DEIM Forum 2023 1a-3-3

BERTを用いたマルチタスク学習における補助タスクの学習に適した層の分析

北村　拓斗†　　鈴木　　優†

† 岐阜大学工学部電気電子・情報工学科　〒 501–1193 岐阜県岐阜市柳戸 1–1
E-mail: †{y3033054@edu.gifu-u.ac.jp，ysuzuki@gifu-u.ac.jp}

あらまし　マルチタスク学習とは，複数のタスクを同時に学習させる手法である．主タスクを解く際のヒントとなる補助タスクを追加することで，主タスク単独で学習させたモデルよりも，モデル全体の性能を向上させることができる．先行研究より，補助タスクとして品詞タグ付けを入力層付近に追加すると，チャンキングの性能向上に有効であることが明らかになった．また，補助タスクとして固有表現抽出を追加すると，分類タスクの性能向上に有効であることが明らかになった．日本語の品詞は，意味を表す内容語と文法的な機能を持つ機能語に大別され，内容語に属する品詞は文章を特徴付ける重要な品詞だと考えられる．そこで，品詞情報を考慮した重要な品詞を予測する補助タスクを提案する．本研究では，BERT を用いたマルチタスク学習による分類タスクにおいて，提案した補助タスクが分類タスクの性能向上に有効か，BERT のどの層で学習させるのが良いのかを明らかにすることを目的とした．実験の結果，分類精度が最大で 1.756%向上し，提案した補助タスクは BERT の 2 層目で学習させるのが良いと分かった．また，品詞情報を考慮した補助タスクの追加により，モデルが意味的なまとまりを認識できるように変化することが分かった．

キーワード　BERT，自然言語処理，機械学習，マルチタスク学習

1　はじめに

マルチタスク学習とは，複数のタスクを同時に学習させ，深層学習モデル全体の性能を向上させる手法 [1] である．マルチタスク学習では，タスクごとに定めた損失関数の重み付き和をモデル全体の損失関数とし，これを最小化するパラメータを探索する．我々が複数のタスクを解くことのできるモデルを構築する場合を考える．この時，モデルが複数のタスクではなく，一部のタスクのみを解けるようになったと仮定する．このモデルは，一部のタスクを解くことはできる．一方，一部のタスクを解けたとしても，残りのタスクも解けるようになるとは限らない．複数のタスクを同時に最適化することで，初めてモデルは複数のタスクを解くことができる．つまり，モデル全体の性能を高めるには，モデルが特定のタスクに特化した知識ではなく，複数のタスクに共通した汎用的な知識を獲得する必要がある．

これまでの研究 [2], [3] では，主タスクを解く際のヒントとなる補助タスクを同時に学習させることによる性能向上が示唆されている．例えば，主タスクが評判分析タスクの場合，肯定的な文章には肯定的な単語，否定的な文章には否定的な単語が含まれやすい．この場合，肯定的，或いは否定的な単語を含んでいることが主タスクを解く際のヒントとなり得る．このことから，文章中に肯定的，或いは否定的な単語が含まれるか否かを予測するといった補助タスクを思い付くのは容易である．ただ，これは主タスクが簡単な分類タスクであるから容易に思い付くだけである．ここで，クラス数の多い分類タスクを解く場合や実験者がタスクに関する知識をあまり持たない場合を考える．この時，何が主タスクを解く際のヒントとなり得るかを考えるのは大変で，補助タスクの設計は困難である．また，補助タスクが変わるたびにラベル付けの必要性が生じ，コストがかかってしまう．そのため，汎用的な補助タスクの開発が必要である．

そこで，自然言語処理の基礎的なタスクである品詞タグ付けを参考にし，品詞情報を考慮した重要品詞抽出タスクを提案する．品詞タグ付けとは，文章中の品詞の文字列位置を推定し，推定した文字列に対して名詞や形容詞といった品詞タグを予測するタスクである．深層学習モデルは，何らかの推論を行う際に判断根拠が曖昧であると，誤った結論を導いてしまう場合が考えられる．日本語の品詞は，意味を表す内容語と文法的な機能を持つ機能語に大別され，内容語に属する品詞は文章を特徴付ける重要な品詞だと考えられる．提案した補助タスクを追加することにより，入力された文章のどの部分が重要であるかを学習させることができる．つまり，入力された文章のどの部分がモデルの推論の判断根拠となり得るか，ヒントとして与えることができる．そのため，主タスクの性能向上に有効な補助タスクとなり得るのではないかと考えた．また，品詞タグの付与を行うための形態素解析器はいくつか公開されており，誰でも利用可能である．こうした解析器を利用することで，コストをかけることなくラベル付けを行うことができると考えた．

先行研究 [4] より，ニューラルネットワークは各層で非線形な変換処理を行っており，各層で獲得した特徴量は各層ごとに異なるものだと考えた．そこで，タスクの性質によっては深い層で学習させるのではなく，浅い層で学習させる方が適している場合もあるのではないかと考えた．実際に，自然言語処理におけるマルチタスク学習の先行研究 [5] によると，モデルの各層の隠れ状態は入力層から出力層に変化するにつれて，より複雑な意味的情報を獲得している傾向があると示唆されている．

また，性質の異なるタスクを同じ層で学習させる場合を考える．この時，複数のタスクに適した仮説関数が存在しない状態で性質の異なる複数のタスクを最適化すると，競合してしまう場合があるのではないかと考えた．

以上のことから，本研究では自然言語処理において基礎的なタスクである品詞タグ付けに着目し，マルチタスク学習における主タスクの性能向上に有効であり，汎用的な補助タスクの開発を目指した．また，複数の自然言語処理タスクにて高い適応性を誇る BERT [6] を用いたマルチタスク学習において，補助タスクを学習させるのに適した層を見つけることを目指した．ただし，本研究では主タスクの性能向上のみを目的としており，補助タスクの性能向上は問わないことに注意されたい．

