対照損失による追加学習が BERT のファインチューニングにもたらす効果

竹中 誠 [1]　瀧 雅人 [2]
[1] 三菱電機株式会社　[2] 立教大学
Takenaka.Makoto@bc.MitsubishiElectric.co.jp taki_m@rikkyo.ac.jp

概要

本研究では，事前学習済み BERT に対する対照損失による追加学習が，下流タスクのファインチューニングに与える影響を実験的に調査する．実験では，事前学習済み BERT と，それを SimCSE で学習したモデルをファインチューニングするときの学習の安定性とモデルの可塑性の二つの観点で分析した．実験の結果，SimCSE で追加学習したモデルでは，(*i*) 大きい学習率でのファインチューニングがより安定的になり，(*ii*) パラメータに関するフィッシャー情報行列の有効ランクが回復することで，モデルの可塑性が向上することがわかった．

1　はじめに

BERT[1] のような事前学習済み言語モデルは，様々な自然言語処理タスクに広く用いられている代表的な手法の一つである．しかし事前学習済みモデルをそのまま用いると，任意の文ペアの類似度が大きくなり十分な表現力を持たないことが知られている [2]．これを改善するために，様々な手法が提案されている．文埋め込みに後処理を施す手法として，文埋め込みを等方的なガウス分布に変換する手法 [2] や，文埋め込みの白色化 [3] などが提案されている．BERT を追加学習する手法としては，対照損失を用いる手法が広く利用されている [4, 5, 6, 7, 8]．これらの多くの手法は，文埋め込みの異方性を解消することを主な目的とするため，STS タスクなどの意味的類似度タスクでは顕著な性能向上をもたらす一方で，事前学習済み BERT を下流タスクでファインチューニングする場合の性能向上は限定的である [9, 5, 6, 10, 7]．そこで一つの疑問が生じる．意味的類似度タスクなど BERT の埋め込みをそのまま利用するタスクで顕著な性能向上をもたらす手法は，ファインチューニングにどのような影響を及ぼすのか．BERT の活用方法としてファインチューニングして用いることは一般的であるにもかかわらず，この疑問にはあまり注意が払われてこなかった．そこで本研究では，対照損失による追加学習が下流タスクのファインチューニングにどのような影響を及ぼすのかを調査することを目的とする．

本研究の貢献は以下である．

- SimCSE による追加学習は，損失関数のヘシアンのランクを回復させる効果を持つことを実験的に示した．
- SimCSE による追加学習は，ファインチューニング中のヘシアンのランクの一様性を維持させる効果を持つことを実験的に示した．
- これらの効果がファインチューニングの安定化や下流タスクにおける性能向上をもたらす可能性があることを定性的に示した．

2　分析方針

本研究では，代表的な対照学習手法である SimCSE[5][1)] が下流タスクのファインチューニングにおよぼす影響を，安定性と可塑性の二つの観点で分析する．

2.1　SimCSE

SimCSE では以下で定義される InfoNCE[11] 損失を最小化する．

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau)}{\sum_{j=1}^{N}\exp(\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j)/\tau)}$$

ここで，N はミニバッチ内のサンプル数を表し，$\mathbf{h}_i$ は入力文 i の埋め込みベクトル，$\mathbf{h}_i^+$ は文 i に対する正例であり SimCSE では同じ入力文に対し異なるドロップアウトマスクを適用した埋め込みを正例とする．また，$\mathbf{h}_j$ は文 i に対する負例であり，教師な

1)　本研究では教師なし SimCSE に限定する．

し SimCSE ではミニバッチ内の他の文の埋め込みを利用する．$\mathrm{sim}(\mathbf{u}, \mathbf{v})$ は，ベクトル $\mathbf{u}$ と $\mathbf{v}$ の類似度で SimCSE ではコサイン類似度が使用される．最後に，τ は温度パラメータである．SimCSE では，正例対（$\mathbf{h}_i$ と $\mathbf{h}_i^+$），つまり似た意味の文の類似度を最大化し，そうでない負例対（$\mathbf{h}_i$ と $\mathbf{h}_j$）との類似度を最小化するように最適化を行う．

2.2 安定性

モデル f の安定性 $S(f)$ は [12] を踏襲し，異なるランダムシード r でファインチューニングして得られる複数のモデル $\{f_r\}$ の評価指標のばらつきとして定義する:

$$S(f) = \sqrt{\mathrm{Var}_r[Accuracy(f_r)]}.$$

本稿では RTE タスクの validation データにおける Accuracy のばらつきで評価する．BERT のファインチューニングは，層数が多くなるほど勾配消失の影響が顕著になり学習は不安定になる [12]．したがって本稿の実験では 24 層の large モデルを分析対象とする．

