Improving Transformers' capabilities in Commonsense Reasoning

Yu Zhou, Yunqiu Han, Hanyu Zhou, Gloria (Yulun) Wu
University of California, Los Angeles

1 Introduction

Masked language models, such as BERT (Devlin et al., 2019) and its further variations (He et al., 2021b; Liu et al., 2019; Lewis et al., 2020), are models pretrained by masking some tokens in a sentence and trying to predict the masked tokens. They provide solid foundations for finetuning on common natural language processing (NLP) tasks.

Endowing computers with human-like commonsense knowledge has remained a challenge of NLP for decades (Sap et al., 2020). Com2Sense was proposed in 2021 as a common sense reasoning benchmark with novel pairwise accuracy metrics. It consisted of natural language sentence pairs labeled True/False, categorized into three dimensions (Singh et al., 2021). Initial experiments revealed that the models did not get high pairwise accuracy on what seemed for granted in human's language understanding, like discerning causal and comparison relationships.

In this project, we studied the impact of different finetuning techniques on the performance of Com2Sense prediction using masked language models, as listed in the following sections.

2 Methods

2.1 Pretrained Language Models

We tested on a variety of pretrained language models, including two variants of BERT models, i.e., $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{base}}$, $\text{DeBERTa}_{\text{base}}$, and two variants of DeBERTaV3, i.e., $\text{DeBERTaV3}_{\text{base}}$, and $\text{DeBERTaV3}_{\text{large}}$.

2.2 Hyperparameter Tuning

We used different learning rates, batch sizes, weight decay, adam epsilon, warmup steps to finetune on Com2Sense and compare their results. The hyperparameters we used and their corresponding results are listed in Table 1.

2.3 Knowledge Transfer

2.3.1 Pretraining on SemEval Dataset

Similar to Com2Sense, Wang et al. (2020) provided a commonsense-related dataset called SemEval. Each instance included a pair of sentences, one of which makes sense while the other does not.

We hypothesized that using a language model pretrained on the SemEval dataset, we are able to achieve better performance than finetuning that model directly on Com2Sense. First, we implemented data loading, data preprocessing and training scripts for the SemEval dataset. Then, we trained the $\text{DeBERTaV3}_{\text{large}}$ model on SemEval dataset to get a checkpoint model. The parameter used for pretraining are: batch size = 48, lr = 4e-5, weight decay = 0.01, adam eps = 1e-6, and trained for 100 steps. Finally, we finetuned the obtained model on the Com2Sense dataset with the same parameters as the model on Line 10 of Table 1 and compare the two models.

2.3.2 Pretraining on QA Dataset

We hypothesized that a language model will achieve higher performance after pretraining on a question-answer dataset. We compared the results from $\text{RoBERTa}_{\text{base}}$ and $\text{DeBERTaV3}_{\text{large}}$ with those from $\text{RoBERTa}_{\text{base}}$-SQuAD2 and $\text{DeBERTaV3}_{\text{large}}$-SQuAD2, respectively, by finetuning them on Com2Sense with the same parameters. We used the parameters in Table 1 for $\text{RoBERTa}_{\text{base}}$ and the best model in Table 1 for $\text{DeBERTaV3}_{\text{large}}$.

2.4 Cross Validation

The training dataset contains 797 pairs of examples and the development set has 398 pairs. We hypothesized that leveraging both datasets for training would yield a more generalized model with a higher level of reliability. We thus employed k-fold cross validation on our best $\text{DeBERTaV3}_{\text{large}}$ model and tested for $k = 2$ and $k = 5$.

