令和 4 年度茨城大学大学院理工学研究科情報工学専攻

修士学位論文

BERT を用いた文書分類タスクにおける Mix-Up 手法の有効性検証

所属　情報工学専攻

著者　菊田尚樹 (21NM720N)

指導教員　新納浩幸教授

令和 5 年 2 月 3 日 (金)

令和 4 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

BERT を用いた文書分類タスクにおける Mix-Up 手法の有効性検証

著者

菊田尚樹 (21NM720N)

指導教員

新納浩幸教授

論文要旨

自然言語処理のタスクを機械学習のアプローチで解決しようとした場合, 訓練データの構築コストが高いことが問題になる．このため従来より様々な試みがなされている．その中で近年のトピックの一つとしてデータ拡張 (Data Augmentation) [1] がある．データ拡張手法は大きく加工と生成の 2 種類に分けられる．例えば画像の識別であれば，訓練データ内の画像を反転させたり，切り取ったりした画像であってもその画像のラベルに変化はないので，そのようにして加工した画像を訓練データに追加することで訓練データを増やすことができる．あるいは GAN などを利用して人工的なデータを生成する手法もデータ拡張の一種といえる．生成を利用したデータ拡張手法の一つである Mix-Up 手法 [2] は，簡単に実装でき効果も高いことが知られている．Mix-Up 手法は画像識別を対象にした手法であるが，自然言語処理にも応用が可能である [3].

本論文では BERT [4] を用いた文書分類タスクに Mix-Up 手法を適用することを試みた．具体的には BERT は 2 文の入力が可能であるため，2 文書の ID 列を連結して入力し，one-hot ベクトルを利用して複数クラスの出力を可能にしたものを教師データとした．livedoor ニュースコーパスを用いた実験で，2 文書を混合させる際の選出方法・混合方法を複数パターンで試し，それぞれについての有効性の検証を行った.

実験の結果，1. 同じ 2 文書の組み合わせ方でも前後の順序によって結果が変わること，2. 混合の割合は 7:3(3:7) よりも 5:5 が適していること，3. 同一ラベルの文書同士を組み合わせた際に分類精度が向上すること，4. 分類精度の低いラベルのデータに特化してデータ拡張を行っても精度の上昇は見られないこと，の 4 点がわかった.

Master's Thesis in Scholastic 2022, Major in Computer and Information Sciences,
Granduate School of Science and Engineering, Ibaraki University

Verification of the effectiveness of the Mix-Up method in document classification tasks using BERT

Author : Naoki Kikuta (21NM720N)

Adviser : Prof. Hiroyuki Shinnou

Abstract

When natural language processing tasks are to be solved with machine learning approaches, the high cost of constructing training data is a problem. One of the recent topics to solve the problem is Data Augmentation [1]. Data Augmentation methods can be divided into two main categories: processing and generation. For example, in the case of image identification, Processed images such as flipped or cropped images can be added to the training data. Artificial data generation using GAN and other methods can also be considered a type of Data Augmentation. Mix-Up method [2], one of the Data Augmentation methods using generation, is known to be easy to implement and highly effective. Mix-Up method was originally developed for image identification, but it can be applied to natural language processing as well [3].

In this paper, we apply Mix-Up method to a document classification task using BERT [4]. Specifically, since BERT allows for the input of two sentences, we concatenated the two documents and used them as the input data, and used the one-hot vector to output multiple classes as the teacher data.

In an experiment using the livedoor news corpus, we tested multiple patterns of mixing methods and verified the effectiveness of each method. Experimental results showed that the results varied depending on the order of the combinations, that mixing ratio of 5:5 is more suitable than 7:3 (3:7), that classification accuracy was improved when documents with the same label were combined and that increasing data with labels that originally have lower classification accuracy than others does not increase accuracy.

目次

表目次

図目次

第 1 章

序論

自然言語処理のタスクを機械学習のアプローチで解決しようとした場合，訓練データの構築コストが高いことが問題になる．このため従来より様々な試みがなされている．その中で近年のトピックの一つとしてデータ拡張 (Data Augmentation) [1] がある．データ拡張手法は大きく加工と生成の 2 種類に分けられる．例えば画像の識別であれば，訓練データ内の画像を反転させたり，切り取ったりした画像であってもその画像のラベルに変化はないので，そのようにして加工した画像を訓練データに追加することで訓練データを増やすことができる．あるいは GAN などを利用して人工的なデータを生成する手法もデータ拡張の一種といえる．生成を利用したデータ拡張手法の一つである Mix-Up 手法 [2] は，簡単に実装でき効果も高いことが知られている．Mix-Up 手法は画像識別を対象にした手法であるが，自然言語処理にも応用が可能である [3].

本論文では BERT [4] を用いた文書分類タスクに Mix-Up 手法の適用を試みた．具体的には BERT は 2 文の入力が可能であるため，2 文書の ID 列を連結して入力し，one-hot ベクトルを利用して複数クラスの出力を可能にしたものを教師データとした．livedoor ニュースコーパスを用いた実験で，2 文書を混合させる際の選出方法・混合方法を複数パターンで試し，それぞれについての有効性の検証を行った.