2.3 可塑性

モデルの可塑性とは，異なるタスクで継続学習するときのモデルの新しいタスクへの適応性のことである．[13] では，損失関数のヘシアン $\mathbf{H} = \nabla^2_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ の有効ランクが大きいほどモデルの可塑性は大きくなることが示されている．有効ランクは $\mathbf{H}$ の固有値 $\{\lambda_i, i = 1, \ldots, d\}$ を降順に並べたときの累積寄与率が 99%を超える最小のインデックスとして定義する:

$$\mathrm{erank}(\mathbf{H}) = \min\left\{ j \;\middle|\; \frac{\sum_{i=1}^{j} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > 0.99 \right\},$$
$$(\lambda_1 > \lambda_2 > \cdots > \lambda_d).$$

直観的には，ヘシアンの有効ランクは損失関数上の最適化方向の有効な自由度と解釈することができる．つまり，継続学習の際の新しいタスクにおける有効ランクは，そのタスクへの適応性を測る指標になり得る．本研究ではこの性質に着目し，損失関数のヘシアンの有効ランクで追加学習の可塑性の特徴付けを試みる．

一般に大規模モデルのヘシアンの計算コストは膨大である．本稿の計算対象である BERT の場合，パラメータ数は $\mathcal{O}(10^8)$ であるためヘシアンを保持するだけでも 10^{16} 相当の RAM やストレージが必要となり計算困難である．そこで本研究では，ヘシアンを経験フィッシャー情報行列 $\hat{\mathbf{F}}$ で近似する．

$$\mathbf{H} \approx \hat{\mathbf{F}} = \frac{1}{N} \sum_{i=1}^{N} \left(\nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta})^\top \right).$$

ここで N はバッチサイズ，$\boldsymbol{\theta} \in \mathbb{R}^d$ はモデルパラメータ，$(\mathbf{x}_i, \mathbf{y}_i)$ はデータ点，$\nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta})$ は各データ点の対数尤度の勾配である．$\mathbf{G} = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}) \in \mathbb{R}^{d \times N}$ とおくと $\hat{\mathbf{F}} = \mathbf{G}\mathbf{G}^\top \in \mathbb{R}^{d \times d}$ と書ける．$\mathbf{G}$ はミニバッチ内の各データ点における d 次元の勾配ベクトルを並べた行列である．$\mathrm{erank}(\hat{\mathbf{F}})$ の評価は $\mathrm{rank}(\hat{\mathbf{F}}) = \mathrm{rank}(\mathbf{G}\mathbf{G}^\top) = \mathrm{rank}(\mathbf{G}^\top\mathbf{G})$ の関係を使うと $\mathbf{G}^\top\mathbf{G} \in \mathbb{R}^{N \times N}$ の固有値分解に帰着することができ計算コストを大幅に低減できる ($\because N \ll d$)．なお，本稿では $\hat{\mathbf{F}}$ は層毎にブロック対角可能 ($\hat{\mathbf{F}} \approx \mathrm{diag}(\hat{\mathbf{F}}_1, \hat{\mathbf{F}}_2, \ldots, \hat{\mathbf{F}}_L)$) であることを仮定し，各層の経験フィッシャー $\hat{\mathbf{F}}_{l \in [1,24]} = \mathbf{G}_l^\top \mathbf{G}_l$ の有効ランクを評価する．BERT のファインチューニングでは下位層に比べて上位層がより変更を受けることが知られており [14, 15, 16]，本稿の実験でも層毎の振る舞いの違いを観察するためである．

3 実験

3.1 SimCSE モデルの学習

事前学習済み BERT モデルは hugging face から入手可能な google-bert/bert-large-uncased [2] を用いる．SimCSE モデルの学習は原著論文 [5] の実装とコーパス[3] を使用し，google-bert/bert-large-uncased に対して追加学習したものを用いる．SimCSE 学習時のハイパーパラメータは [5] に従う．ただし，ドロップアウト率は $p \in \{0.05, 0.1, 0.3, 0.5\}$ としてそれぞれ学習する．[5] では STS タスクでの性能が最大となるように実験的に $p = 0.1$ が提案されている．以降では，google-bert/bert-large-uncased をたんに vanilla，SimCSE モデルは p の値を指定して SimCSE(0.1) などと表記する．

3.2 ファインチューニング

ファインチューニングタスクは RTE[17, 18, 19, 20] を使用する．RTE タスクはデータ数が比較的少なく学習が不安定であることが報告されているため [12]，学習の安定性の分析タスクとして適当であると判断した．バッチサイズを 16 として，3 エ

