卒業論文

BERT におけるタスク同士の階層関係に注目したマルチタスク学習の分析

北村 拓斗

2023 年 2 月 8 日

岐阜大学 工学部 電気電子・情報工学科 情報コース
鈴木研究室

本論文は岐阜大学工学部に
学士（工学）授与の要件として提出した卒業論文である.

北村 拓斗

指導教員：
鈴木 優 准教授

BERT におけるタスク同士の階層関係に注目したマルチタスク学習の分析*

北村 拓斗

内容梗概

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ことにより，タスク単体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学習手法である．先行研究より，LSTM を用いたマルチタスク学習において，品詞タグ付けがチャンキングの性能向上に有効な補助タスクであり，低層で学習させるのが良いと明らかになっている．また，複数の深層学習モデルを用いたマルチタスク学習において，固有表現抽出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ことが明らかになっている．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特に文章を特徴付けていると考えられる．そこで，補助タスクとして品詞情報を考慮した重要品詞抽出タスクを提案する．この補助タスクは，特に文章を特徴付けていると考えられる内容語に属する品詞を予測するタスクである．提案した補助タスクは固有表現抽出タスクと同じ系列ラベリング問題であり，分類タスクの性能向上に有効なタスクとなるのではないかと考えた．本研究では，BERT を用いたマルチタスク学習における分類タスクで，提案したタスクが分類タスクの性能向上に有効か否かを明らかにすることを目的とした．また，補助タスクを 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．提案した補助タスクを組み込んだマルチタスク学習モデル，及び補助タスクを用いないシングルタスク学習モデルを作成し，複数のデータセットに対して実験を行った．補助タスクを追加したことによる分類タスクへの影響を確かめるために，二つのモデルの比較を行った．実験の結果，楽天データセットでは，Accuracy が最大で 0.045% 向上した．Twitter 日本語評判分析データセットでは，Accuracy が 0.383% 向上し，Accuracy の変化に有意差が認められた．

*岐阜大学 工学部 電気電子・情報工学科 情報コース 卒業論文, 学籍番号: 1193033054, 2023 年 2 月 8 日.

livedoor ニュースコーパスでは，Accuracy が最大で 1.756% 向上し，Accuracy の変化に有意差が認められた．提案した補助タスクを用いたマルチタスク学習を行うと，クラス数の多い分類タスクの場合には Accuracy の顕著な向上が見られた．また，補助タスクを BERT の 2 層に追加した場合に，分類タスクの Accuracy が最も向上すると分かった．

キーワード

機械学習，マルチタスク学習，自然言語処理，BERT

目次

図目次 v

表目次 vi

第 1 章 はじめに 1

第 2 章 基本的事項 4

2.1 BERT 4

2.1.1 Attention 5

2.1.2 Transformer 6

2.2 マルチタスク学習 7

2.3 重要品詞抽出タスク 8

2.4 対応あり 2 標本 t 検定 9

2.5 評価指標 10

2.5.1 Accuracy 10

2.5.2 Precision 11

2.5.3 Recall 11

2.5.4 F-score 11

第 3 章 関連研究 12

第 4 章 提案手法 14

4.1 データセット 14

4.1.1 Twitter 日本語評判分析データセット 15

4.1.2 楽天データセット 15

4.1.3 livedoor ニュースコーパス 16

4.2 前処理，ラベル付け 16

4.2.1 前処理 16

4.2.2 主タスクに使用するラベルの付与 17

4.2.3 補助タスクに使用する品詞タグの付与 17

4.3 トークナイザ 18
4.4 モデル作成 19

第 5 章 評価実験 22
5.1 実験手順 22
5.2 実験条件 22
5.3 実験結果，考察 23
5.3.1 Twitter 日本語評判分析データセット 24
5.3.2 楽天データセット 28
5.3.3 livedoor ニュースコーパス 32
5.3.4 全てのデータセットでの比較 35

第 6 章 おわりに 37

謝辞 39

参考文献 40

発表リスト 43

図目次

4.1 ハードパラメータ共有のマルチタスク学習モデル 20
4.2 シングルタスク学習モデル 21

5.1 Attention の可視化の例 24
5.2 5.3.1 項の実験結果における補助タスク追加層の違いによる Accuracy の比較 26
5.3 5.3.1 項の実験結果におけるマルチタスク学習モデルで予測できた例 26
5.4 5.3.1 項の実験結果におけるマルチタスク学習モデルで予測できなかった例 27
5.5 5.3.2 項の実験結果における補助タスク追加層の違いによる Accuracy の比較 30
5.6 5.3.2 項の実験結果におけるマルチタスク学習モデルで予測できた例 30
5.7 5.3.2 項の実験結果におけるマルチタスク学習モデルで予測できなかった例 31
5.8 5.3.3 項の実験結果における補助タスク追加層の違いによる Accuracy の比較 33
5.9 5.3.3 項の実験結果におけるマルチタスク学習モデルで予測できた例 34
5.10 5.3.3 項の実験結果におけるマルチタスク学習モデルで予測できなかった例 34

表目次

2.1 評価指標のための混同行列 10

5.1 5.3.1 項の実験結果における Accuracy の平均値と p 値 25
5.2 5.3.2 項の実験結果における Accuracy の平均値と p 値 29
5.3 5.3.3 項の実験結果における Accuracy の平均値と p 値 32
5.4 マルチタスク学習による各評価指標の変化 35

第 1 章　はじめに

マルチタスク学習とは，主タスクと関係した複数のタスクを同時に学習させる学習手法 [1] である．マルチタスク学習を行うことにより，主タスク単独で学習させるよりも主タスクの性能を向上させ，追加した複数のタスクの性能も向上させることができる．複数のタスクのうち，あるタスクを解くことに特化しても，そのタスクで得られた知識が別のタスクにそのまま流用できるとは限らない．モデル全体の性能を高めるには，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．マルチタスク学習では，タスクごとに設定し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．つまり，複数のタスクを最適化することにより，複数のタスクに共通した知識を獲得しようとしている．

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り，性能が向上すると示唆されている．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，テキスト中に肯定的，或いは否定的な単語が含まれるか否かを分類するというような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や実験者がタスクに関する知識をあまり持たない事例を考える．この場合，何が主タスクを解く上でのヒントとなり得るかを考えるのは大変であり，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てしまい，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，テキスト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ある．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におい

て特に重要な品詞である．重要品詞抽出タスクを追加することにより，入力されたテキスト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，入力されたテキストのどの部分が深層学習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，品詞タグ付け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が行えると考えた．

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，各層ごとに獲得している特徴量は異なるものだと考えた．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存在すると考えた．また，性質の異なるタスクを同じ層で学習させてしまうと複数のタスクに適した仮説関数が存在せず，性質の異なる複数のタスクを最適化しようとして競合してしまう場合が存在す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の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かつ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で高い適応性を誇る BERT[5] 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，形態素解析器の MeCab を用いて品詞タグの付与を行った．また，BERT を用いた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構築し，複数のデータセットを用いて学習させた．そして，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 Accuracy 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，Accuracy が最大で 0.045% 向上したが，Accuracy の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，Accuracy が 0.383% 向上し，Accuracy の変化に有意差が認められた．livedoor ニュースコーパスでは，Accuracy が最大で 1.756% 向上し，Accuracy の変化に有意差が認められた．この補助タスクは，特にクラス数の多い分類タスクの場合に Accuracy の向上に寄与すると分かった．また，補助タスクを BERT の 2 層

に追加した場合に，分類タスクの Accuracy が最も向上すると分かった．

さらに，BERT の Attention[6] の可視化を行い，マルチタスク学習モデルが注目している部分の変化を分析した．Attention とは, BERT や Vision Transformer*のような深層学習モデルに組み込まれた注意機構と呼ばれるもので，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ある．Attention の重みが大きい箇所は推論時に深層学習モデルが注目した部分であり，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 の可視化を行うと，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．

- BERT を用いたマルチタスク学習において，重要品詞抽出タスクはクラス数の多い分類タスクであるほど，Accuracy の向上に寄与することが明らかになった．
- 重要品詞抽出タスクを BERT の 2 層に追加した場合，分類タスクの Accuracy の向上に寄与することが明らかになった．
- 重要品詞抽出タスクを追加すると，モデルが注目する部分を変化させることができ，意味的なまとまりを認識できるように変化することが明らかになった．

本論文の構成は以下の通りである．2 章では，基本的事項について述べる．3 章では，関連研究について述べる．4 章では，提案手法について述べる．5 章では，評価実験について述べる．最後に 6 章では，本論文のまとめと今後の課題について述べる．