本研究では，異なるドメインから収集された複数のデータセットに対し，形態素解析器を用いて品詞タグの付与を自動で行った．また，主タスクのみを解くシングルタスク学習モデル，及び補助タスクを追加したマルチタスク学習モデルを構築し，モデルの学習を行った．両方のモデルのベースには，テキストデータを扱うことに秀でた BERT を用いた．その後，二つのモデルを比較し，補助タスクの追加による主タスクの分類精度の変化が有意であるか否かを対応あり 2 標本 t 検定で調べた．

実験の結果，楽天データセットでは分類精度が 0.045%向上したが，精度変化に有意差は認められなかった．Twitter 日本語評判分析データセットでは分類精度が 0.383%向上し，精度変化に有意差が認められた．livedoor ニュースコーパスでは分類精度が 1.756%向上し，精度変化に有意差が認められた．また，2 層目に補助タスクを追加した場合，分類精度が最も向上することが分かった．

さらに，BERT の Attention [7] の可視化を行い，マルチタスク学習モデルの注目している部分がどのように変化したのかを分析した．Attention は，入力されたデータに応じてニューラルネットワークのユニット同士の接続やパラメータを流動的に変える仕組みで，BERT や Vision Transformer [1]に組み込まれている．Attention の重みが大きい部分は，推論時に深層学習モデルが注目しており，モデルが何故そのように判断したかの判断根拠となり得る．Attention を可視化すると，モデルが意味的なまとまりや文章構造に注目するように変化していた．

本論文の主な貢献は，以下のとおりである．

- 提案した補助タスクを BERT の 2 層目に追加した場合，分類タスクの分類精度の向上に寄与することが明らかになった．
- 提案した補助タスクを追加すると，意味的なまとまりを認識できるようなモデルに変化することが明らかになった．

2 関 連 研 究

マルチタスク学習において，どのような補助タスクが性能向上に有効であるのか，補助タスクをどの層で学習させるのが良いのかについての研究は既にいくつか行われている．

Wu ら [8] は，医療に関する質問の意図分類タスクにおいて，固有表現抽出を同時に学習させることで性能が向上することを示した．オンライン医療に関するコーパスで事前学習した word2vec を単語埋め込みベクトルとして使用し，双方向 LSTM と Attention を組み合わせたマルチタスク学習モデルを提案した．固有表現抽出では，複数の医師によって病名，治療法といった 6 項目に対してアノテーションされたデータを用い，F 値と Recall では BERT と同程度，Precision では BERT 以上の性能を記録した．意図分類タスクでは，提案されたモデルが全ての評価指標で BERT を上回る結果となった．

Benayas ら [9] は，会話型エージェントのための自然言語理解エンジンにおいて，意図分類タスクと固有表現抽出を同時に学習させると意図分類タスクの性能が向上することを示した．Transformer をベースとしたハードパラメータ共有 [1] のモデルを提案した．また，ハードパラメータ共有とソフトパラメータ共有 [10] を融合させたモデルも提案した．

Søgaard ら [11] は，品詞タグ付けのような下位タスクを同時に学習させる場合，出力層付近よりも入力層付近で学習させるのが良いことを示した．主タスクをチャンキング，或いは CCG スーパータグ付けとし，補助タスクを品詞タグ付けとする双方向 LSTM を用いた 3 層のマルチタスク学習モデルを提案した．補助タスクを，主タスクと同じ 3 層目で学習させた場合も主タスクの F 値は向上したが，入力層に近く，主タスクと異なる 1 層目で学習させた場合の方が主タスクの F 値が向上した．

Sanh ら [5] は，複数の自然言語処理タスク間に階層関係があることを示した．1 層目で固有表現抽出，2 層目で言及抽出タスクを解くように双方向 LSTM を積み上げる．その後，3 層目で共参照解析タスクと関係抽出タスクを解くようなマルチタスク学習モデルを構築した．固有表現抽出や言及抽出タスクは，深い言語理解を必要としない下位タスクであるのでモデルの下層で学習させる．一方，共参照解析タスクや関係抽出タスクは上位タスクであるので，モデルの上層で学習させる．下位タスクを下層，上位タスクを上層で学習させることによって，より良い中間表現が得られ，複数のタスクで性能向上が見られた．

先行研究 [8] [9] より，固有表現抽出は，分類タスクの性能を向上させる補助タスクであると知られている．そのため，固有表現抽出と同じ系列ラベリング問題に属する提案した補助タスクも，主タスクの性能向上に有効な補助タスクとなるのではないかと考えた．上記研究とは，BERT を用いたマルチタスク学習での分類タスクにおいて，新たな補助タスクを提案している点，補助タスクの学習に適した層を探索している点が異なる．

3 実 験 条 件

3.1 節では，実験で使用したデータセットの基本的事項について述べる．今回は，3 種類の日本語のデータセットを用いて実験を行った．3.2 節では，使用するテキストデータの前処理，及びラベル付けについて述べる．機械のみを用いて，品詞タグの付与を自動で行った．3.3 節では，トークナイザについて述べる．提案した補助タスクを解くために，既存のトークナイザに対して改良を行った．3.4 節では，マルチタスク学習モデル，

1：https://github.com/lucidrains/vit-pytorch#vision-transformer---pytorch

及びシングルタスク学習モデルの作成方法について述べる．また，ハイパーパラメータや Early Stopping といったモデルの詳細な設定についても述べる．

3.1 データセット

本研究では，提案した補助タスクが分類タスクの性能向上に有効な補助タスクであるか否かを検証するために実験を行う．そのため，使用するデータセットが異なるデータ分布から生成されたものでなければならない．そこで，ドメインの異なる以下のデータセットを用いた．

3.1.1 Twitter 日本語評判分析データセット

Twitter 日本語評判分析データセットとは，岐阜大学工学部鈴木研究室にて提供されているデータセット [2] である．2015 年から 2016 年の期間に収集された電子機器に関するツイートに対し，複数のクラウドワーカーの投票によりポジネガ，ポジティブ，ネガティブ，ニュートラル，無関係の五種類のラベルが付与されている．このデータセットには，約 53 万件のアノテーション済みツイートが含まれる．

