特徴量を組み込んだ深層学習による自動採点モデル

電気通信大学大学院 情報理工学研究科
情報・ネットワーク工学専攻 情報数理工学プログラム

学籍番号　1831077

XIE YIKUAN

主任指導教員 植野 真臣 教授

指導教員 川野 秀一 准教授

yikuan.xie@ai.lab.uec.ac.jp

2020 年 2 月 13 日

目 次

表 目 次

図 目 次

1　はじめに

近年，論理的思考力や創造力などの能力を測定する手法のひとつとして小論文試験が注目されている．一方で，小論文試験の問題として，採点にかかる時間的・経済的コストが高いことや評価者バイアスの影響による採点の信頼性の低さなどが指摘されてきた．小論文自動採点技術 (Automated Essay Scoring, AES) は，これらの問題を解決できる手法の一つとして実用化が期待されており，現在も自然言語処理 (Natural Language Processing, NLP) や人工知能の分野で広く研究が行われている．

現在の自動採点手法は，専門家が事前に設計した特徴量 (Handcrafted features) を用いる特徴量ベースのアプローチと，機械学習モデルを用いてデータから特徴量を獲得するアプローチに大別される．

特徴量ベースのアプローチは，1968 年に Page らが開発した Project Essay Grade™(PEG™)[1, 2] をはじめとして，IEA(Intelligent Essay Assessor)[3] や IntelliMetric[4]，BETSY(Bayesian Essay Test Scoring System)[5] など，様々な手法が開発されてきた．特に ETS(Educational Testing Service) が開発した e-rater[6] は，特徴量ベースの代表的な手法として知られており，TOEFL(Test of English as a Foreign Language) や GRE(Graduate Record Examinations) などで実用されている．特徴量ベースの手法には，特徴量設計が一度完了すれば，様々な小論文データに汎用的に適用できるという利点がある．一方で，高精度を達成するためには，対象とするデータセットの性質に合わせた特徴量のチューニングや再設計が必要であることが指摘されてきた．

この問題を解決するアプローチとして，採点済みの小論文データセットに機械学習を適用することで特徴量を獲得するアプローチが提案されている．特に近年では，深層学習モデルを利用した自動採点手法が多数提案されている [7, 8, 9, 10, 11]．これらの手法は，大量の採点済み小論文データを必要とするものの，データの性質に合わせた特徴量を自動的に獲得することができ，高い精度の自動採点を実現している．

これまで特徴量ベースと深層学習ベースの手法は独立に研究されてき

たが，これらの二つのアプローチは本来は競合する手法ではなく，それぞれに異なる利点を有している．具体的には，深層学習ベースの手法は語彙の出現パタンに基づいて，対象とするデータに合わせた特徴量を獲得できるという利点がある．これに対し，特徴量ベース手法では，長年の研究で有効性が検証されてきた高度な特徴量を利用することで，単語の出現パタンだけでは捉えにくい特徴を扱えるという利点がある．そこで，これらの二つのアプローチを統合したハイブリッド手法が，Dasgupta ら [12] によって提案された．具体的には，深層学習モデルで得られる特徴量と人手で設計した特徴量を同時に利用して予測得点を計算するという手法であり，いずれか片方の特徴量のみを用いるよりも性能を大幅に改善できることが示されている．

Dasgupta らのモデルでは，文単位で設計・抽出した特徴量を深層学習モデルに入力することで文書レベルの特徴量を作成し，それを通常の深層学習自動採点モデルと同様に単語系列から学習した特徴量と統合して得点予測を行う．しかし，このモデルには次の問題がある．

1. 特徴量ベース手法の研究ではこれまでに様々な文書レベルの特徴量が提案され，その有効性が示されてきたが，Dasgupta らのモデルではそのような文書レベルの特徴量を活用できない．

2. Dasgupta らのモデルでは単語系列と特徴量系列の二つの入力を処理する深層学習モデルが内在するため，モデルの学習パラメータやチューニングパラメータが増加する．さらに，基礎モデルとする深層学習モデルを変化させた場合のモデル修正の負担も増加する．

そこで，本研究では，文書レベルの特徴量を利用できる新たなハイブリッド手法を提案する．提案手法は，従来の深層学習モデルで得られる特徴量に，文章レベルで設計した特徴量を直接結合する手法として定式化する．提案手法の特徴は次の通りである．

1. これまでに提案されてきた様々な文書レベルの特徴量を利用できる．

2. 既存の自動採点モデルの出力層に特徴量を加えるだけで実装ができるため，増加するモデルパラメータは出力層の重みパラメータのみであり，新たなチューニングパラメータも必要としない．また，現存の深層学習自動採点モデルの出力層は全て類似した構成となっているため，提案手法は様々な自動採点モデルに容易に組み込むことができる．

本研究では，ベンチマークデータセットを利用して，提案手法の有効性を評価する．

論文の構成は以下のとおりである．第 2 章では特徴量ベース自動採点機について概説する．第 3 章では深層学習ベース自動採点機の枠組みを説明し，第 4 章ではハイブリッド手法について述べ，その問題を整理する．第 5 章では提案手法について説明し，第 6 章では評価実験によって提案手法の有効性を評価する．最後に第 7 章で本論文のまとめと今後の展望について述べる．

2　特徴量ベース自動採点機

本章では，特徴量ベースの自動採点手法について概括する．

特徴量ベースの自動採点手法は，専門家が事前に設計した特徴量 (Handcrafted features) を利用し，重回帰モデルや決定木など機械学習手法によって得点を予測する．世界初の自動採点システムとして知られる PEG(Project Essay Grade)[13] では，エッセイの長さや，前置詞・関連代名詞の数などの単純な特徴量のみを利用していた．しかし，この手法は予測性能が悪いことに加え，表層的な特徴量のみを利用しているため，受験者に仕組みが暴露するとその仕組みを悪用して容易に高得点を得ることができるといった問題が指摘されてきた．これらの問題点を解決するために，近年では，潜在意味解析技術 (Latent Semantic Analysis, LSA) を用いて文書の内容的な意味を考慮できるようにした手法 [3] やエッセイの一貫性を考慮できる手法 [14, 15] や問題文との関連性を加味した手法 [16] なども提案され，高い性能を示している [5, 17]．

以降では代表的な特徴量ベースの手法である，e-rater(Electronic Essay Rater)[6] を紹介する．e-rater は，アメリカの教育試験機関 ETS が実施している共通試験である GRE や TOEFL におけるエッセイの採点に利用されている，最も有名な自動採点機である．e-rater は，Burstein ら [6] が自然言語処理技術 (NLP) と情報検索技術 (Information Retrieval,IR) を用いてそのプロトタイプを開発し，それを 2000 年から ETS Technologies が拡張し，現在は Criterion^{SM} システムに組み込まれ，実用されている．

ETS には大まかに二つのバージョンがある．旧バージョン (v.1.3) では，構造 (Structure)，組織化 (Organization)，内容 (Contents) に関する 57 個の特徴量が定義されており，データに合わせてここから 8〜12 個を選択して利用する．v.2.0 以降は，これらの特徴量の再検証が行われ，以下の特徴量を共通して利用するように修正されている．

