単語埋め込みに効果的なノイズの検証

田屋侑希　小林一郎
お茶の水女子大学
{g1620525,koba}@is.ocha.ac.jp

概要

近年自然言語処理の分野においても，敵対的学習の研究が盛んに行われている．敵対的学習は，ある入力データに対してモデルを騙すような摂動を，正解ラベルに基づいて計算し，求めた摂動を入力データに加えて学習する手法である．敵対的学習ではモデルの頑健性や予測性能の向上が確認されているが，摂動の解釈性が乏しく，摂動の大きさはハイパーパラメータとなっている．そこで，摂動ではなく，入力データ付近に位置するようなノイズでも効果をもたらすのではないかという着眼点から，本研究では，摂動の代わりに，類似している単語の方向に単語埋め込みをずらすことによる精度の向上を検証する．

1　はじめに

近年，自然言語処理の分野において，汎用言語モデル（BERT[1]，RoBERTa[2] 等）は様々なタスクで大きな成功を収めている．一方で，入力データの変化に敏感であることが知られている．例えばクラス分類問題において，入力文のある文字を別の文字に，または一部の単語をその類義語に変換することでモデルの予測結果が変わることが確認されている [3][4][5]．このようなモデルの脆弱性から，入力データの変化に対するモデルの頑健性を向上させるため，敵対的学習という学習手法が提案された [6][7]．敵対的学習は，画像処理分野だけではなく，自然言語処理分野においてもモデルの頑健性と予測性能の向上に有効であることが確認されている [8][9][10]．敵対的学習は，ある入力データに対してモデルを騙すような摂動を，正解ラベルに基づいて計算し，求めた摂動を入力データに加えて学習する手法である．また，摂動を加えていないデータをモデルに入力して，予測した値に基づいて摂動を計算する，仮想敵対的学習（VAT）も提案された [11]．自然言語処理分野では，摂動を計算した後，単語埋め込みに摂動を加算して新しい単語埋め込みを作成しているが，ここで加算される摂動の大きさはハイパーパラメータとなっている．摂動の大きさによって予測性能も変化し，摂動の大きさを大きくすると予測性能は低下する傾向を確認した（付録 B の図 3 参照）．一方で，摂動の大きさが小さい場合（SMART のデフォルト値），汎用言語モデルの低層（$\text{BERT}_{\text{BASE}}$ においては 12 層中 2 層目）で，すでに摂動の効果が見られないことを主成分分析（PCA）の可視化を用いて確認した（付録 A の図 2 参照）．その理由として，Transformer をベースとした言語モデルでは，トークンの埋め込み（Token Embeddings）だけではなく，文埋め込み（Segment Embeddings），位置埋め込み（Position Embeddings）が加算された後，正規化と DropOut を施して単語埋め込みを作成しているため，同じトークンでも入力データによって単語埋め込みが異なり，ばらつきがあることが考えられる．すなわち，RNN 等の従来の言語モデルに比べて，汎用言語モデルは比較的入力データの微量な違いに疎いモデルであるといえる．本研究における手法の説明をする前に用語の定義を行う．本論文では，逆伝播を用いて計算した**摂動**と区別するため，逆伝播を用いずに作成したベクトルを**ノイズ**と記す．本研究では，摂動の代わりに類似している単語の方向にノイズを加えることで，逆伝播を用いて摂動を計算せずに予測性能を維持できるモデルを提案する．また，シソーラスである WordNet [12] を用いた類義語置換と提案手法を比較して，埋め込み空間上で似ている単語の方向にベクトルをずらす効果について検証する．

2　敵対的学習

一般的に，敵対的学習では式 1 を目的関数として用いる．

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\max_{\delta} l(f(x+\delta;\theta),y)] \tag{1}$$

