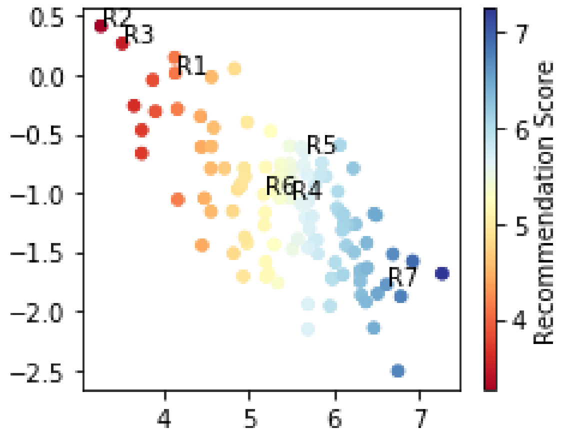


Figure 3: Projections of the output activations of the final layer of *MLP_Senti*. Points are annotated for Reviews from Table 4. X: Predicted Recommendation Scores, Y: Sentiment Activations

Scores	PC
Actual vs Prediction	0.97
Prediction vs Sentiment Activations	-0.93
Actual vs Sentiment Activations	-0.91

Table 4: Pearson Correlation (PC) Coefficient between the *Recommendation Scores* and *Sentiment Activations*. This is to account for the fact that sentiment is actually correlated with the prediction signifying the strength of the model.

Figure 3 shows the output activations[5] from the final layer of *MLP_Senti* against the predicted recommendation scores. We can see that the papers are discriminated into visible clusters according to their recommendation scores. This proves that *DeepSentiPeer* can extract useful features in close correspondence to human judgments. From Figure 3 and Table 4, we see that the sentiment activations are strongly correlated (negatively) with the actual and predicted recommendation scores. Therefore, we hypothesize that our model draws considerable strength if the review text has proper sentiment embedded in it. To further investigate this, we sample the papers/reviews from the ICLR 2017 test set. We consider actual review text and the sentiment embedded therein to examine the performance of the system (See Table 5). We truncate the lengthy review texts and provide the OpenReview links for reference. Appendix A shows the heatmaps of Vader sentiment scores generated for individual sentences corresponding to each paper review in Table 5. We hereby acknowledge that since the scholarly review texts are mostly objective and not straightforward, the score for neutral polarity is strong as opposed to positive, and negative. But still, we can see visible polarities for review sentences which are positive or negative in sentiment. For instance, the second last sentence(s9): *"The paper is not well written either"* from R1 has visible negative weight in the heatmap (Figure 5 in Appendix A). Same can be observed for the other review sentences as well.

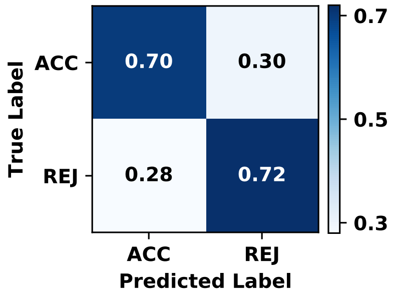


Figure 4: Normalized Confusion Matrix for Accept/Reject Decisions on ICLR 2017 test data with *DeepSentiPeer*(Paper+Review+Sentiment) model.

Besides the objective evaluation of the paper in the peer reviews, the reviewer's opinion in the peer review text holds strong correspondence with the overall recommendation score. We can qualitatively see that the reviews R1, R2, and R3 are polarized towards the negative sentiment (Table 5). Our model can efficiently predict a reasonable recommendation score with respect to human judgment. Same we can say for R7 where the review mostly signifies a positive sentiment polarity. R6 provides an interesting observation. We see that the review R6 is not very expressive for such a