1. Errors in Grammar：総単語数に対する文法誤りの割合

2. Usage：総単語数に対する語の使用法に関する誤りの割合

3. Mechanics：総単語数に対する手順のエラーの割合
4. Style：総単語数に対するスタイルについてのエラーの割合
5. Essay-discourse categories：談話ユニットの数
6. Average discourse element length：各ユニットにおける平均単語数
7. Number of word type to tokens in an essay：全単語数に対する異なり語彙の割合.
8. A measure of vocabulary level：Breland らの単語頻度指標に基づく語彙の難易度
9. Average word length：平均的な単語長さ
10. Score point value：該当エッセイの 6 点法にコサイン類似度が最大となるスコア点
11. Cosine correlation value：最高点 (通常 6 点) を得たエッセイとのコサイン類似度
12. Essay Length：エッセイの長さ

e-rater では，重回帰モデルを利用しており，特徴量の重みは固定されている．

なお，実際のエッセイ採点においては全ての採点業務を e-rater に委ねるわけではない．TOEFL では，個々のエッセイは人間と e-rater が独立に採点し，それらの得点差が 6 点満点中 1.5 点以内の場合，最終得点は二つの得点の平均とする．採点差が 1.5 以上あった場合には別の人間評価者が採点を行い，三つの得点のうち採点差が 1.5 以上の得点を捨て，残り二つの得点の平均点を最終得点としている．

3　深層学習ベース自動採点機

前章で紹介した特徴量ベースの手法には，特徴量設計が一度完了すれば，様々な小論文データに汎用的に適用できるという利点がある．一方で，高精度を達成するためには，対象とするデータセットの性質に合わせた特徴量のチューニングや再設計が必要であることが指摘されてきた．

この問題を解決するアプローチとして，採点済みの小論文データセットに機械学習を適用することで特徴量を獲得するアプローチが提案されている．特に近年では，深層学習モデルを利用した自動採点手法が多数提案され，自動採点手法の主流になりつつある [8, 9, 11, 12, 18, 19].

本章では，これらの近年の研究で基礎モデルとして広く利用されている LSTM(Long short-term memory)[20] モデルといくつかの最先端モデルを紹介する．

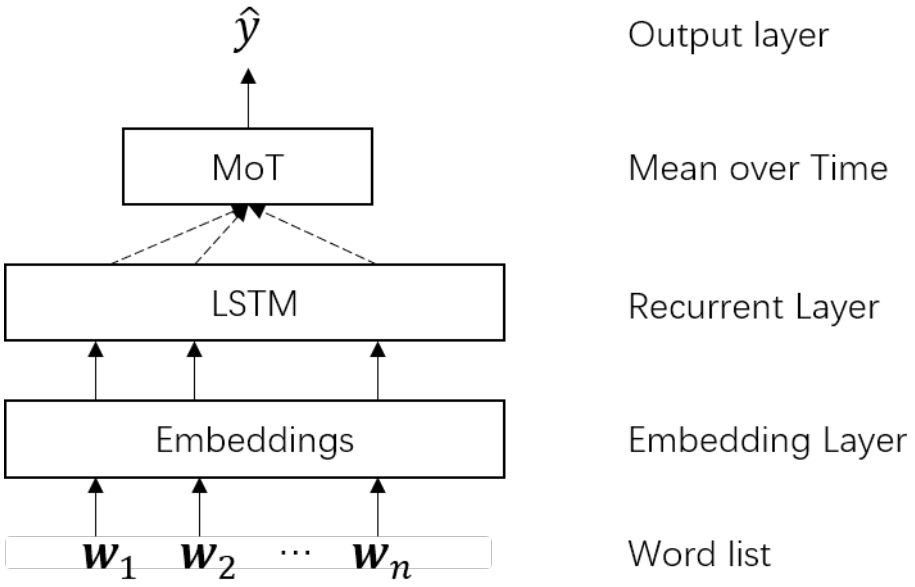
3.1　LSTM を用いた自動採点機

LSTM を用いた自動採点機は Alikaniotis ら [9] が 2016 年に提案した手法である．この自動採点機は，答案の単語系列を入力として受け取り, 多層のニューラルネットワークを通して得点の予測値を出力する. モデルの構成を図 1 に示す．以下でモデルの各層について説明を行う．

1 層目の Embedding Layer では個々の単語を潜在的な意味を表す埋め込みベクトルに変換する．具体的には one-hot ベクトルの系列として表現されたエッセイデータ $(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n) \in \mathbb{R}^{C \times N}$ を埋め込み行列 (Word Embedding matrix)$\boldsymbol{E} \in \mathbb{R}^{D \times C}$ との内積 $\boldsymbol{e}(\boldsymbol{w}) = (\boldsymbol{E} \cdot \boldsymbol{w}_1, \boldsymbol{E} \cdot \boldsymbol{w}_2, \cdots \boldsymbol{E} \cdot \boldsymbol{w}_n)$ で埋め込み表現のベクトルに変換する．

2 層目の Recurrent Layer では，LSTM を用いてエッセイの得点予測に有効な特徴量を抽出する． LSTM は時系列データを処理する深層学習モデルである RNN(Recurrent neural network)[21] の一種であり, 学習時にタイムステップが長くなる場合，指数関数的に勾配が小さくなってしまう勾配消失問題を解決する手法として提案された．LSTM では，入力，出

図 1: LSTM を用いた自動採点機



力，忘却三つのゲートを用いて，勾配を「不変」のまま流れることを可能とするだけでなく，長期依存を維持することもできる．具体的には四つのステップである．まずは Embedding Layer から出力した単語の埋め込みベクトル $\boldsymbol{e}$ の時点 t におけるセル状態から捨てる情報を判定する．式 (1) は入力 $\boldsymbol{e}_t$ と時点 $t-1$ の隠れ層 $\boldsymbol{h}_{t-1}$ を用い，セル状態 $\boldsymbol{c}_{t-1}$ の中の各数値のために 0 と 1 の間の数値を出力する．1 は「完全に維持する」と表し，0 は「完全に取り除く」を表す．次に式 (2) はセル状態で保存する新たな情報を判定する．tanh 層は式 (3) のようにセル状態に加えられる新たな候補値のベクトル $\tilde{\boldsymbol{c}}_t$ を作成し，入力ゲート $\boldsymbol{i}_t$ を用いてどの値を更新するかを判断する．さらに式 (4) のように古い状態に $\boldsymbol{f}_t$ を掛け，先ほど忘れると判定された情報を忘れて，$\boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t$ を加えて，セル状態 $\boldsymbol{c}_t$ を更新する．最後にセル状態のどの部分を出力するかを式 (5)，(6) で判定する．

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \boldsymbol{e}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{1}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{e}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{2}$$

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_c \boldsymbol{e}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \tag{3}$$

$$\boldsymbol{c}_t = \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} \tag{4}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{e}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t) \tag{6}$$

