評価データのクラスタリングを用いた 記述式答案自動採点のためのトランズダクティブ学習

佐藤俊 [1]　佐々木翔大 [2,1]　大内啓樹 [2,1]　鈴木潤 [1,2]　乾健太郎 [1,2]
[1] 東北大学　[2] 理化学研究所
{shun.sato, jun.suzuki, inui}@ecei.tohoku.ac.jp
{shota.sasaki.yv, hiroki.ouchi}@riken.jp

1 はじめに

記述式答案の採点は一般に人手で行われているため，大規模な試験においては，採点のための人的コストの増大が問題となっている．近年では記述式答案の自動採点に関する研究は盛んに行われており，深層学習を用いた自動採点システムが小論文や短答記述試験答案の自動採点において高い性能を示している [6, 7].

自動採点を教師あり学習の枠組みとして学習を行う場合，評価データは完全に未知なデータとして扱われる．手元に全ての答案がある状況が想定される自動採点において，そうした未知のデータを想定するのは過剰に難易度の高い問題設定となっている．したがって記述式答案の自動採点は，学習の過程で評価データの情報を明示的に利用するトランズダクティブ学習 [1] を用いることが適切であると考えられる．

トランズダクティブ学習の先行研究においては，評価データの情報を元にモデルを評価データ全体に特化させる手法が一般的である [3, 5]．しかしながら，記述式答案は同じ得点を付与される答案でも，その内容が多岐にわたることは容易に想像しうる．このことから，評価データ内の全ての答案にモデルを特化させるよりも，評価データを類似した答案ごとに分割した上で，モデルを特化させる方が予測性能の向上が期待できる．そこで本研究では，評価データのクラスタリングを行い，各クラスタに特化させたモデルで答案の点数予測を行う手法を提案し，その予測性能について検証を行う．結果として提案手法によって，複数の問題についてモデルの予測性能が平均的に上昇したが，中には予測性能が上がらない問題も存在した．本稿ではそうした提案手法の抱える問題点についての分析も合わせて報告する．

2 トランズダクティブ学習

トランズダクティブ学習 [1] は，評価データの入力テキストをモデルの学習の際に活用する機械学習の手法である．評価データの正解は既知ではなく，今回の問題設定では，学習データである答案とその点数に加えて，評価データの答案の情報をモデルの学習の際に使うことができる．トランズダクティブ学習は，ある特定の評価データに対する予測性能を向上させることが目的であり，教師あり学習のように未知のデータに対する予測性能については考慮しない．よってトランズダクティブ学習を用いる際には，評価すべきデータが事前に全て手に入っていることが前提となる．

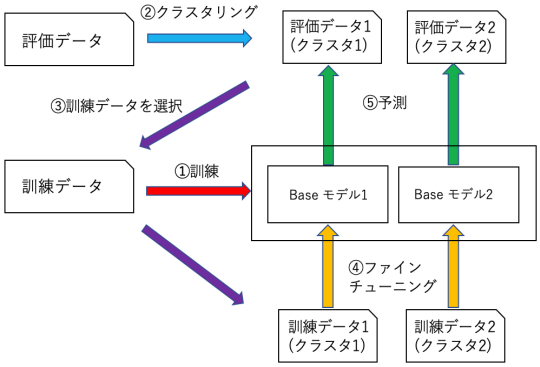

図1: 提案手法の概要図

試験答案の自動採点システムの実際の運用を想定すると，採点すべき答案の情報はすでに手元に全てあると考えられる．また一度回収された試験答案の採点作業には，全データの情報を使い採点システムを訓練するための時間的猶予が十分にある状況が想定されるため，記述式答案の自動採点はトランズダクティブ学習を適用可能な問題であると言える．