クラウドソーシングによるラベル付けでは，複数のクラウドワーカーの投票結果をもとに多数決を行い，投票数の最も多いラベルを採用する．しかし，投票結果次第では投票数の最も多いラベルが複数採用される場合も起こり得る．今回は，採用されたラベルが一意に定まるもののみ使用した．ポジティブ，ネガティブ，ニュートラルの三種類のクラスを用いた．

3.1.2 楽天データセット

楽天データセットとは，国立情報学研究所の情報学研究データリポジトリにて提供されているデータセット [3] である．このデータセットには，約 7,000 万件の商品レビューが含まれる楽天市場や，約 80 万件のレシピ情報が含まれる楽天レシピといったサブセットが存在する．

今回は，サブセットである楽天市場の商品レビューのうち食品カテゴリに該当し，2015 年に投稿されたレビューを使用した．レビューの評価値が 5 と 4 のものをポジティブ，2 と 1 のものをネガティブとして扱い，二種類のクラスを用いた．

3.1.3 livedoor ニュースコーパス

livedoor ニュースコーパスとは，株式会社ロンウィットにより提供されているデータセット [4]である．このデータセットには，NHN Japan 株式会社が運営する livedoor ニュースに関する URL や日付，タイトルや本文が含まれている．ニュース記事に関する九つのカテゴリが存在し，各カテゴリに 512 から 901 件のニュース記事が含まれ，記事の総数は 7,376 件である．

今回は，ニュース記事の本文のみ使用した．映画に関するニュース記事カテゴリをクラス 2，IT に関するニュース記事カテゴリをクラス 5 のように扱い，九種類のクラスを用いた．

3.2 前処理，ラベル付け

3.1 節で述べたデータセットに対し，テキストデータの前処理，主タスクで使用するラベルの付与を行う．その後，補助タスクで使用する品詞タグの付与を行う．

3.2.1 前 処 理

前処理では，数字を全て 0 に置換する，全角文字を半角文字に変換するといったテキストデータの正規化を行う．また，URL を除去する，記号を除去する，改行文字を除去する，空白を除去するといった不要語の除去を行う．

さらに，前処理の後に重複したテキストデータが出現する場合が考えられる．そのため，重複したデータの除去を行う．

3.2.2 主タスクで使用するラベルの付与

ラベル付けの過程で記述内容や投稿日時の偏りが生じないように，前処理後のデータ集合に対してランダムでシャッフルを行う．その後，シャッフル済みのテキストデータを，各クラス間のラベルに偏りが生じないようにインスタンス数を統一してラベル付けを行う．ただし，BERT で扱うことのできるトークンの最大長である 512 トークンを超えないようにする必要がある．そのため，楽天データセットでは 500 単語以上のテキストデータを除外し，livedoor ニュースコーパスではテキストデータの 512 単語を超えた部分の切り捨てを行った．

Twitter 日本語評判分析データセットの各クラスのインスタンス数は，10,000 件とした．実験で使用した全データは 3 クラス × 10,000 件の合わせて 30,000 件である．楽天データセットの各クラスのインスタンス数は，20,000 件とした．実験で使用した全データは 2 クラス × 20,000 件の合わせて 40,000 件である．livedoor ニュースコーパスの各クラスのインスタンス数は，500 件とした．実験で使用した全データは 9 クラス × 500 件の合わせて 4,500 件である．

3.2.3 補助タスクで使用する品詞タグの付与

一般に，機械学習で扱う何らかのデータに対してラベルを付与する場合，人手による作業と機械を用いた作業の二つが考えられる．人手による作業のメリットは，時間や金銭的コストをかけることで，品質の保証されたデータセットの作成ができることである．しかし，デメリットとして品質が作業者の習熟度に依存すること，作業者の作業量が一定ではなくアノテーション済みデータの収集に時間がかかってしまうことが考えられる．

本研究では，補助タスクで使用する品詞タグの付与を行うために，形態素解析器の MeCab を用いて品詞判定を行った．機械を利用することでコストをかけることなく，品質が安定したデータセットの作成を高速かつ自動で行った．MeCab の辞書には，ipadic-NEologd[5]を利用した．

日本語には，名詞や形容詞，接続詞や助詞など全部で十種類の品詞が存在する．その中でも，品詞は意味を表す内容語と文法的な機能を持つ機能語に大別され，名詞や形容詞は内容語，接続詞や助詞は機能語に属する．本研究では，補助タスクの難易度を簡単にするために全種類の品詞を用いず，内容語に属す

2：https://www.db.info.gifu-u.ac.jp/sentiment_analysis/

3：楽天グループ株式会社 (2014):楽天データセット.国立情報学研究所情報学研究データリポジトリ.(データセット).https://doi.org/10.32130/idr.2.0

4：https://www.rondhuit.com/download.html#ldcc

5：https://github.com/neologd/mecab-ipadic-neologd/blob/master/README.ja.md

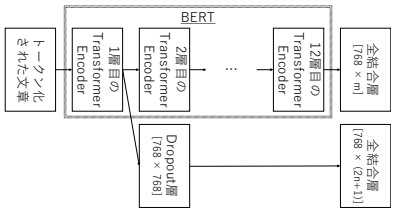

図 1　ハードパラメータ共有のマルチタスク学習モデル

る名詞，副詞，及び形容詞のみを予測するタスクを設計した．

品詞タグは，Inside の I，Outside の O，及び Begin の B を意味するタグを用いる IOB2 形式に倣ってラベル付けを行った．実験で用いる品詞には，B-名詞や I-副詞のように B タグと I タグを付与し，それ以外の品詞には O タグを付与した．