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

第 2 章　基本的事項

2.1　BERT

BERT(Bidirectional Encoder Representations from Transformers) とは，2018 年に Devlin らによって発表された Transformer Encoder を複数積み重ねた自然言語処理に特化したニューラル言語モデルである．従来のモデルでは「岐阜大学は柳戸にある。」という文がある場合，モデルが「岐阜→大学→は→柳戸→に→ある→。」という順方向の文脈しか考慮しなかった．しかし，BERT では「。→ある→に→柳戸→は→大学→岐阜」という逆方向の文脈も加えた双方向の文脈を考慮することで，深い言語理解ができるようになった．また，大規模な事前学習と Fine-tuning により，多くの自然言語処理タスクにおいて当時の最高性能を記録している．

事前学習とは，目的とするタスクを解くために学習させるのではなく，文章の構造のような基礎的な特徴を獲得させるための手法である．BERT の事前学習では，MLM（Masked Language Modeling）と NSP（Next Sentence Prediction）の二つのタスクを学習させている．MLM とは，入力された文章中の単語の中からランダムに 15% を [MASK] トークンに置き換え，[MASK] トークンに置き換えられる前の単語を予測するタスクである．例えば，「岐阜大学は柳戸にある。」という文が「岐阜大学は [MASK] にある。」と置き換えられた場合，元の文で [MASK] の部分にあった「柳戸」という単語を予測できるように学習させている．NSP では，同一文章に含まれる二つの文を結合したもの，ある文章 A に含まれる文と別の文章 B に含まれる文を結合したものをそれぞれ 50% ずつ用意する．その二つの文が同じ文章から生成されたものか，別の文章から生成されたものか予測し，二つの文に意味的な関係性があるか否かを分類するタスクである．これらのタスクは，予めラベルが付与されたテキストデータを用いず，モデル自身がテキストデータから教師ラベルを作成する自己教師あり学習によって学習されている．

Fine-tuning とは，学習済みモデルに目的のタスクを解くための新たな層を追加し，モデルの一部の重みを再学習する手法である．Fine-tuning により，目的のタスクに特化したモデルを構築することができる．また，モデルを一から訓練する必要がなく，学習時間の大幅な短縮や少量データでの学習が可能となる．

2.1.1 Attention

Attention[7] とは，前述の自然言語処理モデルの BERT や画像認識モデルの Vision Transformer に組み込まれた注意機構と呼ばれるものである．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ある．この仕組みを利用することで，入力されたデータのどの部分に注目するか，つまりデータの関係性や類似性を学習することができる．

ここでは，機械翻訳を例にして説明を行う．従来の機械翻訳モデルは，エンコーダとデコーダに RNN や LSTM を用いた Seq2Seq という系列変換モデルであった．RNN とは再帰ニューラルネットワークで，文字列や時系列データのような系列データを扱うことに優れている．LSTM とは RNN に入力ゲート，忘却ゲート，出力ゲートの三種類のゲート機構を組み合わせたニューラルネットワークモデルである．まず，エンコーダ部分では入力テキストの符号化を行う．単語を固定長のベクトルに変換し，逐次的に隠れ層の状態を更新していく．そうすることで，最終的に入力テキスト全体の意味を考慮した固定長のベクトルが作成される．その後，デコーダ部分で復号を行う．出力した単語と隠れ層の状態を用いて，順番に単語を出力して翻訳を行う．ただし，従来のモデルには入力テキストを単一の固定長ベクトルに変換するという問題点が存在する．そのため，入力テキストが長くなると，全体の意味を考慮した固定長ベクトルを表現することはできない．つまり，テキスト全体の意味を記憶しておくことは難しくなる．Bahdanau ら [7] によると，従来のモデルで機械翻訳を行った場合，20 単語以上のテキストになると翻訳性能が悪化すると報告されている．

この問題を解決するために開発されたのが，Attention である．従来のモデルでは，入力テキストを固定長ベクトルに変換して翻訳を行っていた．だが，固定長ベクトルへの変換がボトルネックであった．一方，Attention を用いた場合は，入力テキスト全体の意味を考慮するのに加えて，入力された単語のどこに注目するかを考慮しながら単語を出力している．単語を出力する際に，入力された単語付近の情報も離れた位置の情報も含めた，入力テキスト全体の情報を考慮することができる．つまり，出力時に，入力された情報をより直接的に参照することができる．こうすることで，より正確な翻訳ができるようになり，テキスト全体の意味の記憶が難しいという問題にも対処することができた．

2.1.2 Transformer

Transformer[6] とは，RNN や CNN のようなアーキテクチャを一切用いず，前述の Attention のみを用いた高度なニューラルネットワークアーキテクチャである．Transformer では，Scaled Dot-Product Attention，Multi-Head Attention，Self-Attention の三種類の Attention を組み合わせている．

Scaled Dot-Product Attention とは，クエリ Q とキー K の内積の計算結果を softmax 関数に代入し，代入結果とバリュー V の内積を計算するというものである．Q，K，V とはベクトルで，入力データのベクトルを三つに分解することで高い表現力を得ることができる．Scaled となっているのは，Q と K の内積を K の次元数の平方根の $\sqrt{d_k}$ で除算して標準化を行っているためである．

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1.1}$$

Multi-Head Attention とは，一つの Scaled Dot-Product Attention を計算するのではなく，h 個に分割してそれぞれで Attention を計算し，結合するというものである．複数の head を用意することで，単一の観点から見た類似度を用いるのではなく，異なる観点から見た類似度を用いて Attention を計算することができる．

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \cdots, head_h)W^O \\ where\ \ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.1.2}$$

Self-Attention とは，自己注意機構と呼ばれるものである．2.1.1 項で触れた機械翻訳モデルにおける Attention は，Source-Target Attention と呼ばれるものであった．機械翻訳を例にして説明すると，Source は翻訳前の単語のこと，Target は翻訳後の単語のことである．Source-Target Attention では，注目する部分を決めるために翻訳前と翻訳後の単語のように二つの系列データが必要であった．しかし，Self-Attention は Source-Target Attention とは異なり，自分自身に注目することができる．つまり，入力した系列データのある部分にどれだけ注目するかを計算する際に，入力した系列データ自身そのものを用いて計算するというものである．

従来の RNN では並列化できないこと，CNN では計算量の多さがボトルネックとなっていた．しかし，Transformer では行列の内積で計算できる Attention を用いることで，計算の高速化と並列化を行うことが可能となった．

2.2　マルチタスク学習

マルチタスク学習とは，主タスクと関係した複数のタスクを同時に学習させる学習手法である．複数のタスクの同時学習を行うことで，補助タスクで得られた知識を主タスクを解く際のヒントとして与えることができる．ヒントのある状態で主タスクを解くため，主タスク単独の学習時よりも主タスクの性能を向上させることができる．複数のタスクのうち，一部のタスクを解くことに特化しても，それらのタスクで得られた知識が別のタスクにそのまま流用できるとは限らない．モデル全体の性能を高めるには，特定のタスクに偏った知識ではなく，複数のタスク間で共通した普遍的な知識を獲得しなければならない．マルチタスク学習では，タスクごとに設定した損失関数の重み付き和をモデル全体の損失関数とする．複数のタスクを最適化することで，タスク間で共通した知識を獲得することができる．

マルチタスク学習と似た技術に転移学習 [8] というものがある．転移学習とは，あるドメインで学習されたモデルを別のドメインに適用させる学習手法である．類似点は，あるタスクで獲得した知識を別のタスクに転用することである．相違点は，転移学習の知識を転用する経路は一方向である．タスク A で得た知識をタスク B に転用できるが，タスク B で得た知識をタスク A に転用できない．一方，マルチタスク学習の知識を転用する経路は双方向である．タスク A で得た知識をタスク B に転用するだけでなく，タスク B で得た知識もタスク A に転用できる．

ニューラルネットワークにおけるマルチタスク学習には，ハードパラメータ共有 [1] とソフトパラメータ共有 [9] が存在する．前者は，ベースとなるモデルは一つであり，タスクの数に応じて出力層を増やす手法である．モデルの入力層付近では，アーキテクチャが複数のタスク間で共有されているが，出力層付近ではタスクごとに枝分かれしている．後者は，各タスクそれぞれにベースとなるモデルがあり，モデル同士を並列に接続する．各タスクの各モデル間のパラメータに対して $L2$ 距離 [10] やトレースノルム [11] を用いた正則化を行い，パラメータの更新に制約を設ける手法である．関係するタスクのパラメータは近傍に存在するであろうという考え方を基に，互いのパラメータが近傍から逸脱しないように束縛している．ソフトパラメータ共有は，タスク間でモデルを共有せず，タスクごとにパラメータが異なるので自由度が高い．しかし，モデルを並列に接続するためモデルの構築が複雑であったり，モデルサイズが大きくなるといった技術的な問題も存在する．