ここで，$\boldsymbol{h}_t$ は時点 t における出力ベクトルである．$\boldsymbol{W}_i, \boldsymbol{W}_f, \boldsymbol{W}_c, \boldsymbol{W}_o$ は入力 $\boldsymbol{e}_t$ に対する重みベクトル，$\boldsymbol{U}_i, \boldsymbol{U}_f, \boldsymbol{U}_c, \boldsymbol{U}_o$ は隠れ層 $\boldsymbol{h}_{t-1}$ に対する重みベクトル，$\boldsymbol{b}_i, \boldsymbol{b}_f, \boldsymbol{b}_c, \boldsymbol{b}_o$ はそれぞれのバイアスベクトルである．これらの次元数は，任意に設定する隠れ層の次元数と等しく，全て同時に学習される．ここで，$\circ$ はアダマール積を表している．σ は式 (7) に示すシグモイド関数であり，tanh は式 (8) に示すハイパボリックタンジェントである．

$$\sigma_g(x) = \frac{1}{1 + \exp(-x)} \tag{7}$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{8}$$

なお，1 層の LSTM の代わりに，多層の LSTM や双方向（Bidirectional）LSTM を利用する場合もある．

3 層目の Mean over Time Layer では LSTM Layer の出力 $\mathcal{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n)$ を用いて，次式で平均ベクトルを計算する．

$$\boldsymbol{M}(\mathcal{H}) = \left(\frac{1}{N} \sum_{t=1}^{N} \boldsymbol{h}_t \right) \tag{9}$$

なお，Mean over Time の代わりに LSTM の最後のタイムステップの出力 $\boldsymbol{h}_n$ を用いる Last Pooling を利用する場合もある．

最後に Output Layer では Mean over Time Layer の出力ベクトルから次式により予測得点を計算する．

$$\hat{y}(\boldsymbol{e}) = \sigma(\boldsymbol{W} \circ \boldsymbol{M}(\mathcal{H}) + b) \tag{10}$$

Output Layer の出力はシグモイド関数によって (0,1) の値となるが，実際のデータの得点尺度はこれと異なる場合がある．その場合には，$\hat{y}$ を一

次変換し実データの得点尺度に合わせる．例えば, 実際の得点尺度が 0〜S の S 段階得点の時, $S\hat{y}$ と変換を行う．

3.2 Hierarchical Attention Network

Hierarchical Attention Network は Yang ら [22] が文書分類のために提案したフレームワークであり，Nadeem ら [19] によって自動採点に適用された．

この手法では，文書が階層構造を持つことに着目する．具体的には，単語は文を形成し，文は文書を形成していることに着目する．このモデルでは，単語系列を処理する RNN に一文ずつ入力を行い，文ごとの特徴量を得た上で，文ごとの特徴量の系列を別の RNN に入力するという，階層的なネットワーク構造を持つ．モデルの構造を図 2 に示す．図のようにこのモデルは次の五つの層で構成されている．

1 層目の Word Embedding Layer では，前節の手法と同様に，個々の入力単語を次のように埋め込み表現に変換する．

$$\boldsymbol{e}_{it} = \boldsymbol{E}\boldsymbol{w}_{it} \tag{11}$$

ここで，$\boldsymbol{w}_{it}$ はエッセイにおける i 番目のセンテンスの t 番目の単語である．次に 2 層目の Word Level Recurrent Layer では，埋め込みベクトルの系列を LSTM に入力する．ただし，ここでは，エッセイを文ごとに区切り，各文を独立に入力する．

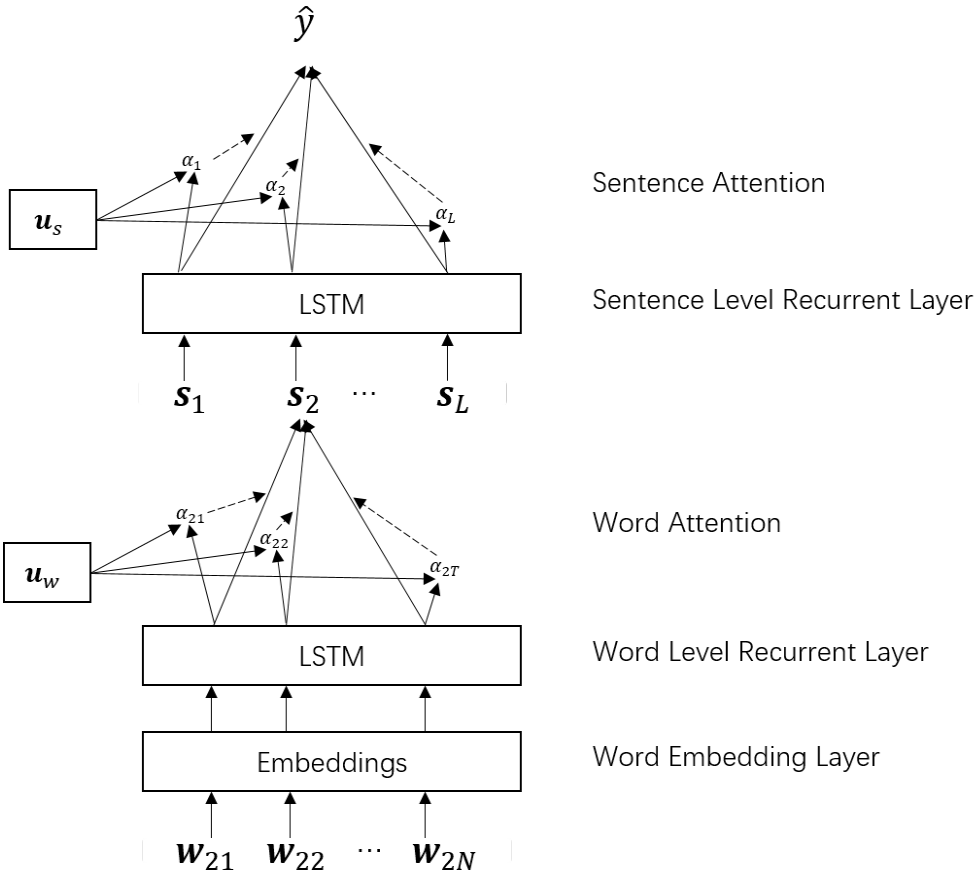
$$\boldsymbol{h}_{it} = LSTM(\boldsymbol{e}_{it}) \tag{12}$$

ここで $\boldsymbol{h}_{it}$ は単語 $\boldsymbol{w}_{it}$ が LSTM 層を通した出力である．

3 層目の Word Attention Layer は Attention モデルを用いて，個々の $\boldsymbol{h}_{it}$ の重要さを考慮して，$\boldsymbol{h}_{it}$ の重みつき平均 $\boldsymbol{s}_i$ を計算する．具体的には次式で計算が行われる．

$$\boldsymbol{u}_{it} = tanh(\boldsymbol{W}_w\boldsymbol{h}_{it} + \boldsymbol{b}_w) \tag{13}$$

図 2: Hierarchical Attention Network



$$\boldsymbol{\alpha}_{it} = \frac{exp(\boldsymbol{u}_{it}^{\top}\boldsymbol{u}_w)}{\sum_t exp(\boldsymbol{u}_{it}^{\top}\boldsymbol{u}_w)}. \tag{14}$$

$$\boldsymbol{s}_i = \sum_t \boldsymbol{\alpha}_{it}\boldsymbol{h}_{it} \tag{15}$$

ここで，α_{it} は一般にアテンション重みと呼ばれる．

4 層目の Sentence Level Recurrent Layer は，Word Attention Layer が学習した重み付き平均 $\boldsymbol{s}_i$ を文の特徴として文レベルの LSTM で処理する．

$$\boldsymbol{h}_i = LSTM(\boldsymbol{s}_i) \tag{16}$$

5 層目の Sentence Attention は 3 層目の Word Attention と同様に，$\boldsymbol{h}_i$ の系列に Attention を適用して，重み付き平均 $\boldsymbol{v}$ を計算する．具体的に

