ストップフレーズ抽出を併用した文書分類

木村　優介†　　駒水　孝裕††　　波多野賢治†††

† 同志社大学大学院文化情報学研究科　〒 610-0394 京都府京田辺市多々羅都谷 1-3
†† 名古屋大学数理・データ科学教育研究センター　〒 464-8601 愛知県名古屋市千種区不老町
††† 同志社大学文化情報学部　〒 610-0394 京都府京田辺市多々羅都谷 1-3
E-mail: †kimura@mil.doshisha.ac.jp, ††taka-coma@acm.org, †††khatano@mail.doshisha.ac.jp

あらまし　深層学習は文書分類においても盛んに利用されており，そのモデルの多くは従来の手法よりも高精度を達成している．文書分類において，入力文のトークン化は重要な役割を果たすが，語彙を適切に設定することが重要となる．文字より細かい単位の文字列を統計的な手法により語彙化するトークナイザは文書分類や文書検索などの幅広いタスクに使われている．一般的に単語はジップの法則に従い，高頻度の単語はその多くが機能語であり，文書分類に寄与しないストップワードとして扱われる．また，サブワードや隣接するサブワードで構成されるサブワードフレーズも単語と同様にジップの法則に従うことから，高頻度なサブワードフレーズはストップフレーズとなることが期待される．文書分類モデルにストップフレーズを考慮させることにより，分類モデルがより分類に寄与する重要な語に注目できるようになり，その精度を向上させる可能性がある．そこで，本研究では文書分類の精度向上を目的としたストップフレーズ抽出を提案する．また，ストップフレーズ抽出と文書分類のマルチタスク学習を行うことで，意図的にストップフレーズの Attention 値を低くすることができ，適切なストップフレーズを選択することができれば文書分類の精度向上の可能性を示した．

キーワード　語彙形成, ストップフレーズ, 文書分類, マルチタスク学習

1　はじめに

デジタル世界で利用可能な文書は急増しており，文書を整理することは重要である [1]．文書分類は文書や文，段落などのテキストのまとまりにラベルやタグを割り振ることを目的とした研究分野である [2]．例えば，感情分析やトピック分類などは文書分類のタスクである [3]．文書分類においてトークン化を工夫すると分類精度が向上することが知られているため，分類精度の向上のためにはまずトークナイザに着目する必要がある [4]．

ニューラル言語モデルを用いた手法の普及により，これまで文法に則って文書を分割していたトークナイザは，後段タスクのために単語や単語より小さな語であるサブワードに分割するようになった [5]．ニューラル言語モデルで用いられるトークナイザはニューラル機械翻訳タスクから発展してきたため，翻訳精度を低下させる未知語をなくすことに着目されてきた．代表的なトークナイザの一つである Byte-Pair Encoding (BPE) は 1 単語内の文字同士の共起頻度を計算し，トレーニングデータにおける文字同士の共起頻度の高い順に結合する．BPE は，この結合プロセスを事前に指定した語彙サイズになるまで繰り返すことで，最短 1 文字から最長 1 単語までのトークンを構築する手法である [6]．また，ユニグラム言語モデルは目的の語彙サイズより大きな語彙をあらかじめ作成しておき，トレーニングデータ全体の損失の増加が最も少ない語を削除していくことで，目的の語彙サイズを作成する手法である [7]．BPE やユニグラム言語モデルは単語に分割するトークナイザと比べ，翻訳精度が高いことが知られている [8]．

しかし，隣接する二つ以上のサブワードから構成されるサブワードの列（サブワードフレーズと呼称する）が後段タスクにどのような影響を及ぼすかは十分に調査されていない．2.3 項で示すように，サブワードの出現頻度はジップの法則に従う．一般に，単語の出現頻度もジップの法則に従うことが知られており，高頻度の単語は文書の意味を表すのに寄与しないストップワードとして扱われる．この単語のストップワードのアイデアに倣い，高頻度で出現するサブワードフレーズをストップフレーズとみなす．文書分類にとって有益でないストップフレーズを明示的に考慮できるようにモデルを訓練することで，文書分類タスクの精度向上につながる可能性がある．

そこで，本稿ではストップフレーズをトークンとして扱った場合，文書分類の精度にどのような影響を及ぼすかを調査する．また，ストップフレーズが文書分類に与える調査結果を基に，文書分類精度を向上させるストップフレーズ抽出タスクを提案する．

