トークン・次元・層の 3 つの観点とクロスファインチューンによる BERT モデル冗長性の解明

福畠 汐音 [1] 狩野 芳伸 [1]
[1] 静岡大学
{sfukuhata, kano}@kanolab.net

概要

特定のタスクのために BERT モデルをファインチューンする場合、最終層の出力の一部を選択し、新しく作成された全結合層に入力することが一般的に行われる。しかし、最終層のどの部分を選択すべきか、また、層の各次元がどのような情報を保持しているかは、よくわかっていない。本研究では、GLUE タスクに対する BERT のファインチューンを通じて、トークンに対応する最終層のベクトル、Transformer の層、ベクトルの次元について、有効性と冗長性を総合的に調査した。その結果、最終層はどのベクトルでも同等の情報を含むこと、ほとんどのタスクは 2-3 次元しか必要としないこと、上位層ではどの Transformer 層でもほとんど差がないことが示された。さらに、異なるタスクで順次ファインチューンを行うクロスファインチューンを実施した。その結果、ファインチューンにより隠れ層が大きく変化すること、複数タスクを同時に学習保持できる冗長性があることが示唆された。

1 はじめに

BERT[1] モデルを特定タスク向けにファインチューンする際は、最終層の一部の出力を選択し、特定タスク向けに新たに作成した全結合層に入力して学習することが一般に行われる [2] [3]。事前学習時および推論時に、入力の先頭に特別なトークン [CLS] を追加し、このトークンに対応する最終層のベクトル（以下、CLS ベクトル）が文全体を表現するとして頻用されている [3] [4] [5]。しかし、最終層のどの入力トークンにあたる部分を選択するのがよいのかは解明されておらず、そもそも最終層の各次元がどのような情報を持っているかも明確ではない [6]。さらに、最終層でなく中間層の出力を利用すると性能が向上することもよくある [7] が、各層の各次元がどのような情報を持つかも明確ではない [8]。

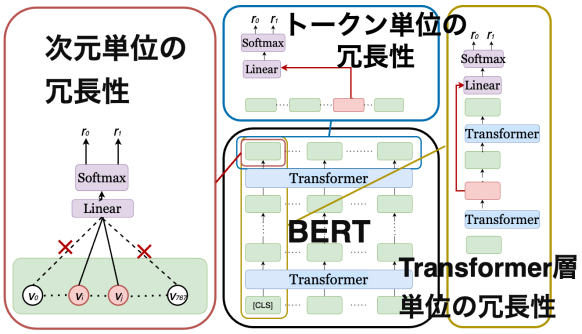

図 1: 提案手法の概要
(**トークン単位の冗長性**: Max-Pooling で選択されたトークンを使用して推論, **次元単位の冗長性**: 少数の次元を選択して推論, **Transformer 層単位の冗長性**: 隠れ層から選択された少数の次元を使用して推論)

事前学習により、モデルは一般的な言語表現を捉えることが期待される [9]。しかし、事前学習済みモデルを汎用的な「言語特徴抽出器」として利用する場合、ファインチューンの際に新しく追加した全結合層が学習の大部分を担う [10] [11] のか、それともある種の破滅的忘却 [12] によってこれらの層が大きく変化するのかは、明らかにされていない [13]。いくつかの研究では、BERT には過剰なパラメータが含まれており、性能に影響を与えることなく刈り込むことができると報告されている [14]。宝くじ仮説 [15] は、高密度でランダムに初期化されたフィードフォワードネットワークの中に、「当たり券」(元のネットワークに匹敵するテスト精度を達成できる疎なサブネットワーク) が存在することを示唆している。この仮説は、コンピュータビジョンの LeNet[16] のような小さなネットワークで実証されており、自然言語処理における LSTM[17] や Transformer にも適用可能である [18] [19] [20]。事前に訓練された BERT モデルでは、タスクの 40%-90% に対応する疎なサブネットワークが確認されている