2.3 重要品詞抽出タスク

品詞タグ付けとは，テキスト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のような品詞を予測するタスクである．例えば，「岐阜大学は柳戸にある。」という文に対して品詞タグ付けを行うと，以下のようになる．

岐阜	大学	は	柳戸	に	ある	。
名詞	名詞	助詞	名詞	助詞	動詞	記号

品詞タグ付けでは，正しい品詞タグを予測するだけではなく，品詞のまとまりであると推定した文字列位置も正解している必要がある．この例文において，「柳戸」ではなく「柳」のみを名詞であると予測した場合は，正しい品詞タグを予測しているが，誤った文字列位置を推定しているため不正解である．

日本語には，名詞，副詞，形容詞，形容動詞，動詞，助詞，助動詞，接続詞，連体詞，感動詞の十種類の品詞が存在している．それらは，名詞のように意味を表す内容語，助詞のように文法的な機能を持つ機能語の二つに分けられる．内容語は意味を表すため，文章の内容を特徴付ける品詞であると考えられる．本研究では，名詞，副詞，及び形容詞を抽出するタスクを重要品詞抽出タスクと呼称する．また，形態素解析器の MeCab を用い，IOB2 表現に倣って品詞タグを付与する．

IOB2 表現とは，Tjong Kim Sang ら [12] によって開発された，トークンのチャンク同定問題におけるトークンに対するラベルの付与方法である．トークンとは，単語やサブワードのような小さい処理単位のことである．チャンクとは，複数のトークンから構成され，複数のトークンを一つにまとめた処理単位のことである．タスクで使用するチャンクが単一のトークンのみで構成される場合，トークンに Begin を意味する B タグを付与する．タスクで使用するチャンクが複数のトークンから構成される場合，最初のトークンには B タグ，残りのトークンには Inside を意味する I タグを付与する．タスクで使用しないチャンクには，Outside を意味する O タグを付与する．例えば，「岐阜大学は柳戸にある。」という文に対して IOB2 表現を用いて重要品詞抽出タスクを行うと，以下のようになる．

岐阜	大学	は	柳戸	に	ある	。
B-名詞	I-名詞	O	B-名詞	O	O	O

2.4　対応あり 2 標本 t 検定

対応あり 2 標本 t 検定とは，ある母集団から抽出された二群間の標本平均に差があるか否かを判断する仮説検定である．対応ありとは，二群間に何らかの対応関係があることを意味する．今回は，二つのモデルを用いて同一のテキストデータ群に対して推論を行い，結果を比較するため，対応あり 2 標本 t 検定を採用した．

次に，仮説検定を行う方法を説明する．

① 帰無仮説 H_0 と対立仮説 H_1 を設定する．

帰無仮説 H_0 とは，二群間に有意差はないという仮説である．対立仮説 H_1 とは，二群間に有意差はあるという仮説である．帰無仮説と対立仮説は相反する仮説で，帰無仮説が決まると自動的に対立仮説も決まる．今回の場合，帰無仮説と対立仮説は以下のようになる．

帰無仮説 H_0：二つのモデル間における Accuracy の平均値に差がない
対立仮説 H_1：二つのモデル間における Accuracy の平均値に差がある

② 有意水準 α を設定する．

有意水準 α とは，帰無仮説を棄却する基準である．一般に，$\alpha = 0.05$，或いは $\alpha = 0.01$ が用いられる．今回は，$\alpha = 0.05$ とする．

③ p 値と検定統計量 T を算出する．

p 値とは，帰無仮説が成立するという仮定の下で、検定統計量 T がある値を取る確率のことである．t 分布表を用いて求めることができる．検定統計量 T は，以下の式で導出することができる．ここで，$\overline{d}$ とは二群間の差の平均，s^2 とは不偏分散，n とは自由度である．

$$T = \frac{\overline{d}}{\sqrt{\frac{s^2}{n}}} \tag{2.4.1}$$

④ p 値と有意水準 α を比較して，両側検定を行う．

算出した p 値と有意水準 α を比較する．p 値が有意水準 α を上回る場合，帰無仮説が採用される．一方，p 値が有意水準 α を下回る場合，帰無仮説が棄却され，対立仮説が採用される．

2.5　評価指標

本論文では，深層学習モデルの性能を測定するために Accuracy と F-score の二つの評価指標を用いる．F-score を導出するために Precision，Recall が必要となるので，こちらも合わせて説明する．説明にあたり，表 2.1 の混同行列を用いる．表中の True Positive とは，陽性ラベルが付与されたものを正しく陽性と予測したものである．False Negative とは，陽性ラベルが付与されたものを誤って陰性と予測したものである．False Positive とは，陰性ラベルが付与されたものを誤って陽性と予測したものである．True Negative とは，陰性ラベルが付与されたものを正しく陰性と予測したものである．

2.5.1　Accuracy

Accuracy とは正解率であり，モデルが予測したラベルのうち，正解ラベルと一致していたものの割合である．予測がどれだけ当たっていたかを測定したい場合，Accuracy を評価指標として採用する．ただし，ラベル間に極端な偏りがある場合，Accuracy のみでモデルの性能を判断することは危険である．例えば，陽性ラベルのデータが 90 件，陰性ラベルのデータが 10 件の場合を考える．この場合，陽性しか予測しないモデルであっても Accuracy は 0.90 と高い値を示してしまう．表 2.1 を用いて説明すると，以下の式で導出することができる．

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.5.1}$$

表 2.1　評価指標のための混同行列

		予測ラベル	
		陽性	陰性
正解ラベル	陽性	True Positive（TP，真陽性）	False Negative（FN，偽陰性）
	陰性	False Positive（FP，偽陽性）	True Negative（TN，真陰性）

2.5.2 Precision

Precision とは適合率であり，モデルが陽性であると予測したもののうち，実際の正解ラベルが陽性であるものの割合である．予測の誤りを少なくすることを重要視する場合，Precision を評価指標として採用する．表 2.1 を用いて説明すると，以下の式で導出することができる．

$$Precision = \frac{TP}{TP + FP} \tag{2.5.2}$$

2.5.3 Recall

Recall とは再現率であり，陽性ラベルが付与されたもののうち，モデルが陽性ラベルであると予測できたものの割合である．予測の漏れを少なくすることを重要視する場合，Recall を評価指標として採用する．表 2.1 を用いて説明すると，以下の式で導出することができる．

$$Recall = \frac{TP}{TP + FN} \tag{2.5.3}$$

2.5.4 F-score

F-score とは Precision と Recall の調和平均である．Precision を高くするには，確実に陽性であるもののみを予測すれば良い．一方，Recall を高くするには，全てを陽性であると予測すれば良い．ただし，Precision か Recall のいずれか一方の評価指標が高くても，F-score が高くなることなはい．F-score を高くするには，両方の評価指標がバランス良く高い値である必要がある．そのため，両方の評価指標を重要視する場合，F-score を評価指標として採用する．表 2.1 を用いて説明すると，以下の式で導出することができる．

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.5.4}$$

第 3 章　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主タスクの性能向上に有効であるかについての研究はいくつか行われている．また，補助タスクの学習に関して，補助タスクをどの階層で学習させるのが良いかについての研究もいくつか行われている．

Wu ら [13] は，医療に関する質問の意図分類タスクにおいて，固有表現抽出タスクを同時に学習させることで性能が向上することを示している．この研究では，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 層と Attention 層を組み合わせたマルチタスク学習モデルを提案している．複数の医師によって病名，治療法，体の部位などの 6 項目に対してアノテーションを行い，固有表現抽出用のデータセットを作成した．提案されたモデルを用いることにより，固有表現抽出タスクにおける F-score と Recall では BERT と同程度の性能，Precision では BERT 以上の性能が報告されている．また，意図分類タスクにおける全ての評価指標で BERT を上回る結果が報告されている．

Benayas ら [14] 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タスクを同時に学習させることで両方のタスクの性能が向上することを示している．この研究では，BERT や RoBERTa といった Transformer をベースとしたハードパラメータ共有のモデルを提案している．また，ハードパラメータ共有にソフトパラメータ共有を融合させたモデルも提案している．この融合モデルでは，意図分類タスクと固有表現抽出タスクで得られたパラメータを，アダマール積や行列積を用いて両方のタスクを考慮したパラメータに変換する機構が組み込まれている．

Søgaard ら [15] は，品詞タグ付けタスクのような下位タスクを同時に学習させる場合，出力層付近で学習させるよりも入力層付近で学習させる方が良いことを示している．この研究では，主タスクをチャンキング，或いは CCG スーパータグ付けタスクとし，補助タスクを品詞タグ付けタスクとした双方向 LSTM を用いた 3 層のマルチタスク学習モデルを提案している．補助タスクを主タスクと同じ 3 層で学習させた場合でも主タスクの F-score は向上するが，補助タスクを 1 層で学習させ