[5] We call them as Sentiment Activations

#	Paper Title	Review Text	Prediction	Actual	Senti_Act
R1	*Multi-label learning with the RNNs for Fashion Search*	—The technical contribution of this paper is not clear. Most of the approaches used are standard state-of-art methods and there are not much novelties. For a multi-label recognition task, there are other available methods, e.g. using binary models, changing cross-entropy loss function, etc. There is not any comparison between the RNN method and other simple baselines. The order of the sequential RNN prediction is not clear either. It seems that the attributes form a tree hierarchy, and that is used as the order of sequence. The paper is not well written either.— https://openreview.net/forum?id=HyWDCXjgx&noteId=B1Mp8grVl	4	3	0.01
R2	*Transformation based Models of Video Sequences*	—While I agree with the authors on these points, I also find that the paper suffer from important flaws. Specifically: -the choice of not comparing with previous approaches in term of pixel prediction error seems very ”convenient”, to say the least. While it is clear that the evaluation metric is imperfect, it is not a reason to completely dismiss all quantitative comparisons with previous work. The frames output by the network on, e.g. the moving digits datasets (Figure 4), looks ok and can definitely be compared with other papers. Yet, the authors chose not to, which is suspicious.— https://openreview.net/forum?id=HkxAAvcxx&noteId=SJE7-lkVx	3	3	0.41
R3	*Efficient Calculation of Polynomial Features on Sparse Matrices*	—Many more relevant papers should be cited from the recent literature.The experiment part is very weak. This paper claims that the time complexity of their algorithm is $O(d^k\ D^k)$, which is an improvement over standard method $O(d^k)$ by a factor d^k But in the experiments, when d=1, there is still a large gap (14s vs. 90s) between the proposed method and the standard one. The authors explain this as ”likely a language implementation”, which is not convincing. To fairly compare the two methods, of course you need to implement both in the same programming language and run experiments in the same environment. For higher degree feature expansion, there is no empirical experiments to show the advantage of the proposed method.— https://openreview.net/forum?id=S1j4RqYxg&noteId=B17Fn04Vg	4	3	0.27
R4	*Efficient Vector Representation for Documents through Corruption*	—While none of the pieces of this model are particularly novel, the result is an efficient learning algorithm for document representation with good empirical performance.Joint training of word and document embeddings is not a new idea, nor is the idea of enforcing the document to be represented by the sum of its word embeddings (see, e.g. see, e.g. ”The Sum of Its Parts”: Joint Learning of Word and Phrase Representations with Autoencoders’ by Lebret and Collobert). Furthermore, the corruption mechanism is nothing other than traditional dropout on the input layer. Coupled with the word2vec-style loss and training methods, this paper offers little on the novelty front.On the other hand, it is very efficient at generation time, requiring only an average of the word embeddings rather than a complicated inference step as in Doc2Vec. Moreover, by construction, the embedding captures salient global information about the document – it captures specifically that information that aids in local-context prediction. For such a simple model, the performance on sentiment analysis and document classification is quite encouraging.Overall, despite the lack of novelty, the simplicity, efficiency, and performance of this model make it worthy of wider readership and study, and I recommend acceptance.—https://openreview.net/forum?id=B1Igu2ogg&noteId=rJBM9YbVg	6	7	-1.04
R5	*R5 Towards a Neural Statistician*	—Hierarchical modeling is an important and high impact problem, and I think that it’s under-explored in the Deep Learning literature.Pros:-The few-shot learning results look good, but I’mm not an expert in this area.-The idea of using a ”double” variational bound in a hierarchical generative model is well presented and seems widely applicable. Questions:-When training the statistic network, are minibatches (i.e. subsets of the examples) used?-If not, does using minibatches actually give you an unbiased estimator of the full gradient (if you had used all examples)? For example, what if the statistic network wants to pull out if *any* example from the dataset has a certain feature and treat that as the characterization.This seems to fit the graphical model on the right side of figure 1. If your statistic network is trained on minibatches, it won’t be able to learn this characterization, because a given minibatch will be missing some of the examples from the dataset.Using minibatches (as opposed to using all examples in the dataset) to train the statistic network seems like it would limit the expressive power of the model— https://openreview.net/forum?id=HJDBUF5le&noteId=HyWm1orEx	6	8	-0.65
R6	*A recurrent neural network without chaos*	The authors of the paper set out to answer the question whether chaotic behaviour is a necessary ingredient for RNNs to perform well on some tasks.For that question’s sake,they propose an architecture which is designed to not have chaos. The subsequent experiments validate the claim that chaos is not necessary.This paper is refreshing. Instead of proposing another incremental improvement, the authors start out with a clear hypothesis and test it. This might set the base for future design principles of RNNs.The only downside is that the experiments are only conducted on tasks which are known to be not that demanding from a dynamical systems perspective; it would have been nice if the authors had traversed the set of data sets more to find data where chaos is actually necessary. https://openreview.net/forum?id=S1dIzvclg&noteId=H1LYxY84l	5	8	-1.01
R7	*Batch Policy Gradient Methods for Improving Neural Conversation Models*	The author propose to use a off-policy actor-critic algorithm in a batch-setting to improve chatbots.The approach is well motivated and the paper is well written, except for some intuitions for why the batch version outperforms the on-line version (see comments on ”clarification regarding batch vs. online setting”).The artificial experiments are instructive, and the real-world experiments were performed very thoroughly although the results show only modest improvement. https://openreview.net/forum?id=rJfMusFll&noteId=H1bSmrx4x	7	7	-1.77

