BERT-CF: Contrastive Flows for MultiTask-BERT

Stanford CS224N Default Project

George Hu
Department of Computer Science
Stanford University
gehu@stanford.edu

Abstract

Generating sentence-level embeddings that generalize across language understanding tasks remains a difficult task, given the shorter context of many sentences. BERT, an architecture for sentence encodings using masked language modeling with bidirectional Transformers, has spawned a variety of methods for robust representations under both unsupervised and supervised contexts (Devlin et al., 2019). Extensions of BERT have used contrastive learning (Gao et al., 2021), which aims to pull together similar sentences and push apart differing sentences, and generative flows (Li et al., 2020), which aim to project embeddings into a more symmetric space, to improve unsupervised semantic similarity. In BERT-CF, we aim to combine these into one pretraining schema, and evaluate in both the unsupervised and multitask supervised finetuning domains.

1 Key Information to include

- Mentor: Candice Laine Penelton

2 Introduction

Recent developments in Natural Language Understanding and Natural Language Generation, spearheaded by the Transformer Architecture introduced in Vaswani et al. (2017), have quantitatively surpasses previous benchmarks in almost every setting, and qualitatively comes close to matching human performance (Raffel et al., 2019).

In the shorter context sentence understanding domain, BERT has found itself as the baseline across a number of tasks such as textual entailment, sentiment analysis, and textual similarity Devlin et al. (2019). Extensions of BERT have integrated various additional methods to improve language understanding performance and dive deeply into minutia on properly fine-tuning BERT, as current baselines still sometimes fail to match human language understanding (Reimers and Gurevych, 2019).

One pertinent failure mode of BERT is its ability to predict unsupervised textual similarity (Gao et al., 2021). In particular, Reimers and Gurevych (2019) show that BERT, when evaluated on SemEval benchmarks without fine-tuning, perform worse that bag of words representations using GloVe vectors (Pennington et al., 2014). One common approach to improve BERT representations has been self-supervision through contrastive learning (Wang and Isola), which learns to align positive pairs of embeddings close to each other and negative pairs of far apart from each other (Chen et al., 2020). This has been applied to BERT in the works of Gao et al. (2021) and Liu et al. (2021), and we will elaborate on these methods in section 3.

A common observation of why sometimes BERT fails to generalize to certain domains is the anisotropy of its predicted sentence embeddings; that is, they are tightly clustered into a hyper-cone rather than distributed more uniformly in the embedding space. Li et al. (2020) identify this issue as a primary cause for poor unsupervised generalization of BERT for various tasks. The authors, inspired by Glow (Kingma and Dhariwal, 2018) from computer vision, show that learning a mapping of BERT sentence embeddings into a gaussian space using generative flows produces embeddings with better generalization. A visualization of BERT's anisotropy can be found in figures 2 and 3 in the appendix.

Another common trend in NLP has been to employ unified, or multitask learning, based upon the fact that many language tasks are quite related to each other. This idea of a unifed text transformer is best exemplified in the T5 model, which has used these cross domain similarities advantageously to generate seemingly human-like text (Raffel et al., 2019). In BERT-CF we further explore this area based upon previous work by Stickland and Murray (2019) and Bi et al. (2022).

3 Related Work

3.1 MirrorBERT

In MirrorBERT, Liu et al. (2021) employ contrastive learning on an unsupervised text corpus to improve BERT's sentence embeddings.Liu et al. (2021) generate positive pairs of sentence embeddings $(f(x_i), f'(x_i'))^+$ for a BERT encoder f by re-randomizing another dropout mask with the same dropout probability p to create the siamese twin encoder f'. Then, they apply character level span-masking to augment the input to x_i'. Similar to other contrastive learning methods on unstructured text, negative pairs are formed by pairing different sentences in the same batch, both with and without augmentation.

MirrorBERT employs the InfoNCE (van den Oord et al., 2018) loss with cosine similarity in order to increase the similarity between positive pairs and push apart the negative pairs. Liu et al. (2021) also uses large positive and negative entailment datasets with a more complex training formulation to further improve their performance, but the idea of BERT-CF is to only focus on general unstructured sentence corpuses, so we do not explore that.

Note that this strategy is equivalent to unsupervised SimCSE by Gao et al. (2021) if no span-masking is employed. Both of these method greatly increase performance on unsupervised textual similarity such as spearman correlation on SemEval Agirre et al. (2013) tasks.