3.3　トークナイザ

提案した補助タスクを解くためには，BERT の既存のトークナイザを改良する必要がある．既存のトークナイザを使用すると品詞のまとまりでサブワードに分かれず，品詞タグの予測がうまくできない可能性が考えられる．

そこで，B タグと I タグの部分に対して先にトークン化を行う．その後，残りの O タグの部分に対してトークン化を行い，トークンを結合するようにトークナイザを改良した．

3.4　モデル作成，及びモデルの詳細な設定

本研究では，自然言語処理に特化した深層学習モデルである BERT を使用する．BERT の事前学習済みモデルは，東北大学の BERT-base モデル [6] を利用した．このモデルをファインチューニングし，実験に用いた．今回の実験では，BERT で扱うことのできるトークンの最大長は 512 とし，バッチサイズは 32 とした．モデルの学習率について，パラメータを訓練可能にした BERT 各層の学習率は 5×10^{-5}，分類層の学習率は 1×10^{-4} である．使用した optimizer は Adam で，学習率の減衰やウォームアップといった学習率の調整は行っていない．

シングルタスク学習モデルとは，一つの入力に対して一つの出力が存在するモデルである．一方，マルチタスク学習モデルとは，一つの入力に対して複数の出力が存在するモデルである．前者は主タスクのみ解くことができるが，後者は主タスクに加えて補助タスクも解くことができる．

例えば，BERT の 1 層目で補助タスクを解き，12 層目で主タスクを解くマルチタスク学習モデルを作成するとする．ここで，補助タスクは n 種類の重要品詞抽出タスクで，主タスクは m クラス分類タスクである．まず，事前学習済み BERT モデルの 1 層目，12 層目のパラメータを訓練可能にする．その後，1 層目にドロップアウト層と全結合層，12 層目に全結合層を追加する．1 層目に追加するドロップアウト層の入力層は 768 次元，出力層は 768 次元であり，ドロップアウト層の出力を初期化する確率 ρ は $\rho = 0.1$ である．ドロップアウト層を経由する全結合層の入力層は 768 次元，出力層は $2n + 1$ 次元である．

6：https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

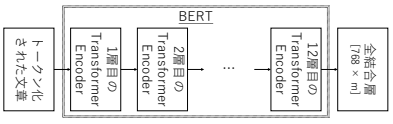

図 2　シングルタスク学習モデル

12 層目に追加する全結合層の入力層は 768 次元，出力層は m 次元である．ハードパラメータ共有のマルチタスク学習モデルを図 1 に示した．上側の全結合層で主タスクを解き，下側のドロップアウト層を経由した全結合層で補助タスクを解いている．

一方，シングルタスク学習モデルは，比較対象のマルチタスク学習モデルと同じ層のパラメータを訓練可能にする．今回の例では，事前学習済み BERT モデルの 1 層目，12 層目のパラメータを訓練可能にする．その後，12 層目に全結合層を追加してシングルタスク学習モデルを作成する．12 層目に追加する全結合層の入力層は 768 次元，出力層は m 次元である．シングルタスク学習モデルを図 2 に示した．

シングルタスク学習モデルで 1 層目，12 層目のパラメータを訓練可能にするのは，マルチタスク学習モデルのシングルタスク学習モデルに対する優位性を検証する際に，訓練可能なパラメータ数の増加による性能への影響を排除するためである．

モデル全体の損失関数 L_{all} は以下のように設定した．L_{main} は主タスクの損失関数，λ_{main} は主タスクの損失関数の重み付けパラメータである．また，L_{sub} は補助タスクの損失関数，λ_{sub} は補助タスクの損失関数の重み付けパラメータである．

$$L_{all} = L_{main} \times \lambda_{main} + L_{sub} \times \lambda_{sub}$$

マルチタスク学習では，λ の値を変えてタスク間の Loss を重み付けすることで，どのタスクをどの程度重要視するか調整することができる．例えば，Kendal ら [12] が Homoscedastic Uncertainty に基づいたタスク間の重み付けを行うことで，タスク間の重要度を調整している．今回は，λ_{main} を 1，λ_{sub} を 0.5 としたが，最適な λ の比率の探索は行っていないため，今回設定した条件より適した比率が存在することは十分に考えられる．また，各タスクの損失関数は Cross Entropy Loss である．

過学習を防ぐために，Early Stopping を行った．検証用データの Loss の最小値が 10epoch 更新されなかったら学習停止し，最小値を記録した時点のモデルを保存する．予備実験にて，検証用データの Loss の最小値が 100epoch 更新されなかったら学習停止するように設定して実験を行ったが，10epoch の場合の結果と変わらなかったため，10epoch で十分であると判断した．

4　実　　験

実験の目的は，主に二つある．一つ目は，BERT を用いたマルチタスク学習において，提案した補助タスクが主タスクの性能向上に有効な補助タスクであるか否かを調べることである．二つ目は，提案した補助タスクを学習させるのに適した層は，BERT のどの層であるのかを調べることである．

4.1 実験手順

まず，3.2 節で述べた方法でラベル付けを行い，交差検証を行うために全てのテキストデータを 10 個に分割する．訓練用，検証用，テスト用データの割合は 8：1：1 とした．

次に，3.4 節で述べたようにハイパーパラメータを設定し，マルチタスク学習モデル，及びシングルタスク学習モデルを作成する．その後，同一の訓練用，検証用データを用いて二つのモデルの学習を行い，Early Stopping を行う．

最後に，同一のテスト用データを用いて推論を行う．その際，マルチタスク学習モデルとシングルタスク学習モデル間の主タスクの分類精度を比較し，補助タスクの追加による分類精度の変化が有意であるか否かを対応あり 2 標本 t 検定で調べる．

4.2 実験結果，考察

4.2 節では各データセットでの実験結果と全体的な結果を示す．その後，テスト用データで誤った推論を行った以下の事例について，BERT の Attention を可視化して要因を分析する．