[19] が、プルーニングによってモデルパラメータが減少する一方で、どのサブネットワークが効果的なのか、あるいは各次元がどのような情報を含んでいるのかは未だ明らかになっていない。

本研究では、GLUE ベンチマーク [21] を使用し、次元ごと、トークンごと、および層ごとの比較を行いながら、事前学習された BERT モデルのファインチューンの結果何が起こるかを包括的に検証する (図 1)。本研究の貢献は以下の通りである：

- CLS ベクトル以外のトークンに対応する最終層ベクトルにも、同等の情報が含まれることを明らかにした (**トークン単位の冗長性**)
- 最終層の CLS ベクトルの 2-3 次元のみを使用することで、CLS ベクトル情報の冗長性を明らかにした (**次元単位 の冗長性**)
- それぞれの隠れ層の CLS ベクトルの 2 次元のみを使用することで、隠れ層上位層の冗長性を明らかにした (**Transformer 層単位の冗長性**)
- ファインチューンにより大きく重みが更新されるにもかかわらず、隠れ層の冗長性により複数のタスクに同時に適応できることを明らかにした (**クロスファインチューン**)

2 手法

事前学習済みモデルをファインチューンすることで、GLUE ベンチマーク [21] の様々な下流分類タスクを実行することを、実験における共通の前提とする。[1)]

トークン単位の冗長性 CLS 以外のトークンベクトルを用いた推論との比較で、CLS ベクトル情報の冗長性を検証する。具体的には、各トークンに対応するベクトルのノルムの大きさ[2)]に基づいて MaxPooling で選択されたベクトルを使用して推論を行う。CLS ベクトルのみに含まれる重要な情報がある場合、MaxPooling による代替で性能が大幅に低下するはずである。

次元単位の冗長性 下流タスクで用いるベクトルの次元が冗長かどうかを調査する。DropConnect[22] により CLS ベクトルの任意の 2 次元の組み合わせのみを使用しほかの次元は無効化して推論を行う。

推論性能に重要な次元の特定 個々の次元の貢献を評価する。まず様々な組み合わせの 3 つの次元で推論し、次に 3 つの次元のうちそれぞれ一つを除去して性能低下を観察する。特定の次元の除去で性能が著しく低下すれば、その次元が重要と判断する。その検証として、重要と判断した次元を組み合わせ 2 次元で推論を行い、高い性能を示した組み合わせに共通する次元を真に重要な次元と判断する。

Tranformer 層単位の冗長性 隠れ層の冗長性を評価する。N 層目の CLS ベクトルから 2 次元を選択し、最終全結合層に直接入力する。高い性能が得られれば上位層の冗長性を示唆する。

事前学習済み層の凍結 事前学習済み層の重みを凍結し、ファインチューンのために 2 つの全結合層を追加した場合と、凍結せずに 1 つの全結合層を追加した場合の性能を比較する。

クロスファインチューン あるタスクでファインチューニングした後、別タスクでさらにファインチューニングし、性能を評価する。これを クロスファインチューンと呼ぶ。

事前学習の結果、汎用的で重要な次元が生じると考えられる。ファインチューンの結果ネットワーク全体の重みが更新されるとしたら、これら重要な次元も更新されタスク特有の形に変更される可能性がある。そうするとクロスファインチューンではもはや事前学習で得られた情報を復元できず、直接ファインチューンする場合より性能が低下するはずである。

しかし、ネットワークが冗長で「利用されていない余り」の情報量があるならば、その「余り」で新規タスクの学習結果を保持し、重要な次元は変更せずに済むかもしれない。この仮説が正しければ、2 番目のタスクで直接ファインチューニングした場合と同等の性能が得られるはずである。

3 実験

本実験では、標準的な事前学習モデルとして BERT の bert-base-uncased[23] [3)]を使用した。下流タスクには GLUE ベンチマーク（CoLA、SST-2、MRPC、STS-B、QQP、MNLI、QNLI、RTE）を採用した。公開訓練データと検証データを使用し、データを訓練・検証・テストの 8:1:1 に分割した。訓練時には、事前学習済み最終層の CLS ベクトルまたは