た場合は 3 層で学習させた場合より F-score が向上することが報告されている．

Sanh ら [16] は，複数の自然言語処理タスクを組み合わせたモデルを構築し，タスク間に階層関係があることを示している．この研究では，1 層目では固有表現抽出タスク，2 層目では言及抽出タスク，3 層目では共参照解析タスクと関係抽出タスクを解くように双方向 LSTM を積み上げたマルチタスク学習モデルを提案している．固有表現抽出タスクや言及抽出タスクのようなタスクは，深い言語理解を必要としない下位タスクであるので，モデルの入力層付近で学習させるようにした．一方，共参照解析タスクや関係抽出タスクのようなタスクは，深い言語理解を必要とする上位タスクであるので，モデルの出力層付近で学習させるようにした．下位タスクを入力層付近，上位タスクを出力層付近で学習させることによって良い中間表現が得られ，複数のタスクで性能の向上が見られたと報告されている．

先行研究 [13][14] より，分類タスクにおいて固有表現抽出タスクは有効な補助タスクであると知られている．そのため，重要品詞抽出タスクも固有表現抽出タスクと同じ系列ラベリング問題であり，分類タスクの性能向上に有効なタスクとなるのではないかと考えた．先行研究との類似点は，系列ラベリング問題に属するタスクを補助タスクとして用いている点，品詞情報を考慮した補助タスクを用いている点である．一方，先行研究との相違点は，BERT を用いたマルチタスク学習での分類タスクにおいて，重要品詞抽出タスクという新しい補助タスクを用いている点，補助タスクを学習させるのに適した層を探索している点である．

第 4 章　提案手法

本章では，使用するデータセット，ラベル付けの方法について述べる．また，使用する深層学習モデル，マルチタスク学習モデルとシングルタスク学習モデルの作成方法についても述べる．

4.1 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．本研究では，三種類の日本語のデータセットを使用した．4.2 節では，使用するテキストデータの前処理，ラベル付けについて述べる．補助タスクで使用する品詞タグの付与は，人手による作業を一切行わず，機械のみを用いて作業を行った．4.3 節では，トークナイザについて述べる．重要品詞抽出タスクを解くためには，日本語用の BERT のトークナイザでは対応できない場合が考えられる．そのため，既存のトークナイザに対して改良を行った．4.4 節では，モデルの作成方法にについて述べる．マルチタスク学習モデル，シングルタスク学習モデルの二種類のモデルを作成した．マルチタスク学習モデルは，ハードパラメータ共有とソフトパラメータ共有の二つに大別される．今回は，モデルの構築が容易なハードパラメータ共有のマルチタスク学習モデルのみ構築し，実験を行った．

4.1　データセット

本研究では，BERT を用いたマルチタスク学習での分類タスクにおいて，重要品詞抽出タスクが主タスクの性能向上に有効で，汎用的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，似たようなジャンルや領域のテキストデータが含まれたデータセットを使用するべきではない．そこで，ドメインが異なる以下の複数のデータセットを用いた．

4.1.1 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* 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によって投票が行われた．投票の選択肢には，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用意されている．集約された投票結果をもとに，ツイートに対して五種類のラベルのいずれかが付与されている．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ている．

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，投票結果によっては，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．例えば，五人のクラウドワーカーによりポジネガに二票，ポジティブに二票，ニュートラルに一票が投票された場合を考える．この時，ポジネガラベルとポジティブラベルの二種類のラベルが採用されてしまう．そのため，採用されたラベルが一意に定まるもののみ使用した．ただし，採用されたラベルが総投票数の過半数を超えているとは限らない．

4.1.2 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† 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や約 3 億件の商品情報が含まれる楽天市場，約 80 万件のレシピ情報が含まれる楽天レシピ，約 700 万件のレビューや約 3 万件の施設情報が含まれる楽天トラベル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，2015 年 1 月から 12 月に投稿されたレビューで食品カテゴリに該当するものを使用した．使用する他のデータセットが，電子機器に関するレビューやニュース記事といった特定のジャ

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

†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

ンルで構築されているため，使用するカテゴリは食品カテゴリに限定している．また，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うようにし，評価値が 3 のものは使用していない．

4.1.3 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[‡]である．NHN Japan 株式会社が運営する livedoor ニュースの URL や日付，タイトルや本文を収集したものである．

このデータセットには，トピックニュースや IT ライフハック，MOVIE ENTER といった九つのカテゴリが存在する．また，各カテゴリには 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．

4.2 前処理，ラベル付け

4.1 節で述べた複数のデータセットに対し，テキストデータの前処理を行い，主タスクの分類タスクで使用するラベルの付与を行う．その後，補助タスクの重要品詞抽出タスクで使用する品詞タグの付与を行う．

4.2.1 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った．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った．

さらに，前処理の後に字面が同じテキストデータが複数出現する場合が考えられる．字面は全く同じであるのに，付与されたラベルが異なるテキストデータが存在してしまう可能性がある．そのため，重複したテキストデータは除去を行い，使用しないようにした．

[‡] https://www.rondhuit.com/download.html#ldcc

4.2.2　主タスクに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，前処理後のテキストデータ集合に対してランダムでシャッフルを行った．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った．ただし，BERT で扱え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を行った．また，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では，Twitter の仕様で 140 文字を超えるツイートは存在しないため，文字数を調整する処理を行っていない．

Twitter 日本語評判分析データセットでは，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．各クラスのインスタンス数は 10,000 件で統一した．今回の実験で使用した全データは，三種類のクラス × 10,000 件の合わせて 30,000 件である．

楽天データセットでは，ポジティブ，ネガティブの二種類のクラスを用いた．各クラスのインスタンス数は 20,000 件で統一した．今回の実験で使用した全データは，二種類のクラス × 20,000 件の合わせて 40,000 件である．

livedoor ニュースコーパスでは，トピックニュースや IT ライフハック，MOVIE ENTER といった九種類のクラスを用いた．各クラスのインスタンス数は 500 件で統一した．今回の実験で使用した全データは，九種類のクラス × 500 件の合わせて 4,500 件である．

4.2.3　補助タスクに使用する品詞タグの付与

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．一般に，機械学習で扱う何らかのデータに対してラベル付けを行う場合，人手による作業と機械を用いた作業の二通りの方法が考えられる．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，デメリットとしてデータセットの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，MeCab 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った．また，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[§]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している．さらに，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している．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，内容語に属する名詞，副詞，及び形容詞のみ分類するタスクを設計した．

Inside の I，Outside の O，Begin の B を意味するタグを用いる IOB2 形式に倣い，品詞タグの付与を行った．実験に用いる三種類の品詞には，B-名詞，I-名詞，B-副詞のように B タグ，及び I タグの付与を行った．それ以外の品詞には，O タグの付与を行った．詳細は，2 章の基本的事項にある 2.3 節の重要品詞抽出タスクを参照されたい．

4.3 トークナイザ

トークナイザとは，文字情報のような数値情報でないデータを深層学習モデルで扱うために，数字に変換するモジュールである．BERT の既存のトークナイザ[¶]では，IPA 辞書を用いて形態素解析器の MeCab で単語分割を行う．その後に，WordPiece[17] のアルゴリズムを用いてサブワードに分割する仕様である．BERT の既存のトークナイザの語彙サイズは 32,000 である．サブワードとは，単語よりも小さな処理単位のことである．WordPiece とは，出現頻度が高い文字は一文字として扱い，出現頻度が低い文字はサブワードとして扱うアルゴリズムである．

重要品詞抽出タスクのような系列ラベリング問題を解くためには，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トークンに分かれず，品詞タグの予測がうまくできない可能性が考えられる．例えば，「岐阜大学は」というテキストから「岐阜大学」という名詞を抽出した

[§]https://github.com/neologd/mecab-ipadic-neologd/blob/master/README.ja.md
[¶]https://github.com/cl-tohoku/bert-japanese/blob/main/tokenization.py

い場合，トークナイザによって「岐阜」,「大学は」のようにトークン化されてしまうと，正しく重要品詞抽出タスクを行うことができない．そこで，B タグ，I タグを付与した品詞のかたまりだけ先にトークン化処理を行う．その後，残りの O タグを付与した部分に対してトークン化処理を行ったのち，トークンを結合するようにトークナイザを改良した．つまり,「岐阜大学は」というテキストであれば，名詞である「岐阜大学」の部分だけ先にトークン化を行ったのち，残りの「は」の部分のトークン化を行った．

改良後のトークナイザを用いた場合，4.1.1 項のデータセットでは 30,000 件のテキストのうち，22.3% のテキストで既存のトークナイザによるトークン化と異なる結果となった．4.1.2 項のデータセットでは 40,000 件のテキストのうち，36.2% のテキストで異なる結果となった．4.1.3 項のデータセットでは 4,500 件のテキストのうち，14.0% のテキストで異なる結果となった．