この式は，データセット D の入力 x に摂動 δ を加

算したベクトルをモデル f（パラメータ θ）に入力し，正解ラベル y との誤差が最大になる摂動 δ を求め，この摂動 δ を加えた入力データ $x+\delta$ を作成し，モデル f の全体の損失 $\mathbb{E}$ が最小になるパラメータ θ を求めることを意味する．この式 1 における，l を敵対的損失と呼び，PGD（Projected Gradient Descent）[7] を解くことで δ を求める．本研究では提案手法の比較対象として SMART [9] を利用した．SMART は仮想敵対的学習 [11] を用いており，目的関数は式 2 のとおりである．

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[l(f(x;\theta),y)+ \alpha \max_{\delta} l(f(x+\delta;\theta),f(x;\theta))] \tag{2}$$

式 1 と異なる点は，正解ラベル y ではなく，自身のモデルの予測結果 $f(x;\theta)$ との誤差に対して摂動を求めている点である．仮想敵対的学習における敵対的損失 l は，入力データ x とその近傍のデータ $x+\delta$ はモデルが同じ予測をするという制約を意味する正則化項とみなすことができる．SMART での摂動 δ の求め方を簡単に記す．まず初めに，正規分布に従いランダムに生成したノイズ ($\|init\_noise\|_\infty \leq \epsilon$, ϵ はハイパーパラメータ) を δ として入力データに足し合わせ，敵対的損失が最大になるような摂動を計算する．求めた摂動 ($\frac{\delta}{\|\delta\|_\infty}$) に係数（SMART では 0.001）を掛けて，初めにランダムに生成したノイズ（$init\_noise$）に足し合わせて摂動 δ とする．また，式 2 の α は一般的な損失と敵対的損失のトレードオフを制御するハイパーパラメータで，本研究では SMART と同様，1.0 に設定した．

3 提案手法

提案手法の概要図は図 1 に示す．図 1 の Step1 は通常のファインチューニングであり，Step2 で式 2 における敵対的損失を求めている．まず，提案手法に必要な**事前準備**について述べた後，図 1 における Step2 の Token Embeddings として効果的な単語埋め込みを検証するため，提案手法を**実験 1** から**実験 3** に分けて説明する．

3.1 事前準備

BERT の出現語彙（30,522 単語）の単語埋め込み（Token Embeddings）同士のコサイン類似度を総当たりで計算する．後の手順の利便性のため，単語ごとにコサイン類似度が降順になるように単語 ID を並び替えた状態で保存しておく．また，RoBERTa（出現語彙数：50,265 単語）においても同様の処理を行う．

3.2 実験 1：類似単語の方向にノイズ付加

Step1 で入力したそれぞれの Token Embeddings（$\boldsymbol{E}_{\text{original}}$）に対して，最もコサイン類似度が高い単語の Token Embeddings（$\boldsymbol{E}_{\text{similar}}$）を求めて，これらの差ベクトル（$\boldsymbol{E}_{\text{diff}}$）を計算する．

$$\boldsymbol{E}_{\text{diff}} = \boldsymbol{E}_{\text{similar}} - \boldsymbol{E}_{\text{original}}$$

Step2 の全部（BERT では [CLS]，[SEP]，[PAD]，RoBERTa では <s>，</s>，<pad> 以外）の Token Embeddings に差ベクトル（$\boldsymbol{E}_{\text{diff}}$）を足し，新しい Token Embeddings（$\boldsymbol{E}_{\text{new}}$）を作成する（**提案手法 1-1**）．

$$\boldsymbol{E}_{\text{new}} = \boldsymbol{E}_{\text{original}} + noise\_size \times \boldsymbol{E}_{\text{diff}}$$