3.2 BERT-flow

In BERT-flow, Li et al. (2020) attempt to tack the anisotropy of BERT embeddings directly by emulating the work of Kingma and Dhariwal (2018) to project BERT embeddings into a standard gaussian space. Let $\mathcal{Z}$ be the standard multivariate gaussian with distribution p_Z, and $\mathcal{B}$ be the space for the pretrained BERT output with unknown distribution p_B.

The flow mapping $g : \mathcal{Z} \rightarrow \mathcal{B}$ is defined as the composition of k flow blocks $g = g_1 \circ g_2 \circ \ldots g_k$. Each flow block is designed as a dimensionally permutation invariant invertible scaling so that the inverse function g^{-1} maps the vectors to gaussian $\mathcal{Z}$. During training, the BERT encoder is frozen and we learn g by maximizing the log-likelihood of the resulting vector in $\mathcal{B}$. For optimal performance, BERT-flow can be trained on task-specific sentences, but Li et al. (2020) show that generalized pretraining also performs well.

4 Approach

In BERT-CF, we aim to take advantage of the distributional effects of generative flow and the similarity heuristics of contrastive learning by combining MirrorBERT and BERT-flow pretraining. The central idea we employ is first doing self-supervision with MirrorBERT to induce sentence similarity generalization, and then freeze the BERT encoder and jointly train MirrorBERT and BERT-flow. This joint training is essentially learning the flow mapping of BERT-flow while using MirrorBERT as a regularizer.

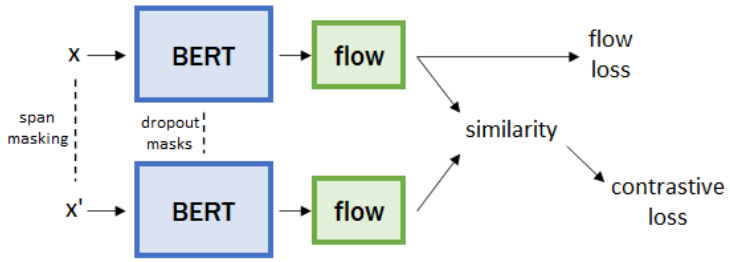

Figure 1: Self-supervised pretraining architecture for BERT-CF generative flow with contrastive regularization. The initial round of contrastive learning has no flow blocks nor flow loss.

The end result of this pretraining allows for unsupervised evaluation of BERT-CF, which we will compare to just MinBERT, along with just standalone MirrorBERT, SimCSE, and BERT-flow adaptations. Then, our supervised fine-tuning head will simply use jointly optimized task-specific linear projections for sentiment analysis, paraphrase identification, and semantic similarity.

4.1 Contrastive Learning

We employ a pretraining formulation almost identical to MirrorBERT in section 3.1, but change character-level span masking to token-level span masking. This is inspired by Joshi et al. (2019), which modifies the masked language modeling of BERT to mask spans of tokens, and in general we do not find much reasoning behind masking out an arbitrary spans of $s = 5$ characters so that the word tokens are cut to non-existing words in Liu et al. (2021).

Thus, for sentence batches $\{x_i\}_{i=1}^{b}$, we similarly generate the set of positive and negative targets for x_i to be $N(x_i) = f'(x_i') \cup \bigcup_{j \neq i} \{f'(x_j'), f(x_j)\}$ with $|N(x_i)| = 2b - 1$. f and f' have differing dropout masks with dropout ratio p_d, and x_i' is formed by token span masking under the a geometric distribution with success probability p_g. The pretraining objective is the InfoNCE loss, which we can expressed as

$$L_{\text{mirror}} = -\frac{1}{b} \sum_{i=1}^{b} \log \frac{e^{\text{sim}(f(x_i), f'(x_i'))/\tau}}{\sum_{f_n \in N(x_i)} e^{\text{sim}(f(x_i), f_n)/\tau}}$$

where sim is the cosine similarity $\text{sim}(u, v) = \langle u, v \rangle / (||u||_2 ||v||_2)$ and τ is the temperature scale.