4.4 モデル作成

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 を使用した．BERT の事前学習済みモデルは，東北大学の BERT-base モデル[‖] を利用した．このモデルに対してファインチューニングを行い，実験を行った．この BERT-base モデルは，Transformer Encoder を 12 層積み重ねたモデルである．

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あるモデルである．このモデルは出力が一つであるので，主タスクのみ解くことができる．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あるモデルである．このモデルは出力が複数あるので，主タスクに加えて補助タスクも解くことができる．

まず，マルチタスク学習モデルの作成方法を述べる．例えば，BERT の 12 層目で主タスクを解き，1 層目で補助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，主タスクは m 種類のクラスの分類タスク，補助タスクは n 種類の重要品詞抽出タスクである．はじめに，事前学習済み BERT モデルの 1 層と 12 層のパラメータを訓練可能にする．そして，1 層目にドロップアウト層と全結合

[‖] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

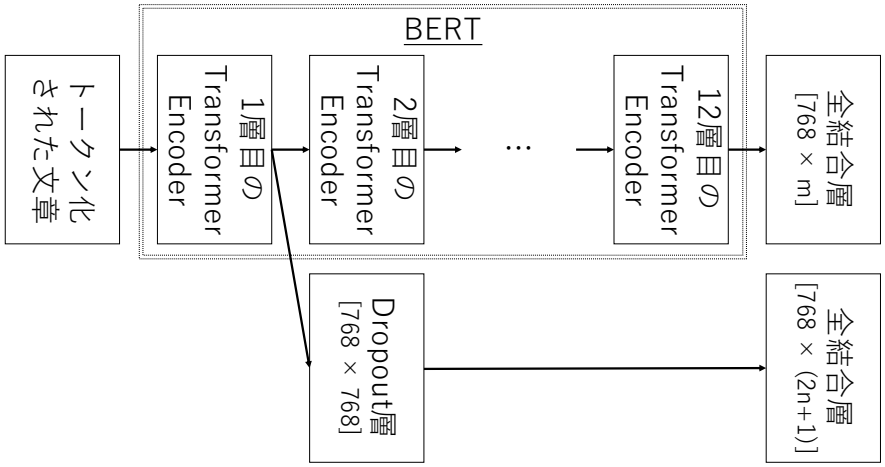

図 4.1　ハードパラメータ共有のマルチタスク学習モデル

層，12 層目に全結合層を追加する．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．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．品詞タグは IOB2 表現でラベル付けされており，n 種類の品詞それぞれに B タグと I タグを用いているため，$2 \times n$ 種類のタグ，及び O タグが必要となる．そのため，補助タスクを解く下側の全結合層の出力層が $2n+1$ 次元である．また，系列ラベリング問題を解くための BertForSequenceClassification モデルにおいて，ドロップアウト層が追加されていたため，本研究でも同様に追加した．BertForSequenceClassification モデルの詳細な構造については，こちらのサイト**を参考にされたい．ハードパラメータ共有のマルチタスク学習モデルの模式図を，図 4.1 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．

次に，シングルタスク学習モデルの作成方法を述べる．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においてパラメータを訓練可能にした層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

**https://huggingface.co/transformers/v3.5.1/_modules/transformers/modeling_bert.html

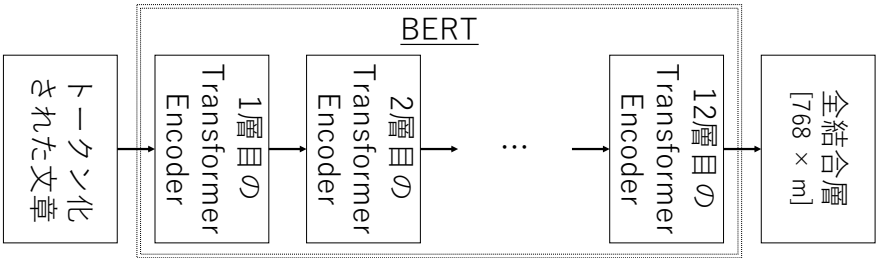

図 4.2　シングルタスク学習モデル

デルの 1 層と 12 層のパラメータを訓練可能にする．そして，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の模式図を，図 4.2 に示した．

シングルタスク学習モデルでは，1 層と 12 層のパラメータを訓練可能にしている．これは，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有意性を検証する際に，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．

第 5 章　評価実験

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での分類タスクにおいて，重要品詞抽出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検証することである．二つ目は，重要品詞抽出タスクを学習させるのに適した層が存在するのかを調べることである．

5.1　実験手順

まず，4.2 節で述べた方法でラベル付けを行い，k-分割交差検証を行うために全てのテキストデータを k 個に分割する．今回は，$k = 10$ の 10 分割交差検証を行った．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，4.4 節で述べた方法で，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．使用した BERT-base モデルは，Transformer Encoder を 12 層積み重ねたモデルである．今回は，主タスクを解く層は 12 層目で固定化し，補助タスクをどの層に追加するか探索する．そのため，訓練可能なパラメータの組み合わせは 12 通りある．また，作成するモデルは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の二種類ある．そのため，12 通り × 二種類の合わせて 24 個のモデルを作成した．ハイパーパラメータや損失関数のようなモデルの詳細な設定については，次の 5.2 節で述べる．

最後に，同一の訓練用，検証用データで二つのモデルの学習を行い，同一のテスト用データで推論を行う．主タスクの推論結果から Accuracy を算出し，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間の Accuracy を比較する．その際，補助タスクの追加による Accuracy 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．

5.2　実験条件

今回の実験では，BERT で扱えるトークンの最大長は 512 とし，バッチサイズは 32 とした．モデルのハイパーパラメータに関して，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} とした．使用した optimizer は Adam で，学習率の減衰やウォームアップのような学習率スケジューリングは行っていない．

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．

$$L_{all} = \lambda_{main} \times L_{main} + \lambda_{sub} \times L_{sub} \tag{5.2.1}$$

マルチタスク学習では，λ の値を変えてタスク間の Loss の重み付けを行い，どのタスクをどの程度重要視するかを調整することができる．例えば，Kendal ら [18] が Homoscedastic Uncertainty に基づいたタスク間の Loss 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．そのため，設定した条件よりも適したタスク間の Loss の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．Early Stopping とは，検証用データを用いてモデルの性能評価を行い，改善が見られない場合は学習を停止し，検証用データの Loss が最小値を記録した時点のパラメータを選択する手法である．検証用データでの Loss が増加する前のパラメータを用いることで，テスト用データに対するモデルの性能を良くする可能性がある．今回は，検証用データの Loss の最小値が 10epoch 更新しなかったら学習を停止し，検証用データの Loss が最小値を記録した時点のモデルを保存した．予備実験にて，検証用データの Loss の最小値が 100epoch 更新しなかったら学習を停止するように設定して実験を行った．しかし，10epoch の場合の実験結果と変わらなかったため，10epoch で十分であると判断した．

5.3 実験結果，考察

5.3 節では各データセットでの実験結果を示す．その後，テスト用データの推論を行った際に予測を誤った以下の二つの場合について，BERT の Attention の重みを可視化することで要因を分析する．

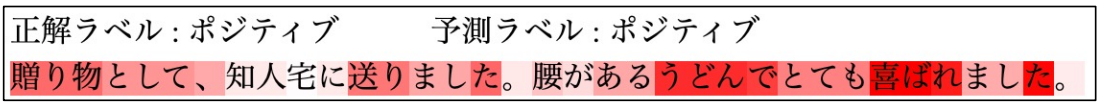

図 5.1　Attention の可視化の例

① マルチタスク学習モデルでは予測できたが，シングルタスク学習モデルでは予測できなかったもの
② マルチタスク学習モデルでは予測できなかったが，シングルタスク学習モデルでは予測できたもの

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．12 個の Multi Head Attention で得られる Attention の重みの相加平均をとる．この Attention の重みの相加平均を w とする．Attention の重みの相加平均をとることで，重みの相加平均 w は 0 から 1 の範囲にスケーリングされる．その後，以下の式を用いて，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．

$$R = 255,\ G = 255 \times (1 - w),\ B = 255 \times (1 - w) \tag{5.3.1}$$

Attention の可視化を行った例を図 5.1 に示す．Attention の重みの相加平均 w が 1 である場合，RGB は (255，0，0) で表現され，赤色を示す．図 5.1 の「喜ばれ」や「た」の部分は Attention の重みの値が大きいため，濃い赤色で示されている．一方，Attention の重みの相加平均 w が 0 である場合，RGB は (255，255，255) で表現され，白色を示す．図 5.1 の「知人」や「宅」の部分は Attention の重みの値が小さいため，薄い赤色や白色で示されている．