は，次式で計算される．

$$\boldsymbol{u}_i = tanh(\boldsymbol{W}_s \boldsymbol{h}_i + \boldsymbol{b}_s) \tag{17}$$

$$\boldsymbol{\alpha}_i = \frac{exp(\boldsymbol{u}_i^\top \boldsymbol{u}_s)}{\sum_t exp(\boldsymbol{u}_i^\top \boldsymbol{u}_s)} \tag{18}$$

$$\boldsymbol{v} = \sum_t \boldsymbol{\alpha}_i \boldsymbol{h}_i \tag{19}$$

最後に，前節の Output Layer と同様に，シグモイド関数を活性化関数とする一層の全結合層を通して，スコアに対応するスカラ値を求める．

3.3 BERT

BERT(Bidirectional Encoder Representation from Transformers)[23] は 2018 年に Google が提案した最先端の深層学習モデルの一つであり，質問応答 (Question Answering) や自然言語推論 (Multi Natural Language Inference) などの 11 種の自然言語処理タスクにおいて現在最高性能を達成している手法である．

BERT は，RNN と同様に文字列のような時系列データを扱うモデルであるが，RNN とは異なり Attention メカニズムを利用した構成されている．具体的には，双方向 Transformer[24] と呼ばれる機構によって，入力文字列の単語を一度に読み込み，単語の系列全体を解釈して文脈を学習している．

BERT の学習は大量のコーパスを用いる事前学習 (pre-trained) と比較的少量のデータを用いるファインチューニング (fine-tuning) の二段階で構成される．事前学習は「Masked Language Model」と「Next Sentence Prediction」の二つの学習タスクでモデルのトレーニングを行なっている．

- Masked Language Model (MaskedLM)

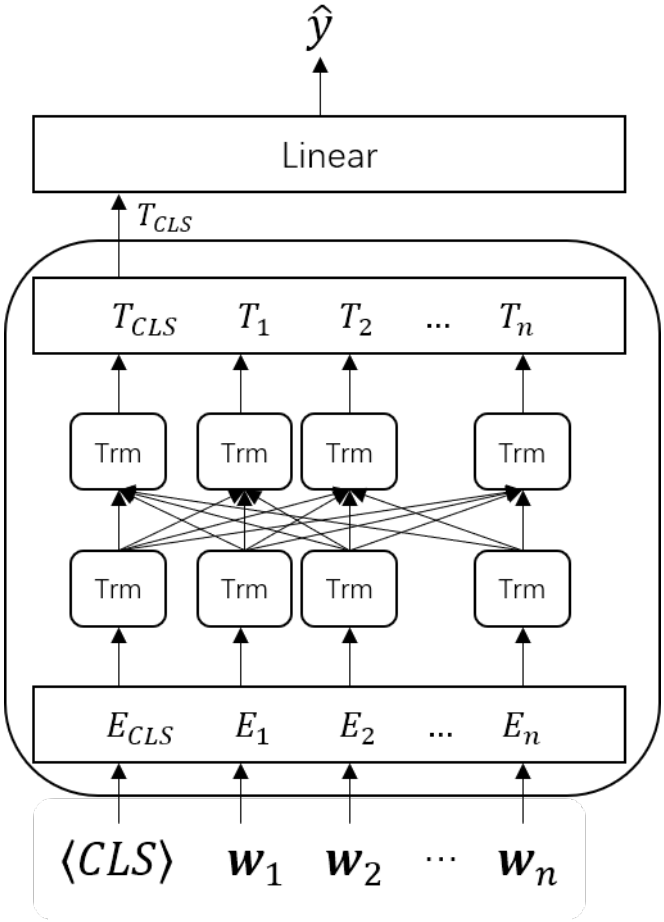
  文字列の 15% の単語を [Mask] トークンに置き換え，マスクされた単語を予測するタスク

- Next Sentence Prediction (NSP)

  入力文字列のペアを受け取り，それらが連続した文であるかを予測するタスク

ファインチューニングでは，事前学習で得られたパラメータを初期値として，対象とする言語処理タスクに関するラベル付き学習データを用いてモデルの再学習を行う．このとき，文書の最初に特殊なタグ $< CLS >$ を設置し，回帰や分類問題では，それに対応する出力 T_{cls} を特徴量として用いる．モデルの構造を図 3 に示す．

図 3: BERT Fine-tuning



4　ハイブリッド手法

特徴量ベースと深層学習ベースの手法は独立に研究されてきたが，これらの二つのアプローチは本来は競合する手法ではなく，それぞれに異なる利点を有している．具体的には，深層学習ベースの手法は語彙の出現パタンに基づいて，対象とするデータに合わせた特徴量を獲得できるという利点がある．これに対し，特徴量ベース手法では，長年の研究で有効性が検証されてきた高度な特徴量を利用することで，単語の出現パタンだけでは捉えにくい特徴を扱えるという利点がある．そこで，これらの二つのアプローチを統合したハイブリッド手法が，Dasgupta ら [12] によって提案された．具体的には，深層学習モデルで得られる特徴量と人手で設計した特徴量を同時に利用して予測得点を計算するという手法であり，いずれか片方の特徴量のみを用いるよりも性能を大幅に改善できることが示されている．

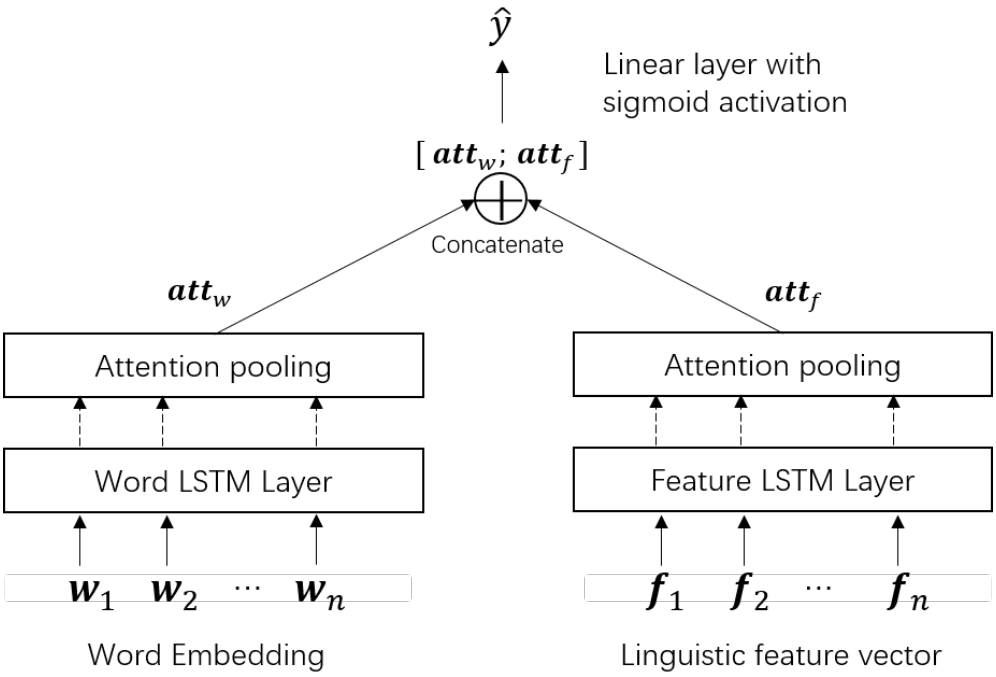
モデルの構成を図 4 に示す．このモデルは一般的な LSTM を用いた深層学習自動採点モデルに，文レベルの特徴量を処理する LSTM を結合したモデルとなっている．具体的には，文単位で設計・抽出した人手での特徴量 $\boldsymbol{F}^s = (\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n)$ を，3.2 で紹介したアテンション付き LSTM モデルに入力することで文書レベルの特徴量 $\boldsymbol{att}_f$ を作成する．同時に，通常の深層学習自動採点モデルと同様に，単語系列を入力としてアテンション付き LSTM モデルで特徴量 $\boldsymbol{att}_w$ を作成する．最後に，この二つの特徴量を結合 (Concantenation) $[\boldsymbol{att}_w; \boldsymbol{att}_f]$ し，これを全結合層に通して得点予測を行う．しかし，このモデルには次の問題がある．