事例	マルチタスク学習モデル	シングルタスク学習モデル
①	正しく予測できた	正しく予測できなかった
②	正しく予測できなかった	正しく予測できた

BERT の Self Attention には 12 個の Multi Head Attention が存在し，それぞれで異なる Attention の重みが得られる．そこで，Attention の重みの相加平均をとり，256 段階の RGB スケールと対応させることで可視化を行った．

4.2.1 Twitter 日本語評判分析データセット

3.1.1 項で述べた Twitter 日本語評判分析データセットを用いて実験を行った．実験結果，及び検定結果を図 3，表 1 に示す．実験の結果，補助タスクを追加したほぼ全ての層で，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が高くなった．表 1 の検定結果より，補助タスクを 2 層目，6 層目，7 層目に追加した場合に p 値が有意水準 0.05 を下回り，有意差が認められた．つまり，補助タスクの追加により分類精度が向上したと言える．また，2 層目，6 層目，7 層目に補助タスクを追加した場合，マルチタスク学習モデルはシングルタスク学習モデルと比較して，分類精度がそれぞれ 0.383%，0.337%，0.373%上昇している．これらから，補助タスクを入力層に近い 2 層目に追加した時に分類精度が最も良くなり，補助タスクを学習させるのに適した層は 2 層目だと言える．この帰結は，品詞タグ付けのような下位タスクは入力層付近で学習させるのが良いという先行研究 [11] での知見とも一致する結果となった．

表 1　4.2.1 項の実験結果における分類精度の平均値と p 値
(注：表中の太字は，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が高くて p 値が有意水準 0.05 を下回ったもの，下線は，p 値が有意水準を下回らなかったものを示す．)

補助タスク追加層	分類精度 [%]		p 値
	Multi Task	Single Task	
1 層目	81.240 ± 0.572	81.243 ± 0.575	0.5053
2 層目	**81.627 ± 0.520**	81.240 ± 0.488	**0.0289**
3 層目	81.513 ± 0.524	81.303 ± 0.487	0.1046
4 層目	81.560 ± 0.476	81.520 ± 0.626	0.4185
5 層目	81.500 ± 0.540	81.350 ± 0.395	0.1192
6 層目	**81.370 ± 0.727**	81.033 ± 0.444	**0.0333**
7 層目	**81.390 ± 0.673**	81.017 ± 0.610	**0.0278**
8 層目	80.790 ± 0.538	80.600 ± 0.735	0.1666
9 層目	80.153 ± 0.623	79.730 ± 0.837	0.0596
10 層目	79.913 ± 0.798	79.727 ± 0.639	0.1043
11 層目	79.463 ± 0.714	79.157 ± 0.853	0.1496
12 層目	78.187 ± 0.912	78.277 ± 0.706	0.7125

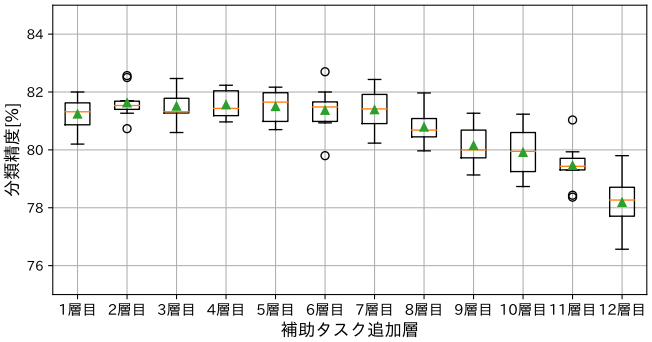

図 3　補助タスク追加層の違いによるマルチタスク学習の分類精度比較
(4.2.1 項の実験結果)

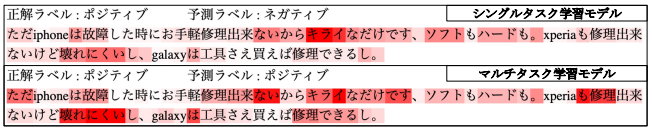

図 4　事例 ①(4.2.1 項の実験結果の可視化)

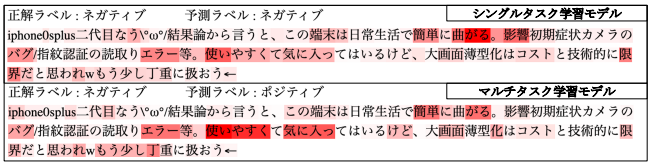

図 5　事例 ②(4.2.1 項の実験結果の可視化)

分類精度に変化が見られた要因を分析するために，Attention の可視化を行う．使用したデータは，分類精度が最も良くなった補助タスクを 2 層目に追加したマルチタスク学習モデル，及び比較対象となる 2 層目のパラメータを訓練可能にしたシングルタスク学習モデルを用いた際の分類結果である．

① の場合の例を図 4 に示す．ポジティブラベルが付与された文章をシングルタスク学習モデルではネガティブクラスと誤って予測し，マルチタスク学習モデルではポジティブクラスと正しく予測したものである．濃い赤色の部分は Attention の重みが大きく，モデルが注目している．一方，薄い赤色の部分は Attention の重みが小さく，モデルがあまり注目していない．シングルタスク学習モデルでは，否定的な意味の「キライ」や「にくい」に注目している．一方，マルチタスク学習モデルでは「キライ」の部分にかかっている Attention の重みは小さくなり，ツイートの投稿主が何故嫌いなのかについて言及している「修理出来ないから」の部分にも注目するように変化した．また，シングルタスク学習モデルでは「にくい」の部分に注目していたが，マルチタスク学習モデルでは特に「壊れにくい」