5.3.1　Twitter 日本語評判分析データセット

4.1.1 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 5.2，表 5.1 に示す．表中の太字は，シングルタスク学習モデルの Accuracy よりマルチタスク学習モデルの Accuracy が高く，p 値が有意水準 0.05 を下回り，有意差が認められたものを示す．表中の下線は，シングルタスク学習モデルの Accuracy よりマルチタスク学習モデルの Accuracy が高く，p

値が有意水準 0.05 を下回らず，有意差が認められなかったものを示す．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 Accuracy が高くなった．

表 5.1 の検定結果より，補助タスクを 2 層，6 層，7 層に追加した場合に 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，補助タスクを追加したマルチタスク学習により Accuracy が向上したと言える．また，2 層，6 層，7 層に補助タスクを追加した場合において，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，Accuracy がそれぞれ 0.383%，0.337%，0.373% 上昇した．Accuracy が最も向上したのは補助タスクを 2 層に追加した場合で，0.383% の向上が見られた．これらのことから，補助タスクを学習させるのに適した層は 2 層であると言える．この帰結は，品詞タグ付けタスク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 [15] での知見とも一致する結果となった．

Accuracy に変化が見られた要因を分析するために Attention の可視化を行った．可視化に使用したデータは，Accuracy が最も良くなった補助タスクを 2 層に追加

表 5.1　5.3.1 項の実験結果における Accuracy の平均値と p 値

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

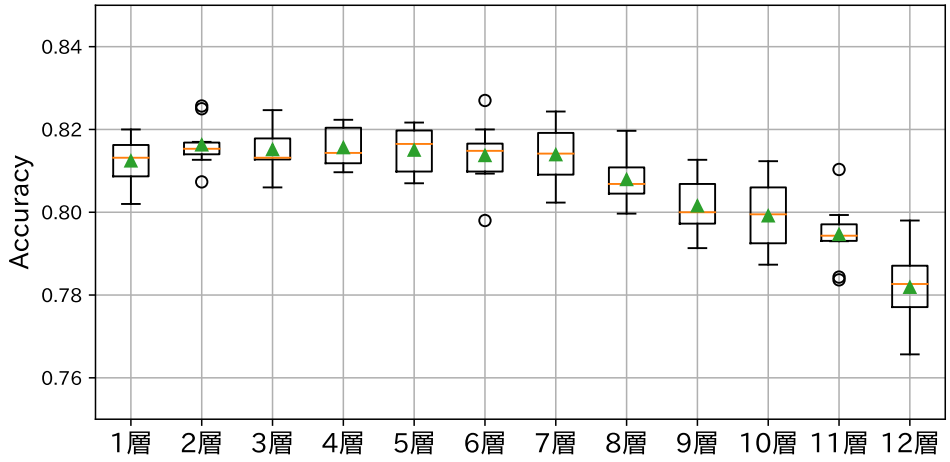

図 5.2　5.3.1 項の実験結果における補助タスク追加層の違いによる Accuracy の比較

正解ラベル : ポジティブ　　予測ラベル : ネガティブ　　シングルタスク学習モデル ただiphoneは故障した時にお手軽修理出来ないからキライなだけです、ソフトもハードも。xperiaも修理出来ないけど壊れにくいし、galaxyは工具さえ買えば修理できるし。 正解ラベル : ポジティブ　　予測ラベル : ポジティブ　　マルチタスク学習モデル ただiphoneは故障した時にお手軽修理出来ないからキライなだけです、ソフトもハードも。xperiaも修理出来ないけど壊れにくいし、galaxyは工具さえ買えば修理できるし。
正解ラベル : ネガティブ　　予測ラベル : ポジティブ　　シングルタスク学習モデル xperiaz0動作が遅いtt機種変更前の方が動きが良かったとか。。。 正解ラベル : ネガティブ　　予測ラベル : ネガティブ　　マルチタスク学習モデル xperiaz0動作が遅いtt機種変更前の方が動きが良かったとか。。。

図 5.3　5.3.1 項の実験結果におけるマルチタスク学習モデルで予測できた例

したマルチタスク学習モデル，比較対象となる 2 層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 の場合の一例を図 5.3 に示した．上段は，ポジティブラベルが付与されたテキスト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．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」に注目した．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

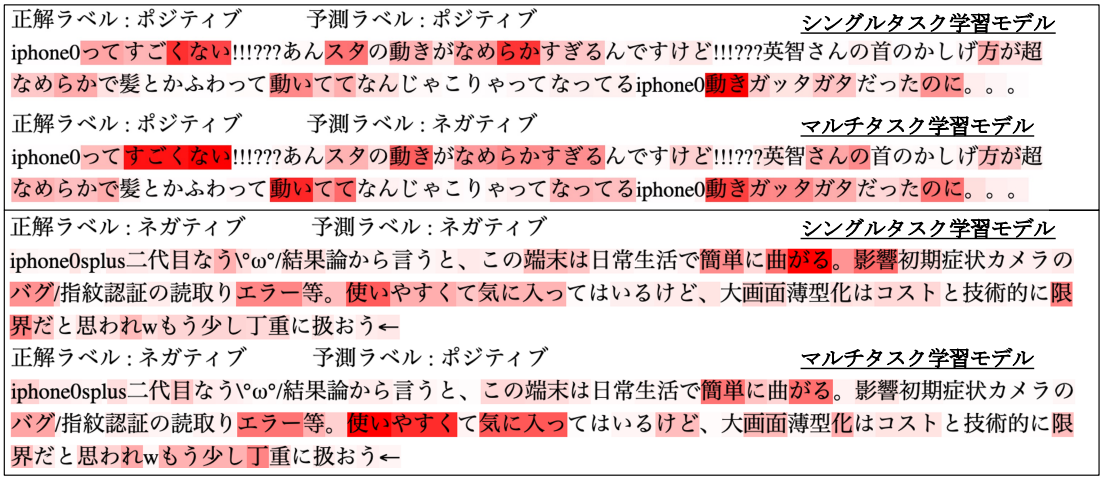

図 5.4　5.3.1 項の実験結果におけるマルチタスク学習モデルで予測できなかった例

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」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テキスト中の肯定的な意味を認識することで，ポジティブクラスと正しく予測できるようになったと考えられる．

下段は，ネガティブラベルが付与されたテキスト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，肯定的な意味の「動きが良かっ」に注目しているが，否定的な意味の「動作が遅い」にはあまり注目していない．一方，マルチタスク学習モデルでは「動きが良かっ」の部分にかかっている Attention の重みは小さくなり，「動作が遅い」の部分に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あまり注目していなかった「機種変更前」の部分に対する Attention の重みが，マルチタスク学習モデルでは僅かではあるが大きくなっている．これにより「動きが良かっ」のみでは肯定的な意味だが，「機種変更前の方が動きが良かっ」となることで否定的な意味になる．マルチタスク学習モデルではテキスト中の否定的な意味を認識することで，ネガティブクラスと正しく予測できるようになったと考えられる．

② の場合の一例を図 5.4 に示した．上段は，ポジティブラベルが付与されたテキストをシングル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，マルチ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たものである．マルチタスク学習モデルでは，「すごくない」と「ガッタガタ」の部分に注目するように変化した．高木 [19] によると，「すごくない!!!???」のように「否定形 + 疑問形」で表される表現は同意表現と呼ばれ，話し手が聞き手に対して発言の同意を求めるものであり，否定的な意味を喪失している．マルチタスク学習モデルでは，「すごくない」の部分を意味的なまとまりとして認識できるようになった．しかし，否定的な意味を喪失した同意表現ではなく否定的な意味と認識されたため，ネガティブクラスと誤って予測されたと考えられる．

下段は，ネガティブラベルが付与されたテキスト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てしまったと考えられる．

以上のことから，重要品詞抽出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，重要品詞抽出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 Accuracy 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てしまった場合もいくつか確認された．

5.3.2 楽天データセット

4.1.2 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 5.5，表 5.2 に示す．表中の二重下線は，シングルタスク学習モデルの Accuracy よりマルチタスク学習モデルの Accuracy が高く，p 値が有意水準 0.5 を下回り，有意差が認められたものを示す．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 Accuracy

が低くなった．10 層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 Accuracy が 0.045% 上昇した．

しかし，表 5.2 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の Accuracy の変化に有意差は見られなかった．つまり，補助タスクを追加したマルチタスク学習により Accuracy が向上も低下もしなかったと言える．

Accuracy に変化が見られなかった要因を分析するために Attention の可視化を行った．可視化に使用したデータは，先ほどのデータセットと同じく補助タスクを 2 層に追加したマルチタスク学習モデル，2 層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 の場合の一例を図 5.6 に示した．上段は，ポジティブラベルが付与されたテキスト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．上段の例に関して，シングルタスク学習モデルでは「残念」の部分に注目している．一方，マル