本稿の 2 節で基礎分析としてストップフレーズをトークンとして扱う有効性について説明する．3 節では，基礎分析から得られた結果を基に事前学習済みニューラル言語モデルでストップフレーズを考慮して文書分類モデルを学習する手法を提案する．4 節では，ストップフレーズを考慮した文書分類モデルと文書分類のみを学習したモデルとの比較実験を行い，その結果に対する考察を行う．5 節では，既存のマルチタスク学習を考慮した文書分類手法との差異を説明し，6 節では本稿が行ったことをまとめて説明する．

2 基礎分析

基礎分析では，文書分類にとって有用なトークン化を明らかにするためにサブワードフレーズをトークナイザの語彙に追加し，その語彙を用いて文書をトークン化することで文書分類にどのような影響を与えるのかを明らかにする．

サブワードフレーズをトークンとして扱うためには，まず，トレーニングデータに対し，トークナイザを学習する．次に，サブワードの N-gram ($2 \leqq N \leqq 10$) で構成されるサブワードフレーズをトレーニングデータの各文書から抽出し，トレーニングデータにおける出現頻度順にトークナイザの語彙に N 語加える．本稿では，これら N 語のトレーニングデータにおいて高頻度のサブワードフレーズをストップフレーズと呼ぶ．

2.1 ストップフレーズを考慮した文字列の分割

この手法の実装に関しては Algorithm 1 を用いる．Algorithm 1 は，既存のサブワード区切りのトークナイザで入力文書をトークン化した後，ストップフレーズを構成する複数のトークンを一つのトークンとして扱う手法である．

Algorithm 1 の 1 行目では，サブワード区切りのトークナイザを定義する．2 行目では，事前にトレーニングデータから抽出したサブワードフレーズの出現頻度上位 N 語を stopphrase_list に格納する．3 行目では，トレーニングデータやテストデータに含まれる文書を格納する．4 行目では，トークン化された結果を格納する空の配列を定義する．5 〜20 行目では，まずトレーニングデータやテストデータに含まれる各文書に 2 行目で定義したストップフレーズが格納された配列 stopphrase_list に含まれるサブワードフレーズが出現している場合，そのサブワードフレーズより前の部分を 1 行目で定義したトークナイザでトークン化して配列 token_list に格納する．次に，ストップフレーズのリストに含まれるサブワードフレーズを配列 token_list に格納し，そのサブワードフレーズより前の文字を削除する．このサブワードフレーズより前を削除する処理は文書に含まれるサブワードフレーズがなくなるまで行い，サブワードフレーズがなくなった場合，残りの文字をトークナイザでサブワード区切りにし，配列 `token_list` に格納する．21 行目では，これらの処理を全文書に行い，その結果を配列 tokenized_document_list に格納する．最終的に得られる配列 tokenized_document_list には，各文書のサブワードとサブワードフレーズの分割結果が出現順で格納されている．

Algorithm 1 ストップフレーズを考慮したトークン化

```
Input: a document
Output: a tokenized document
 1: tokenizer ← e.g.  SentencePiece
 2: stopphrase_list ← list of top N stopphrases
 3: corpus ← list of documents
 4: tokenized_document_list ← []
 5: for i=0 to len(corpus) do
 6:   text ← corpus[i]
 7:   token_list ← []
 8:   for i=0 to len(stopphrase_list) do
 9:     if stopphrase[i] in text then
10:       idx ← stopphrase[i]'s index of the text
11:       subwords ← tokenize text[:idx] using tokenizer
12:       token_list.append(subwords)
13:       stopphrase ← text[idx:idx + len(stopphrase[i])]
14:       token_list.append(stopphrase)
15:       delete text[:idx + len(stopphrase[i])]
16:     else
17:       subwords ← tokenize text using tokenizer
18:       token_list.append(subwords)
19:     end if
20:   end for
21:   tokenized_document_list.append(token_list)
22: end for
```

