Enhancing Multi-Task Text Classification with Contrastive Learning and Dataset Augmentation in BERT-like Models

Stanford CS224N Default Project

Phillip Yao-Lakaschus
Department of Computer Science
Stanford University
yaol@stanford.edu

Abstract

Contrastive learning objectives have shown great promise in achieving state-of-the-art performance metrics in various natural language processing (NLP) tasks. Contrastive learning provides two main benefits: first, it effectively augments the available data, which is often sparse, and second, it adds an additional learning objective that can aid in learning more robust representations. In this report, an unsupervised Ansatz for contrastive learning is employed by applying two different dropout masks on the same example, creating positive example pairs [1]. Additionally, a novel supervised contrastive learning approach is explored by defining examples from the SST-5 dataset with the same sentiment labels as positive pairs. While initially thought to only benefit sentiment classification tasks, it is observed that this approach is also beneficial for other tasks. However, when fine-tuning on transfer tasks, most of the benefits from contrastive learning are lost. Furthermore, a novel technique for augmenting available datasets with ChatGPT is investigated, which entails certain risks of overfitting and elimination of inherent biases in the dataset. The proposed extensions on top of the uncased BERT base model yields results comparable to those of state-of-the-art methods for the SST-5, STS, and QQP datasets individually, with scores of 0.526, 0.869, and 0.873, respectively. Furthermore, the multi-task model, capable of performing all of these tasks concurrently, attained scores of 0.520, 0.868, and 0.858 for the SST-5, STS, and QQP tasks.

1 Key Information to include

- Mentor: Drew Kaul
- External Collaborators (if you have any): None
- Sharing project: None

2 Introduction

It has been six years since the development of the first transformer model [2] and five years since the infamous BERT paper [3] that demonstrated the potential of transformers. Although the transformer-encoder model BERT has been greatly outperformed by newer and larger models [4, 5, 6], it is nevertheless of great importance for academia because it is conceptually simple and serves as playing ground for many explorative studies. It is also of great pedagogical value as it can be trained on affordable GPUs making it accessible for students and even interested laymen.

Given the aforementioned benefits of BERT, such as conceptual simplicity and computational feasibility, this model was chosen as the foundation for the CS224N default project. The goal of

the default project is to build a multitask classifier on top of the uncased BERT base model that is able to perform well on three tasks simultaneously, a sentiment classification task using the Stanford Sentiment Treebank (SST-5) dataset [7], a paraphrase detection task based on the Quora question pairs dataset (QQP) [8], and a regression task based on the SemEval STS dataset (STS) [9]. The difficulty here comes from the fact, that all of these tasks differ in terms of task type (classification vs. regression), number of classes, domains and dataset size. The state-of-the-art results for the given downstream tasks are as follows: 59.8% accuracy for SST-5-5[1], 92.4% accuracy for the quora paraphrase dataset[2], and 92.9% Pearson correlation for STS[3]. Note however, that these results were obtained using significantly larger models and therefore a comparison to the results presented here are not fair. Also, it is generally harder to train a multi-task model that performs well on many tasks instead of fine-tuning one single tasks specifically.

In this project report various improvements will be attempted to score well on all tasks simultaneously but one special focus will be set on contrastive sentence embeddings (CSE). Concretely, the contributions in this report are summarized as follows:

- Implementation of an unsupervised contrastive learning method following Ref. [1].
- Implementation of a supervised contrastive learning objective using sentiment class labels.
- Augmenting the SST-5 and STS datasets using ChatGPT3.5.
- Comparing different multi-task learning strategies, in particular, sequential learning and joint task learning
- Exploring various hyperparameter optimizations

3 Related Work

Contrastive learning has gained significant attention in recent years due to its success in various domains, particularly in computer vision [10, 11]. It has been demonstrated that contrastive learning can lead to robust representations by maximizing the similarity between semantically similar data points while minimizing the similarity between dissimilar data points. Recently, the field of NLP has followed suit and started to apply contrastive learning to natural language processing tasks, with promising results, see also the excellent primer by Rethmeier and Augenstein [12].

The foundation of this project is predominantly influenced by Ref. [1]. In that paper, the authors introduced an innovative and remarkably simple unsupervised contrastive learning objective by applying two distinct dropout masks to the same input sentence and defining the resulting two embeddings as positive pairs. This learning objective directly benefits text similarity tasks, such as those based on the STS datasets, without the need for explicit fine-tuning. They call their approach "Simple Contrastive Learning of Sentence Embeddings" (SimCSE), which in this report will also be referred to as SimCSE.