表 5.2　5.3.2 項の実験結果における Accuracy の平均値と p 値

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

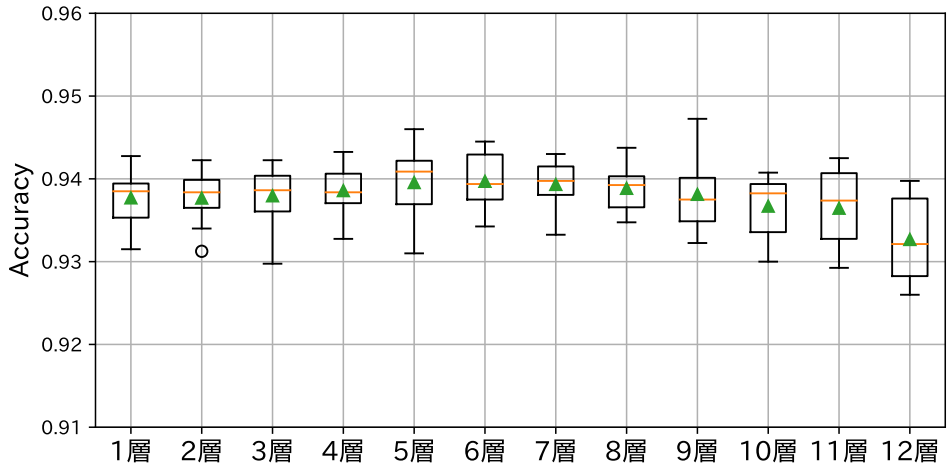

図 5.5　5.3.2 項の実験結果における補助タスク追加層の違いによる Accuracy の比較

正解ラベル : ポジティブ　　予測ラベル : ネガティブ 何より安いし、梱包もよかったです。少し白いお米が多くて残念でした。	シングルタスク学習モデル
正解ラベル : ポジティブ　　予測ラベル : ポジティブ 何より安いし、梱包もよかったです。少し白いお米が多くて残念でした。	マルチタスク学習モデル
正解ラベル : ネガティブ　　予測ラベル : ポジティブ サラダ等にのせて食べました。味はまあまあ。値段の割にはカニの身が寂しい感じはした。	シングルタスク学習モデル
正解ラベル : ネガティブ　　予測ラベル : ネガティブ サラダ等にのせて食べました。味はまあまあ。値段の割にはカニの身が寂しい感じはした。	マルチタスク学習モデル

図 5.6　5.3.2 項の実験結果におけるマルチタスク学習モデルで予測できた例

チタスク学習モデルでは，「安く」や「よかった」の部分にも注目するように変化した．

下段は，ネガティブラベルが付与されたテキスト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．下段の例に関して，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た．注目すべき場所をヒントとして与えることで，正しく予測できるようになったと考え

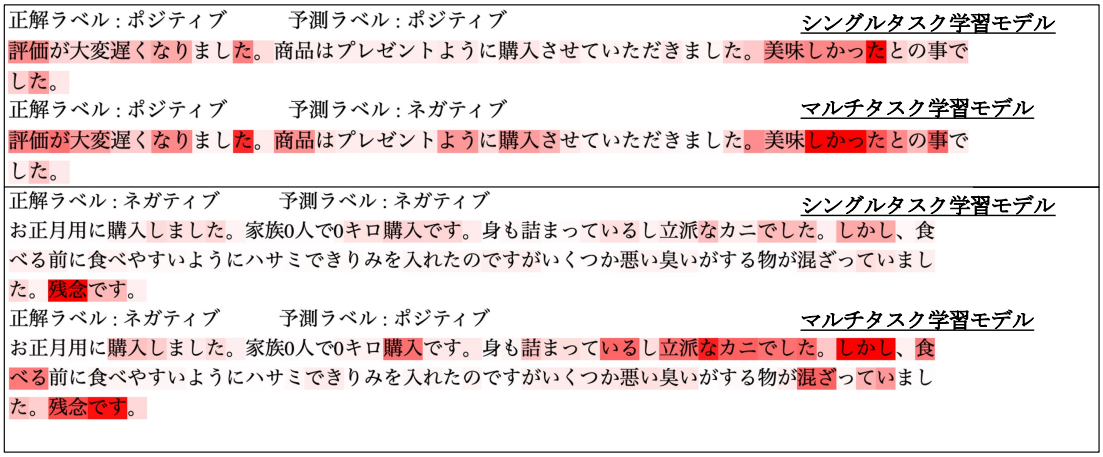

図 5.7　5.3.2 項の実験結果におけるマルチタスク学習モデルで予測できなかった例

られる．

② の場合の一例を図 5.7 に示した．上段は，ポジティブラベルが付与されたテキストをシングル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，マルチ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たものである．上段の例に関して，どちらのモデルも文章全体に満遍なく注目している．マルチタスク学習モデルでは品詞を推定する補助タスクを学習させることで，名詞の「商品」,「購入」,「事」の部分にも注目するように変化した．シングルタスク学習モデルでは正しく予測できたが，マルチタスク学習モデルでは誤った予測を行った．

下段は，ネガティブラベルが付与されたテキスト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．下段の例に関して，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」や「しかし」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 Accuracy 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ことが考えられる．このデータセットでは，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」のようにポジティブな内容とネガティブな内容の両方の表現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，ポジティブな内容とネガティブな内容の両方の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 Accuracy の向上に寄与するかどうかを再度調査する必要がある．

5.3.3 livedoor ニュースコーパス

4.1.3 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 5.8，表 5.3 に示す．表中の太字は，シングルタスク学習モデルの

表 5.3　5.3.3 項の実験結果における Accuracy の平均値と p 値

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

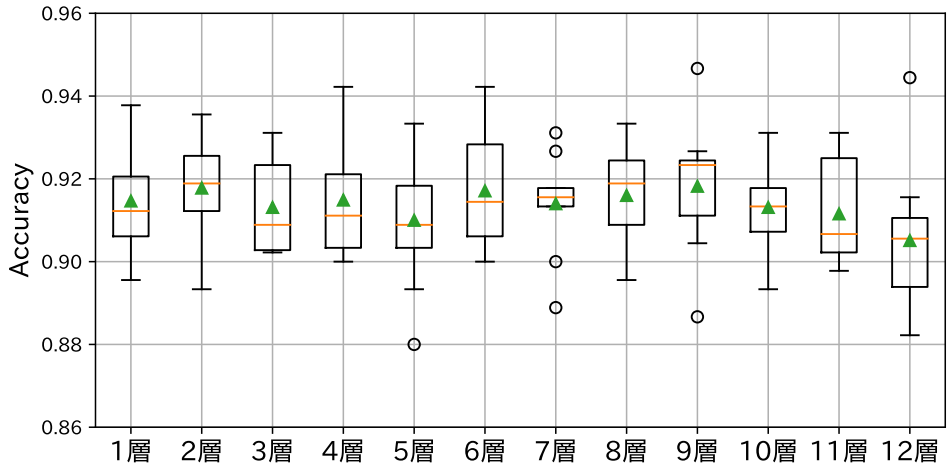

図 5.8　5.3.3 項の実験結果における補助タスク追加層の違いによる Accuracy の比較

Accuracy よりマルチタスク学習モデルの Accuracy が高く，p 値が有意水準 0.05 を下回り，有意差が認められたものを示す．表中の下線は，シングルタスク学習モデルの Accuracy よりマルチタスク学習モデルの Accuracy が高く，p 値が有意水準 0.05 を下回らず，有意差が認められなかったものを示す．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 Accuracy が高くなった．

表 5.3 の検定結果より，補助タスクを 5，12 層以外に追加した場合に 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，補助タスクを追加したマルチタスク学習により Accuracy 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，Accuracy が最も向上したのは 2 層に追加した場合で，Accuracy が 1.756% 上昇した．これらのことから，補助タスクを学習させるのに適した層は 2 層であると言える．この帰結は，先行研究 [15] での知見とも一致する結果となった．

Accuracy の変化の要因を分析するために Attention の可視化を行った．可視化に使用したデータは，補助タスクを 2 層に追加したマルチタスク学習モデル，2 層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．