Line No.	Model	best step	batch size	lr	weight decay	adam ϵ	warmup step	Pairwise Acc %	F1 score
1	$\text{BERT}_{\text{base}}$	1020	32	1e-5	0	1e-8	0	3.01	0.4073
2	$\text{BERT}_{\text{large}}$	60	64	5e-5	0	1e-8	0	2.51	0.3720
3	$\text{RoBERTa}_{\text{base}}$	1040	64	1e-5	0.01	1e-8	0	18.84	0.5463
4	$\text{DeBERTa}_{\text{base}}$	6000	32	1e-5	0	1e-8	500	17.84	0.5302
5	$\text{DeBERTaV3}_{\text{base}}$	4500	48	1e-5	0.01	1e-6	500	48.74	0.7145
6	$\text{DeBERTaV3}_{\text{base}}$	1500	48	3e-5	0.01	1e-6	100	52.76	0.7219
7	$\text{DeBERTaV3}_{\text{base}}$	2500	48	3e-5	0.01	1e-6	500	49.00	0.7057
8	$\text{DeBERTaV3}_{\text{base}}$	1000	48	9e-6	0.01	1e-6	500	45.48	0.6767
9	$\text{DeBERTaV3}_{\text{large}}$	750	64	9e-6	0.01	1e-6	500	67.84	0.8090
10	**$\text{DeBERTaV3}_{\text{large}}$**	**1900**	**48**	**9e-6**	**0.01**	**1e-6**	**500**	**68.34**	**0.8103**
11	$\text{DeBERTaV3}_{\text{large}}$	1000	48	8.5e-6	0.01	1e-6	500	67.34	0.8111
12	$\text{DeBERTaV3}_{\text{large}}$	450	48	9.5e-6	0.01	1e-6	500	66.33	0.7990
13	$\text{DeBERTaV3}_{\text{large}}$	1000	48	9e-6	0.01	1e-6	300	67.59	0.8059
14	$\text{DeBERTaV3}_{\text{large}}$	1400	48	9e-6	0.01	1e-6	750	66.58	0.8029

Table 1: Summary of hyper-parameter tuning with results calculated on the dev dataset, the experiments are focused on finding the best model backbone, model size and ideal values for hyper-parameters. The best performing model and ideal hyper-parameter group is highlighted in bold.

2.5 Contrastive Learning

The Come2Sense dataset is complementary in nature. That is, for each statement, there is a complementary statement that is constructed with small perturbation on certain words making it concerning similar common sense concepts but with different (opposite) labels. This unique setting makes Com2Sense an ideal case for the use of contrastive learning.

We hypothesized that for the model to capture the semantic difference between commonsensical inputs vs their syntactically similar counterparts, it would be beneficial if we can push apart the hidden representation of each complementary input pair in the embedding space.

In practice, inspired by the InfoNCE Contrastive Loss by van den Oord et al. (2018), we propose a Pairwise Contrastive Loss (PCL) function:

$$\mathscr{P}^{\mathrm{x_i},\mathrm{x_j}}(\mathrm{W}) = \frac{\mathrm{e}^{\mathrm{sim}\left(\mathrm{g}(\mathrm{x_i}),\mathrm{g}\left(\mathrm{x_j}\right)\right)/\tau}}{\sum_{\mathrm{k}=1,\mathrm{k}\neq\mathrm{i}}^{2\ \mathrm{N}} \mathrm{e}^{\mathrm{sim}(\mathrm{g}(\mathrm{x_i}),\mathrm{g}(\mathrm{x_k}))/\tau}} \quad (1)$$

Here for each complementary input sample pair $(\mathrm{x_i}, \mathrm{x_j})$ with embedding vectors $\mathrm{g}(\mathrm{x_i}), \mathrm{g}(\mathrm{x_j})$, where $\mathrm{sim}\left(\mathrm{g}\left(\mathrm{x_i}\right), \mathrm{g}\left(\mathrm{x_j}\right)\right)$ is the dot product of the L2 normalised inputs and τ is the constant temperature parameter which we set to 0.5.

The total contrastive loss, $\mathcal{L}$, is defined as the arithmetic mean over all pairs in the batch of the cross entropy of their normalised similarities, i.e.

$$\mathcal{L}_{\text{total}} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{j}=1}^{\mathrm{N}} \log \mathscr{P}^{\mathrm{x_j},\mathrm{x_j}}(\mathrm{W}) \quad (2)$$

2.6 Model Ensemble and Random Perturbation

Due to the complementary nature of the Com2Sense dataset, each input data pair should have one positive sample and one negative sample. With this fact in mind, we propose a posterior model ensemble pipeline that aims to reduce the number of Same-Output Pairs where two prediction labels are the same (either both positive or both negative). This method further helps our ensembled model distinguish between syntactically similar sentence pairs that represent different ideas.