2) https://huggingface.co/google-bert/bert-large-uncased
3) https://github.com/princeton-nlp/SimCSE

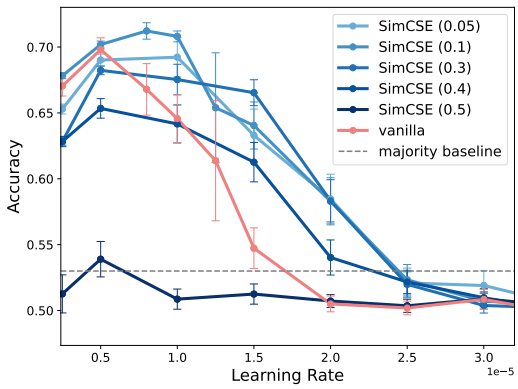


図 1: RTE タスクにおける正解率の学習率依存性．学習率を上げていくと SimCSE(0.05) や SimCSE(0.1) に比して vanilla がより早く減衰する．エラーバーは 30 回試行の 95%信頼区間．

表 1: Performance Comparison on different learning rates

Learning rate	1e−5				2e−5			
Model	$S(f)$	mean	max	#failure	$S(f)$	mean	max	#failure
vanilla	9.6	64.6	74.0	7/30	3.2	50.5	56.7	25/30
SimCSE (0.1)	2.2	70.8	74.0	0/30	9.5	58.3	72.9	14/30

ポック学習する．最大学習率は [5e−6, 3e−5] の区間から実験に応じて何点か指定する．学習率のスケジューリングは，0.3 エポック時点で最大学習率となるように 0 から線形ウォームアップ，その後 3 エポック時点で 0 になるように線形減衰させる．オプティマイザーは AdamW[21]，ハイパーパラメータは transformers.AdamW クラス[4] のデフォルト値を指定する．ただしバイアス補正は無効化する．これは，バイアス補正による学習の安定化によって SimCSE の効果が見えづらくなることを防ぐためである．

ファインチューニングの成否

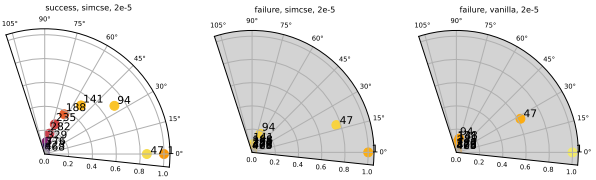
[12] を踏襲し，検証データにおける majority baseline をファインチューニング成否の閾値とする．majority baseline とは，学習データに含まれる最多ラベルを常に予測結果としたときの評価値のことで，RTE タスクの場合は正解率 0.53 以下のときファインチューニング失敗と定義する．

4 結果

SimCSE は学習率の安定領域が広い

学習率と正解率の関係を**図 1** に示す．また，**表 1** に最大学習率 1e−5, 2e−5 における，正解率の平均，

4) https://huggingface.co/docs/transformers/v4.44.2/en/main_classes/optimizer_schedules#transformers.AdamW



(a) 成功 (SimCSE)　(b) 失敗 (SimCSE)　(c) 失敗 (vanilla)

図 2: ファインチューニングの最適化軌跡．各点の横の数値はステップ数，色は訓練損失値を表す．47 ステップまでが学習率のウォームアップ期間であり，それ以降は学習終了時に学習率 0 となるように線形減衰する．灰色の背景は学習失敗を意味する．

標準偏差，最大値，失敗回数を示す．1e−5 では，正解率の最大値は同程度であるが標準偏差と平均値は SimCSE (0.1) が有利である．2e−5 では，vanilla の方が標準偏差は小さいが，これは vanilla ではほとんどの試行でファインチューニングに失敗しているためである．以上の結果より SimCSE (0.1) はより大きな学習率に対してもより安定であるといえる．

SimCSE の学習初期の最適化軌跡は等方的

図 2 にファインチューニング中の軌跡を $(r_t, \varphi_t) = \left(\frac{\|\Delta\theta_t\|}{\|\Delta\theta_{init}\|}, \cos^{-1}\left(\frac{\Delta\theta_t \cdot \Delta\theta_{init}}{\|\Delta\theta_t\|\|\Delta\theta_{init}\|}\right)\right)$ の 2 次元極座標で表示する．ここで，t はファインチューニングのステップ数，$\Delta\theta_t = \theta_t - \theta_{fin}$ である．この可視化手法は [22] で提案された方法で，学習の終点からみたときの初期値に対するパラメータ空間の軌跡を表している．**図 2b** **図 2c** より，学習失敗時は学習初期において φ 方向に大きく変位していることがわかる．学習率のスケジュールは 0 から最大学習率まで線形にウォームアップするため学習初期においては学習率は小さい．よって，ウォームアップ中の大きな偏角はパラメータ空間の特定方向に偏って変位したことを意味する．この振る舞いはパラメータ空間の曲率が大きいことを示唆する．一方，学習成功時の軌跡**図 2a** は，学習初期には等方的に θ_{fin} に向かう．これは学習初期にパラメータ空間がより平坦であることを示唆する．