1. これまでの特徴量ベース手法の研究では，様々な文書レベルの効果的な特徴量が提案されてきたが，このモデルではそのような文書レベルの特徴量を活用できない．

2. 単語系列と特徴量系列の二つを処理するネットワーク機構が必要であるため，モデルの学習パラメータやチューニングパラメータが増加するとともに，基礎モデルとする深層学習モデルを変化させた場

合のモデル修正の手間も増加する.

図 4: Qualitatively Enhanced Deep Neural Network



5　提案手法

前節で述べた問題を解決するために，本研究では，文書レベルの特徴量を扱うことができる新たなハイブリッド手法を提案する．提案手法は，従来の深層学習モデルで得られる特徴量に，文章レベルで設計した特徴量を直接結合する手法として定式化する．

5.1　モデル構成

提案手法では，任意の深層学習モデルを利用できる．利用する深層学習モデルにおいて，Output layer への入力となる特徴量を $\boldsymbol{M}$ とし，文全体の特徴量 (overall features) を $\boldsymbol{F}^o$ とすると，提案手法では次式のように $\boldsymbol{M}$ と $\boldsymbol{F}^o$ を結合する．

$$\boldsymbol{C}(\boldsymbol{M}, \boldsymbol{F}^o) = [\boldsymbol{M}; \boldsymbol{F}^o] \tag{20}$$

そして，この特徴量を，前節で紹介してきた深層学習自動採点モデルと同様に，シグモイド関数を活性化関数とする一層の全結合層を通して，スコアに対応するスカラ値を求める．

$$\hat{y} = \sigma((\boldsymbol{W} \circ \boldsymbol{C}) + b) \tag{21}$$

提案手法で増加するパラメータは $\boldsymbol{F}^o$ と同数の出力層の重み $\boldsymbol{W}$ パラメータのみであり，チューニングパラメータはない．

例として，図 5〜8 に 3 章と 4 章で紹介した LSTM，Hierarchical Attention Network，BERT，ハイブリッド・モデルに，提案手法のアプローチで特徴量を組み込んだモデル図を示す．

提案手法の特徴は次の通りである．

1. これまでに提案されてきた様々な文書レベルの特徴量を利用できる．
2. 既存の自動採点モデルの出力層に特徴量を加えるだけで実装ができるため，この拡張で増加するモデルパラメータは出力層の重みパラ

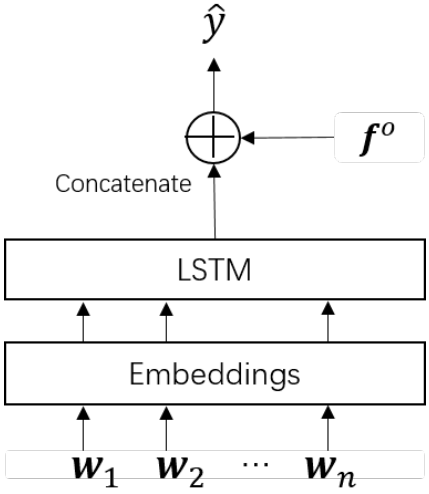


図 5: LSTM を用いた提案モデル

メータのみであり，新たなチューニングパラメータも必要としない．また，出力層は現存の自動採点モデルは全て類似した構成となっているため，様々な自動採点モデルに容易に組み込むことができる．

5.2 特徴量

本研究で利用する特徴量は次の 25 個である．

1) 長さに関する特徴量 (Length Features) : 単語数，文数，単語の平均長さ，文の平均長さ

2) 頻度に関する特徴量 (Occurrence Features) : レンマ数，句読点数，ストップワード数とスペルミスの数

3) POS に基づく特徴量 (Part-of-speech Features) : 名詞 (Nouns)，動詞 (Verbs)，形容詞 (Adjectives)，副詞 (Adverbs)，接続詞 (Conjunctions) の数

4) 可読性指標 (Readability Features) : 音節，単語，文などの数を用い

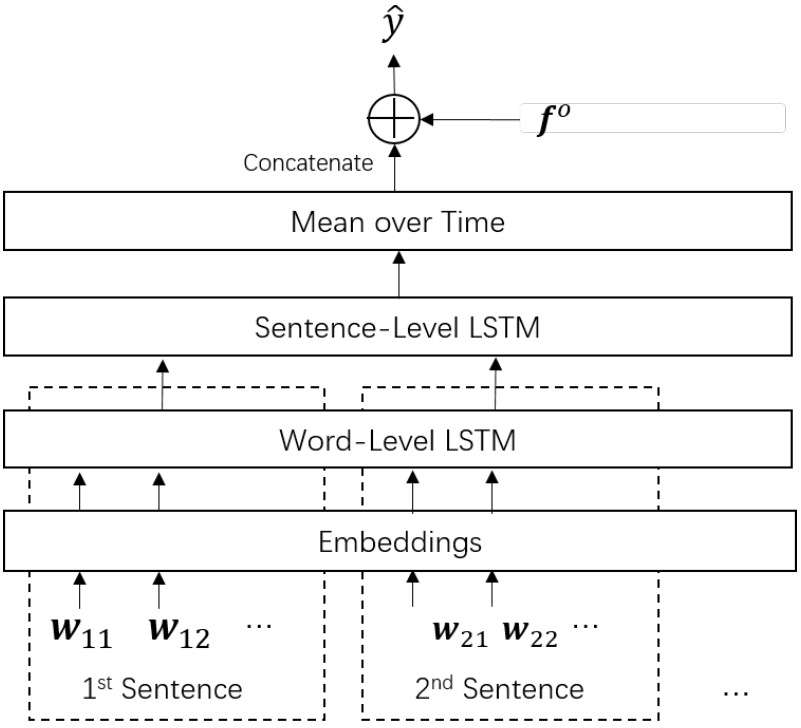


図 6: Hierarchical Network を用いた提案モデル

て，エッセイの可読性を評価するために設計された特徴量であり，様々な指標が提案されている．本研究では，以下の指標を利用する．

1. Flesch-Kincaid Readability[25]：エッセイをどれほど理解し難しいかを示すために設計された可読性指標であり，Flesch Reading Ease と Flesch Kincaid Grade に大別される．

   Flesch Reading Ease：文章の読みやすさを表し，次式で定義される．

$$206.835 - 1.015(\frac{\text{単語数}}{\text{文数}}) - 84.6(\frac{\text{音節数}}{\text{単語数}}) \tag{22}$$