4 Approach

As outlined in the introductory section of this report, this variant of the default project aims to leverage contrastive methods to enhance the performance of the baseline BERT model for the tasks specified in Table 1. To this end, the uncased BERT base model[4] was provided, which has 12 layers with 12 attention heads each and a hidden size of 768, resulting in 110M parameters. The context length is 512 tokens. Increasing the deepness and hidden size of the model is usually considered the most promising approach for improving the performance on downstream tasks. In this report however, it was attempted to squeeze out as much as possible from the base model without changing its fundamental architecture.

In the following, the several steps on developing the multitask-classifier model are elaborated.

[1] `https://paperswithcode.com/sota/sentiment-analysis-on-sst-5-fine-grained`

[2] `https://paperswithcode.com/sota/paraphrase-identification-on-quora-question`

[3] `https://paperswithcode.com/sota/semantic-textual-similarity-on-sts-benchmark`

[4] https://huggingface.co/bert-base-uncased

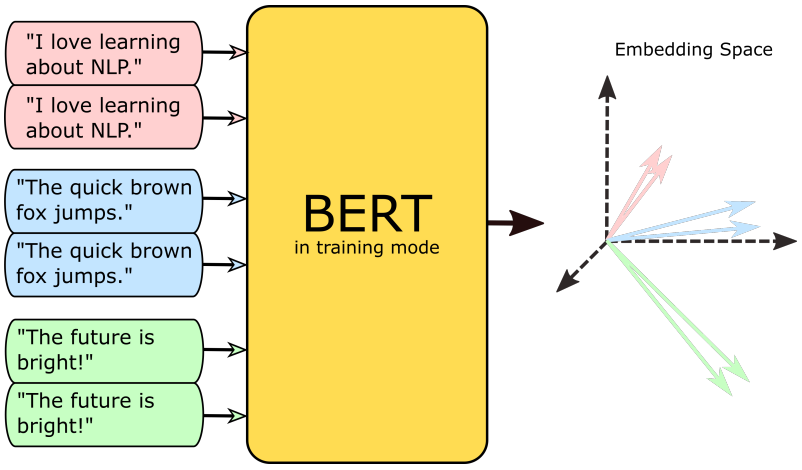

Figure 2: During training the dropout masks are randomized, i.e., two same input sentences will yield slightly two different embeddings in the forward passes. These embedding pairs represent the positive pairs in the unsupervised SimCSE learning objective.

Implementation of transfer tasks. As shown in Table 1 the QQP, SST-5 tasks are both classification tasks with 2 and 5 classes, respectively. However, both tasks differ with respect to the input.

While SST-5 consists of single sentences that can be directly classified, the QQP contains of two questions per example and thus need to be concatenated first before being able to be classified. The concatenation strategy $(u, v, |u - v|)$ was chosen, where u and v are the encodings for the two questions. The reason for this concatenation is simply that best results were reported in Ref. [13] for this concatenation.

The STS task also consists of two input sentences per example and the goal is to determine the similarity between them. For this task, concatenation is not needed as the similarity of the input sentences can be directly computed using the cosine-similarity metric. We also multiply the output of the cosine-similarity by five in order to match the regression scale of the origin STS dataset.

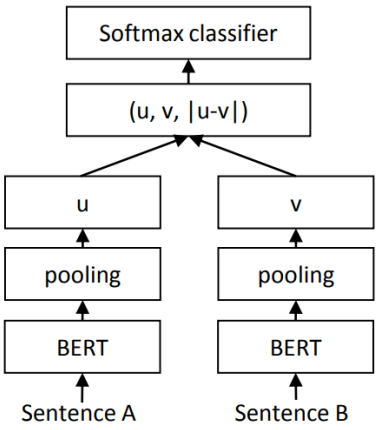

Figure 1: Visualization of the classification strategy for two input sentences. Figure taken from Ref. [13]

Unsupervised contrastive learning. Using the unsupervised SimCSE method of Ref. [1] the goal is to improve the base model such that it leads to better sentence embeddings by aligning them and making them more uniform. The method comprises of generating positive sentence pairs by applying dropout twice on the same input sentence leading to slightly different embeddings. The objective is then simply minimizing

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'})/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'})/\tau}}, \tag{1}$$

where i corresponds to the i-th batch, N is the batch size, $(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'})$ are two different sentence embeddings obtained by applying two different dropout masks, z_i and z_i', using the same input

sentence. sim is the cosine similarity, and τ a temperature hyperparamter. Note that in practice these two different dropout masks are obtained by simply doing two forward passes on the BERT base model during training. See Fig. 2 for an illustration of this method.