4.2 Generative Flow with Contrastive Regularization

For learning a generative flow, we use the same implementation as BERT-flow as discussed in section 3.2. We learn flow mapping head $g : \mathcal{Z} \to \mathcal{B}$, defined as the composition of k flow blocks $g = g_1 \circ g_2 \circ \dots g_k$, and each flow block $g_i(u) = v$ where $u, v \in \mathbb{R}^D$ does the following:

$$u = s \odot u + b \qquad \text{(ActNorm)}$$

$$u = Wu \qquad \text{(Invertible 1x1 Conv)}$$

$$v_{1:d} = u_{1:d} \quad , \quad v_{d+1:D} = \text{AC}(u_{d+1:D}) \qquad \text{(Additive Coupling)}$$

for embedding dimension D and hyperparameter d normally set to $d = \lfloor D/2 \rfloor$. The ActNorm employs learnable s and b initialized as scale and bias for standardizing u, and the additive coupling AC is a 3-layer fully connected network with ActNorm normalizations. Further details on these transformations can be found in Kingma and Dhariwal (2018), and we use an implementation from this implementation used in TSDAEWang et al. (2021).

Given text input x, we want to maximize the log-likelihood of $p_{\mathcal{B}}(v)$, and we employ the change of variables inverse-mapping that Li et al. (2020) and Kingma and Dhariwal (2018) use, deriving the probability from the standard gaussian $p_{\mathcal{Z}}$. Thus, the pretraining negative log-likelihood loss is $L_{\text{flow}} = -\log(p_{\mathcal{Z}}(g^{-1}(f(x)))) - \log \left| \frac{\partial g^{-1}(f(x))}{\partial f(x)} \right|$.

Putting this together with contrastive regularization, we pass the final output $F(x) = g(f(x))$ into the MirrorBERT module and jointly optimize

$$L_{joint} = \lambda_{\text{mirror}} L_{\text{mirror}} + L_{\text{flow}}$$

to learn just the flow blocks g, freezing the BERT-encoder f.

4.3 Multitask Fine-tuning

We fine-tune BERT-CF on sentiment analysis, paraphrase prediction, and semantic similarity using jointly optimized linear projection heads that take in the BERT-CF output $f(x) \in \mathbb{R}^D$.

- For sentiment analysis on C classes, we use a linear layer $W_{\text{sent}} \in \mathbb{R}^{D \times C}$ to get class logits, and use cross entropy loss to obtain L_{sent}.
- For paraphrase prediction on two inputs x_1 and x_2, we pass each $f(x_1)$ and $f(x_2)$ through a linear layer $W_{\text{para}} \in \mathbb{R}^{D \times e_{\text{para}}}$, and use the cosine similarity of the resulting paired embeddings, as done in SBERT (Reimers and Gurevych, 2019). We find it useful to use a lower bound to connote dissimilarity where anything under that margin is equivalently dissimilar, inspired by the margin proposed in Wilkinson and Brun (2016). We call this hinge h_{para}, and clamp the cosine similarities to $[h_{\text{para}}, 1]$ so that $y_{\text{para}} = \max(\text{sim}(f(x_1), f(x_2)), h_{\text{para}})$.

We then use the cosine embedding loss in Wilkinson and Brun (2016) to obtain $L_{\text{para}} = \mathbb{1}_{\{y_{true}=1\}}(1 - y_{\text{para}}) + \mathbb{1}_{\{y_{true}=-1\}}(y_{\text{para}} - h_{\text{para}})$

- For semantic similarity, the approach is similar to paraphrase prediction, projecting the embeddings using the linear layer $W_{\text{sem}} \in \mathbb{R}^{D \times e_{\text{sem}}}$. We use the hinge cosine similarity $y_{\text{sem}} = \max(\text{sim}(f(x_1), f(x_2)), h_{\text{sem}})$ as our output, and scale the semantic similarity ground truth labels linearly to $[h_{\text{sem}}, 1]$. The training objective L_{sem} used for this regression task is mean square error.

Thus, we perform multitask gradient updates combining these three losses with appropriate scaling to fine tune BERT-CF.

$$L_{FT} = \lambda_{\text{sent}} L_{\text{sent}} + \lambda_{\text{para}} L_{\text{para}} + \lambda_{\text{sem}} L_{\text{sem}}$$

5 Experiments

5.1 Data

For unsupervised pretraining, we use the same 10^6 randomly sampled sentences from English Wikipedia (Wiki1M) that was used in SimCSE (Gao et al., 2021), which is uploaded to HuggingFace. We find that these sentences actually oftentimes are just short section headers with very little context, so we remove all sentences length 5 or less, leaving us with 916,110 sentences.