2.2 基礎分析の評価

評価実験では，本稿で提案した手法がどれだけ文書分類タスクに有効かを明らかにする．データセットとして，映画に対するレビューサイト IMDb から得られたレビュー文書とそのレビューを書いたユーザがその映画に対してポジティブ感情を持っているか，ネガティブ感情を持っているかの二値のラベルが付与されたデータセットを用いる [9]．データセット IMDb は 25,000 件の学習用データと 25,000 件のテスト用データで構成されており，ラベルの偏りはない．また，検証用データはトレーニングデータの 10 分の 1 を利用する．

Algorithm 1 で使用するトークナイザには SentencePiece のユニグラム言語モデルを使用する．比較対象として，ユニグラム言語モデルと音声認識タスクのサブワードフレーズを考慮した BPE と比較実験を行う．評価指標には式 (1) の $F1$ 値を用いる．

$$F1 = \frac{2P \cdot R}{P + R} \tag{1}$$

ただし，

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}$$

TP は分類モデルがポジティブのラベルが付与されたレビューをポジティブと推測できた数，FP はネガティブのラベルが付与されたレビューをポジティブと推測した数，FN はポジティブのラベルが付与されたレビューをネガティブと推測した数を表す．

2.3 サブワードフレーズにおけるジップの法則の検証

本節では，サブワードがジップの法則に従うかどうかを確認する．ジップの法則に従うかどうかの確認には，サブワードのトレーニングデータにおける出現頻度を縦軸に設定し，出現頻度順にすることで得られるランクを横軸にした両対数グラフが右肩下がりになっているかどうかで確認する．また，サブワー

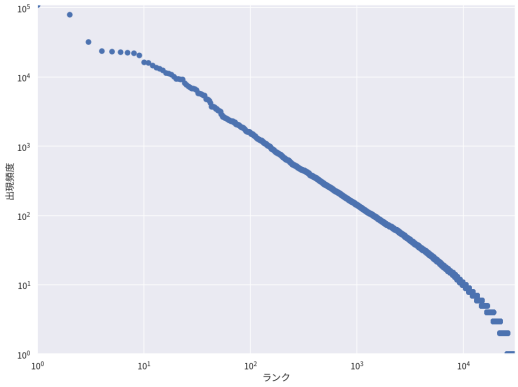

図 1　サブワードの出現頻度順に基づいた両対数グラフ

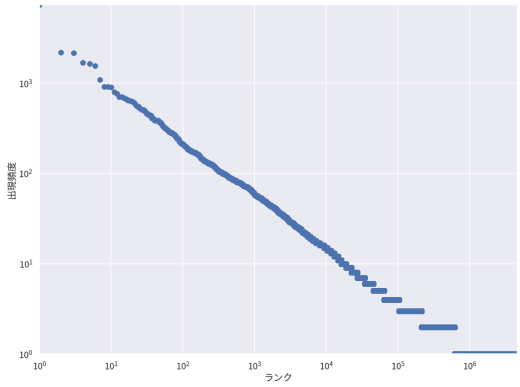

図 2　サブワードフレーズの出現頻度順に基づいた両対数グラフ

表 1　異なるトークン化による実験結果 (IMDb)

手法名	$F1$ 値
ユニグラム言語モデル	0.917
単語境界をまたぐ BPE	0.642
語彙へのストップフレーズの追加	**0.933**

ドフレーズでも同様に確認する．

図 1，2 から，サブワードフレーズとサブワードはどちらも右肩下がりになっていることから，両者ともにジップの法則に従っていると考えられる．

2.4　語彙サイズ・モデルの設計

ハイパーパラメータの違いで分類精度に差を出さないために，語彙サイズは 35,200 で固定して実験を行う．ストップフレーズは，まず語彙サイズ 32,000 語に設定したユニグラム言語モデルを IMDb のトレーニングデータで学習する．次に，IMDb のテストデータに含まれる各文書をトレーニング済みのユニグラム言語モデルでトークン化し，サブワードフレーズの頻度順上位 3,200 語を語彙に格納することで語彙サイズ 35,200 語にする．

また，入力次元を 300 次元で設定した Embedding 層を用いた 1 層の Transformer で実験を行う [10]．Transformer の設計にはバージョン 1.8.2 の PyTorch を用いて実装する．IMDb に対する Transformer の学習には，エポック数は 60，バッチサイズは 64 で実験を行う．