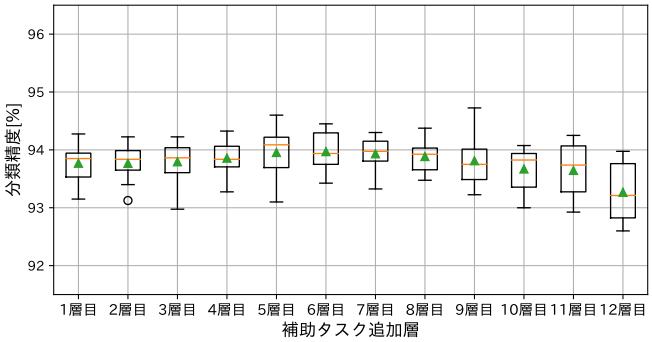

図 6　補助タスク追加層の違いによるマルチタスク学習の分類精度比較 (4.2.2 項の実験結果)

の部分に注目するように変化した．これにより「にくい」のみでは否定的な意味であるが,「壊れにくい」となることで肯定的な意味になる．マルチタスク学習モデルでは文章中の肯定的な意味を認識することで，正しく予測できたと考えられる．

② の場合の例を図 5 に示す．ネガティブラベルが付与された文章をシングルタスク学習モデルではネガティブクラスと正しく予測し，マルチタスク学習モデルではポジティブクラスと誤って予測したものである．マルチタスク学習モデルでは,「使いやすく」や「気に入っ」のような意味的なまとまりを捉えられるように変化した．しかし，この部分は肯定的な意味を持つため，ポジティブクラスであると予測を誤ったと考えられる．

以上から，提案した補助タスクを同時に学習させることで，モデルが単語間の関係性を認識し，より正確に文章の構造や意味のまとまりを捉えられるようになったと考えられる．すなわち，提案した補助タスクがモデルの言語理解能力を高めることに寄与し，マルチタスク学習モデルの分類精度の向上に繋がったと考えられる．しかし，意味のまとまりを捉えることが逆に予測を誤る要因となった場合もいくつか確認された．

4.2.2　楽天データセット

3.1.2 項で述べた楽天データセットを用いて実験を行った．実験結果，及び検定結果を図 6，表 2 に示す．実験の結果，補

表 2　4.2.2 項の実験結果における分類精度の平均値と p 値
(注：表中の二重下線は，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が高く，p 値が有意水準 0.5 を下回ったものを示す．)

補助タスク追加層	分類精度 [%]		p 値
	Multi Task	Single Task	
1 層目	93.770 ± 0.316	93.797 ± 0.310	0.6352
2 層目	93.767 ± 0.309	93.927 ± 0.464	0.9649
3 層目	93.795 ± 0.355	93.912 ± 0.325	0.8655
4 層目	93.858 ± 0.314	93.955 ± 0.430	0.9154
5 層目	93.955 ± 0.447	93.995 ± 0.355	0.6468
6 層目	93.970 ± 0.358	94.065 ± 0.244	0.8246
7 層目	93.932 ± 0.297	93.987 ± 0.415	0.6761
8 層目	93.885 ± 0.273	93.953 ± 0.403	0.7648
9 層目	93.812 ± 0.448	93.905 ± 0.411	0.7702
10 層目	93.670 ± 0.372	93.625 ± 0.578	0.3994
11 層目	93.645 ± 0.451	93.775 ± 0.441	0.8783
12 層目	93.267 ± 0.481	93.430 ± 0.450	0.9576

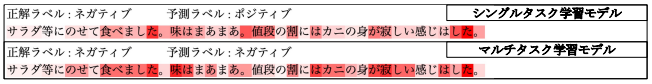

図 7　事例 ①(4.2.2 項の実験結果の可視化)

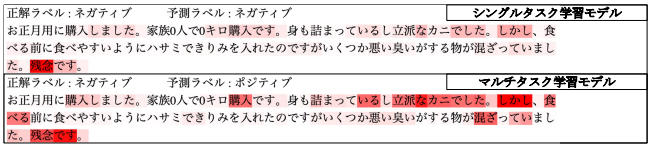

図 8　事例 ②(4.2.2 項の実験結果の可視化)

助タスクを追加したほぼ全ての層で，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が低くなった．10 層目に補助タスクを追加した場合のみ，シングルタスク学習モデルと比較して分類精度が 0.045%上昇している．しかし，表 2 の検定結果より，p 値が有意水準 0.05 を下回るものはなく，シングルタスク学習モデルとマルチタスク学習モデル間における分類精度の変化に有意差は見られなかった．つまり，補助タスクの追加により分類精度が向上も低下もしなかったと言える．

分類精度に変化が見られなかった要因を分析するために，Attention の可視化を行う．使用したデータは，先ほどのデータセットでの実験と同じく，補助タスクを 2 層目に追加したマルチタスク学習モデル，及び 2 層目のパラメータを訓練可能にしたシングルタスク学習モデルを用いた際の分類結果である．

① の場合の例を図 7 に示す．ネガティブラベルが付与された文章をシングルタスク学習モデルではポジティブクラスと誤って予測し，マルチタスク学習モデルではネガティブクラスと正しく予測したものである．シングルタスク学習モデルでは，文章全体に満遍なく注目している．一方，マルチタスク学習モデルでは「味はまあまあ」や「カニの身が寂しい」の部分に注目するように変化した．補助タスクの追加により，モデルが文章を特徴付ける他の部分にも注目するように変化し，正しく予測できるようになったと考えられる．

② の場合の例を図 8 に示す．ネガティブラベルが付与された文章をシングルタスク学習モデルではネガティブクラスと正しく予測し，マルチタスク学習モデルではポジティブクラスと誤って予測したものである．シングルタスク学習モデルでは，特に「残念」の部分に注目している．一方，マルチタスク学習モデルでは「立派なカニでした」の部分にも注目するように変化した．また，シングルタスク学習モデルでは逆接の接続詞である「しかし」の部分にあまり注目しなかったが，マルチタスク学習モデルでは注目するように変化した．一見すると，マルチタスク学習モデルでは文章全体の構造を捉えているように思われるが，誤った予測を行った．