1) 本研究では、英語テキストで事前学習された BERT モデルを検証した。同様の冗長性は他の言語にも存在すると考えられるが、その検証は今後の研究課題である。

2) 以下、ベクトルのノルムはユークリッドノルムを指すこととする。

3) https://huggingface.co/google-bert/bert-base-uncased

MaxPooling で選択されたベクトルを、新たに追加した全結合層に入力した。テストでは、全結合層に入力する次元や層を選択した。詳細な設定は付録 A.1 に記載する。

3.1 トークン単位の冗長性の検証

ノルムが最大のトークンベクトルを MaxPooling で選択して推論し、CLS ベクトルを用いる場合と比較した。

3.2 次元単位の冗長性の検証

少数の次元のみを使った推論 最終層の CLS ベクトルからランダムに選んだ 2〜3 次元のみを用いて推論を行い、十分な性能が得られるか検証した。[4)]

有効な次元数の特定 各タスクに効果的な次元を特定するため、検証データに対する推論性能上位 10 セットの 3 次元セットそれぞれについて、各次元を 1 つずつ除去した場合の性能低下を調査した。性能が大幅に低下する次元を有効次元と判断し、その中から 2 次元を組み合わせて推論結果を評価した。

DropOut の影響 Dropout が次元の冗長性を誘導する可能性があるため、訓練中の Dropout[24] 適用有無が次元ごとの冗長性に及ぼす影響を検証した。

3.3 隠れ層を使用した推論

最終層以外の CLS ベクトルからランダムに選んだ 2 次元を使用し、下位層の出力が最終層と同等の精度を達成できるか検証した。

3.4 事前学習済み層の凍結

事前学習済み層を凍結した場合と凍結しなかった場合の性能を比較した。

3.5 クロスファインチューン

クロスファインチューンした場合の性能を、2 番目のタスクのみでファインチューンしたベースラインと比較した。

4 結果

4.1 トークン単位の冗長性の検証

表 1 に、CLS ベクトルと MaxPooling 選択ベクトルを用いた性能を示す。RTE では MaxPooling 選択ベクトルの性能が低下したが、他のタスクでは顕著な差は見られなかった。

	MNLI Acc	QQP Acc	QNLI Acc	STS-B Corr	MRPC Acc	RTE Acc	SST-2 Acc	CoLA Mcc
CLS	83.2	90.1	88.7	88.8	77.9	67.5	94.8	53.1
MaxPooling	83.4	89.8	90.0	88.1	77.9	56.3	94.8	58.2

表 1: CLS と MaxPooling の性能比較

4.2 次元単位の冗長性の検証

少数の次元のみを使った推論 表 2 に、検証データで最高精度を示した上位 5 つの 2 次元セットの性能を示す。MNLI を除き、最良の次元セットを用いた推論性能は全次元使用時と性能誤差の範囲で同等だった。[5)]

有効な次元数の特定 有効次元の組み合わせのうち最も性能が高かった推論結果を表 3 に示す。MRPC, RTE, SST-2, CoLA では、この方法による 2 次元推論は全次元使用時と同等の性能であった。この結果から、単純なタスクでは 2〜3 次元が有効であると示唆された。

DropOut の影響 DropOut 適用時には高性能な次元組み合わせが増加したが、その差はわずかだった（図 A.3）。

4.3 隠れ層を使用した推論

QQP、MRPC、SST-2、STS-B、CoLA では、12 層と 11 層とで同等の性能だった。一方、MNLI、QNLI、RTE では、11 層使用時に性能が大幅に低下した。また、一部タスクでは特定層を下回ると急激に性能が低下した（例: 層 5-4 間の STS-B、層 10-9 間の SST-2）。

4.4 事前学習済み層の凍結

事前学習済み層を凍結し 2 層の全結合層を追加した場合、凍結せず 1 層の全結合層を追加した場合より性能が低下した（表 4）。これは、事前学習済みの Transformer 層がファインチューンにおいても大きく貢献していることを示唆する。