2.5　基礎分析の実験結果・考察

表 1 に基礎分析の実験結果を示す．表 1 からストップフレーズのトークン化が文書分類タスクに有効であることが分かった．

自動音声認識タスクで有効であった単語境界をまたぐ BPE が文書分類に有効ではなかった理由として，語彙に含まれる単語境界をまたぐトークンの量を調整できない点にあると考えられる．実際に，本稿では語彙内のストップフレーズの数を 3,200 語ではなく，6,400 語に増大させた語彙サイズ 38,400 語で実験したところ，文書分類の F 値は 0.635 になり，35,200 語で構築したどのモデルよりも低下した．この事実から，ストップフレーズは良くも悪くも分類精度に影響を与えることが分かった．また，ユニグラム言語モデルとの差異は文書分類に重要ではないと考えられる複数のトークンを一つのトークンにまとめた点であり，既知の情報を考慮したトークン化は分類精度を向上させる可能性を示した．

3　提 案 手 法

本研究では 2 節の文書分類タスクの精度を向上させる可能性があるストップフレーズの抽出タスクを提案する．2 節の 1 層の Transformer で行った実験では，ストップワードと同様に文書分類に寄与しないストップフレーズをトークンとして扱うことで文書分類の精度を向上させることができた．しかし，2 節のストップフレーズを考慮したトークナイザはサブワードだけで語彙を構築するよりも語彙サイズが増加し，各タスクのトレーニングにおける計算コストが高くなる [11]．

そのため，語彙に加えることなく既存の事前学習済みニューラル言語モデルでストップフレーズを考慮する単純な方法として，図 3 のストップフレーズ抽出と文書分類のマルチタスク学習を採用する．マルチタスク学習とは，あるタスクの汎化性能を他の関連タスクを用いて向上させる方法である [12]．本研究の場合，主となるタスクが文書分類であり，その補助となるタスクがストップフレーズ抽出である．

ストップフレーズ抽出は，文書分類のために利用する事前学習済みニューラル言語モデルのトークナイザを用いて, 隣接するサブワード N-gram の出現頻度の降順上位 N 個（$1 \leqq N \leqq 10$）をストップフレーズとして抽出する．ストップフレーズの N 個は，全サブワードフレーズの総出現頻度の $k\%$ になるまで，サブワードフレーズをその出現頻度の降順で採用した数である．

ストップフレーズを抽出するデータセットの構築は，固有表現抽出で使われる Inside-outside-beginning (IOB2) タグを用いて各トークンにアノテーションを行う [13]．IOB2 タグの B タグはストップフレーズの先頭に位置するトークンに該当し，I タグは二つ以上のサブワードで構成されるサブワードフレーズの先頭以外のトークンに該当し，サブワードフレーズに含まれないトークンには O タグを付与する．

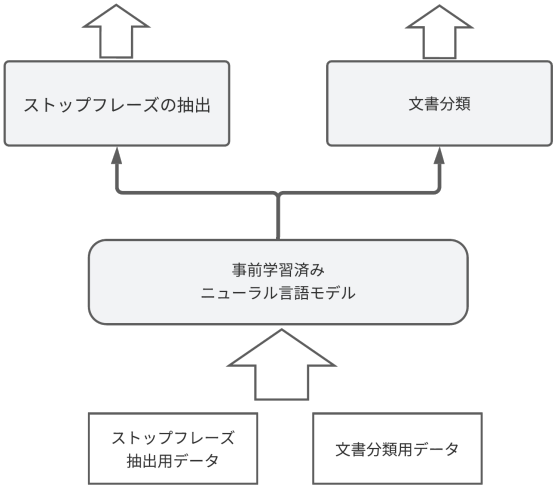

図 3　ストップフレーズを考慮した文書分類のためのマルチタスク学習

表 2　各種データセットの内容

	トレーニング用	検証用	テスト用
Amazon	22,500	2,500	5,000
IMDb	22,500	2,500	25,000
LiMiT	21,204	2,355	1,000
Yelp	22,500	2,500	25,000