A reason why unsupervised SimCSE works is that it makes the representations more uniform in embedding space while not degrading with respect to alignment. It is known that pre-trained word embeddings tend to suffer from anisotropy [14, 15], which is somewhat ameliorated using unsupervised SimCSE. Alignment here is defined as follows:

$$\ell_{\text{align}} \triangleq \underset{(x,x^+)\sim p_{\text{pos}}}{\mathbb{E}} ||f(x) - f(x^+)||^2 \,, \tag{2}$$

where $f(x)$ is the embedding obtained by a forward pass through the encoder and (x, x^+) are positive example pairs sampled from a distribution of positive pairs p_{pos}. On the other hand, uniformity is defined as

$$\ell_{\text{uniform}} \triangleq \log \underset{(x,y)\ i.i.d.}{\mathbb{E}} e^{-2||f(x)-f(y)||^2} \,. \tag{3}$$

Both of these objectives correlate with the objective of contrastive learning, namely that positive instances should be close while random embeddings should be uniformly distributed in order to preserve maximal information [16].

In contrast to Ref. [1] where $1e6$ sentences from Wikipedia were used to fine-tune the model, here $1e6$ examples from the Openwebtext dataset [17] were taken. The rationale behind this is that sentences from Wikipedia are associated with a specific, more formal writing style, while the Openwebtext dataset is more broad in terms of topics and writing style. In addition to that, a masked language modeling (MLM) objective is supplemented during the fine-tuning of SimCSE embeddings in order to prevent catastrophic forgetting of token-level knowledge which might hurt performance on transfer tasks.

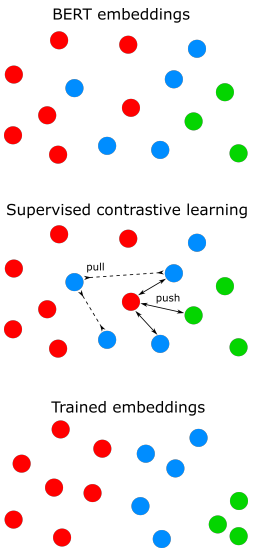

Figure 3: Visualization of the supervised contrastive learning process. The different colors correspond to different sentiment class labels.

Sentiment-based Supervised contrastive learning. Initially, the intention was to implement the supervised SimCSE objective of [1], which consists of leveraging natural language inference datasets and defines positive examples as those that both are classified as "entailment" and negative pairs as those with the "contradiction" label. However, here I wanted to primarily improve the SST-5 task score because I felt there is more potential as the difference of my baseline (0.49) to the SOTA results (0.60) is quite large (0.11). The idea thus was to use the training datasets for the SST-5 task and define examples with the same class label as positive and examples with different class labels as negatives. This idea was also inspired by Ref. [18]. The effect on the embeddings during training is illustrated in Fig. 3.

Multi-task learning. As a first Ansatz, a sequential multi-task model was employed. This model simply learns one task after another. This is not the most sophisticated approach, however, for the purpose here it serves as a baseline. Also, this approach better reveals in what way different tasks lead to positive or negative transfer. In this project it has been observed, that the STS and QPP tasks do not affect each other by much, but that these tasks interfere destructively with the SST-5 task. That means training SST-5 leads to negative transfer and forgetting with the STS and QPP tasks and vice versa.