For supervised fine-tuning, we use the datasets provided in the Default Project repository.

- Sentiment Analysis: Stanford Sentiment Treebank (SST-5), by Socher et al. (2013). This consists of single sentences that need to be classified into $C = 5$ sentiment classes.
- Paraphrase Identification:, Quora Question Pairs (QQP), by Dey et al. (2016). This consists of sentence pairs that are either paraphrases of each other, or not, as a binary label.
- Semantic Similarity, SemEval 2013 (STS-13), by Agirre et al. (2013). This consists of sentence pairs with semantic similarity labeled continuously from 0 to 5.

5.2 Evaluation method

5.2.1 Unupervised Evaluation

We evaluate BERT-CF in an unsupervised way on the STS-13 development set to compare the robustness of embeddings without finetuning. To do this, for input pairs (x_1, x_2) and BERT encoder f, we just calculate the cosine similarity $\text{sim}(f(x_1), f(x_2))$ of the encoded representations. This can be directly fed into the correlation metric with the actual 0 through 5 similarity labels. We choose spearman correlation here to measure ordinal rather than the linear relationship of pearson correlation, as we cannot guarantee that unsupervised similarities match linearly without label calibration (Reimers and Gurevych, 2019).

5.2.2 Supervised Evaluation

- For SST-5, we calculate total accuracy from the class-wise argmax of the logits.
- For QQP, we employ the cosine similarity with the hinge h_{para}, similar to training. Given input pair (x_1, x_2), we have $y_{\text{para}} = \max(\text{sim}(f(x_1), f(x_2)), h_{\text{para}})$. Negative predictions are represented by $y_{\text{para}} \in [h_{\text{para}}, (1 + h_{\text{para}})/2)$, and positive predictions are represented by $y_{\text{para}} \in [(1 + h_{\text{para}})/2, 1]$. The evaluation metric is accuracy.
- For STS-13, we similarly get $y_{\text{sem}} = \max(\text{sim}(f(x_1), f(x_2)), h_{\text{sem}})$ and linearly scale this from $[h_{\text{sem}}, 1]$ to $[0, 5]$ to get the predicted similarity. The evaluation metric is pearson correlation r.

5.3 Experimental details

Whereas Devlin et al. (2019) design the [CLS] token output in BERT to be the encoder output, we find that using the mean of the last transformer block sequence output, masked over valid tokens, such as what SimCSE and MirrorBERT do, provides better results (Gao et al., 2021) (Liu et al., 2021). For all experiments, $f(x)$ refers to the mean of the last transformer block.

For MirrorBERT pretraining, we use batch size $b = 16$, learning rate $\alpha = 10^{-4}$, dropout $p_d = 0.1$, span-masking geometric distribution rate $p_g = 0.3$, and hard-cap the span-mask length to 5. For SimCSE experiments, we just cap the span-masking to length 0 so that we only have dropout.

For BERT-flow pretraining, we use batch size $b = 16$, learning rate $\alpha = 3 \times 10^{-4}$, and 3 flow blocks, the same number as in Li et al. (2020). When doing joint optimization of both MirrorBERT and BERT-flow, we keep the same hyperparamters and use $\lambda_{\text{mirror}} = 0.3$ to match the MirrorBERT learning rate.

In all pretraining tasks, we choose the model checkpoint that maximizes the STS-13 spearman correlation, and we empirically find that not many optimizer steps are needed for the performance to plateau, so we perform $t = 10^4$ steps of training on Wiki1M.

For supervised fine-tuning, we use $e_{\text{para}} = 768$ and $e_{\text{sem}} = 768$ embedding sizes, $h_{\text{para}} = 0.7$, and $h_{\text{sem}} = 0$ hinge values. We scale the constituent losses as $\lambda_{\text{sent}} = 5$, $\lambda_{\text{para}} = 20$, and $\lambda_{\text{sem}} = 1$ with learning rate $\alpha = 10^{-5}$. Each gradient update contains a batch of size 32 from SST-5, a batch of size 96 from QQB, and a batch of size 32 for STS-13, and we have all layers unfrozen. For the sentiment analysis layer we have a dropout of 0.3; for the paraphrase identification layer we have no dropout, and for the semantic similarity layer we have a dropout of 0.1. We keep the BERT-encoder dropout of 0.1, as in the original paper (Devlin et al., 2019). We fine-tune for 5000 steps, and pick the model that has the highest average development set metrics.