4　評 価 実 験

本稿では，ストップフレーズ抽出タスクを補助タスクとしたマルチタスク学習が文書分類に対して有効かを明らかにするために，評価実験を行う．

4.1　実 験 設 定

本稿の評価実験では，表 2 の四つのデータセットを使用する．Amazon は EC サイトの Amazon に投稿された多言語の商品レビューと 20 種類の商品カテゴリが付与されている [14]．評価実験では他のデータセットと条件を合わせるために英語で記載されたレビューのみを用いた．LiMiT は自然言語で書かれた文とその文が比喩か実際の動作かの二値のラベルが付与されている [15]．Yelp は口コミサイト Yelp に投稿されたレビューとその店に対する 5 段階の星評価が付与されている [16]．

今回の実験では，多種多様なデータセットでの評価を優先し，Amazon と Yelp に関しては IMDb のデータサイズに合わせてリサイズを行う．ただし，元々5,000 件しかない Amazon のテスト用データに関してはリサイズを行わない．今回使用するデータセットの 1 文書の単語数は図 4 である．

ストップフレーズはトレーニングデータに出現するサブワードフレーズの全出現頻度の 5% を占める語を設定する．この設定における各データセットのストップフレーズは，Amazon で 38 語，IMDb で 56 語，LiMiT で 33 語，Yelp で 52 語が選択された．

本稿の提案手法との比較対象として，文書分類タスクのみのモデルと比較を行う．事前学習済みニューラル言語モデルには bert-base-uncased を用いた [17]．また，トレーニングは 30 エポック，バッチサイズ 16 で実験を行い，検証用データで最良なモデルを選択する．

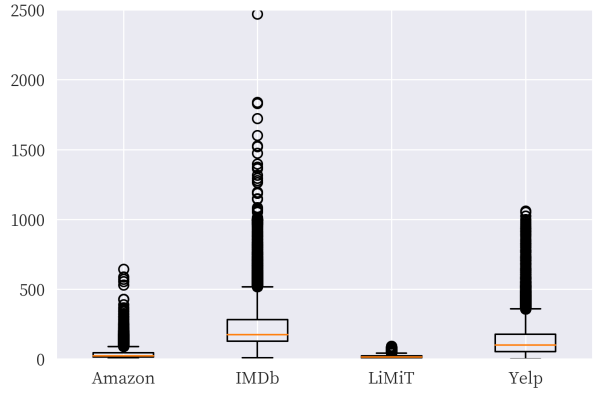

図 4　各種データセットの文書長

表 3　各種データセットでの重み付き $F1$ 値による評価実験

データセット名	文書分類モデル	ストップフレーズ抽出と文書分類のマルチタスク学習
Amazon	0.316	**0.340**
IMDb	0.836	**0.848**
LiMiT	**0.700**	0.673
Yelp	**0.540**	0.523

4.2　結　　果

表 3 に評価実験の結果を示す．Amazon や IMDb のデータセットでは本稿の提案手法であるマルチタスク学習を行った文書分類モデルの精度が文書分類モデルよりも高かった．その一方で，LiMiT や Yelp のデータセットでは文書分類用ラベルのみを学習した文書分類モデルよりも分類精度が低下した．

4.3　考　　察

ストップフレーズ抽出が文書分類に与えた影響を考察するために，ストップフレーズ抽出を併用することで文書分類ラベルだけを学習した分類モデル BERT の Attention にどのような影響を及ぼしたかを調査した．本稿で行う BERT の Attention の分析は既存研究を参考に行った [18]．

ストップフレーズの Attention の値が文書分類ラベルだけを学習した分類モデルと本稿の提案手法でどのような違いがあるかは次の方法で調べた．bert-base の Attention は 12 層の Transformer と各層の Transformer は 12 個の Head から構成されているため，1 トークンから得られる Attention の値は $12 \times 12 = 144$ 個得られる．本稿の評価実験で使用した文書分類モデルは，BERT-base の 12 層すべてをファインチューニングしているため，各層によってストップフレーズの Attention の値は異なる可能性がある．そこで，層ごとでストップフレーズを構成するトークンの Attention 値の平均値を算出した．ただし，分類結果に悪影響を及ぼす可能性のあるトークンが持つ Attention の値を取り除くため，各分類モデルが正しく文書分