SimCSE の erank($\hat{\mathbf{F}}$) はより大きい

各モデルのファインチューニング前の各層の erank($\hat{\mathbf{F}}_l$) を**図 3** に示す．ここで，有効ランクは最大ランクで正規化している．図より，SimCSE(0.05) や SimCSE(0.1)，SimCSE(0.3) はすべての層で vanilla と比べて有効ランクは大きい．また，SimCSE(0.5) は，出力側で erank($\hat{\mathbf{F}}_l$) が大きく減少し層方向の非一様性が増大している．この結果と**図 1** より，可塑性は

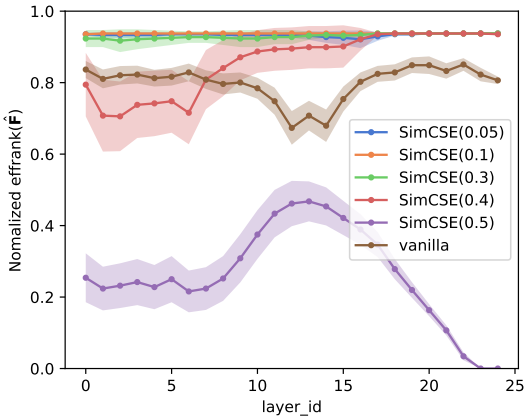


図 3: erank($\hat{\mathbf{F}}_l$) の平均（30 回試行）．帯は 95%信頼区間．最大ランクで規格化している．横軸は BERT の各層に対応しており数字が小さいほど入力側に近いことを意味する．

表 2: BERT 全体の $\hat{\mathbf{F}}$ の有効ランク．SimCSE(0.1) などは単に 0.1 と表記する．

Model	0.05	0.1	0.3	0.4	0.5	vanilla
erank($\hat{\mathbf{F}}$)	23.20	23.43	23.39	21.23	8.17	20.06

erank($\hat{\mathbf{F}}_l$) の大きさや非一様性に依存していることが示唆される．以上より，適切なドロップアウト率によって SimCSE で追加学習することは erank($\hat{\mathbf{F}}_l$) の回復をもたらし結果として可塑性が向上していることが示唆される．また，BERT 全体の erank($\hat{\mathbf{F}}$) は**表 2** となる[5]．p が小さい領域で erank($\hat{\mathbf{F}}$) は vanilla より大きくなり可塑性が大きいことが示唆される．この結果は**図 1** の結果と概ね整合していることがわかる．

ファインチューニングの成功時は erank($\hat{\mathbf{F}}$) は保存される

ファインチューニング中の erank($\hat{\mathbf{F}}_l$) の進化を**図 4** に示す．ここでは，{SimCSE(0.1), vanilla} × {1e−5, 2e−5, 3e−5} の 6 パターンを示す．図より，最も性能が良い (SimCSE(0.1), lr=1e-5) のときは，erank($\hat{\mathbf{F}}_l$) の値と層方向の一様性は学習後も保存されている．最大学習率を上げると層方向の非一様性が増大し，(vanilla, 2e−5) と 3e−5 の両モデルで学習が失敗する．学習失敗時では，ある時刻を皮切りに中層〜上位層において層方向の非一様性がさらに増大し erank($\hat{\mathbf{F}}_l$) が崩壊する．これより，ファインチューニングの失敗は erank($\hat{\mathbf{F}}_l$) のランク崩壊として現れることがわかる．上記の結果より，SimCSE による追加学習には erank($\hat{\mathbf{F}}_l$) の層方向の非一様性の増大を抑制し，結果的に崩壊を遅らせる効果があることが示唆される．

5) $\hat{\mathbf{F}}$ がブロック対角可能の仮定のもとでは，rank($\hat{\mathbf{F}}$) = $\sum_l$ rank($\hat{\mathbf{F}}_l$)