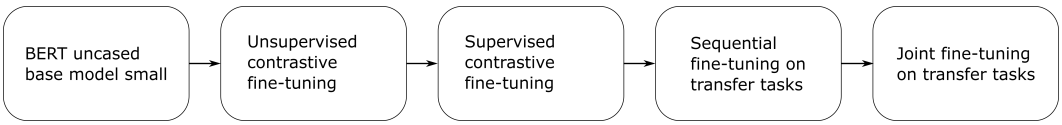

Figure 4: The full pipeline

Subsequently, we also develop a joint learning strategy, by which all tasks are trained simultaneously. This is done by batch-wise backpropagating on the sum of the losses for the three transfer tasks like so:

$$\mathcal{L}_{\text{total}} = w_{\text{SST-5}}\,\mathcal{L}_{\text{SST-5}} + w_{\text{STS}}\,\mathcal{L}_{\text{STS}} + w_{\text{QQP}}\,\mathcal{L}_{\text{QQP}}\,, \tag{4}$$

where the coefficients of the losses are weights that enables the priorization of the tasks. These are subject to hyperparameter tuning.

Training pipeline. This project involves multiple components, leading to multiple combinations that can be studied. To obtain the final result submitted to the test and dev leaderboards, the following steps were employed: First, the small uncased BERT base model was finetuned based on the unsupervised contrastive learning objective, using $1e6$ randomly sampled examples from the openwebtext dataset. Subsequently, the finetuned model checkpoint was used to train using the supervised contrastive learning objective. As direct joint multitask learning encountered difficulties in finding a minimum where all tasks had comparatively good scores, sequential fine-tuning was used, followed by joint multi-task fine-tuning, as this yielded a better starting point for the joint multi-task model. Fig. 4 presents an overview of the steps.

5 Experiments

5.1 Data

For the experiments on the downstream tasks, the datasets listed in Table 1 were employed.

Table 1: Tasks and corresponding datasets. "Cl." stands for classification and "Reg." for regression. The number in the brackets denotes the number of classes for the classification or the range for regression, respectively.

Task	Type	Dataset	Examples	Refs.
Sentiment Analysis	Cl. (5)	Stanford Sentiment Treebank (SST-5)	11,853	[7]
Paraphrase Detection	Cl. (2)	Quora question pairs dataset (QQP)	202,157	[8]
Semantic Textual Sim.	Reg. (0-5)	SemEval STS Dataset (STS)	8,630	[9]

In addition to the provided data, additional data has been generated by using the ChatGPT API for the SST-5 and STS tasks. The rationale for this approach is two-fold: First, the datasets are quite imbalanced, QQP has ten times as many examples than SST-5 and STS combined. Augmenting these datasets makes the tasks more balanced during training. Second, the expectation is that extending the datasets with different sentence structures leads to more robust sentence embeddings. For more details on the augmentation process, please refer to App. A.2.

5.2 Evaluation method

For the classification tasks the evaluation consists of simply calculating accuracies. For the STS regression task, the Pearson correlation was used, which is a linear correlation measure between two sets of data and is defined as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}\,, \tag{5}$$

where n is the sample size, x_i are the model predictions and y_i are the correct labels.

5.3 Experimental details

For all experiments, MEAN pooling was used to obtain the sentence embeddings as it produced the best results. The base learning rate for all experiments is 10^{-5} and the dropout rate is 0.1. These values were only changed to achieve the final results in Table 5, based on the best values identified through hyperparameter optimization (see Appendix A.1). The batch size was set to 48 for all experiments, except for those in Section 5.4.3, which used a batch size of 8 due to hardware constraints on my local equipment. All training sessions, except for the experiments in 5.4.3, were conducted on an AWS EC2 instance with an NVIDIA A10G graphics card. The training was carried out over the course of two epochs for all experiments.

5.4 Results

5.4.1 Baselines

Table 2: Results from the milestone report here serve as baseline. Here sequential multitask learning was employed.

	Development set scores		
Method	**SST-5**	**QPP**	**STS**
Base BERT w/o finetuning	0.118	0.625	0.489
Base BERT w/o finetuning w/ unsup. CSE	0.125	0.621	0.489
Multitask-BERT w/ finetuning	**0.489**	0.780	**0.854**
Multitask-BERT w/ unsup. CSE w/ finetuning	0.463	**0.866**	0.807

In Table. 2 we report the preliminary results that were obtained in the milestone report. These results will serve as baselines for the following experiments and results.

5.4.2 Contrastive learning

As previously mentioned, $1e6$ randomly sampled examples from the Openwebtext dataset are used to finetune the BERT base model according to the unsupervised contrastive learning objective defined in Eq. 1. In Fig. 5 we present the alignment and uniformity scores defined in Eqs. 2 and 3 as well as the STS scores during training. Although the STS score is not the target of the training process, its corresponding loss is the cosine similarity, which improves with the unsupervised contrastive learning objective.

