Empirical Analysis of Model Structures and Hyper-parameters

We experimented with alternative classification heads and hyperparameter values of ASSORT$_S$ and measured their accuracy. We used the same training data for these experiments, consisting of 2,424 answer posts and 14,165 sentences.

I. Classification head

As Reviewer A suggested, we experimented with different classification heads on top of ASSORT$_S$'s sentence representation, including random forests, decision trees, linear regression, Ada boost, logistic regression, and Naive Bayes Classifier (NBC). Table I shows the model accuracy after replacing the original feedforward neural network (FNN) with each alternative classification head. Column F1 shows the F1 score, and Column Δ shows the model accuracy difference in comparison to the original design (i.e., FNN).

As shown in Table I, using FNN achieved significantly better accuracy, 0.71 in the F1 score. This indicates that FNN has better learnability than other models. In future work, we plan to experiment with more advanced neural network architectures such as LSTM.

TABLE I: Model Accuracy with Different Classification Heads

	F1	**Δ**
Feedforward NN	**0.71**	**-**
Random forest	0.56	-0.15
Decision tree	0.54	-0.17
Linear regression	0.65	-0.06
Logistic regression	0.63	-0.08
Ada boost	0.59	-0.12
Naive Bayes Classifier	0.62	-0.09

II. Number of hidden layers

We experimented with different numbers of hidden layers in the FNN classification head of ASSORT$_S$. Table II shows the result. It turns out that a "deeper" neural network with more hidden layers does not necessarily increase the accuracy of ASSORT$_S$. For the simplicity of the model structure and to avoid increasing the risk of overfitting, we chose to include only one hidden layer in the classification head. Furthermore, adding one layer on top of a deep pre-trained language model is a common practice in NLP. For example, in the original BERT paper [1], the authors only used one hidden layer in their classification head. Similarly, the authors in the BERTSum paper [2] also used one hidden layer.

TABLE II: Model Accuracy with Different Numbers of Hidden Layers in the FNN

Number of hidden layers	**Precision**	**Recall**	**F1**
1	**0.73**	**0.69**	**0.71**
2	0.69	0.71	0.70
3	0.73	0.71	0.72
4	0.73	0.70	0.71

III. Learning rate

Another hyper-parameter we have experimented with is the learning rate. Figure 1 shows the model accuracy change over training epochs when using four different learning rates. On the one hand, when setting a learning rate bigger than the current design (1e-5), the model accuracy fluctuates more during training. On the other hand, when selecting a learning rate that is too low, the training process takes more epochs and more computational power and time to reach the optimal accuracy.

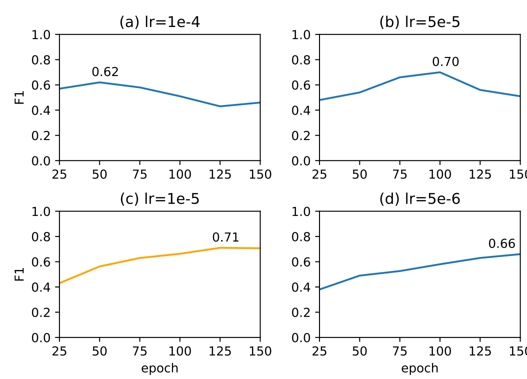

Fig. 1: Model accuracy with different learning rates (lr).

References

[1] J. Devlin, M.-W. Chang *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345*, 2019.