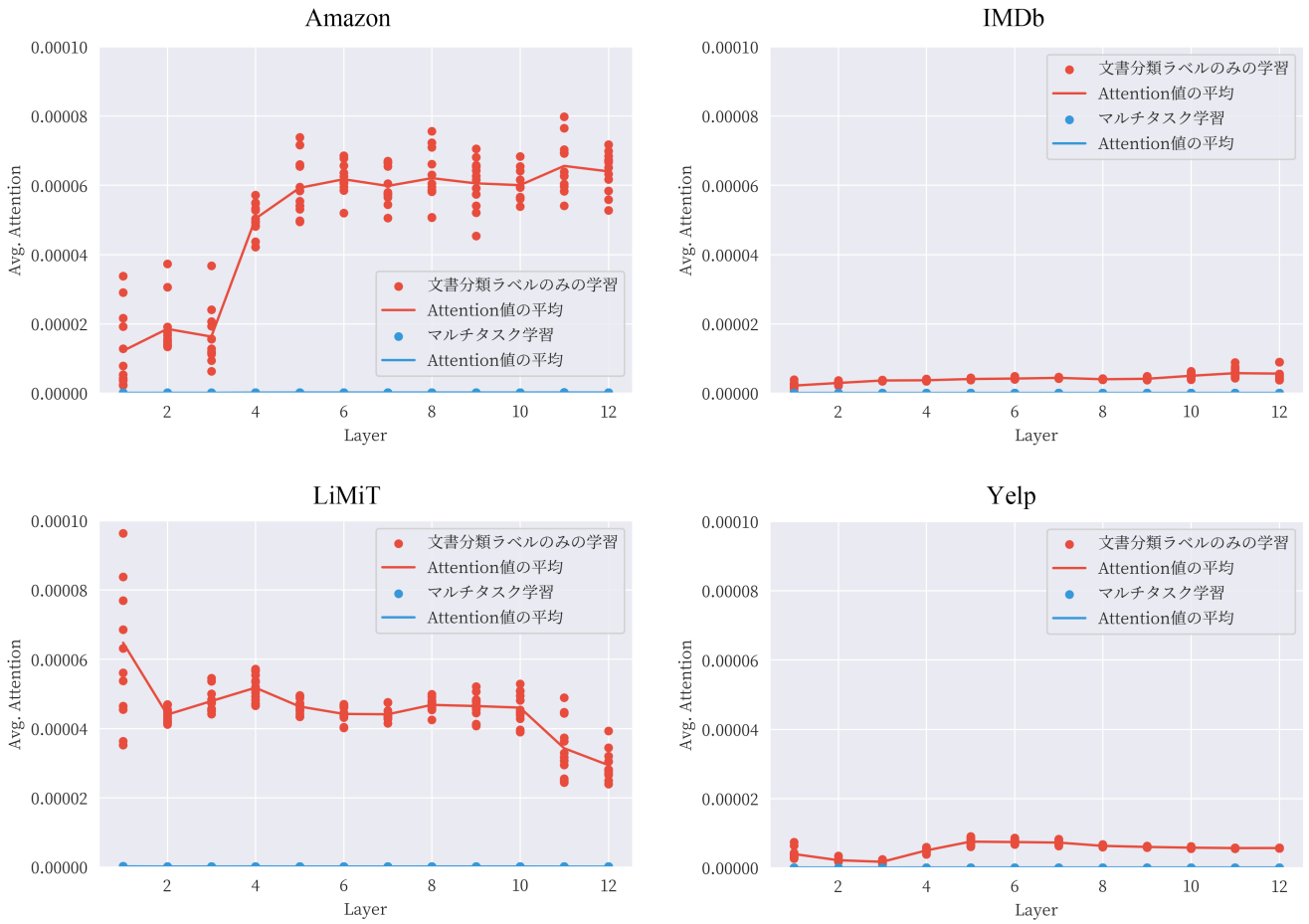

図 5　各種データセットのストップフレーズの Attention 値

表 4　ストップフレーズ抽出の精度

	検証用データ			テスト用データ		
	適合率	再現率	$F1$ 値	適合率	再現率	$F1$ 値
Amazon	0.982	0.984	0.983	0.983	0.985	0.984
IMDb	0.896	0.940	0.918	0.895	0.940	0.917
LiMiT	1.00	1.00	1.00	0.996	0.998	0.997
Yelp	0.903	0.953	0.927	0.916	0.960	0.938