ここで，類似度が高い単語の Token Embeddings にどの程度近づけるかを，$noise\_size$ 係数（$0 \leq noise\_size \leq 1$）を用いて決定する．$noise\_size$ が 1 の場合は，コサイン類似度が高い単語の Token Embeddings（$\boldsymbol{E}_{\text{similar}}$）を Step2 に入力していることを意味する．本手法では，$noise\_size$ を 0.5 に設定した．つまり Step2 の Token Embeddings は Step1 の Token Embeddings（$\boldsymbol{E}_{\text{original}}$）とコサイン類似度が高い Token Embeddings（$\boldsymbol{E}_{\text{similar}}$）の中間に位置するベクトルである．$noise\_size$ のアブレーションスタディの結果は付録 B に載せる．また，本研究では，Step1 の Token Embeddings に加算される $noise\_size \times \boldsymbol{E}_{\text{diff}}$ を**ノイズ**と定義する．提案手法 1-1 では，常時コサイン類似度が一番高い単語が選ばれてしまうため，コサイン類似度が高い 10 単語の中からランダムに選択して $\boldsymbol{E}_{\text{similar}}$ とする設定でも実験を行った（**提案手法 1-2**）．

3.3 実験 2：部分的な単語にノイズ付加

次に全部の単語ではなく一部（1 文の 15%）の単語にノイズを加える実験を行なった．ノイズを加える単語を部分的にすることの効果を確認する．ノイズの加え方は提案手法 1-1 と同様の手法を用いる．新しい Token Embeddings（$\boldsymbol{E}_{\text{new}}$）に置き換える単語の決め方は，下記の 3 通りを実験した．

提案手法 2-1　ランダム
提案手法 2-2　Saliency の値が高い単語
提案手法 2-3　Saliency の値が低い単語

提案手法 2-2，2-3 における Saliency [13] の値とは，Step1 の損失を用いて，Step1 の Token Embeddings に

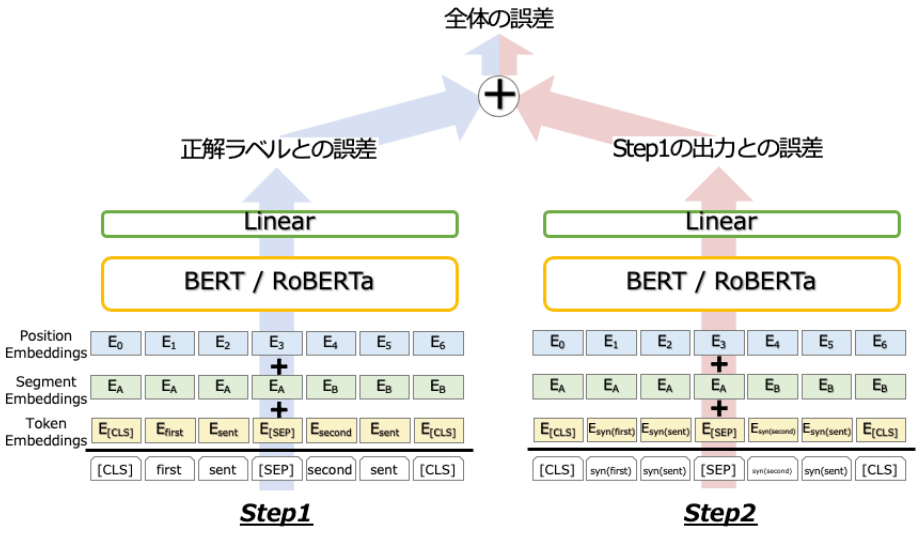

図 1　提案手法の概要図

対する勾配を計算し，勾配の絶対値を単語埋め込みの次元数分足し合わせたものである．一般的に，Saliency の値は，予測結果の判断根拠を解釈するために利用されている．つまり，Saliency の値が高いほど，予測結果に影響を与えている単語であると解釈することができる．