   Flesch Kincaid Grade：文章の困難度を表し，次式で定義される．

$$0.39(\frac{\text{単語数}}{\text{文数}}) + 11.8(\frac{\text{音節数}}{\text{単語数}}) - 15.59 \tag{23}$$

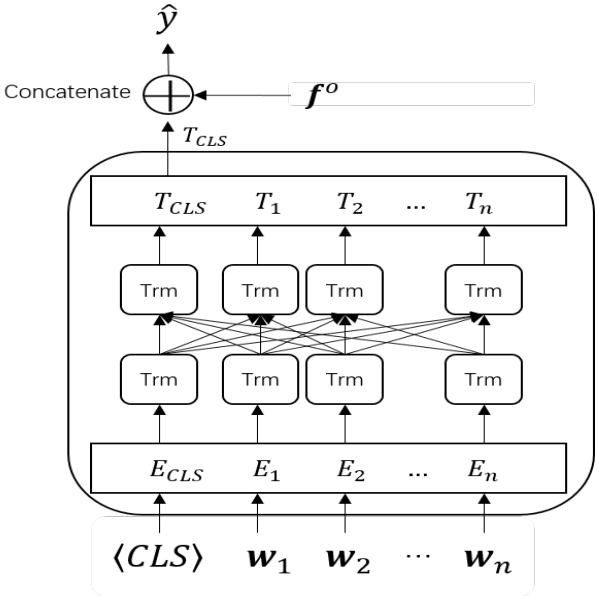


図 7: BERT を用いた提案モデル

2. Syllable count：エッセイにおける音節数

3. Smog(Simple Measure of Gobbledygook) Index : 文章を理解するために必要な教育レベルを表す尺度 [26]．例えば，あるエッセイの Smog Index は 6.3 の場合，6 年生が理解できるレベルであることを意味する．この指標は次式で定義される．

$$1.0430\sqrt{\text{多音節数} \times \frac{30}{\text{文数}}} + 3.1291 \tag{24}$$

4. Coleman-Liau Index[27]：

$$0.0588L - 0.296S - 15.8 \tag{25}$$

ここで，L は 100 単語あたりの平均文字数であり，S は 100 単語あたりの平均文数．

5. Automated Readability Index (ARI)[28] : 他の可読性テストと異なり，単語ごとの音節数ではなく，単語ごとの文字数に依存する指標

$$4.71(\frac{\text{文字数}}{\text{単語数}}) + 0.5(\frac{\text{単語数}}{\text{文数}}) \tag{26}$$

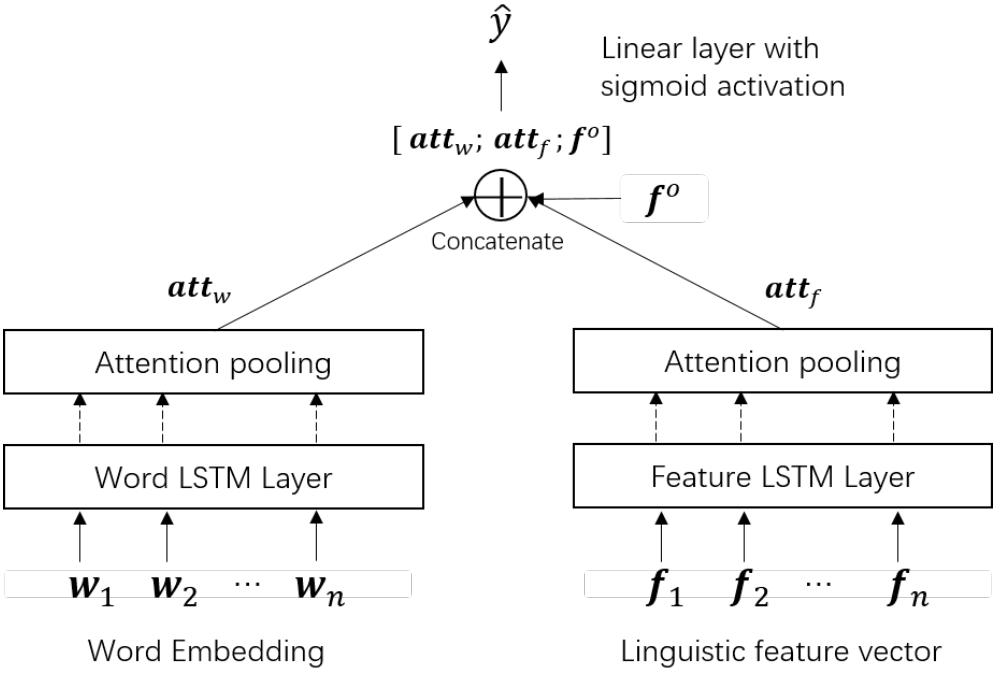


図 8: Qualitatively Enhanced DNN を用いた提案モデル

6. Difficult Words：難語数

7. Dale-Chall Readability Score[29]：アメリカ 4 年生が理解できる 3000 単語のリストを使用し，リスト以外の単語の割合に基づく指標.

$$0.1579(\frac{\text{読みにくい単語数}}{\text{単語数}} \times 100) + 0.0496(\frac{\text{単語数}}{\text{文数}}) \quad (27)$$

8. Gunning Fog Index[30]：

$$0.4\left[(\frac{\text{単語数}}{\text{文数}}) + 100 \times (\frac{\text{複雑な単語の数}}{\text{単語数}})\right] \quad (28)$$

一般に，幅広い読者向けの文章は得点を 12 点以下となる.

9. Linsear Write Formula：

$$\begin{cases} \frac{P}{S}/2 & if \quad \frac{P}{S} > 20 \\ (\frac{P}{S}/2) - 1 & otherwise \end{cases} \quad (29)$$

ここで，S は文数，P は $P^A + P^B$ とし，P^A は 2 音節以下で構成される単語数，P^B は 3 音節以上の単語数 $\times 3$ である．

6　評価実験

本章では, ベンチマークデータセットを用いて, 提案手法の有効性を評価する. ニューラネットワークの実装は Tensorflow[1] と Keras[2] を用いる.

6.1　データセット

本研究では，自動採点研究でベンチマークデータとして広く利用されている ASAP(Automated Student Assessment Prize) データセットを使用した [31]．ASAP データセットは八つの課題に対するエッセイで構成されており，各課題におけるエッセイは元々Grade 7 から Grade 10 の学生によって書かれたものである．データの基礎統計量を表 1 に示す.