In practice, we take n finetuned models and rank them by their pairwise accuracy score on the dev set to represent our confidence in the model. Then we use the highest performing model as a base predictor to generate predictions on the test set, which would contain a number of Same-Output Pairs. For each Same-Output Pair, we move down the list of ranked models and by confidence and generate their predictions. If the new model can differentiate

Line No.	Model and Method	Trial No.	Pairwise Acc %	Standard Acc %
1	UnifiedQA-3B (Previous SOTA)	N/A	51.26	71.31
2	DeBERTaV3$_{\text{large}}$ (our baseline with best hyperparameters)	02	63.40	77.87
3	DeBERTaV3$_{\text{large}}$ + KT	04	64.19	78.10
4	DeBERTaV3$_{\text{large}}$ + KT + CV	09	66.74	79.39
5	DeBERTaV3$_{\text{large}}$ + KT + CV + Contrastive + RP	10	82.07	82.07
6	DeBERTaV3$_{\text{large}}$ + KT + CV + Contrastive + Ensemble(5) + RP	11	82.62	82.62
7	**DeBERTaV3$_{\text{large}}$ + KT + CV + Contrastive + Ensemble(8) + RP**	**12**	**83.69**	**83.69**
8	Human	N/A	95.00	96.50

Table 2: Summary of our results on the Com2Sense test set: KT stands for Knowledge Transfer, CV stands for Cross Validation, RP stands for Random Perturbation, Ensemble(5) stands for a 5-model ensemble between DeBERTaV3$_{\text{large}}$ and DeBERTaV3$_{\text{base}}$, Ensemble(8) stands for an 8-model ensemble between DeBERTaV3$_{\text{large}}$, DeBERTaV3$_{\text{base}}$, and RoBERTa$_{\text{base}}$. The best model and method is highlighted with bold texts.

the two samples (generating one positive and one negative), then we adopt the new model's prediction. In the end, we have an (ideally very small) number of test pairs where all models are unable to differentiate between, in which case we randomly assign different prediction values to the pair.

3 Results

3.1 Pretrained Language Models

To find the best model backbone architecture, we compare the results of BERT$_{\text{base}}$, RoBERTa$_{\text{base}}$, DeBERTa$_{\text{base}}$, and DeBERTaV3$_{\text{base}}$ with the best finetuning parameters used by their respective authors. Our results show DeBERTaV3$_{\text{base}}$ to be the best structure with 48.74% pairwise acc, while DeBERTa$_{\text{base}}$ and RoBERTa$_{\text{base}}$ share similar performance at ∼18%. BERT base is the lowest performing model at ∼3%

To find the best model size, we conduct multiple experiments with DeBERTaV3$_{\text{base}}$ and DeBERTaV3$_{\text{large}}$ under best finetuning parameters used by He et al. (2021a). The results show that DeBERTaV3$_{\text{large}}$ reaches 68.34% pairwise acc while DeBERTaV3$_{\text{base}}$ reaches 52.76%. This supports the hypothesis that larger models have stronger common sense reasoning ability.

3.2 Hyperparameter Tuning

After choosing the best performing model DeBERTaV3$_{\text{large}}$ as our base model, we perform hyperparameter tuning on model parameters including: batch size (equivalent batch size after gradient accu- mulation), learning rate, and warmup steps. In each case, we fix all other parameters and test the effect of different values for the parameter under investigation. The testing results are documented in Table 1 lines 9-14, and the best set of parameters are highlighted in line 10.

3.3 Knowledge Transfer

3.3.1 Pretraining on SemEval Dataset

Model	Pairwise %	F1 Score
best DeBERTaV3$_{\text{large}}$	68.34	0.8103
SemEval-pretrained	68.84	0.8139

As shown in the table, the transferred model performs better than the best model directly applied to Com2Sense, with a 0.364% improvement on pairwise accuracy.

3.3.2 Pretraining on QA Dataset

Model	Pairwise %	F1 Score
RoBERTa$_{\text{base}}$	18.84	0.5463
RoBERTa$_{\text{base}}$-SQuAD2	13.56	0.5542
DeBERTaV3$_{\text{large}}$	68.34	0.8103
DeBERTaV3$_{\text{large}}$-SQuAD2	65.83	0.7893

It can be seen from the results that models pretrained on question answering data did not perform as well as those that have not. This can be because question answering is a vastly different task than binary classification.

3.4 Cross Validation

As shown in Table 2 lines 3-5, cross validation helped to improve test performance by 2.5% empirically by incorporating the dev set for finetuning, which added around 50% more training data, resulting in improved performance. Consequently, the

training time escalates with the number of folds: 2-fold cross validation took around 20 hours to train and 5-fold took more than 50 hours.