3.4　実験 3：WordNet を用いたノイズ付加

次に，Step2 に入力する単語として，類義語またはコサイン類似度が高い単語のどちらが効果的かを確認する．図 1 の Step1 で入力した単語の類義語を WordNet[1] で検索し，類義語が存在する場合は類義語を Step2 に入力した（**WordNet_1.0**）．類義語が複数存在する場合は，ランダムに 1 つ選んで入力した．この時，Step2 の Token Embeddings は Step1 の単語の類義語そのものを意味するため，実験 1 における *noise_size* は 1.0 と解釈できる．また，提案手法 1-1 で *noise_size* を 1.0 とした場合，つまり Step2 でコサイン類似度が最も高い単語を入力した場合の実験（**提案手法 1-1_1.0**）も行い比較する．また，WordNet を用いて，Step2 に入力する類義語を決定し，*noise_size* を 0.5 に設定した実験も行った（**WordNet_0.5**）．

4　実験

使用するデータセット　データセットは ANLI（Adversarial Natural Language Inference）[2] [14] を利用した．ANLI は汎用言語モデル（BERT，RoBERTa）が予測を誤るデータを積極的に検証データとテストデータに集めたものである．また，データセットは A1・A2・A3 から成り，A1，A2，A3 の順にモデルが予測を誤りやすいものになっている．データ数は表 1 の通りである．

表 1　ANLI：データ数

	訓練	検証	テスト
A1	16,946	1,000	1,000
A2	45,460	1,000	1,000
A3	100,459	1,200	1,200
ANLI	162,865	3,200	3,200

評価方法　評価指標は正解率（Accuracy）を用いる．また，SMART の実験方法に倣い，ANLI（A1・A2・A3）の訓練データを用いて学習し，A1・A2・A3 それぞれの検証データ，テストデータに対して評価を行った．

実験設定　実装は，MT-DNN[3] [15][16] をもとに行い，事前学習済み言語モデルは $\text{BERT}_{\text{BASE}}$(uncased)[1]，及び $\text{RoBERTa}_{\text{LARGE}}$[2] を用いた．ファインチューニングの設定として，学習率は 2×10^{-5}，バッチサイズは 32（$\text{BERT}_{\text{BASE}}$），16（$\text{RoBERTa}_{\text{LARGE}}$），エポック数は 6，最適化手法は Adam [17] を用いた．また，全学習ステップのうち始めの 10% で，学習率を 0 から設定した学習率まで線形に学習率を増加させる warm up を用いた．先行研究の SMART [9] における摂動の大きさ等はデフォルトの設定を利用した．

1）https://wordnet.princeton.edu/
2）https://dl.fbaipublicfiles.com/anli/anli_v1.0.zip
3）https://github.com/namisan/mt-dnn

4.1 実験結果・考察

実験結果は表 2，表 3 に示す．A1・A2・A3 それぞれのテストデータで評価し，その算術平均を取った値を表 2，表 3 に，各データセット（A1・A2・A3）に対する正解率は付録 C の表 4 に記す．また，実験結果は異なるシード値で 6 回実験を行い，その評価結果の平均を取った値である．表 2，表 3 の 1，2 行目は通常のファインチューニングと SMART の再実験結果である．以降，テストデータにおける正解率について言及する．

表 2 実験結果（実験 1・実験 2 における正解率）

	$\text{BERT}_{\text{BASE}}$		$\text{RoBERTa}_{\text{LARGE}}$	
	検証	テスト	検証	テスト
fine-tuning	48.9	48.8	57.5	56.2
+SMART	49.0	49.1	58.2	56.5
+提案手法 1-1	48.8	49.5	57.7	56.2
+提案手法 1-2	48.8	**49.8**	57.0	56.3
+提案手法 2-1	48.7	49.2	57.6	56.6
+提案手法 2-2	48.4	49.5	57.3	56.2
+提案手法 2-3	48.9	48.7	58.1	**56.7**