Table 5: A qualitative study of the effect of sentiment in the overall recommendation score prediction. Prediction → is the overall recommendation score predicted by our system, Actual → is the recommendation score given by reviewers. **Senti_Act** are the output activations from the final layer of *MLP_Senti* which are augmented to the decision layer for final recommendation score prediction. The correspondence between the sentiment embedded within the review texts and Sentiment Activations are fairly visible in Figure 3. Kindly refer to Appendix A for polarity strengths in individual review sentences. The OpenReview links in the table above give the full review texts.

high recommendation score 8. It starts with introducing the authors work and listing the strengths and limitations of the work without much (and necessary) details. Our model hence predicts 5 as the recommendation score. Whereas R4 can be seen as the case of a usual well-written review, expressing the positive and negative aspects of the paper coherently. Our model predicts 6 for an actual recommendation score of 7. These validate the role of the reviewer's opinion and sentiment to predict the recommendation score, and our model is competent enough to take into account the overall polarity of the review-text to drive the prediction. Figure 4 presents the confusion matrix of our proposed model on ICLR 2017 test data for Task 2.

7 Conclusion

Here in this work, we show that the reviewer sentiment information embedded within peer review texts could be leveraged to predict the peer review outcomes. Our deep neural architecture makes use of three information channels: the paper full-text, corresponding peer review texts and the sentiment within the reviews to address the complex task of decision making in peer review. With further exploration, we aim to mould the ongoing research to an efficient AI-enabled system that would assist the journal editors or conference chairs in making informed decisions. However, considering the sensitivity of the topic, we would like to further dive deep into exploring the subtle nuances that leads into the grading of peer review aspects. We found that review reliability prediction should prelude these tasks since not all reviews are of equal quality or are significant to the final decision making. We aim to include review reliability prediction in the pipeline of our future work. However, we are in consensus that scholarly language processing is not straightforward. We need stronger, pervasive models to capture the high-level interplay of the paper and peer reviews to decide the fate of a manuscript. We intend to work upon those and also explore more sophisticated techniques for sentiment polarity encoding.

Acknowledgements

The first author, Tirthankar Ghosal, acknowledges Visvesvaraya PhD Scheme for Electronics and IT, an initiative of Ministry of Electronics and Information Technology (MeitY), Government of India for fellowship support. The third author, Asif Ekbal, acknowledges Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia). We also thank Elsevier Center of Excellence for Natural Language Processing, Indian Institute of Technology Patna for adequate infrastructural support to carry out this research. Finally, we appreciate the anonymous reviewers for their critical evaluation of our work and suggestions to carry forward from here.

References

Lutz Bornmann and Hans-Dieter Daniel. 2010. Reliability of reviewers' ratings when using public peer review: a case study. *Learned Publishing*, 23(2):124–131.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174.

Laurent Charlin and Richard Zemel. 2013. The toronto paper matching system: an automated paper-reviewer assignment system.

Tirthankar Ghosal, Ravi Sonam, Sriparna Saha, Asif Ekbal, and Pushpak Bhattacharyya. 2018a. Investigating domain features for scope detection and classification of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2018b. Investigating impact features in editorial pre-screening of research papers. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*, pages 333–334.

Clayton J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1647–1661.

John Langford and Mark Guzdial. 2015. The arbitrariness of reviews, and advice for school administrators. *Commun. ACM*, 58(4):12–13.