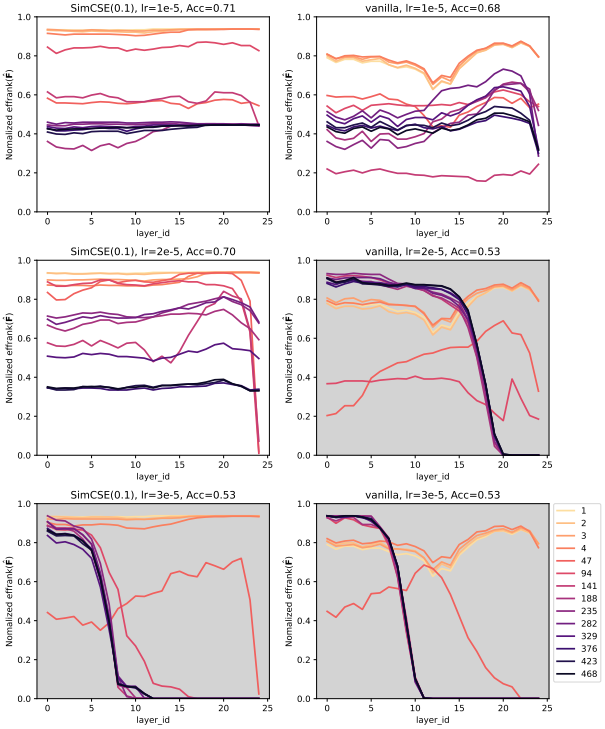


図 4: ファインチューニング中の erank($\hat{\mathbf{F}}_l$) の進化．横軸は BERT の各層に対応し，数字の小さい方が入力側を意味する．左列が SimCSE(0.1)，右列が vanilla，上から最大学習率が 1e−5, 2e−5, 3e−5 のときの結果である．また線の色はステップ数，灰色の背景は学習失敗を表す．

5 おわりに

本研究では，BERT に対する SimCSE による追加学習がファインチューニングにもたらす影響を実験的に調査した．実験では，パラメータに関する曲率情報を保持するフィッシャー情報行列の有効ランクに着目し，安定性と可塑性の観点で分析した．実験の結果，適切な設定での SimCSE による追加学習には (*i*) 大きい学習率でのファインチューニングの安定化効果と (*ii*) erank($\hat{\mathbf{F}}$) の回復をもたらすことで可塑性を向上させる効果があることがわかった．今後の課題としては，(*i*) フィッシャー情報行列のブロック対角近似の妥当性の議論，(*ii*) 他のモデルやタスク，追加学習手法による検証，(*iii*) 本稿の実験では簡単のため学習率のみ変更したが，バッチサイズとはどのような関係にあるのかの調査，(*iv*) 対照損失による追加学習が有効ランクの回復をもたらすメカニズムの原理的解明，が挙げられる．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **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)**, pp. 4171–4186, 2019.

[2] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 9119–9130, 2020.

[3] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening Sentence Representations for Better Semantics and Faster Retrieval. **arXiv preprint arXiv:2103.15316**, 2021.

[4] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 5065–5075, 2021.

[5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 6894–6910, 2021.

[6] Jiahao Xu, Wei Shao, Lihui Chen, and Lemao Liu. SimCSE++: Improving contrastive learning for sentence embeddings from two perspectives. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 12028–12040, 2023.

[7] Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 4892–4903, 2022.

[8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3982–3992, 2019.

[9] Mingxin Li, Richong Zhang, and Zhijie Nie. Towards better understanding of contrastive sentence representation learning: A unified paradigm for gradient. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 14506–14521, 2024.

[10] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. WhiteningBERT: An easy unsupervised sentence embedding approach. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 238–244, 2021.

[11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. **arXiv e-prints**, p. arXiv:1807.03748, 2018.

[12] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In **International Conference on Learning Representations**, 2021.

[13] Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of Curvature as an Explanation for Loss of Plasticity. **arXiv e-prints**, p. arXiv:2312.00246, 2023.

[14] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to BERT embeddings during fine-tuning? In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupa la, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, **Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP**, pp. 33–44, 2020.

[15] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Investigating learning dynamics of BERT fine-tuning. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, **Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing**, pp. 87–92, 2020.

[16] Yichu Zhou and Vivek Srikumar. A closer look at how fine-tuning changes BERT. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1046–1061, 2022.

[17] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, **Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment**, pp. 177–190, 2006.

[18] Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. 2006.

[19] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini, editors, **Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing**, pp. 1–9, 2007.

[20] Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In **Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009**, 2009.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations**, 2019.

[22] Namuk Park and Songkuk Kim. How do vision transformers work? In **International Conference on Learning Representations**, 2022.