4.5 クロスファインチューン

クロスファインチューニングは、直接ファインチューニングに匹敵する性能を達成した（表 5）。

4) 性能比較にあたり誤差範囲を見積もるため、付録 A.2 に示す複数のランダムサンプリング率をテストした。

5) 性能誤差の見積もりについて、異なるシード値を用いた全次元推論の結果を表 A.2 に記載する。

Dimension Set	MNLI Accuracy		QQP Accuracy		QNLI Accuracy		STS-B Pcc		MRPC Accuracy		RTE Accuracy		SST-2 Accuracy		CoLA Mcc	
	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test
All (baseline)	N/A	83.2	N/A	90.1	N/A	88.7	N/A	88.8	N/A	77.9	N/A	67.5	N/A	94.8	N/A	53.1
Set 1	75.1	75.1	89.1	89.0	88.5	87.9	83.2	86.1	78.4	76.5	70.4	65.3	94.8	94.2	60.3	54.3
Set 2	73.4	72.5	88.6	88.7	88.1	87.7	82.7	86.0	78.2	77.2	66.8	61.4	94.7	94.4	58.1	54.0
Set 3	73.1	73.2	88.6	88.6	88.1	87.4	82.6	85.8	77.7	78.9	66.4	62.8	94.6	94.8	57.7	55.3
Set 4	73.0	72.9	88.4	88.4	88.0	87.2	82.6	85.7	77.7	73.8	66.1	64.6	94.6	94.2	57.6	51.1
Set 5	72.8	72.2	88.1	88.0	87.9	87.3	82.6	85.8	77.7	77.7	66.1	66.1	94.6	94.6	57.6	53.9

表 2: 2 次元と全次元を使用した上位 5 セットの検証セットとテストセットでの評価スコア (Pcc：ピアソン相関係数、Mcc：マシューズ相関係数)

Task	Metric	Dimension Set	valid	test
MNLI	Acc	[632, 734]	72.4	72.2
QQP	Acc	[71, 268]	78.4	77.9
QNLI	Acc	[216, 245]	73.8	73.2
STS-B	Pcc	[94, 476]	57.3	58.8
MRPC	Acc	[260, 591]	70.8	76.0
RTE	Acc	[270, 336]	54.9	63.5
SST-2	Acc	[294, 606]	93.2	93.6
CoLA	Mcc	[223, 313]	61.1	58.8

表 3: 有効次元の組み合わせのうち最も性能の高かった組み合わせによる推論性能。**Dimension Set** の列の数字は、一意の次元 ID を表す。

	MNLI Acc	QQP Acc	QNLI Acc	STS-B Pcc	MRPC Acc	RTE Acc	SST-2 Acc	CoLA Mcc
凍結なし	83.2	90.1	88.7	88.8	77.9	67.5	94.8	53.1
凍結あり	54.5	73.5	69.1	19.7	69.6	59.6	78.4	0.0

表 4: 事前学習済み層の凍結有無による性能比較

5 考察

5.1 次元単位の冗長性

適切な次元セットを選択すれば、2 次元だけでも全次元使用時と同等の精度が得られることが確認された。特に SST-2 の感情分類や CoLA の文法判定など単純なタスクではこの傾向が顕著であった。また、特定の次元を除去すると性能が著しく低下する一方、その次元を使用すると高い性能が得られた。これらの結果は、単純なタスクでは推論に用いる次元数が冗長な可能性を示している。

5.2 Transformer 層単位の冗長性

多くの場合で最終層に出力させる Transformer の層を上位にするほどに性能が徐々に向上する傾向にあった。特定タスクでは例外もあり、SST-2 や CoLA では、最終層から特定の上位層までの利用をやめると性能が著しく低下し、特定層までの情報が重要であることが示された。