All experiments use the AdamW optimizer with weight decay $\gamma = 0.01$ (Loshchilov and Hutter, 2017) and the default random seed 11711.

5.4 Results

Encoder	*Unsupervised*	*Supervised*		
	STS-13 (dev ρ)	**SST-5** (dev)	**QQP** (dev)	**STS-13** (dev r)
MinBERT	0.514	0.512	0.821	0.848
MinBERT-SimCSE	0.745	0.511	0.825	0.852
MinBERT-Mirror	0.736	0.522	0.82	<u>0.858</u>
MinBERT-flow	0.517	0.506	0.821	0.857
BERT-CF	<u>0.760</u>	<u>0.525</u>	<u>0.841</u>	0.857

Table 1: Unsupervised and supervised development set metrics for MinBERT encoders. We can see that BERT-CF outperforms the other baselines.

Taking a look at the unsupervised performance on STS-13, the base MinBERT performance is quite poor, and when we just learn a generative flow on top, there is minimal increase. This makes sense, as the flow only uses the geometry of sentence embeddings and does not add any informational heuristic on sentence similarities. On the other hand, the contrastive learning modules add significant performance boosts, with BERT-CF, our method, achieving an impressive spearman correlation of 0.760, a 47.9% increase over the base model.

When we multitask fine-tune the models on SST-5, QQP, and STS-13, there is not that drastic performance change, but in general all the pretraining methods marginally improved performance, with BERT-CF improving the most, averaging a 1.6% increase over the base fine-tune. This is not too surprising, as much of the fine-tuning performance comes from the last few transformer layers adapting, regardless of the improved pretrained embeddings (Liu et al., 2019).

Encoder	**SST-5** (test)	**QQP** (test)	**STS-13** (test r)
BERT-CF	0.522	0.840	0.853

Table 2: Test set supervised metrics for BERT-CF

6 Analysis

6.1 Ablation Studies

Replacing MirrorBERT with SimCSE

We try replacing MirrorBERT with SimCSE in both the contrastive learning and joint flow pretraining, as standalone SimCSE seems to perform comparably, if not better than standalone MirrorBERT. But

interestingly enough, we get that the joint flow addition to SimCSE does not improve performance much.

Encoder	**STS-13** (dev ρ)
MinBERT-SimCSE	0.745
MinBERT-SimCSE-joint	0.746
MinBERT-Mirror	0.736
MinBERT-Mirror-joint (BERT-CF)	0.760

We suspect that due to the span-masking in MirrorBERT, the contrastive learning affects a greater variety of sentences, so learning the additional flow-mapping affecting the distribution of the whole BERT output space provides more benefit than for the narrower SimCSE pretraining.

Different Formulations of Combining Contrastive Learning and Generative Flow

Something we also explore is different ways of combining MirrorBERT and BERT-flow. In particular, we attempt to just learn the flow with contrastive regularization jointly while unfreezing the BERT encoder for the contrastive update (call this MinBERT-joint), and also to do the contrastive learning, then learn the flow without contrastive regularization (call this MinBERT-Mirror-flow).

Encoder	**STS-13** (dev ρ)
BERT-CF	0.760
MinBERT-joint	0.629
MinBERT-Mirror-flow	0.739

We find that only doing the joint optimization fails at both optimizaing the contrastive objective and flow mapping, as the contrastive signal through the flow blocks is reduced, and the flow likelihood objective is essentially a moving target. For MinBERT-Mirror-flow without the contrastive regularization when learning generative flow, we surmise that there is nothing inherently wrong with this approach, but just that adding the joint objective further optimizes the flow blocks.

6.2 Qualitative Analysis

Here, we show some qualitative examples on STS-13 of how BERT-CF can improve semantic similarity completely unsupervised.