3 提案手法

図1に本論文の提案手法の概要図を示した．提案手法では先行研究に倣い，評価データと文ベクトルが類似した訓練データを，その評価データに対するモデルの予測性能に関わる重要なデータだと考えた．その上で，それらを用いてモデルをファインチューニングすることで，評価データに関する予測性能が向上することを期待している．また，評価データ全体に対してモデルをファインチューニングさせるのではなく，評価データを意味のまとまりに分割した上でファインチューニングを行うこと

で予測性能が向上することを期待している．

まず，訓練データ $D^{\text{train}} = \{(X_i^{\text{train}}, Y_i^{\text{train}})\}_{i=1}^{N}$ を用いて，自動採点モデルのパラメータ Θ の学習を行う．

$$\Theta' = \arg\min_{\Theta} L(\Theta | D^{\text{train}}) \tag{1}$$

次に，評価データのクラスタリングを行う．評価データ $D^{\text{test}} = \{X_i^{\text{test}}\}_{i=1}^{M}$ に含まれる各答案 $X = w_1, w_2, \cdots, w_n$ に対して，単語ベクトル $\boldsymbol{v}(w)$ の平均を取ることで文ベクトル $\boldsymbol{v}(X)$ を作る．

$$\boldsymbol{v}(X) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}(w_i) \tag{2}$$

この文ベクトルを使い，コサイン類似度を距離尺度とした k-medoids++ アルゴリズム[*1]を用いて評価データを T 個に分割する．

$$\{D_t^{\text{test}}\}_{t=1}^{T} = \text{medoids}\ (D^{\text{test}}, T) \tag{3}$$

各クラスタ D_t^{test} の中心点 $\mathbf{c}_t^{\text{test}}$ を計算し，この中心点とコサイン類似度が大きい訓練データを K 個選択する．

$$D_t^{\text{train}} = \{X_k^{\text{train}}\}_{k=1}^{K} = \text{top}K(\mathbf{c}_t^{\text{test}}, D^{\text{train}}) \tag{4}$$

この D_t^{train} を用いて訓練済みの自動採点モデルのファインチューニングを行う．

$$\Theta''_t = \arg\min_{\Theta'} L(\Theta' | D_t^{\text{train}}) \tag{5}$$

パラメータ Θ''_t を用いて，クラスタ t の評価データの各答案 $X_{t,j}^{\text{test}} \in D_t^{\text{test}}$ に対して点数 $s_{t,j}$ の予測を行う．

$$s_{t,j} = f_{\Theta''_t}(X_{t,j}^{\text{test}}) \tag{6}$$

$f_{\Theta''_t}$ は点数を返す採点モデルであり，4.2 節で詳述する．

4 実験

評価データをクラスタリングし，クラスタ毎にファインチューニングを行う提案手法の効果を検証するための実験を行う．

4.1 データセット

実験では記述式答案のデータセットである kaggle の ASAP-SAS[*2]を使用する．ASAP-SAS は，解答者が英語で書かれた長文を読んだ上でそれについての記述式の設問に解答した答案テキストと，それに対して採点者が付けた点数のペアのデータから成る．このデータセットは 10 問の異なる試験問題で構成され，各問題は平均で 2226 個の答案テキストを含む．本研究ではこれらを訓練データ (平均 1363 個)，開発データ (平均 341 個)，評価データ (平均 522 個) の 3 つに分割して使用する．各問題の配点は 2 点もしくは 3 点である．図2にこのデータセットの設問と答案の具体例を示す．モデルの学習には，問題中の長文の情報は一切用いず，答案とそのスコアのみを用いる．

問題 8: During the story, the reader gets background information about Mr. Leonard. Explain the effect that background information has on Paul. Support your response with details from the story.	
Answer(2点)	The new information motivates Paul to do well, and even to help Mr. Leonard learn how to read.
Answer(1点)	The effect of the background info on Paul is that he realizes he is not alone.
Answer(0点)	It made Paul curious of why Mr. Leonard didn't still run track.