		Target							
		MNLI Acc	QQP Acc	QNLI Acc	STS-B Pcc	MRPC Acc	RTE Acc	SST-2 Acc	CoLA Mcc
Source	**MNLI**	83.2	88.9	90.0	88.2	79.9	68.6	95.1	54.6
	QNLI	83.1	88.7	89.9	87.2	78.9	65.3	94.9	54.4
	QQP	82.8	87.8	90.1	88.2	81.1	67.1	94.9	50.6
	STS-B	82.8	88.9	90.4	88.8	78.7	64.2	94.7	52.3
	MRPC	82.3	88.6	89.8	88.3	77.9	65.0	95.1	53.4
	RTE	82.3	88.8	89.9	87.9	76.7	67.5	95.0	53.4
	SST-2	83.2	88.7	90.0	88.0	79.7	68.6	94.8	54.7
	CoLA	83.1	89.0	90.0	87.8	81.6	63.2	95.1	53.1

表 5: クロスファインチューニングの結果。各セルにはモデルを Source タスクでファインチューンした後、Target タスクでさらにファインチューンを行った場合の Target タスクに対する推論精度が記載されている。対角成分は直に Target タスクに対してファインチューンを行った場合の推論精度。

5.3 冗長な次元の影響

クロスファインチューンの結果、タスク A で学習したモデルをタスク B で追加ファインチューンしても、タスク B を直接学習した場合と性能差がなかった。一つのタスクのファインチューンに必要な次元数（情報量）に対して、モデルが学習しうる潜在的な情報量が多く、タスクごとに別の次元セットを学習に用いることで、破滅的忘却なく複数タスクの学習ができていると考えられる。

6 結論

BERT がトークンベクトル、次元、層にわたって冗長性があることを示し、次元を大幅に削除しても性能が維持されることを示した。ファインチューンにより Transformer 層の重みが変化するが、この冗長性により以前に学習された情報は保持され破滅的忘却が起きないと考えられる。これらの結果は、BERT モデルのさらなる最適化の可能性を示唆している。なお、Transformer 機構を共有する GPT モデルにも同様の冗長性が存在する可能性があるが、これを調査することは今後の課題である。

謝辞

本研究はJSPS科研費（JP22H00804）、JSTさきがけ（JPMJPR2461）、JST AIP加速課題（JPMJCR22U4）、およびセコム科学技術財団特定領域研究助成の支援をうけた。

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. **arXiv preprint arXiv:1706.03762**, 2017.

[2] 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**, 2019.

[3] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. **arXiv preprint arXiv:2002.12327**, 2020.

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

[5] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? **arXiv preprint arXiv:1905.05583**, 2019.

[6] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupa la, Yonatan Belinkov, and Dieuwke Hupkes, editors, **Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[7] Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference. **arXiv preprint arXiv:2002.04815**, 2020.

[8] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. **arXiv preprint arXiv:2002.00737**, 2020.

[10] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. **arXiv preprint arXiv:1801.06146**, 2018.

[11] Alexander Irpan, Kanishka Rao, Konstantinos Bousmalis, Chris Harris, Julian Ibarz, and Sergey Levine. Off-policy evaluation via off-policy classification. Vol. 32, 2019.

[12] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In **Psychology of learning and motivation**, Vol. 24, pp. 109–165. Elsevier, 1989.

[13] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 4297–4308, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[14] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**, 2019.

[15] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. **arXiv preprint arXiv:1803.03635**, 2018.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Vol. 86, No. 11, pp. 2278–2324, 1998.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. **Neural Computation**, Vol. 9, No. 8, pp. 1735–1780, 1997.

[18] Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. **arXiv preprint arXiv:1906.02768**, Vol. 1, No. 1, pp. 1–10, 2020.

[19] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. **Advances in Neural Information Processing Systems**, Vol. 33, No. 1, pp. 15834–15846, 2020.

[20] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. **arXiv preprint arXiv:2005.00561**, 2020.

[21] Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. **arXiv preprint arXiv:1804.07461**, 2018.