実験 1・実験 2 表 2 より，$\text{BERT}_{\text{BASE}}$ において提案手法 1-1・1-2 は，通常のファインチューニング・SMART と比較して正解率が向上した．提案手法 2-1・2-2・2-3 の結果からは部分的にノイズを加えるより文全体にノイズを加える方（提案手法 1-1）が効果的であることが確認できた．また，Saliency の値が低い単語より高い単語にノイズを加える方が効果があった．$\text{RoBERTa}_{\text{LARGE}}$ では，提案手法 1-1・1-2 は通常のファインチューニングと同等の結果となった．一方，部分的にノイズを加えた提案手法 2-1・2-3 は SMART より正解率が向上した．また，$\text{BERT}_{\text{BASE}}$ とは異なり，Saliency の値が高い単語よりも低い単語にノイズを加えた方が効果的であることが確認できる．

表 3 実験結果（実験 3 における正解率）

	$\text{BERT}_{\text{BASE}}$		$\text{RoBERTa}_{\text{LARGE}}$	
	検証	テスト	検証	テスト
fine-tuning	48.9	48.8	57.5	56.2
+SMART	49.0	**49.1**	58.2	56.5
+WordNet_1.0	47.7	48.7	57.0	56.1
+提案手法 1-1_1.0	48.0	49.0	56.7	55.2
+WordNet_0.5	49.0	48.9	58.1	**57.1**

実験 3 表 3 から，$\text{BERT}_{\text{BASE}}$ において Step2 に入力する単語を，WordNet を用いて類義語にした（WordNet_1.0）より Step1 の単語埋め込みとのコサイン類似度が高い単語に置き換えた方（提案手法 1-1_1.0）がやや効果があることが確認できた．また，提案手法 1-1 と同様，WordNet を用いた場合においても，類義語そのもの（WordNet_1.0）より類義語との中間に位置するように Step2 の Token Embeddings を作成した方（WordNet_0.5）が正解率が高いことが確認できた．$\text{RoBERTa}_{\text{LARGE}}$ においては，$\text{BERT}_{\text{BASE}}$ の結果とは異なり，コサイン類似度が高い単語よりも WordNet における類義語の方が正解率が高い結果となった．WordNet を用いて類義語に置き換える際，類義語が存在しない単語もあるため，結果的に 1 文の約 40%（$\text{BERT}_{\text{BASE}}$ は約 30%）の単語が置換されている．実験 2 の結果も踏まえて，$\text{RoBERTa}_{\text{LARGE}}$ では部分的にノイズを加えることに効果があると考えられる．*noise_size* に関しては，$\text{BERT}_{\text{BASE}}$ と同様，0.5 にしたほうが正解率は高くなることが確認できた．また，WordNet_0.5 は SMART よりも大きく精度が向上した．

ノイズの大きさ 付録 B の図 3 において，*noise_size* 0.5 の時が一番精度が良くなる理由として，全ての単語にできるだけ大きなノイズを加える場合，どの単語からもちょうど中間となる 0.5 のサイズが大きなノイズになると考えた．

また，単語埋め込みに対する摂動またはノイズの大きさ（L2 ノルム）の割合を計算したところ，SMART における摂動は約 0.003%，提案手法 1-1 のノイズは約 30%，WordNet_0.5 のノイズは約 70%であり，ノイズの大きさを大きくしてもノイズの方向性に意味がある場合，精度が低下しないことが確認できた．

5 まとめ

本研究では，敵対的学習における摂動の代わりに，類似している単語の方向にノイズを加えることで SMART と同等の精度が出ることを確認した．また，通常のファインチューニングと比較して順伝播が 2 回，逆伝播が 1 回多い敵対的学習を，順伝播が 1 回のみ多いモデルに置き換えることができ，計算時間を約 23%削減することができた．敵対的学習において，必ずしも摂動が重要ではないことを示唆する一つの手法を提案した．一方，提案手法のノイズの大きさが，モデルの予測性能向上に有効である証拠が不十分であるため，ノイズの有効性について検証していきたい．本研究では，$\text{BERT}_{\text{BASE}}$ と $\text{RoBERTa}_{\text{LARGE}}$ を用いて実験を行ったが，それぞれで異なる実験結果が確認できたため，各モデルの特徴や埋め込み空間の調査を継続したい．

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, 2019.