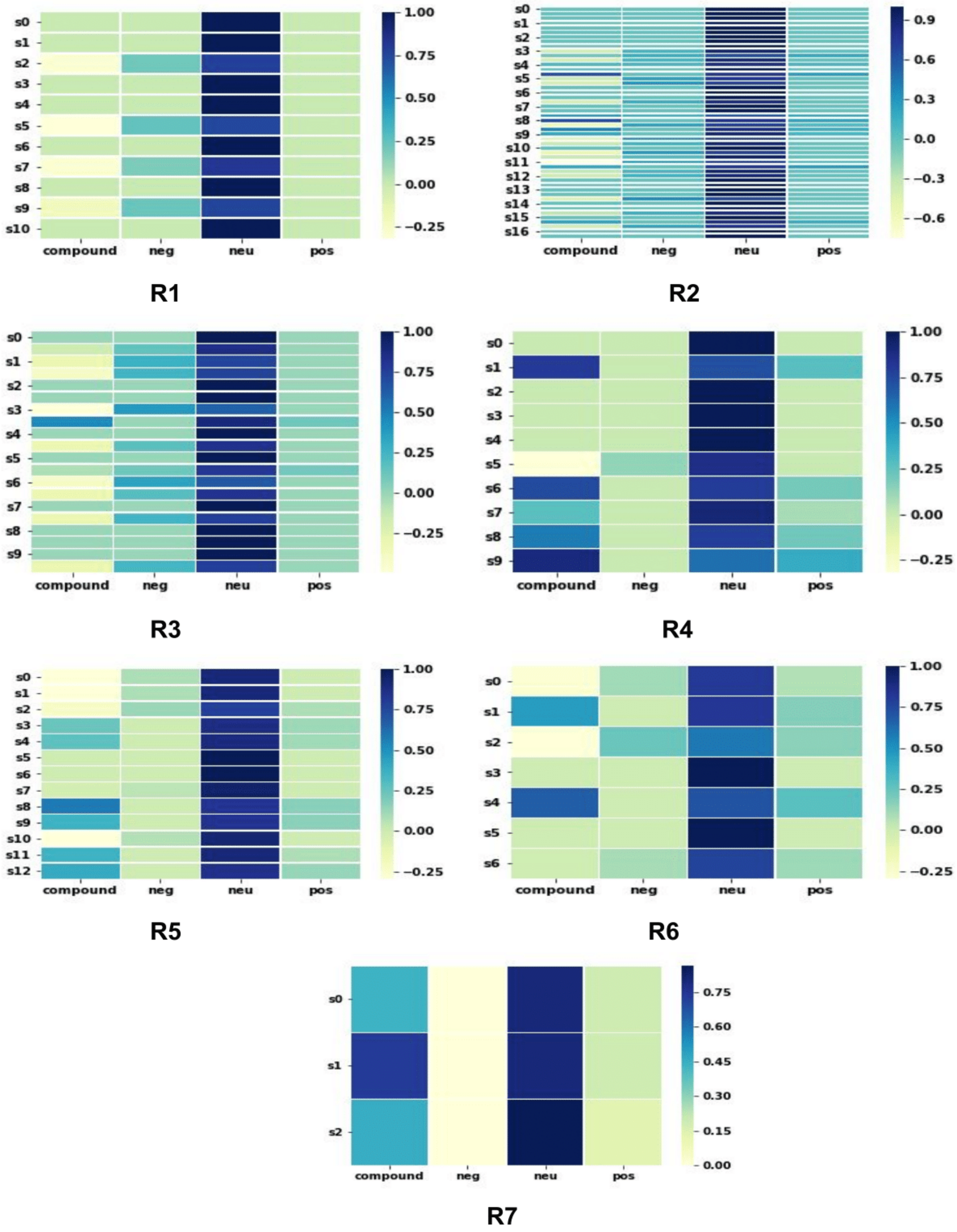
Maciej J Mrowinski, Piotr Fronczak, Agata Fronczak, Marcel Ausloos, and Olgica Nedic. 2017. Artificial intelligence in peer review: How can evolutionary computation support journal editors? *PloS one*, 12(9):e0184711.

Simon Price and Peter A. Flach. 2017. Computational support for academic peer review: a perspective from artificial intelligence. *Commun. ACM*, 60(3):70–79.

Richard Smith. 2006. Peer review: a flawed process at the heart of science and journals. *Journal of the royal society of medicine*, 99(4):178–182.

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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 175–184.

A Heatmaps Depicting Sentiment Polarity in Review Texts



R1

R2

R3

R4

R5

R6

R7

Figure 5: Heatmaps of the sentence-wise VADER sentiment polarity of reviews considered in Table 4. Reviews generally reflect the polarity of the reviewer towards the respective work. *s0...sn* → are the sentences in the peer review texts.