[22] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In **Proceedings of the 30th International Conference on Machine Learning**, 2013.

[23] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of naacL-HLT**, Vol. 1, p. 2. Minneapolis, Minnesota, 2019.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, Vol. 15, No. 1, pp. 1929–1958, 2014.

	STS-B Corr		MRPC Acc		RTE Acc		SST-2 Acc		CoLA Mcc	
Rank	valid	test	valid	test	valid	test	valid	test	valid	test
All (baseline)	N/A	88.8	N/A	77.9	N/A	67.5	N/A	94.8	N/A	53.1
1	83.7	86.5	80.9	76.7	68.6	69.0	94.9	94.7	61.2	58.5
2	83.6	86.9	79.7	77.5	68.2	64.3	94.7	94.3	60.9	58.5
3	83.5	86.3	79.2	78.9	68.2	61.4	94.6	94.5	60.2	58.9
4	83.5	86.5	79.2	79.2	68.2	63.2	94.6	94.3	60.1	54.4
5	83.5	86.4	79.2	74.0	67.9	62.8	94.6	94.4	59.3	55.6

(a) sampling: 75,202

	STS-B Corr		MRPC Acc		RTE Acc		SST-2 Acc		CoLA Mcc	
Rank	valid	test	valid	test	valid	test	valid	test	valid	test
All (baseline)	N/A	88.8	N/A	77.9	N/A	67.5	N/A	94.8	N/A	53.1
1	83.5	86.3	79.7	77.5	67.5	62.1	94.9	94.7	60.1	54.4
2	83.3	86.3	78.9	77.9	67.1	65.7	94.7	94.3	58.9	56.8
3	82.9	85.5	78.2	77.9	66.8	61.0	94.6	94.5	58.7	52.3
4	82.9	86.3	77.7	75.5	66.4	63.9	94.6	94.3	58.1	52.6
5	82.9	85.5	77.7	77.5	66.4	58.8	94.6	94.4	58.1	53.5

(b) sampling: 7,520

	MNLI Acc		QQP Acc		QNLI Acc		STS-B Corr		MRPC Acc		RTE Acc		SST-2 Acc		CoLA Mcc	
Rank	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test
All (baseline)	N/A	83.2	N/A	90.1	N/A	88.7	N/A	88.8	N/A	77.9	N/A	67.5	N/A	94.8	N/A	53.1
1	79.3	78.81	88.9	88.7	88.9	88.3	82.6	85.9	76.7	75.7	66.4	63.9	94.4	94.3	56.3	50.8
2	77.6	77.0	88.6	88.5	88.6	87.8	82.5	84.5	76.5	77.0	66.1	60.6	94.4	94.2	56.2	52.4
3	77.1	76.8	88.3	87.9	88.1	87.7	81.9	85.2	76.2	75.0	65.3	59.6	94.4	94.3	56.2	52.4
4	76.04	75.8	88.3	87.9	87.8	86.9	80.8	83.4	76.0	78.9	64.6	62.1	94.3	94.2	55.9	49.4
5	75.5	75.3	88.2	88.4	87.6	87.0	80.1	83.4	76.0	75.0	64.6	62.1	94.2	93.5	55.4	51.9

(c) sampling: 752

	MNLI Acc		QQP Acc		QNLI Acc		STS-B Corr		MRPC Acc		RTE Acc		SST-2 Acc		CoLA Mcc	
Rank	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test
All (baseline)	N/A	83.2	N/A	90.1	N/A	88.7	N/A	88.8	N/A	77.9	N/A	67.5	N/A	94.8	N/A	53.1
1	71.7	71.2	87.8	87.5	88.1	87.7	79.7	83.6	75.23	77.23	64.6	62.1	94.4	94.3	51.2	49.5
2	70.3	70.3	87.5	87.3	87.1	86.7	79.6	80.8	75.0	76.2	61.0	60.3	94.3	94.2	50.2	46.8
3	70.2	70.0	87.4	87.2	86.7	86.2	78.5	81.7	74.3	73.3	60.7	56.0	94.2	93.7	50.0	43.7
4	69.2	68.9	87.3	86.7	86.6	86.2	78.0	80.6	73.8	73.0	60.7	62.1	93.3	93.2	45.7	46.9
5	66.8	66.8	87.3	87.1	85.6	84.9	75.6	79.1	73.3	75.0	60.7	58.1	93.0	92.9	45.6	42.0