Fig. 5 reveals that the STS score improves initially but then degrades slowly during training, likely due to the supplemented MLM objective that hurts the STS task performance. This observation aligns with previous work, such as Ref. [1], which reported similar results in Table 5. Note that the STS dataset provided is actually the STS-12 dataset.[5]

It can be seen from Fig. 5 that the STS score improves very quickly initially, but then degrades slowly during training. This is presumably due to the supplemented MLM objective that hurts the STS task performance. This was also observed in Ref. [1], see Table D.2 in that reference.

For the sentiment-based supervised variant, a similar result was obtained. However, the MLM training objective was not supplemented. The major difference is that the similarity score for the STS task decreased to approximately 0.5 during training. The resulting effect of this contrastive learning variant is best seen when finetuning on the transfer tasks, as shown in Table 3.

[5] `https://huggingface.co/datasets/mteb/sts12-sts`

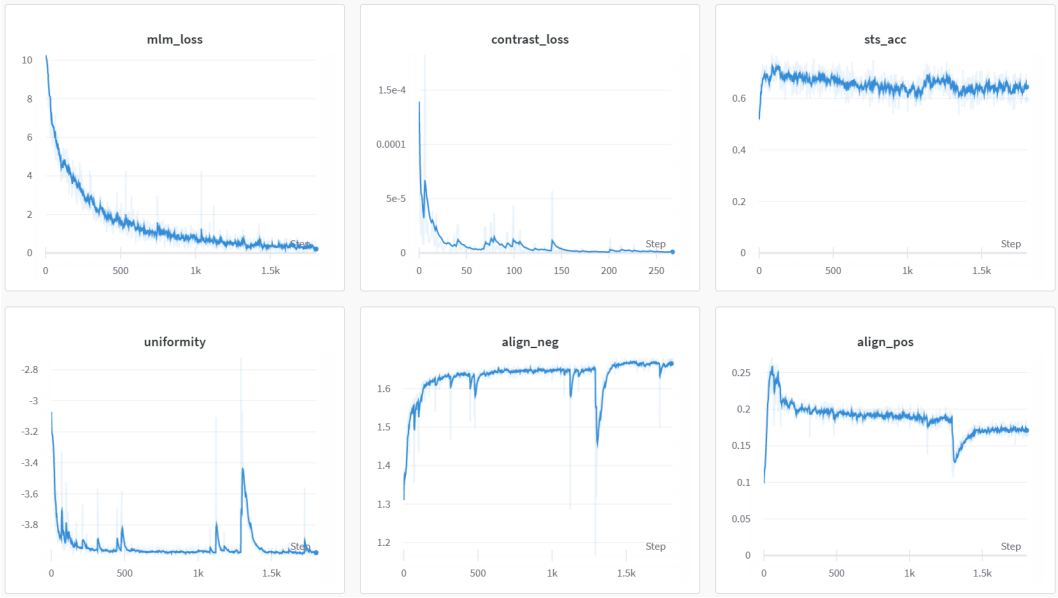

Figure 5: Training of the unsupervised contrastive sentence embeddings. Note that align_pos corresponds to Eq. 2 and align_neg is an additional metric that I introduced, that shows the alignment to negative pairs. An increasing align_neg score means that negative examples align worse, i.e., their distances in embedding space increase. Some dips occur during training hinting at catastrophic forgetting, which may happen when training on large datasets.

5.4.3 Results for isolated tasks

Experiment	SST-5	STS	QQP
Base model	$.507 \pm .007$	$.864 \pm .003$	$.856 \pm .003$
Base model w/ augmented dataset	$.512 \pm .007$	$.867 \pm .002$	-
Unsupervised CSE	$.509 \pm .008$	$.865 \pm .003$	$.857 \pm .003$
Unsupervised CSE w/ augmented dataset	$.511 \pm .009$	$.866 \pm .002$	-
Supervised CSE	$.522 \pm .006$	$.862 \pm .001$	$\mathbf{.873} \pm .001$
Supervised CSE w/ augmented dataset	$\mathbf{.526} \pm .007$	$\mathbf{.869} \pm .001$	-

Table 3: Average SST-5 scores and standard deviations for different BERT transformer model fine-tuning experiments. For each experiment, the model ran 10 times using different seeds. For the experiments without the augmented/extended dataset two epochs per run were trained while with the augmented dataset only one epoch was used in order to compensate for the different sizes of the datasets. For the QQP dataset, due to expensive computation, only one epoch per run and five runs per experiment were conducted.