表 1: Automated Student Assessment Prize Dataset の基礎統計量

課題	# of essays	Score essays	Max Word count	Max Sentence count	Max Word count in Sentence
1	1783	2–12	600	40	35
2	1800	1–6	700	40	40
3	1726	0–3	256	15	40
4	1772	0–3	200	10	50
5	1805	0–4	256	15	40
6	1800	0–4	256	15	40
7	1569	0–30	350	27	35
8	723	0–60	870	65	37

[1]https://www.tensorflow.org/
[2]https://keras.io/

6.2 実験手順

実験は 5 分割交差検証法により行った．評価指標には，自動採点の研究で広く利用される重み付きカッパ係数 (Quadratic Weighted Kappa，QWK)[32] を用いた．QWK は次式で定義できる．

$$k = 1 - \frac{\sum_{i,j} w_{i,j} x_{i,j}}{\sum_{i,j} w_{i,j} m_{i,j}} \tag{30}$$

$$w_{i,j} = (i-j)^2 \tag{31}$$

ここで，$x_{i,j}$ は真値は i，推定値は j の場合の割合．$w_{i,j}$ は x_{ij} の重み．$m_{i,j}$ は真値は i の確率と推定値は j の確率の積である．

モデルは，基礎モデルとして 3 章と 4 章で紹介した図 1,2,3,4 のモデルと，それらに文章特徴量を加えた提案手法 (図 5,6,7,8,9) を利用し，それらの性能を比較する．LSTM を用いたモデルでは，1 層の LSTM に加えて，2 層の LSTM と Bidirectional LSTM も比較した．また，1 層の LSTM では，Mean over Time polling と Last Pooling でも実験を行った．また，文章特徴量にロジスティック回帰を適用した特徴量ベース手法との比較も行った．ここで，Dasgupta らの手法で利用する文レベルの特徴量としては，5.2 で定義した特徴量を用いる．ただし，文単位では定義できない，文数と文の平均長さ，Readability Features の一種である Smog Index は除外した．また，文の相対位置を表す特徴量として，以下の特徴量を追加した．

$$max(\frac{1}{i}, \frac{1}{N-i+1}) \tag{32}$$

また，埋め込み行列を利用するモデルでは，事前訓練された 50 次元の Glove[33] を用いる．LSTM モデルについては隠れ層を 300 層に設定する．そして，過剰適合を避けるためにドロップアウトを採用する．外部からの入力は 50%ドロップアウト，リカレント内部は 10%ドロップアウトする．層と層の間も 50%のドロップアウトを行う．BERT については事前学習モデルを利用し，768 次元の出力ベクトルと特徴量をオンカットする．損失関数 (loss function) は平均二乗誤差 (MSE) とし，Adam アルゴリズム

(学習率を 0.001，減衰は $1e^{-6}$)[34] を利用して誤差逆伝搬法で最適化する．このとき，ミニバッチサイズ (mini-batch size) を 32，モデルを 50 エポック (epoch) でトレーニングする．また，計算量削減のために，エッセイの最大単語数，最大文数，文における最大単語数を表 1 のように制限した．

6.3 実験結果

表 2: 比較実験の結果

	課題									
	1	2	3	4	5	6	7	8	Avg.	p
LSTM （Last pooling）	0.373	0.407	0.516	0.773	0.753	0.767	0.635	0.174	0.550	0.018
＋ Overall features	0.801	0.621	0.602	0.778	0.771	0.777	0.761	0.645	0.720	
2-layer LSTM（Last pooling）	0.435	0.414	0.530	0.791	0.698	0.768	0.639	0.163	0.555	0.017
＋ Overall features	0.778	0.620	0.592	0.779	0.779	0.769	0.762	0.643	0.715	
Bidirectional LSTM	0.484	0.419	0.500	0.777	0.738	0.721	0.625	0.218	0.560	0.014
＋ Overall features	0.779	0.597	0.582	0.778	0.762	0.765	0.756	0.661	0.710	
LSTM （MoT）	0.717	0.522	0.616	0.775	0.796	0.783	0.749	0.562	0.690	0.015
＋ Overall features	0.821	0.649	0.617	0.790	0.787	0.807	0.794	0.694	0.745	
Hierarchical Attention Network	0.731	0.611	0.628	0.786	0.790	0.781	0.765	0.585	0.710	0.031
＋ Overall features	0.808	0.645	0.620	0.792	0.800	0.784	0.780	0.673	0.738	
BERT	0.829	0.391	0.762	0.886	0.876	0.584	0.818	0.540	0.711	0.021
＋ Overall features	**0.852**	0.651	**0.804**	**0.888**	**0.885**	**0.817**	**0.864**	0.645	**0.801**	
従来のハイブリッド手法	0.729	0.635	0.631	0.787	0.802	0.793	0.773	0.693	0.730	0.073
＋ Overall features	0.823	**0.674**	0.601	0.795	0.790	0.811	0.806	**0.714**	0.752	
特徴量ベース手法	0.822	0.648	0.666	0.704	0.783	0.672	0.724	0.600	0.702	-

実験結果を表 2 に示す．既存モデル間で比較すると，平均エッセイ長の短い課題 4，5，6 が，エッセイ長が長い課題 1，2，8 に比べて精度が高いことがわかる．また，LSTM ベースの手法では，Bidirectional LSTM や Mean over Time を利用した方が，高い性能を示している．これは，長期の単語依存性を考慮しやすいためと解釈でき，先行研究の結果と同様の傾向となっている．また，BERT を利用した場合が，最も精度が高くなっており，これも近年の研究結果と一致した傾向となっている．

次に人手での特徴量を加えた場合との比較を行うと，全ての基礎モデルにおいて，特徴量を加えたときに平均的な精度が大幅に改善している．ここで，特徴量の有無で性能に有意性あるかを検証するために，対応のあるt検定を適用した．結果を表2の「p値」列に示す．この結果から，ほとんどの場合において，特徴量を加えたときに5%有意に精度が向上していることが確認できる．最先端のハイブリッド手法についても，特徴量を加えたときに10%有意に精度が向上している．また，人手での特徴量のみを利用した場合と比較すると，平均の精度では全ての場合に提案手法が高い性能を示した．全ての中で最も精度が高かった手法は，BERTに特徴量を加えた場合であり，平均予測精度は0.801であることがわかる．

ここで，人手での特徴量の効果を確認するために，表3にBERTに基づく提案手法における出力層の重みを示した．重みが大きい特徴量は問題によって異なるが，いずれの特徴量も得点に一定の寄与をしていることが読み取れる．