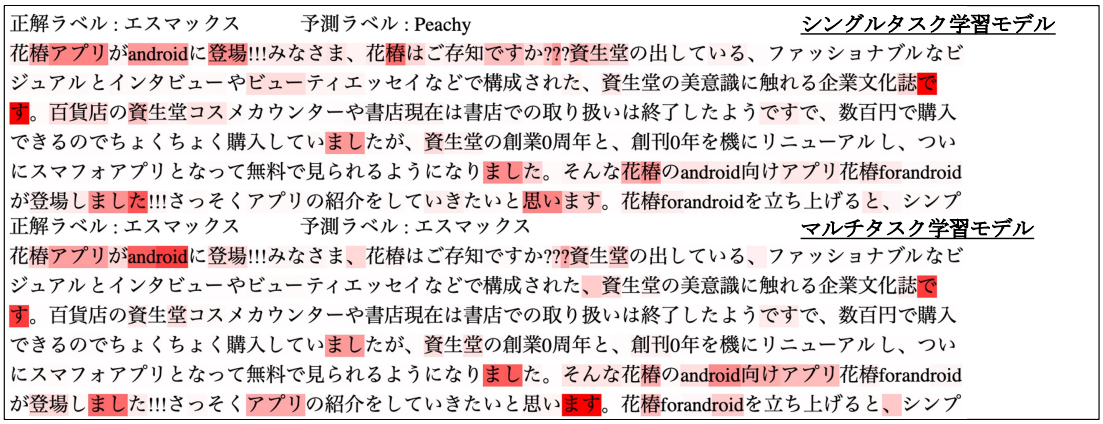

図 5.9　5.3.3 項の実験結果におけるマルチタスク学習モデルで予測できた例

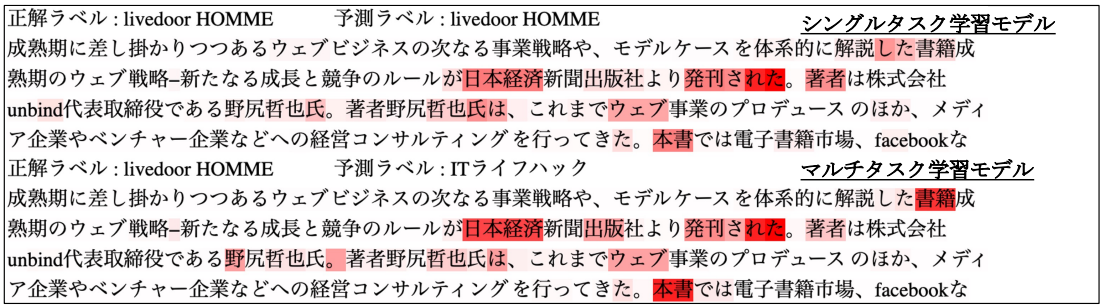

図 5.10　5.3.3 項の実験結果におけるマルチタスク学習モデルで予測できなかった例

① の場合の一例を図 5.9 に示した．エスマックスラベルが付与されたテキスト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，マルチタスク学習によってモデルが注目する部分を変化させることができ，正しく予測できるようになったと考えられる．

② の場合の一例を図 5.10 に示した．livedoor HOMME ラベルが付与されたテキスト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．

5.3.4　全てのデータセットでの比較

全体的な結果を表 5.4 に示す．この表は，テスト用データの分類を行った際，シングルタスク学習モデルでの全ての評価指標と比較して，マルチタスク学習モデルでの全ての評価指標がどれほど向上しているかを示している．楽天データセットでは，Accuracy が最も向上した場合の結果を用いている．Twitter 日本語評判分析データセットと livedoor ニュースコーパスでは，Accuracy の変化に有意差が認められたもののうち，Accuracy が最も向上した場合の結果を用いている．2 クラス分類である楽天データセットでは，Accuracy が最大で 0.045%，F-score が最大で 0.045% 向上した．3 クラス分類である Twitter 日本語評判分析データセットでは，Accuracy が 0.387%，F-score が 0.430% 向上した．9 クラス分類である livedoor ニュースコーパスでは，Accuracy が最大で 1.756%，F-score が最大で 1.759% 向上した．

これらのことから，今回の補助タスクはクラス数の少ない分類タスクの場合には，

表 5.4　マルチタスク学習による各評価指標の変化

データセット	Accuracy[%]	Precision[%]	Recall[%]	F-score[%]
楽天 (2 クラス)	+0.045	+0.041	+0.045	+0.045
Twitter(3 クラス)	+0.387	+0.448	+0.387	+0.430
livedoor(9 クラス)	+1.756	+1.608	+1.756	+1.759

モデルの性能の向上にあまり寄与しないことが分かった．クラス数の少ない分類タスクでは，補助タスクをヒントとして与えなくても既に主タスクのみでうまく学習できており，性能の向上に寄与しなかったと考えられる．だが，クラス数の多い分類タスクであればあるほど，性能の向上に寄与することが分かった．クラス数の多い分類タスクでは，主タスクのみでも学習できているが，どの部分が判断根拠となるか明示的に与えることで，更なる性能の向上に寄与したと考えられる．

第 6 章　おわりに

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であり，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，補助タスクの設計は実験者の主観や経験に頼る場合が多く，設計には多くの時間やコストが費やされていた．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，自然言語処理において基礎的なタスクである品詞タグ付けに注目し，補助タスクとして重要品詞抽出タスクを提案する．提案した補助タスクを用いたマルチタスク学習を行い，補助タスクが主タスクの性能向上に寄与するかどうか実験を行った．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．BERT を用いた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構築し，学習を行った．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 Accuracy 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，Accuracy が最大で 0.045% 向上した．しかし，Accuracy の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，Accuracy が 0.383% 向上した．この場合は Accuracy の変化に有意差が認められた．livedoor ニュースコーパスでは，Accuracy が最大で 1.756% 向上した．この場合も Accuracy の変化に有意差が認められた．今回の補助タスクは BERT の 2 層に追加して学習させるのが良いということ，特にクラス数が多い場合に Accuracy の向上に寄与することが分かった．また，BERT の Attention の可視化を行い，モデルが注目している部分を比較することで，Accuracy が向上した理由を分析した．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，他の自然言語処理タスクで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，関係抽出タスクのような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 以外の解析器を用いてラベル付けを行いたいと考えている．MeCab は，ipadic や ipadic-NEologd という辞書の更新がされておらず新語に対応できない，他の解析器と比較して解析速度は速いが解析性能は劣るという問題点も存在する．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が行える 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により，どのような補助タスクが性能向上に有効であるかを突き止める一助となれば良いと考える．

謝辞

卒業研究を進めるにあたり，指導教員の鈴木優准教授には多くのことでご指導ご鞭撻を賜りました．研究を順調に進めていた時期には，この先どのように進めていけば良いかアドバイスをして下さいました．研究に行き詰まっていた時期には，面談で何度か相談に乗っていただきました．また，毎週のゼミ冒頭の雑談では，面白い話をしていただきました．実は，私はあの雑談を楽しみにしていたりします．良い設備や環境を提供して下さったおかげで，配属当初から研究活動に没頭することができたと思います．本当にありがとうございました．

次に，研究室に所属する方々には，研究に関する様々な意見をいただいたり，協力をしていただきました．特に，研究室の同期の皆様には様々な場面でお世話になりました．他愛のない雑談を通して，配属当初と比べると，見違えるほど楽しく研究活動を行える環境に変化したと思います．また，お互いの研究に関する相談を重ねることで，より良い研究に昇華させていくことができたと思います．ありがとうございました．

次に，配属当初から昨年十月ごろまでは佐野さんに，十月以降は井尾さんに，事務補佐員として書類の作成や事務的な手続きをして下さいました．円滑に研究活動を行えるようにサポートしていただきました．ありがとうございました．

そして，家族は大学卒業までの約二十年もの間，経済的にも精神的にも私を支えて下さいました．私の性格故にご迷惑をおかけし，振り回してしまったことが多々あったと思います．何度も心配をかけ，悩ませてしまった時期もあったと思います．それでも，私が何をやるにしても，いつも前向きに応援して下さったことは忘れません．感謝してもしきれません．ありがとうございました．そして，これからも温かく私を見守っていただけると幸いです．

さらに，研究を進めるにあたり，楽天グループ株式会社様にはデータセットを提供していただきました．ありがとうございました．

最後になりますが，本研究を無事に終えることができたのは，皆様の援助があったからこそだと思っております．改めて，ありがとうございました．この場を借りて感謝申し上げます．

参考文献

[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] 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.

[6] 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.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, pp. 1345–1359, 2010.

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

[10] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pp. 845–850, 2015.

[11] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *5th International Conference on Learning Representations*, 2017.

[12] Erik F Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of EACL'99, Bergen, Norway*, pp. 173–179, 1999.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus

Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[18] 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.

[19] 高木千恵. 関西若年層の用いる同意要求の文末形式クナイについて. 日本語の研究, Vol. 5, No. 4, pp. 1–15, 2009.

発表リスト

[1] 北村拓斗，鈴木優『クラウドソーシングにおけるマルチタスク学習の補助タスクの設定手法』，東海関西データベースワークショップ，2022
[2] 北村拓斗，鈴木優『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』，第 15 回データ工学と情報マネジメントに関するフォーラム，2023