図2: ASAP-SAS データセットの設問と答案の例

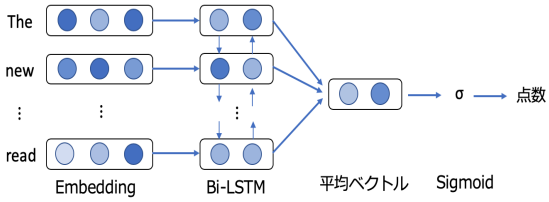

図3: ベースモデルの概要図

4.2 ベースモデル

本研究では文献 [6] で提案された自動採点モデルをベースモデルとして採用する．図3にベースモデルの概要図を示す．このモデルでは，はじめに答案中の単語を単語ベクトルで表現した後，Bi-LSTM を用いて単語長分の隠れベクトルを得る．次にそれらを単語長に関して平均をとることで文ベクトルとし，最後に 1 次元への線形変換を施すことで答案に対する点数を予測する．また学習に用いる損失には，全ての訓練データについてシステムの予測した点数と正解の点数の間の二乗誤差を用いる．

4.3 実験設定

自動採点モデルの予測性能の評価指標には，文献 [6, 7] に倣い，Quadratic Weighted Kappa (QWK) を用いる．QWK はラベルに答案の点数のような順序関係がある場合の一致率を測るための評価指標であり，-1 から 1 までの範囲の値をとる．

ベースモデルに関する設定 単語ベクトルとして事前学習された 300 次元の GloVe のベクトル [4] を用い，訓練中に単語ベクトルの値の更新も行った．Bi-LSTM 層は次元数を 256，積層数を 2 とし，層間で 0.1 の確率でドロップアウトを実行した．256 次元から 1 次元への線形変換の際に 0.5 の確率でドロップアウトを実行した．損失の最適化には Adam [2] を用い，学習率の初期値は $\rho = 0.001$ とした．

[*1] https://scikit-learn-extra.readthedocs.io/en/latest/generated/sklearn_extra.cluster.KMedoids.html

[*2] https://www.kaggle.com/c/asap-sas

表1: 実験結果

	問題	1	2	3	4	5	6	7	8	9	10	平均
	ベース	0.734	0.703	0.645	0.594	0.737	0.758	0.567	0.553	0.735	0.701	0.673
クラスタリングなし	提案手法	0.742	**0.714**	**0.643**	**0.602**	**0.753**	**0.755**	**0.568**	0.563	**0.741**	0.702	**0.678**
	ランダム	**0.747**	0.709	0.640	0.593	0.739	0.745	0.568	**0.570**	0.739	**0.708**	0.676
2 クラスタ	提案手法	**0.742**	**0.715**	**0.648**	**0.600**	**0.757**	0.755	0.570	0.561	0.740	**0.710**	**0.680**
	ランダム	0.741	0.713	0.635	0.591	0.750	**0.758**	0.570	**0.574**	**0.749**	0.709	0.679

提案手法に関する設定　ファインチューニングする際には Adam の学習率を $\rho = 0.0001$ とした．ファインチューニングに用いる訓練データの数 K は，10 個の問題それぞれについて $K = 50, 100, 200, 300, 400, 500$ の中から開発データでの QWK がもっとも高かった時の値を用いた．ファインチューニングの回数についても 1~5 回の中で，ある 1 つの K の条件についてもっとも開発データにおける予測性能が高い時の値を採用した．この K の選択に用いる開発データは各クラスタの中心から訓練データと同様の手法で 100 個選択したものを用いた．クラスタリングの分割数は 2 とし，また全ての計測について異なる 5 つの乱数のシード値を用いた実験結果の平均値を最終的な計測値とした．また，提案手法における評価データのクラスタ分割の効果を検証するため，クラスタリングを行わずにクラスタ数 2 の場合と同じ実験を行った．

提案手法による QWK の変化とファインチューニングによる QWK の変化を区別するため，訓練データをランダムに選択した場合についても同様の実験を行なった．ランダムに選択する場合では，クラスタ毎に 100 個ランダムに選択したものを開発データとし，開発データでの予測性能がもっとも高かった時の K の値を評価データに対して用いた．訓練データに関しても提案手法と同様に $K = 50, 100, 200, 300, 400, 500$ の数の訓練データをランダムに選択してモデルのファインチューニングに用いた．