以上の結果から，人手での特徴量を加えることで，大幅に精度を改善できたことがわかる．

表 3: 出力層における人手で作成した特徴量の重み

	課題							
	1	2	3	4	5	6	7	8
Length features								
# of words	0.018	-0.087	0.393	0.123	-0.117	-0.296	0.366	-0.196
# of sentences	-0.123	0.151	0.078	0.033	0.209	0.130	0.335	0.050
avg. word length	0.351	0.013	0.081	-0.253	0.234	0.163	-0.353	0.060
avg. sentence length	0.076	0.017	-0.106	-0.152	-0.012	0.033	0.007	-0.035
Occurrence features								
# of lemmas	0.073	0.026	0.168	-0.149	0.159	0.406	0.387	0.219
# of spelling errors	0.001	-0.058	-0.077	0.014	0.038	-0.165	-0.085	-0.043
# of stop-words	-0.113	0.039	-0.147	-0.062	0.446	0.291	-0.126	-0.335
# of commas	0.055	0.048	0.060	-0.022	0.030	0.002	0.043	0.041
# of exclamation marks	0.021	-0.005	-0.046	-0.108	0.003	-0.020	0.003	-0.019
# of question marks	0.062	0.012	-0.040	-0.026	0.003	0.008	-0.061	-0.034
POS features								
# of nouns	0.226	-0.002	0.012	0.321	0.280	0.285	-0.009	-0.089
# of verbs	0.140	0.111	0.041	-0.003	0.098	0.079	-0.061	0.115
# of adjectives	0.031	-0.010	-0.037	0.271	-0.011	0.344	0.000	0.046
# of adverbs	0.060	0.035	-0.032	-0.084	0.020	0.140	-0.020	0.045
# of conjunctions	0.012	-0.027	0.138	-0.002	0.047	-0.133	0.000	0.057
Readability features								
Automated readability index	0.019	0.238	0.286	0.307	0.147	-0.100	-0.005	-0.038
Coleman-Liau index	-0.366	0.049	-0.159	0.144	-0.053	-0.072	0.293	-0.134
Dale-Chall readability score	0.009	-0.207	0.043	0.096	-0.002	-0.031	0.044	0.003
Difficult word count	0.139	0.202	0.315	0.279	-0.171	0.140	-0.005	0.076
Flesch reading ease	0.078	-0.166	-0.042	0.219	-0.058	-0.219	-0.050	-0.035
Flesch-Kincaid grade	-0.002	0.134	-0.076	-0.019	-0.182	0.135	-0.030	0.082
Gunning fog	-0.075	-0.301	0.002	-0.210	0.296	-0.195	-0.010	-0.038
Linsear write formula	0.032	-0.067	-0.151	0.195	-0.163	-0.007	-0.054	-0.021
Smog index	0.090	0.063	-0.046	0.203	0.054	0.081	0.106	0.071
Syllables counts	0.166	0.048	0.261	0.506	-0.055	-0.339	-0.352	0.289

7　むすび

本論文では文書レベルの特徴量を利用できる新たなハイブリッド手法を提案した．提案手法はシンプルなフレームワークにも関わらず，精度の向上に大きく寄与することを明らかにした．

なお，提案手法では，従来の深層学習モデルで得られる特徴量に，文章レベルで設計した特徴量を直接結合する形として定式化したが，結合方法には他にパタンが考えられる．また文特徴量の組み込み方も様々なバリエーションが考えられる．こちらの結合パタンについて今後は検討したい．また，提案手法では，文章レベルの特徴量として 26 個の特徴量を用いたが，PEG など大規模自動採点機はすでに 500 以上の特徴量を定義され，他の先行研究でも様々な特徴量が提案されている．今度は特徴量の数を増やして，その有効性を評価したい．さらに，本論文では出力層における特徴量の重みについて報告したが，それぞれの特徴量が予測精度にどのように影響しているかを直接には分析できていない．本研究で提案したフレームワークでは，深層学習モデルでは学習が難しいような特徴量を人手で作成した特徴量として利用することで，性能の大幅な改善が見込める．今後は，具体的にどのような特徴量が，本フレームワークにおいて有効に機能するのかを詳細に分析していきたい．また，現時点での自動採点についての研究はほとんどは英語のデータを対象にしており，日本語や中国語の研究は少ない．今後は，他言語データでの性能評価も行なっていきたい．

謝辞

本論文を作成するにあたり，指導教員の植野真臣教授から，丁寧かつ熱心なご指導を賜りました．ここに感謝の意を表します．また，日頃から親身になって研究を支えていただいた宇都雅輝助教に深謝いたします．そして，ゼミや日常の議論を通じて多くの示唆や知識を頂いた川野秀一准教授，西山悠准教授，研究室の先輩・同期・後輩に感謝いたします．

参考文献

[1] Ellis B Page. The use of the computer in analyzing student essays. International review of education, pages 210–225, 1968.

[2] Ellis Batten Page. Computer grading of student prose, using modern concepts and software. The Journal of experimental education, 62(2):127–142, 1994.

[3] Thomas Landauer, Darrell Laham, and Peter Foltz. The intelligent essay assessor. Intelligent Systems, IEEE, 15:27–31, 09 2000.

[4] Vantage Learning. How intellimetric™ works, 2005.

[5] Lawrence Rudner and Tahung Liang. Automated essay scoring using bayes' theorem. Journal of Technology, Learning, and Assessment, 1, 08 2002.

[6] Yigal Attali and Jill Burstein. Automated essay scoring with e-rater® v. 2.0. ETS Research Report Series, 2004(2):i–21, 2004.

[7] Fei Dong and Yue Zhang. Automatic features for essay scoring – an empirical study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1072–1077, Austin, Texas, November 2016. Association for Computational Linguistics.

[8] 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, pages 1882–1891, 2016.

[9] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic text scoring using neural networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics

(Volume 1: Long Papers), pages 715–725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[10] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. Skipflow: incorporating neural coherence features for end-to-end automatic text scoring. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[11] Jiawei Liu, Yang Xu, and Lingzhe Zhao. Automated essay scoring based on two-stage learning. arXiv preprint arXiv:1901.07744, 2019.

[12] Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, pages 93–102, 2018.

[13] Ellis B Page. The imminence of... grading essays by computer. The Phi Delta Kappan, 47(5):238–243, 1966.

[14] Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. Evaluating multiple aspects of coherence in student essays. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 185–192, 2004.

[15] Isaac Persing and Vincent Ng. Modeling prompt adherence in student essays. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1534–1543, 2014.

[16] Annie Louis and Derrick Higgins. Off-topic essay detection using short prompt texts. In proceedings of the NAACL HLT 2010

fifth workshop on innovative use of NLP for building educational applications, pages 92–95. Association for Computational Linguistics, 2010.

[17] Jill Burstein, Martin Chodorow, and Claudia Leacock. Automated essay evaluation: The criterion online writing service. Ai Magazine, 25(3):27–27, 2004.

[18] Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. Neural automated essay scoring and coherence modeling for adversarially crafted input. arXiv preprint arXiv:1804.06898, 2018.

[19] Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. Automated essay scoring with discourse-aware neural models. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 484–493, 2019.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[21] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In Eleventh annual conference of the international speech communication association, 2010.

[22] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pages 1480–1489, 2016.

[23] 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, 2018.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[25] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.

[26] Paul R Fitzsimmons, BD Michael, Joane L Hulley, and G Orville Scott. A readability assessment of online parkinson's disease information. The journal of the Royal College of Physicians of Edinburgh, 40(4):292–296, 2010.

[27] Meri Coleman and Ta Lin Liau. A computer readability formula designed for machine scoring. Journal of Applied Psychology, 60(2):283, 1975.

[28] RJ Senter and Edgar A Smith. Automated readability index. Technical report, CINCINNATI UNIV OH, 1967.

[29] Edgar Dale and Jeanne S Chall. A formula for predicting readability: Instructions. Educational research bulletin, pages 37–54, 1948.

[30] Mary Whisner. When judges scold lawyers. Law Libr. J., 96:557, 2004.

[31] Mark D Shermis and Jill Burstein. Handbook of automated essay evaluation: Current applications and new directions. Routledge, 2013.

[32] Jacob Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychological bulletin, 70(4):213, 1968.

[33] 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 (EMNLP), pages 1532–1543, 2014.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.