3.5 Contrastive Learning and Random Perturbation

From Table 2 lines 5-8, we observe that in practice contrastive learning together with the Random Perturbation helped to improve test performance by 16%. In this case, we count that random perturbation changed a total of 371 pairs in the test set (total 2790 pairs). While this can have a maximum influence of 13.29% if all changed pairs turn out to be correct, since it is purely random perturbation, on average it should have improved pairwise accuracy by 6.65%.

After removing the benefits of Random Perturbation, we conclude that Contrastive Learning on average yields an improvement of 8.77%, while in the worst case it yields an improvement of 2.04%. The improvement can be attributed to the fact that Com2Sense dataset comes in a natural contrastive fashion, with similar true/false pairs that need to be differentiated from each other.

3.6 Model Ensemble

From Table 2 lines 6-8, we observe that in practice model ensemble as a post-processing technique helps the model perform better compared to straight-through Random Perturbation, likely because Random Perturbation only has a 50% chance of correctly predicting a pair while models used in the ensemble have a much higher accuracy.

In addition, the ensemble between DeBERTaV3$_{\text{large}}$, DeBERTaV3$_{\text{base}}$, and RoBERTa$_{\text{base}}$ models outperforms the ensemble between DeBERTaV3$_{\text{large}}$ and DeBERTaV3$_{\text{base}}$ models by 1.07% pairwise acc. This results supports the common understanding that diversity in model structure is beneficial for the ensemble.

4 Discussion

4.1 Analysis

We make use of the domain, scenario, and numeracy dimensions of common2sense, taking the best performing model of BERT$_{\text{base}}$, DeBERTaV3$_{\text{base}}$, DeBERTaV3$_{\text{large}}$, and DeBERTV3$_{\text{large}}$ pretrained on SemEval dataset, then calculate the pairwise accuracy on common2sense dev dataset in every possible combination of the three dimensions.

In general BERT$_{\text{base}}$ gives very low pairwise accuracy, less than $\frac{1}{10}$ of its F1 score as shown in Table 1, which indicates its poor performance on identifying the sentence pairs and thus the common sense logic underlying. Figure 1 top graph gives further information, revealing that the model correctly predicts none of the data with numeracy. Comparatively it gives better predictions on sentences with comparison than with causal relationship; get higher pairwise accuracies on temporal sentences than physical and lastly social.

DeBERTaV3$_{\text{base}}$ gets boosting pairwise accuracy. From Figure 1 bottom graph, it performs better on comparative data than causal for all domains. On the other hand, we get slightly better results with numeracy than without numeracy for the physical domain, but in reverse in forthe social domain. The pattern of the temporal domain is more mixed: data with numeracy information has higher pairwise accuracy for comparisons, but decreases for causal scenario.

DeBERTaV3$_{\text{large}}$ improves the pairwise accuracy for all categories, but in particular more salient for social domain and numeracy data, as shown in Figure 2 top graph. The patterns of DeBERTaV3$_{\text{base}}$ are mostly preserved, with the only exception that the model performs better on data with numeracy information for temporal, causal sentences.

The result of our attempt on transfer learning displayed as Figure 2 bottom graph, DeBERTV3$_{\text{large}}$ pretrained on SemEval dataset, in general performs better on social domain than physical, lastly temporal; also better on data without numeracy. We do observe though, an exception of high performance on temporal, comparative, and numeric sentences, probably indicating SemEval's effect. We also noticed that the pretraning might improves DeBERTV3$_{\text{large}}$'s ability to learn causal reasoning as the pairwise accuracy of pretrained model increases for causal scenario.

4.2 Areas of Improvement

There were some areas that we could have done better during the training and finetuning stages of our project. Firstly, due to different hardware limitations on each of our VMs, we were not able to keep the per-GPU batch size the same throughout the project. While some of the trials used 6-instance batches over 8 accumulation steps, others were only able to use 4-instance batches over 12 accumulation steps. This might have slight impacts on our final

result. Secondly, due to time constraints, we only tested a limited range of hyperparameters which was not guaranteed to be the global optimum.

5 Conclusion

In this project, we trained large NLP models on the downstream Com2Sense text classification task. We explored a variety of finetuning techniques such as hyperparameter tuning, knowledge transfer on SemEval and SQuAD v2.0, cross validation, contrastive learning, and model ensemble. We first found the best-performing hyperparameters, and by using them as the base model, we found that all the techniques except pretraining with question-answering improved the performance over the base model. In particular, we were able to achieve a 83.69% pairwise accuracy (a 63% increase from previous SOTA) by applying a combination of techniques including knowledge transfer from SemEval, cross validation, contrastive learning, random perturbation, and model ensembling.