[2] 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.

[3] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics.

[4] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment, 2020.

[5] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6193–6202, Online, November 2020. Association for Computational Linguistics.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. **arXiv preprint arXiv:1412.6572**, 2014.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. **arXiv preprint arXiv:1706.06083**, 2017.

[8] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for language understanding. **arXiv preprint arXiv:1909.11764**, 2019.

[9] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. **arXiv preprint arXiv:1911.03437**, 2019.

[10] Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. Adversarial training for large neural language models. **arXiv preprint arXiv:2004.08994**, 2020.

[11] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. **IEEE transactions on pattern analysis and machine intelligence**, Vol. 41, No. 8, pp. 1979–1993, 2018.

[12] George A. Miller. Wordnet: A lexical database for english. **Commun. ACM**, Vol. 38, No. 11, p. 39–41, nov 1995.

[13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[14] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics.

[15] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4487–4496, Florence, Italy, July 2019. Association for Computational Linguistics.

[16] Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, and Jianfeng Gao. The microsoft toolkit of multi-task deep neural networks for natural language understanding. **arXiv preprint arXiv:2002.07972**, 2020.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. **ICLR (Poster) 2015**, 2015.

A　単語埋め込みの可視化

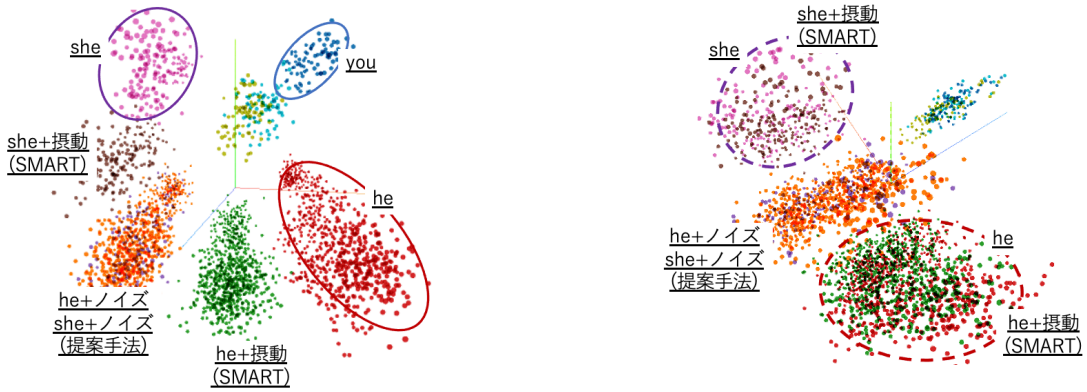

図 2　主成分分析（PCA）による単語埋め込みの可視化：she，he，you の 3 単語を $\text{BERT}_{\text{BASE}}$ に入力して作成した単語埋め込み（BERT Embeddings）（丸で囲まれている点集合），SMART を用いた摂動を加えた単語埋め込み，提案手法 1-1 を用いてノイズを加えた単語埋め込みの可視化結果（左）．$\text{BERT}_{\text{BASE}}$ の 2 層目の出力の単語埋め込みの可視化（右）．左右の図を比較して，$\text{BERT}_{\text{BASE}}$ の 2 層目の出力（右）は SMART の摂動を加えた単語埋め込みと摂動を加えていない単語埋め込みの区別がつかないことが分かる．可視化には TensorBoard の Embeddings Projector を利用．