4.4　実験結果

表1に実験結果を示す．表中の「ベース」はファインチューニング前の訓練済みモデルを表す．クラスタリングなし，クラスタ数 2 のいずれの場合でも，ランダムのモデルよりも提案手法のモデルの方が平均的に QWK が上昇した．このことから，記述式答案の自動採点モデルの学習においてトランズダクティブ学習が有効であることがわかった．また提案手法のクラスタリングなしとクラスタ数 2 の結果を比較すると，クラスタ数 2 の QWK がクラスタリングなしの QWK を平均的に上回った．このことから，クラスタリングによる評価データの分割によって，評価データに対するモデルのより緻密なファインチューニングが可能となり，モデルの予測性能向上につながることがわかった．しかし，提案手法の性能がランダムよりも低くなる問題や，差が微小な問題も存在し，手法自体に改善の余地があることがわかった．

表2: 2 つのクラスタ内に含まれる評価データの数とファインチューニング前後の性能差．性能差は提案手法の QWK からランダムに選択した時の QWK を引いたスコアを表している．

	クラスタ 1	クラスタ 2	比率	性能差
問題 1	547	10	55:1	0.0012
問題 2	407	19	21:1	0.0024
問題 3	318	88	4:1	0.0135
問題 4	148	147	1:1	0.0090
問題 5	445	153	3:1	0.0073
問題 6	433	166	3:1	-0.0022
問題 7	547	52	11:1	0.0000
問題 8	579	20	29:1	-0.0128
問題 9	587	12	49:1	-0.0089
問題 10	448	98	5:1	0.0004

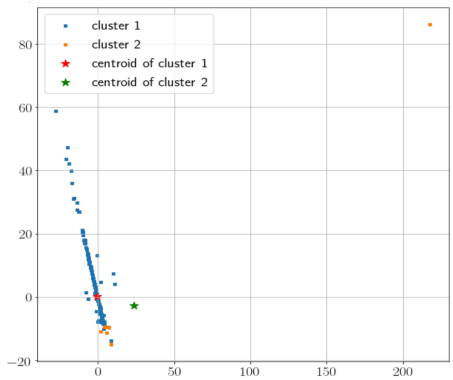

図4: 問題 9 の 2 つのクラスタの様子．各点は 1 つの答案の文ベクトルに主成分分析を実行し 2 次元に変換したものを表し，青とオレンジの点の集合がそれぞれクラスタになっている．

5　考察

本章ではクラスタ数 2 の時の提案手法の QWK がランダムに選択した時の QWK よりも低くなる問題や変化しない問題が見られた原因について考察する．

表2に 2 つに分割したクラスタに含まれる評価データの数とその問題における提案手法とランダムに選択した時のモデルの性能差を示す．比較的 QWK がランダムに選択した場合よりも上昇した問題 3，4，5 などは他の問題に比べて各クラスタ内の評価データの数の偏りが少ないことがわかる．

ここで次にランダムに選択した場合の QWK スコアよりも提案手法の QWK スコアが下がった問題の中で，クラスタ間の要素数の差が大きかった問題 9 について観

表3: 問題 6 の点数分布とクラスタに関連する点数の分布

		0 点	1 点	2 点	3 点
[1]	訓練データ	1216	130	54	37
	評価データ	498	56	30	15
[2]	クラスタ 1 の訓練データ	44	4	2	0
	クラスタ 2 の訓練データ	44	5	0	1
	クラスタ 1 の評価データ	342	51	27	13
	クラスタ 2 の評価データ	156	5	3	2