Sentence Pairs	**MinBERT Cos Sim**	**BERT-CF Cos Sim**	**Label**
'some guy sitting on a couch watching television .'	0.862	0.890	5.0
'a guy is sitting on the couch watching tv'			
'a girl is eating a cupcake .'	0.967	0.927	2.6
'a woman is eating a cupcake .'			
'more than 1 ,000 inmates escape from libya 's al-kweifiya prison'	0.740	0.634	4.4
'1000 prisoners escape from libyan jail'			

Table 3: Sentence pairs sampled from the STS-13 development set, and their unsupervised cosine similarities on base MinBERT and BERT-CF

In the first and second examples, BERT-CF pretraining correctly increases and decreases the similarity respectively. We hypothesize that "girl" and "woman" are represented well in Wiki1M, so the differentiation between "girl" and "woman" is learned from the contrastive pretraining. And the same would apply to the similarity between "television" and "tv". In the third example, both MinBERT and BERT-CF incorrectly have lower similarity than desired. It is likely that the difficulties in equating "1 ,000" with "1000" due to tokenization and "libya 's al-kewifiya prison" with "libyan jail" due to the very specific information is not improved by BERT-CF.

And after fine-tuning, some of these failure modes persist, but BERT-CF still shows impressive semantic similarity performance.

Sentence Pairs	BERT-CF Prediction	Label
'russian officials have called for a conference on the conventional forces in europe treaty to discuss ratification of the amended treaty .'	3.651	3.2
'antonov spoke the day before a conference on the conventional forces in europe treaty .'		
'more than 1 ,000 inmates escape from libya 's al-kweifiya prison'	3.310	4.4
'1000 prisoners escape from libyan jail'		

Table 4: Sentence pairs sampled from the STS-13 development set, and their BERT-CF fine-tune predictions

In first example, BERT-CF is surpsingly decent at assessing the moderate amount of similarity between the treaty statements, and we think this to be partially coincidence, but also the model having longer context. For the second example, BERT-CF still performs poorly, as the fine-tuning still seems to fail to provide the specific jail name information.

Sentence Pairs	BERT-CF Prediction	Label
'what is the answer to this question ? (see description)'	0	1
'what are the answers to these questions ?'		
'does reporting fake names on quora do anything ?'	1	1
'is it worth it to report fake names on quora ?'		

Table 5: Sentence pairs sampled from the QQP development set, and their BERT-CF fine-tune predictions

The fine-tune performance on QQP shows how BERT-CF is able to correctly assign similar question connotations together, as with the second example. The 5-gram "report fake names on quora" does help, but the rest of the question is quite dissimilar, so assigning is as a correct paraphrase is impressive. However, in the first example which is quite related, BERT-CF fails to note that "(see description)" does not change the question's meaning.

Sentence	BERT-CF Prediction	Label
'in a way , the film feels like a breath of fresh air , but only to those that allow it in .'	1	4
'no one goes unindicted here , which is probably for the best .'	2	2
'unlike the speedy wham-bam effect of most hollywood offerings , character development – and more importantly , character empathy – is at the heart of italian for beginners .'	4	4

Table 6: Sentence pairs sampled from the SST-5 development set, and their BERT-CF fine-tune predictions

The sentiment analysis dataset SST-5, derived from movie reviews, is particularly difficult due to the sardonic tone of many movie critics. In the above first example, BERT-CF likely misses far here because the "but only" portion is an often setup for a contradiction that would scathe the movie. But in the next two examples, BERT-CF does a good job, correctly not focusing on "best" in the second

example to output a negative sentiment, and not falling into the contradiction trap in the 3rd example with the "unlike" clause.

6.3 Alignment and Uniformity

Wang and Isola introduce the notions of alignment and uniformity for contrastive self-supervision methods in order to respectively measure how aligned positive pairs are and how uniformly distributed all the embeddings are. In our case, we use the QQP dataset, evaluating our encoders without fine-tuning in an unsupervised fashion on the training set, since QQP has ground truth positive pairs and spans many different types of questions. We emulate the formulation in Gao et al. (2021) for a dataset of positive and negative sentence pairs $\mathcal{D} = \mathcal{P} \cup \mathcal{N}$.

$$\text{Alignment} = \frac{1}{|\mathcal{P}|} \sum_{x_i, x_j \in \mathcal{P}} ||f(x_i) - f(x_j)||_2^2$$

$$\text{Uniformity} = \log \left(\frac{1}{|\mathcal{D}|} \sum_{x_i, x_j \in \mathcal{D}} \exp\left(-2||f(x_i) - f(x_j)||_2^2\right) \right)$$

For both these metrics, lower is better.