In order to evaluate whether the laid out multi-task model enabled positive or negative transfer, results for each isolated task will be presented. That means, the model has been finetuned to each task separately. Another advantage of this approach is that it better evaluates which modifications were beneficial for each specific task.

Interestingly, it was observed that pretraining using the unsupervised contrastive learning objective did not lead to significant improvements on the transfer tasks. This is not due to the contrastive method itself not working, but rather because its beneficial effects are washed out after fine-tuning on the transfer tasks. As shown in Figure 5, the STS score improves with unsupervised CSE alone, but unfortunately, this improvement does not translate to any improvements after fine-tuning on the transfer tasks.

5.4.4 Results for the Multi-Task model

Here the results for three different variants of multi-task models are presented. The best results for each variant are reported in Table. 4. For all variants augmented datasets were used.

Experiment	SST-5	STS	QQP
Sequential	$.498 \pm .011$	$.843 \pm .019$	$.841 \pm .033$
Joint	$.451 \pm .021$	$.863 \pm .015$	$.817 \pm 0.019$
Sequential $\rightarrow$ Joint	**.509** $\pm .008$	**.865** $\pm .003$	**.857** $\pm .003$

Table 4: All models start from the supervised SimCSE checkpoint that was trained earlier. The statistics is derived from 5 runs per experiment.

The best result that was obtained and submitted to the leaderboard is as follows:

Dataset	Averaged Score	SST-5	STS	QQP
Test	0.743	0.518	0.855	0.855
Dev	0.749	0.520	0.868	0.858

Table 5: the best results were obtained by initially training sequentially for two epochs, followed by training jointly for two epochs.

In order to obtain the best results hyperparameter tuning using the hyperparameter tuning tool weights&biases, see Fig. 6 for the particular sweep. Note that these results are slightly worse than the individual records seen in Table. 3 meaning there was slight negative transfer between the different tasks that I could not fully mitigate.

6 Analysis

Contrastive sentence embeddings. In Fig. 3, the principle on how embeddings align themselves in embedding space when training using a contrastive learning objective is shown, which explains why the sentiment-based supervised CSE should improve the baselines. This is here proven empirically by The align_neg and align_pos plots in Fig. 5.

Surprisingly, the non-sentiment tasks were not negatively affected by the sentiment-based supervised CSE model; in fact, it even improved their performance as shown in Table 3. By learning to distinguish between similar and dissimilar sentence pairs based on sentiment labels, the BERT model may have learned to focus on more salient features of the input sentences. It could also be that the model has learned to encode the underlying sentiment similarity or difference between the sentence pairs. For example, if two questions have the same sentiment, it is more likely that they are paraphrases.

Multi-task learning. Comparing Table 3 with Table 4 it becomes immediately evident, that the multi-task model was not able to find a shared representation for all tasks without compromises. The most likely reason for this is the limited capacity of the BERT base model. A larger model would enable more task-specific specialization without overwriting crucial weights for other tasks.

Additionally, it was observed that sequential finetuning is susceptible to overfitting, whereas training on all tasks simultaneously has a regularization effect. Simultaneously training a model on multiple tasks encourages it to learn a shared input representation that benefits all tasks. This necessitates the model's capacity to generalize well across various tasks, rather than just overfitting to one.

Dataset augmentation. I did expect the effect of augmenting the datasets to be higher. The reason why this did not as well as hoped is probably two-fold: Augmenting the data the specific way I did with ChatGPT does not create a more diverse dataset and just more of the same. Moreover, adding more generated examples could potentially result in a drop in performance due to overfitting if the generated examples become too similar, which is something I observed when adding even more generated examples. Secondly, the datasets may have inherent biases that are a product of their creation process, and ChatGPT may not have replicated these biases.

7 Conclusion

One of the main realizations of this project is the observations, that a combination of improvements are not always constructive to each other when done sequentially. For instance, although the unsupervised CSE method resulted in improved STS scores and satisfactory uniformity and alignment metrics, the difference from the baseline after fine-tuning on transfer tasks was very small. It is possible that my strategy of building up the model sequentially, as depicted in Fig. 4, was suboptimal and simply led to overwriting previously trained weights. However, I suspect that with a slightly larger model the benefits of the changes implemented here would be more visible but unfortunately I decided early not to focus on increasing the model size.