References

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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Shikhar Singh, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. COM2SENSE: A commonsense reasoning benchmark with complementary sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 883–898, Online. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

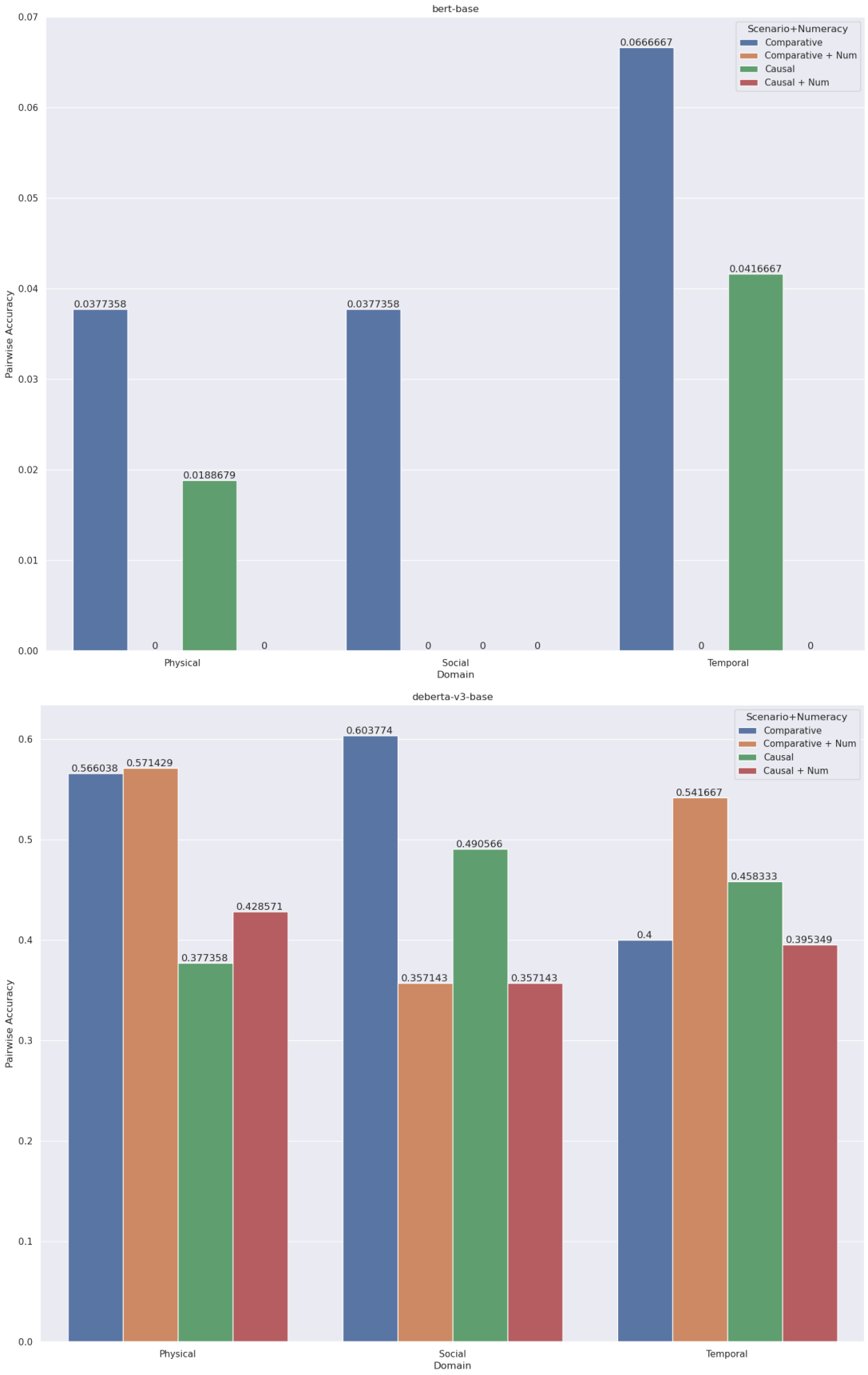

Figure 1: The top graph shows the pairwise accuracy of different dimension combinations of $\text{BERT}_{\text{base}}$, and the graph below shows the pairwise accuracy of different dimension combinations of $\text{DeBERTaV3}_{\text{base}}$.