Encoder	Alignment	Uniformity
MinBERT	0.216	-1.425
MinBERT-SimCSE	0.397	-3.264
MinBERT-Mirror	0.391	-2.990
MinBERT-flow	0.499	-2.857
BERT-CF	0.416	-3.198

Table 7: Unsupervised Alignment and Uniformity on QQP Train

We interestingly find that all methods increase alignment from the original MinBERT embeddings, which is likely be due to the extremely high anisotropy initially. In learning the generative flow, we decrease alignment significantly as we expand out all the vectors, thus decreasing uniformity. We can also see that the contrastive learning methods actually perform better at uniformity than flow, indicating how separating the negatives from direct cosine similarity provides an extremely robust method to induce isotropy into embeddings, as Gao et al. (2021) and Liu et al. (2021) both note. The final alignment and uniformity for BERT-CF is strictly worse than MinBERT-SimCSE surprisingly, and throughout this paper we rely upon the assumption that unsupervised STS-13 performance is the best method for evaluating pretrained embeddings. All other experiments and the fine-tuning results agree with BERT-CF producing more generalizable embeddings, so further research into this area is needed.

7 Conclusion

In BERT-CF, we test different formulations of contrastive learning and generative flows as additional se;f-supervised training for MinBERT, and develop a multitask fine-tuning framework to evaluate sentence embedding quality. We show that our novel method of doing contrastive learning first, and then learning a flow mapping while keeping the contrastive objective as a regularizer, produces robust embeddings that have nice distributional and important semantic qualities without needing to look at any data labels. Our evaluation methods show how BERT-CF improves upon just contrastive and just flow approaches, and we find that the common failure modes of BERT-CF generally deal with legitimately difficult text. Since the literature for semantic similarity is mainly focused on SemEval datasets, we focus on them as well, but our results possibly show that additional analysis methods should be standard practice, such as alignment and uniformity. We find this area of doing additional pretraining with a focus on both semantic and distributional properties for MinBERT one still ripe to unveil, and hope that our work inspires further exploration.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *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. Association for Computational Linguistics.

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2016. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2880–2890, Osaka, Japan. The COLING 2016 Organizing Committee.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *CoRR*, abs/2011.05864.

Fangyu Liu, Ivan Vulic, Anna Korhonen, and Nigel Collier. 2021. Fast, effective and self-supervised: Transforming masked languagemodels into universal lexical and sentence encoders. *CoRR*, abs/2104.08027.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. 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. Association for Computational Linguistics.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. *CoRR*, abs/2104.06979.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere.

Tomas Wilkinson and Anders Brun. 2016. Semantic and verbatim word spotting using deep neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 307–312.

A Appendix

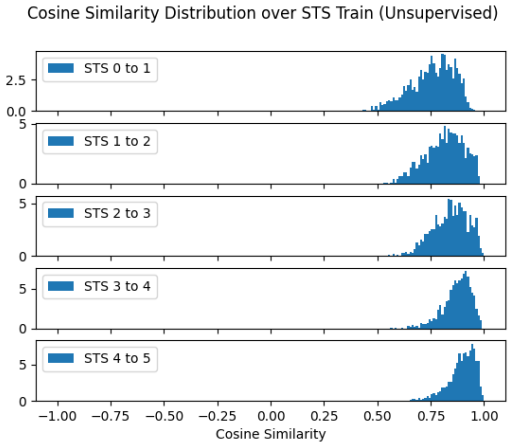

Figure 2: Distribution of cosine similarities for sentence pairs of STS-13, using unsupervised MinBERT. We can see how the sentence pairs all have high cosine similarity, indicating anisotropy.

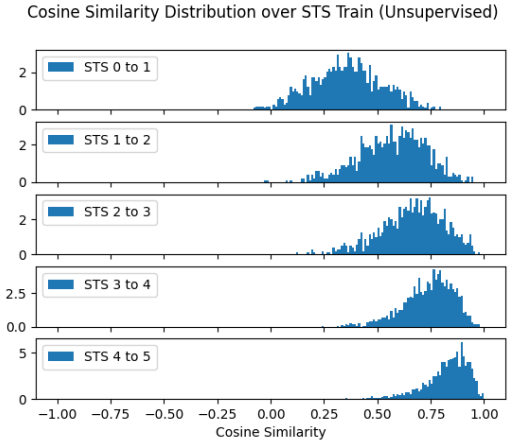

Figure 3: Distribution of cosine similarities for sentence pairs of STS-13, using unsupervised BERT-CF. We can see how the dissimilar sentence pairs have much lower cosine similarity.