楽天データセットを用いた実験で分類精度が向上しなかった要因として，データの作成方法が適切ではなかったと考えられる．本実験では，評価値が 5 と 4 のものをポジティブクラス，2 と 1 のものをネガティブクラスとして扱った．ただ，評価値が 2 や 4 のレビューには,「コスパは良いが配送が遅い」や「見た目は良くないけど味は良かった」のような逆接表現が使用さ

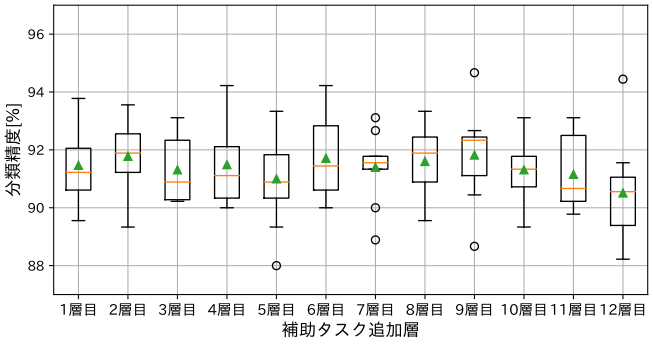

図 9　補助タスク追加層の違いによるマルチタスク学習の分類精度比較 (4.2.3 項の実験結果)

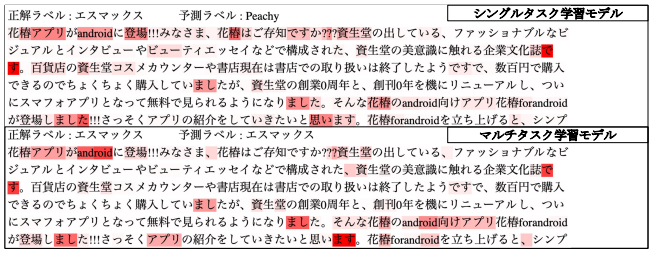

図 10　事例 ①(4.2.3 項の実験結果の可視化)

れている場合が多く，ポジティブな内容とネガティブな内容の両方を含むレビューが多い．今回の実験で誤った予測を行ったデータには逆接表現を含む場合が多く，たまたま一方のモデルでは予測でき，もう一方のモデルでは予測できなかったといった偶然性が否めない．評価値が 2 や 4 のものを使用しない，或いは別のクラスに分けて実験を行い，補助タスクが分類精度の向上に寄与するかどうかを再度調査する必要がある．

4.2.3　livedoor ニュースコーパス

3.1.3 項で述べた livedoor ニュースコーパスを用いて実験を行った．実験結果，及び検定結果を図 9，表 3 に示す．実験の結果，補助タスクを追加した全ての層で，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が高くなった．表 3 の検定結果より，補助タスクを 5 層目，12 層目以外に追加

表 3　4.2.3 項の実験結果における分類精度の平均値と p 値
(注：表中の太字は，シングルタスク学習モデルよりマルチタスク学習モデルの分類精度が高くて p 値が有意水準 0.05 を下回ったもの，下線は，p 値が有意水準を下回らなかったものを示す．)

補助タスク追加層	分類精度 [%]		p 値
	Multi Task	Single Task	
1 層目	**91.467 ± 1.262**	90.400 ± 1.051	**0.0099**
2 層目	**91.778 ± 1.116**	90.022 ± 1.489	**0.0033**
3 層目	**91.311 ± 1.144**	90.400 ± 1.132	**0.0010**
4 層目	**91.489 ± 1.341**	90.467 ± 1.049	**0.0119**
5 層目	91.000 ± 1.566	90.667 ± 1.225	0.2579
6 層目	**91.711 ± 1.334**	90.244 ± 0.875	**0.0006**
7 層目	**91.400 ± 1.146**	90.489 ± 0.788	**0.0116**
8 層目	**91.600 ± 1.299**	90.356 ± 1.043	**0.0236**
9 層目	**91.822 ± 1.516**	90.578 ± 1.202	**0.0092**
10 層目	**91.311 ± 1.059**	90.511 ± 1.265	**0.0154**
11 層目	**91.156 ± 1.264**	90.311 ± 1.164	**0.0415**
12 層目	90.511 ± 1.651	89.556 ± 0.905	0.0577

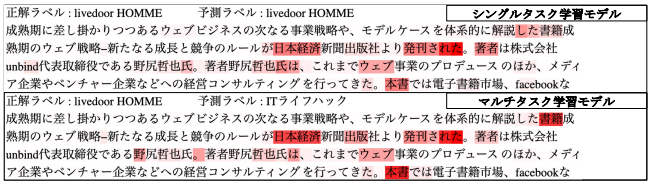

図 11　事例 ②(4.2.3 項の実験結果の可視化)

した場合に p 値が有意水準 0.05 を下回り，有意差が認められた．つまり，補助タスクの追加により分類精度が向上したと言える．また，有意差が認められた補助タスク追加層について，分類精度が最も向上したのは補助タスクを 2 層目に追加した場合である．これらのことから，補助タスクの学習に適した層は 2 層目だと言える．この帰結は，先行研究 [11] での知見とも一致する結果となった．

分類精度の変化の要因を分析するために，Attention の可視化を行う．使用したデータは，補助タスクを 2 層目に追加したマルチタスク学習モデル，及び 2 層目のパラメータを訓練可能にしたシングルタスク学習モデルを用いた際の分類結果である．

① の場合の例を図 10 に示す．エスマックスラベルが付与された文章をシングルタスク学習モデルでは Peachy クラスと誤って予測し，マルチタスク学習モデルではエスマックスクラスと正しく予測したものである．エスマックスクラスにはスマートフォンを中心としたモバイルに関する生活に役立つ情報の記事，Peachy クラスには女性向けの記事が含まれている．シングルタスク学習モデルでは,「資生堂」や「ビューティ」といった女性向けの記事に関連しそうな単語に注目している．一方，マルチタスク学習モデルでは「android」や「アプリ」の部分によく注目するように変化した．特に,「android 向けアプリ」の部分を一つのまとまりとして捉えられるように変化した．これらのことから，マルチタスク学習モデルではモデルの注目する部分が変化し，正しく予測できるようになったと考えられる．