B　摂動およびノイズの大きさのアブレーションスタディ

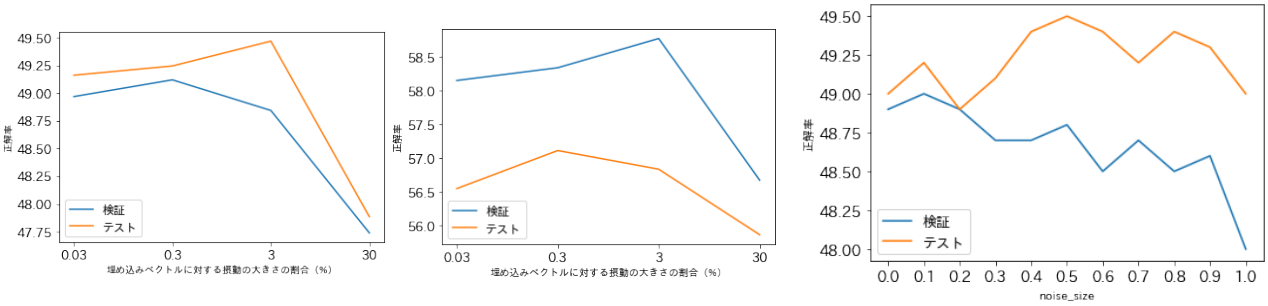

図 3　左 2 つの図：SMART において摂動の大きさを変化させたときの正解率のグラフ（左：$\text{BERT}_{\text{BASE}}$，右：$\text{RoBERTa}_{\text{LARGE}}$）．横軸は埋め込みベクトルの L2 ノルムに対する摂動の L2 ノルムの割合（%）．右の図：提案手法 1-1 における *noise_size* のアブレーションスタディ（$\text{BERT}_{\text{BASE}}$）．

C　実験結果詳細

表 4　各データセット（A1，A2，A3）における実験結果：**太字**は各データセットにおいて一番正解率が高い値，下線は SMART より正解率が高い値を表す．提案手法は，ANLI データセットの中で難易度が高い A3 のデータセットに対する正解率が向上している傾向がみられる．

	$\text{BERT}_{\text{BASE}}$								$\text{RoBERTa}_{\text{LARGE}}$							
	正解率（検証）				正解率（テスト）				正解率（検証）				正解率（テスト）			
	A1	A2	A3	All	A1	A2	A3	All	A1	A2	A3	All	A1	A2	A3	All
	55.8	45.7	45.7	48.9	55.8	46.0	45.4	48.8	72.7	51.5	49.7	57.5	71.8	49.9	48.4	56.2
SMART	55.6	46.0	45.9	49.0	56.1	46.5	45.6	49.2	73.5	52.6	50.0	58.2	72.0	50.7	48.5	56.5
提案手法 1-1	55.1	45.6	46.1	48.8	56.9	46.2	46.0	49.5	72.7	51.9	50.0	57.7	71.1	50.2	48.7	56.2
提案手法 1-2	55.2	45.4	46.2	48.8	**57.7**	**46.8**	45.7	**49.8**	71.3	51.1	50.0	57.0	71.2	50.0	49.0	56.3
提案手法 2-1	55.4	45.4	45.8	48.7	56.4	45.9	46.0	49.2	72.6	51.5	50.1	57.6	72.1	50.5	48.8	56.6
提案手法 2-2	54.8	45.2	45.8	48.4	56.7	46.0	**46.5**	49.5	72.4	51.2	49.9	57.3	71.3	49.6	**49.2**	56.2
提案手法 2-3	55.7	45.6	46.1	48.9	55.5	45.6	45.7	48.7	73.1	52.4	50.4	58.1	72.0	50.7	49.0	56.7
WordNet_1.0	54.3	44.1	45.3	47.7	56.4	45.1	45.4	48.7	72.1	51.2	49.3	57.0	70.9	**51.1**	48.0	56.1
提案手法 1-1_1.0	54.3	44.8	45.4	48.0	56.7	45.0	45.9	49.0	70.8	51.0	49.6	56.7	70.6	49.3	47.4	55.2
WordNet_0.5	56.9	45.0	45.7	49.0	55.7	45.5	46.0	48.9	73.6	52.2	50.0	58.1	**72.9**	50.9	49.1	**57.1**