察を行ってみる．図4に問題 9 における評価データのクラスタの様子を示す．図4では，クラスタ 2 の評価データの数がクラスタ 1 の評価データの数に比べて少なくなっていることがわかる．図4のクラスタ 2 には，右側に大きく外れた地点に外れ値があり，クラスタ 2 内の要素数が少ないためにその外れ値の影響を大きく受け，訓練データを選ぶために用いる中心点の位置が実際の評価データがまとまっている地点から離れてしまっていることがわかる．上記の観察から，クラスタ間で要素数に偏りがあることによって評価データの予測に有益な訓練データが取得できず，モデルの予測性能に悪影響を及ぼす可能性があると考えられる．今後の課題として，要素数の偏りを軽減したクラスタリングの方法や外れ値に頑強なデータ選択の手法を考える必要がある．

次に問題 6 の結果について考察する．問題 6 は表2においてクラスタ間の要素数の偏りが小さいにも関わらず，QWK はベースモデルのスコアよりも下がる結果となった．訓練データと評価データにおける各点数の分布を表3[1] に示す．問題 6 は 3 点満点の問題であるが，全体に占める 0 点の割合が高いことがわかる．提案手法において，問題 6 では評価データの評価をクラスタの中心から選択された 50 個の訓練データを使ってファインチューニングされたモデルで行っており，その選択された 50 個の訓練データおよびクラスタ内の評価データの点数分布は表3[2] のようになっている．選択された訓練データにも点数の偏りが存在し，大半が 0 点となっている．クラスタ 1 の 3 点やクラスタ 2 の 2 点など，クラスタ内の評価データには含まれるにも関わらず，ファインチューニングに用いる訓練データには含まれていない点数があることがわかる．実際に，1 つのモデルのファインチューニング前後の予測を比較すると，ファインチューニング後には全体としてファインチューニング前よりも点数を低く予測する傾向があった．具体的には，ファインチューニング前に評価データの 3 点のデータへの予測は 66% 正解していたのに対し，ファインチューニング後には 33% まで低下していた．このように，提案手法では，ファインチューニングに用いる訓練データの選択に文ベクトルの類似度のみを用いているため，データ全体の点数分布に偏りがあると，うまく機能しない場合があると考えられる．

6 関連研究

トランズダクティブ学習を用いた先行研究として，ニューラルネットワークを用いた機械翻訳のタスクに適用したものがある [5]．この論文では，評価データの情報を用いて選択された訓練データを使ってモデルをファインチューニングすることで翻訳性能が向上がみられたと報告されている．文献 [3] では，述語項解析のタスクについて大規模コーパスで訓練済みの言語モデルを評価データについてファインチューニングし，そのモデルで述語項解析の訓練を行うことで予測性能が向上した．

7 おわりに

本稿では，記述式答案の自動採点を教師あり学習の問題として解くことが過度に難易度の高い問題を解いていると考え，新たにトランズダクティブ学習の問題として捉え直した．その上で，評価データをクラスタリングを用いて分割し，それぞれのクラスタに対して自動採点モデルのファインチューニングを行う新たな手法を提案し，その効果を検証した．評価データへのモデルのファインチューニングおよび評価データの分割には一定の効果があったものの，クラスタリングの手法および訓練データの選択の仕方にはまだ改善の余地があると言えるため，今後の課題としたい．

謝辞

本研究は JSPS 科研費 JP19K20351, JP19H04162 の助成を受けたものです．

参考文献

[1] A. Gammerman, V. Vovk, and V. Vapnik. “Learning by Transduction”. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. 1998.

[2] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations*. 2015.

[3] Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. “Transductive Learning of Neural Language Models for Syntactic and Semantic Analysis”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019.

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014.

[5] Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. “Transductive Data-Selection Algorithms for Fine-Tuning Neural Machine Translation”. In: *Proceedings of The 8th Workshop on Patent and Scientific Literature Translation*. 2019.

[6] Brian Riordan et al. “Investigating neural architectures for short answer scoring”. In: *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. 2017.

[7] Kaveh Taghipour and Hwee Tou Ng. “A Neural Approach to Automated Essay Scoring”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016.