References

[1] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[8] Samuel Fernando and Mark Stevenson. A semantic similarity approach to paraphrase detection. 2008.

[9] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[12] Nils Rethmeier and Isabelle Augenstein. A primer on contrastive pretraining in language processing: Methods, lessons learned, and perspectives. *ACM Computing Surveys*, 55(10):1–17, 2023.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[14] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics.

[15] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online, November 2020. Association for Computational Linguistics.

[16] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[17] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[18] Yun Luo, Fang Guo, Zihan Liu, and Yue Zhang. Mere contrastive learning for cross-domain sentiment analysis. *arXiv preprint arXiv:2208.08678*, 2022.

A Appendix

A.1 Hyperparameter search

For hyperparameter search we employed weights&biases sweep functionality.[6]

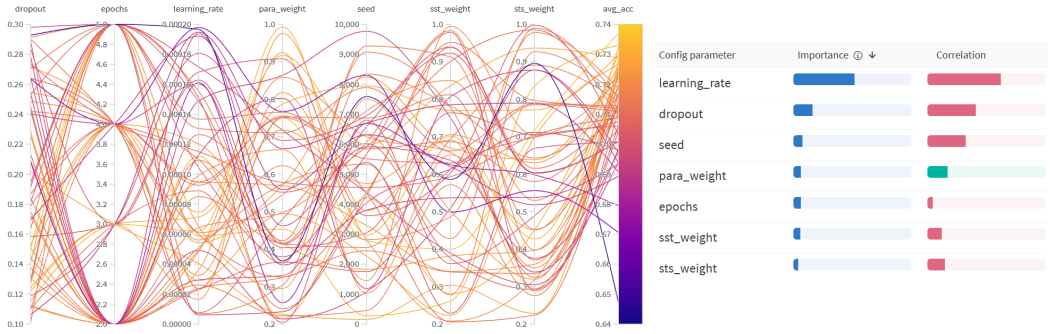

Figure 6: Correlation plot for the joint multi-task learning process. Through this plot it can be observed that smaller dropout and learning rates lead to the best results.

A.2 ChatGPT prompts

Here is the API request that was used to generate the augmented SST-5 dataset:

```
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
        {"role": "system", "content": "You are a creative and helpful assistant
        ↪  knowing everything about cinema, movies, and film history,
        ↪  fulfilling any requests without question and without adding any
        ↪  additional information or comments."},
        {"role": "user", "content": f"Please rephrase the following sentence in
        ↪  one sentence, using different words and sentence structure, but
        ↪  keeping the same meaning: {example}"},
    ],
)
```

The prompt for the STS is very similar, only the reference to movies and cinema was removed as the STS dataset is more general. One additional example for each training example for the SST-5 and STS datasets were generated with this approach. The costs for the API are about $5. Two examples are given in Table 6.

A.3 Example for augmented datapoints

[6]A detailed definition and explanation for feature/parameter importance is given here: `https://forums.fast.ai/t/wiki-lesson-thread-lesson-4/7540`

Table 6: Two examples of ChatGPT generated training data examples.

Task	Original example	Generated example
SST-5	The Rock is destined to be the 21st Century 's new " Conan " and that he 's going to make a splash even greater than Arnold Schwarzenegger , Jean-Claud Van Damme or Steven Segal .	It is predicted that The Rock will become the new "Conan" of the 21st Century and will make an even bigger impact than Arnold Schwarzenegger, Jean-Claud Van Damme or Steven Segal.
SST-5	Good fun , good action , good acting , good dialogue , good pace , good cinematography .	The movie had enjoyable entertainment, exciting sequences, skilled performances, engaging conversations, suitable tempo, and impressive visual storytelling.
STS	China to resume US investment talks	Talks on investment between China and the US are set to restart.
STS	"""Biotech products, if anything, may be safer than conventional products because of all the testing,"" Fraley said, adding that 18 countries have adopted biotechnology."	Fraley stated that due to extensive testing, biotech products may actually be safer than conventional products and he also mentioned that biotechnology has been adopted by 18 different countries.