類に成功した場合のトークンが持つ Attention の平均値を算出した．

各層の Attention 値を算出した結果が図 5 である．図 5 から全データセットの全レイヤーで一貫してストップフレーズの Attention 値は，文書分類ラベルのみを学習した分類モデルよりもマルチタスク学習を行った手法の方が低くなった．また，表 4 のストップフレーズ抽出の精度はすべてのデータセットで $F1$ 値が 0.9 を超えており，高い精度でストップフレーズを抽出できた．これらの結果から，文書分類にストップフレーズ抽出を併用することで，ストップフレーズの Attention 値を意図的に低くすることができると考えられる．

5　先 行 研 究

これまでの文書分類の研究では，分類精度を向上させるために他タスクとのマルチタスク学習を提案している．例えば，文書分類の一つである発話の意図を分類するタスクでは，意図分類に寄与すると考えられる固有表現抽出と文書分類を組み合わせることで分類精度が向上することが報告されている [19, 20]．

本稿の提案手法と既存のマルチタスク学習手法との差異は，人手で付与された文書分類以外のラベルが必要ない点にある．また，本稿の手法は文書分類における特定のタスク以外でも分類精度の向上を目的としている．

6　お わ り に

これまでのニューラル言語モデルのトークナイザが単語より小さな単位を考慮した手法であるのに対し，本研究では複数のサブワードで構成されるサブワードフレーズの中でも高頻度のサブワードフレーズ（ストップフレーズ）が文書分類タスクに与える影響を検証した．ストップフレーズの有効性を明らかにするために，ストップフレーズをトークナイザの語彙に追加して行う実験と，ストップフレーズの抽出と文書分類を行うマルチタスク学習での実験を行った．二つの評価実験の結果，ストップフレーズを考慮することで，既存のトークナイザから得られるサブワードを用いて文書分類を行うモデルよりも高い精度で文書分類が可能になった．また，ストップフレーズ抽出によって，文書の意味を表さないストップフレーズの Attention

値を意図的に低くすることができた．

今後は，頻度に基づくストップフレーズの選択では，その文書において重要なサブワードフレーズが含まれている可能性がある．これは，ストップワードのような機能語とは異なり，語の意味を変換する接頭辞や接尾辞がストップフレーズに含まれる可能性が高まるためである．そのため，頻度とは異なる視点での重要度を付与することが文書分類においては重要である．その一つのアイデアとして，ストップフレーズがその文の意味を表現しないことを考慮する．その方法として，タスクの一つとして Masked Language Model タスクをマルチタスク学習に追加し，マスクされたサブワードフレーズの重要度を取り入れることを考えている．

謝　　辞

本研究の一部は JSPS 科研費 JP18H03342, JP19H04218 および JP21H03555 の助成を受けたものである．

文　　献

[1] Nihar Ranjan, Abhishek Gupta, Ishwari Dhumale, Payal Gogawale, and Rugved Gramopadhye. A survey on text analytics and classification techniques for text documents. *International Journal of Development Research*, Vol. 5, pp. 5952–5955, 2021.

[2] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, Vol. 54, No. 3, 2021.

[3] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A survey on text classification: From shallow to deep learning. *CoRR*, Vol. abs/2008.00364, , 2020.

[4] Tatsuya Hiraoka, Hiroyuki Shindo, and Yuji Matsumoto. Stochastic tokenization with a language model for neural text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1620–1629. Association for Computational Linguistics, 2019.

[5] Elizabeth Salesky Colin Raffel Manan Dey Matthias Gallé Arun Raja Chenglei Si Wilson Y. Lee Benoît Sagot Samson Tan Sabrina J. Mielke, Zaid Alyafeai. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. In *CoRR*, 2021.

[6] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725. Association for Computational Linguistics, 2016.

[7] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75. Association for Computational Linguistics, 2018.

[8] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71. Association for Computational Linguistics, 2018.

[9] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30, pp. 1–11. Curran Associates, Inc., 2017.

[11] Sebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, pp. 1–10. Association for Computational Linguistics, 2015.

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

[13] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995.

[14] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

[15] Irene Manotas, Ngoc Phuoc An Vo, and Vadim Sheinin. LiMiT: The literal motion in text dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 991–1000. Association for Computational Linguistics, 2020.

[16] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, p. 649–657. MIT Press, 2015.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

[18] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention. In *BlackBoxNLP@ACL*, 2019.

[19] 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.

[20] 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.