(d) sampling: 75

表 A.1: 3 次元のみ使用した推論の検証データセットにおける上位 5 つの評価結果と、4 つの異なる組み合わせサンプリング数のテストデータセットにおける対応する推論性能の結果

Seed	MNLI Acc		QQP Acc		QNLI Acc		STS-B Corr		MRPC Acc		RTE Acc		SST-2 Acc		CoLA Mcc	
	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test
42	83.5	83.2	90.1	90.1	89.6	88.7	85.8	88.7	77.5	77.9	65.7	67.5	95.2	94.8	55.8	53.1
0	83.2	83.1	90.2	90.0	89.5	89.0	84.2	88.2	79.9	79.7	61.7	65.7	95.1	95.0	56.3	50.3
331	83.1	83.1	90.2	90.0	89.5	89.0	85.6	88.7	78.4	78.2	65.7	67.2	94.3	94.3	56.7	54.0
17	83.2	83.3	90.1	90.0	89.5	88.7	84.8	88.5	79.2	80.4	66.8	67.2	94.5	94.9	58.0	54.3
31	83.3	83.2	90.0	90.0	89.5	88.8	83.9	88.3	78.9	78.2	66.8	64.6	95.0	94.9	56.4	54.5

表 A.2: シード値を変えた場合の各タスクの推論性能

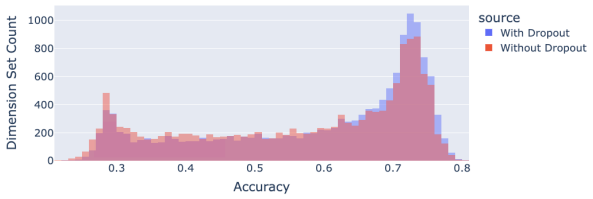

表 A.3: 2 次元を用いた MRPC データセットにおける推論性能の分布（Dropout の有無の比較）。縦軸は次元の組み合わせ数、横軸は推論性能。

A　実験設定

A.1　ファインチューニング時の実験設定

Dropout rate: 0.1, Learning rate: 5e-5, Batch size: 64, Maximum Training Epochs: 5, Random seed: 42

A.2　2 次元・3 次元のみ使用する推論のサンプリング率

実行時間を考慮し、データセットによって異なるサンプリングレートを使用した。[6)]

A.2.1　2 次元のみを使用した推論

合計で ${}_{768}C_2 = 294,528$ 個の次元の組合せがあり、以下の値の 100%に相当する。
各タスクに対して、以下のサンプリングレートを実行した：
MLNI, QQP, QNLI: 1% (2,945)　　STS-B, MRPC, RTE, SST-2, CoLA: 5% (14,726)

A.2.2　3 次元のみを使用した推論

合計で ${}_{768}C_3 = 75,202,816$ 個の次元の組合せがあり、以下の値の 100%に相当する。
各タスクに対して、以下のサンプリングレートを実行した：
MLNI, QQP, QNLI: 0.001% (752) / 0.0001% (75)
STS-B, MRPC, RTE, SST-2, CoLA: 0.1% (75,202) / 0.01% (7,520) / 0.001% (752) / 0.0001% (75)

6)　本研究では、2 次元または 3 次元のみを使用した推論が、ランダムにサンプリングされた組み合わせで実施された。いくつかの次元の組み合わせはテストされなかったが、本研究の主張は、最適な次元のセットを選択することで、すべての次元を使用するのと同等の性能を達成できるというものである。したがって、現在の実験セットアップで十分であると考えられる。