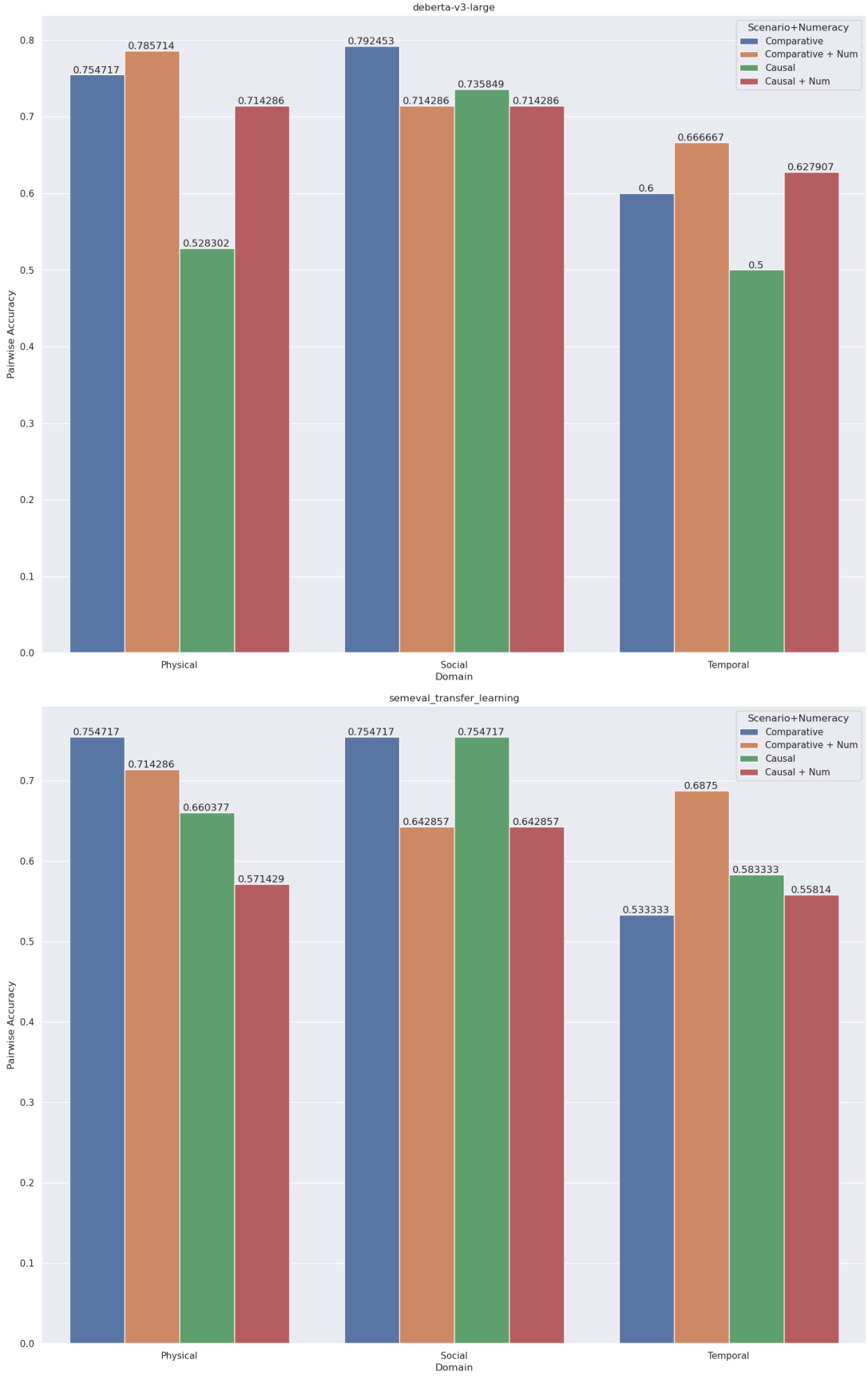

Figure 2: The top graph shows the pairwise accuracy of different dimension combinations of $\text{BERT}_{\text{large}}$, and the graph below shows the pairwise accuracy of different dimension combinations of $\text{DeBERTaV3}_{\text{large}}$ pretrained on SemEval dataset.