② の場合の例を図 11 に示す．livedoor HOMME ラベルが付与された文章をシングルタスク学習モデルでは livedoor HOMME クラスと正しく予測し，マルチタスク学習モデルでは IT ライフハッククラスと誤って予測したものである．どちらのモデルも，全体的にモデルが注目している部分に差はないように見える．変化点としては，シングルタスク学習モデルと比較して，マルチタスク学習モデルでは「書籍」や「日本経済」,「本書」の部分における Attention の重みが大きいと言える．しかし，予測を誤った要因は Attention の重みの変化だけとは考えにくく，他にも要因があると考えられる．今後，Attention の可視化以外の方法で要因を解明する必要がある．

4.3　BERT の 2 層目で補助タスクを学習させるのが適していた理由の考察

4.2 節の実験結果より，品詞情報を用いた補助タスクを BERT の 2 層目に追加した場合，主タスクの性能を最も向上させることが明らかとなった．先行研究 [5] によると，モデルの各層の隠れ状態は入力層から出力層に変化するにつれて，より複雑な意味的情報を獲得している傾向があると示唆されている．今回

補助タスクで用いた品詞情報は，複雑な意味的情報ではなく，文章の表面的な情報であると考えられる．そのため，複雑な意味的情報ではない，表面的な情報を持つと考えられる入力層付近で補助タスクを学習させるのが適しており，主タスクの性能を最も向上させるという結果が得られたと考えられる．

5 ま と め

本研究は，マルチタスク学習における主タスクの性能向上に有効かつ汎用的な補助タスクの開発を目的とした．また，BERTを用いたマルチタスク学習において，補助タスクを学習させるのに適した層を見つけることを目的とした．これまでのマルチタスク学習における補助タスクの設計では，主タスクが変わる度に補助タスクが考案されていた．また，補助タスク設計は実験者の主観や経験に頼る場合が多かった．そのため，主タスクに依存しない汎用的な補助タスクの開発が必要である．そこで，自然言語処理の基礎的なタスクである品詞タグ付けに注目し，重要品詞抽出タスクを提案する．提案した補助タスクが，主タスクの性能向上に寄与するか否かを検証するために実験を行った．複数のデータセットに対し，形態素解析器を用いて品詞タグの付与を行った．シングルタスク学習モデル，及びマルチタスク学習モデルを構築し，モデルの学習を行った．その後，二つのモデルを比較し，補助タスクの追加による主タスクの分類精度の変化が有意であるか否かを対応あり t 検定で調べた．

実験の結果，楽天データセットでは，分類精度が 0.045%向上した．しかし，精度変化に有意差は認められなかった．Twitter日本語評判分析データセットでは，分類精度が 0.383%向上した．この場合は精度変化に有意差が認められた．livedoor ニュースコーパスでは，分類精度が 1.756%向上した．この場合も精度変化に有意差が認められた．提案した補助タスクは，BERTの 2 層目に追加して学習させるのが良いということが分かった．また，BERT の Attention の可視化を行い，モデルの注目している部分を比較することにより，分類精度が向上した要因を分析した．Attention の可視化を行うと，深層学習モデルが意味的なまとまりや文章構造に注目するように変化していた．

今後の展望として，他のタスクにおいても，提案した補助タスクの有効性を検証したいと考えている．本研究では，主タスクが分類タスクの場合のみ実験を行った．しかし，他のタスクでは実験を行っていない．他のタスクでも性能向上に有効であるかを確かめるために，追加で実験を行う必要がある．

また，MeCab 以外の形態素解析器を用いてラベル付けを行いたいと考えている．MeCab には，辞書の ipadic や ipadic-NEologd が古くて新しい語彙に対応できない，並びに他の解析器と比較して解析速度は速いが解析精度は劣るという問題点が存在する．辞書の継続的な更新が保障されている Sudachi，高精度な形態素解析を行うことができる JUMAN++のような他の解析器を用いることで問題点を解消できると考えている．

さらに，主タスクの性能向上に有効でラベル付けコストのかからない，今回のような補助タスクを他にも発見したいと考えている．マルチタスク学習において，主タスクの性能を向上させる補助タスクの特性については解明されていないことが多い．新たな補助タスクの発見が，どのような補助タスクが性能向上に有効であるのかを突き止める一助になれば良いと考える．

謝辞　本研究の一部は JSPS 科研費 19H04218 および越山科学技術振興財団の助成を受けたものです．本研究では、国立情報学研究所の IDR データセット提供サービスにより楽天グループ株式会社から提供を受けた「楽天データセット」（`https://rit.rakuten.com/data_release/`）を利用した．

文　献

[1] Rich Caruana. Multitask learning. *Machine learning*, Vol. 28, No. 1, pp. 41–75, 1997.

[2] Jeremy Barnes, Erik Velldal, and Lilja Øvrelid. Improving sentiment analysis with multi-task learning of negation. *Natural Language Engineering*, Vol. 27, No. 2, pp. 249–269, 2021.

[3] Hao Cheng, Hao Fang, and Mari Ostendorf. Open-domain name error detection using a multi-task rnn. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 737–746, 2015.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[5] Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 6949–6956, 2019.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[7] 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*, pp. 5998–6008, 2017.

[8] Chaochen Wu, Guan Luo, Chao Guo, Yin Ren, Anni Zheng, and Cheng Yang. An attention-based multi-task model for named entity recognition and intent analysis of chinese online medical questions. *Journal of Biomedical Informatics*, Vol. 108, p. 103511, 2020.

[9] Alberto Benayas, Reyhaneh Hashempour, Damian Rumble, Shoaib Jameel, and Renato Cordeiro De Amorim. Unified transformer multi-task learning for intent classification with entity recognition. *IEEE Access*, Vol. 9, pp. 147306–147314, 2021.

[10] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[11] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–235, 2016.

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.