第 2 章

関連事項

2.1　埋め込み表現

機械学習にて自然言語を処理する際，テキストをそのままの形で用いることはできず，ベクトルの形に表現し直す必要がある．ベクトル化することを埋め込みといい，埋め込みによってできたベクトルを埋め込み表現と呼ぶ．埋め込み表現を獲得するモデルとして代表的なものに Word2Vec [5] がある．

2.2　ニューラルネットワーク

人間の脳の神経細胞ネットワークを元に考えられた数理モデルであり，複数のニューロンから構成される．(図 2.1 参照) 基本的に入力層，中間層，出力層の三層構成で，入力に重み (w) をかけた後，バイアス (b) を加えたものが次の層への出力される．w と b を最適化し，理想的な出力値 y を求めることを目的とする．また，このニューラルネットワークを多層化し，さらに複雑になったものがディープニューラルネットワークであり，それを用いた学習がディープラーニングと呼ばれる．

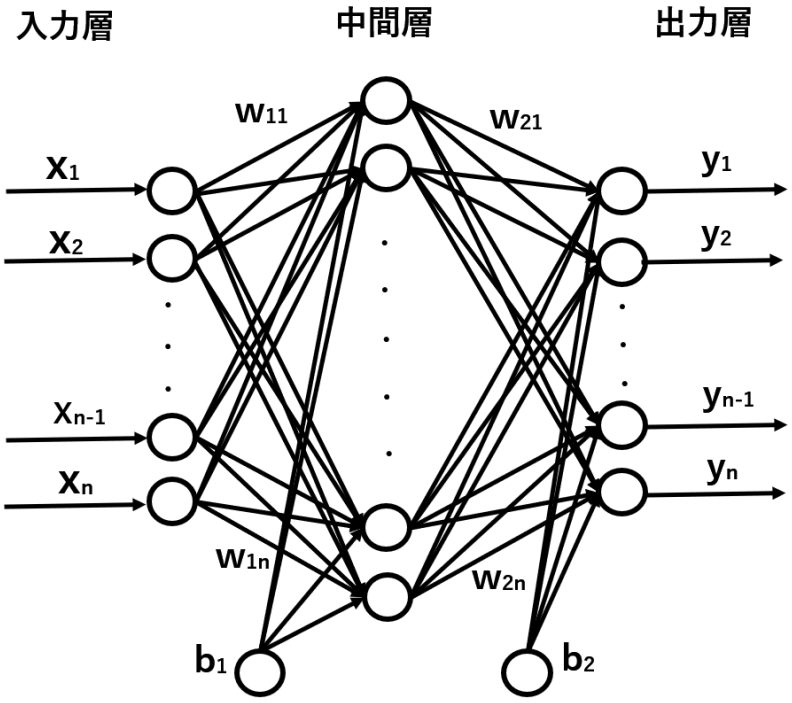

図 2.1: ニューラルネットワークの構造

2.3　RNN

文脈を正しく理解する際, 文章中の時系列を正しく捉える必要がある．RNN(Recurrent Neural Network) はニューラルネットワークを拡張し，時系列データを扱えるようにしたものである．RNN の構造を図 2.2 に示す．RNN は，中間層がループする構造となっており，中間層が前の時刻の中間層の影響を受けるので，ニューラルネットワークが以前の時刻における情報を保持できる．そのため，過去の情報を用いて判断を下すことが可能になっている．しかし RNN では長期の記憶を保持することに難点がある．

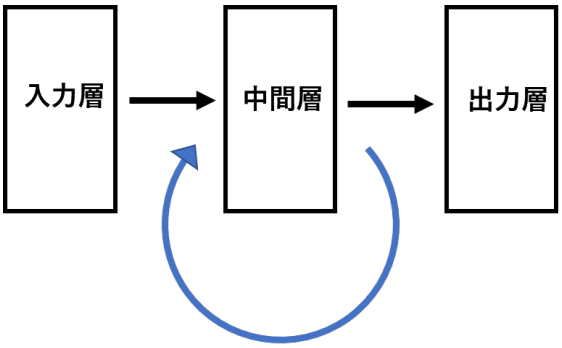

図 2.2: RNN の構造

2.4　LSTM

RNN の長期記憶を保持することが難しい点を克服したのが LSTM(Long short-term memory) である．LSTM はゲートと呼ばれる仕組みを採用することで，RNN と違い，長期と短期両方の記憶を保持することが可能となっている．LSTM の構造を図 2.3 に示す．LSTM は中間層の代わりに LSTM ブロックを配置し，過去の情報を忘れるか保持するかを判断することで，必要な情報のみを次の時刻に引き継ぐことを可能にしている．

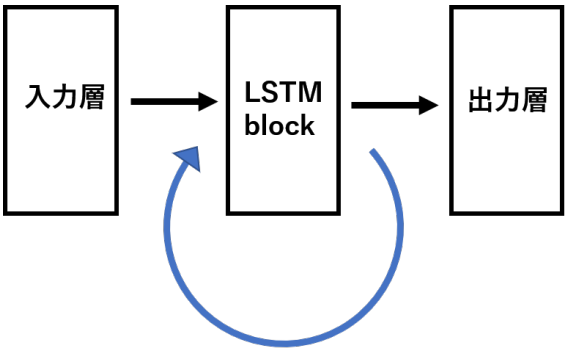

図 2.3: LSTM の構造

2.5　Transformer

2017 年に Ashish Vaswani らによって発表された深層学習モデル [6] であり，自然言語処理の多くのタスクに利用できる．Transformer は Attention 機構と呼ばれる仕組みに基づき作られており，それ以前に時系列データの分析に使われていた RNN や LSTM の性能を上回り，代替となるものである．RNN などでも Attention 機構を追加したものが利用されていたが，Transformer では Attention 機構のみを利用して単語間の関係の分析，文脈判断を行うことが可能である．モデルの構造を図 2.4 に示す．

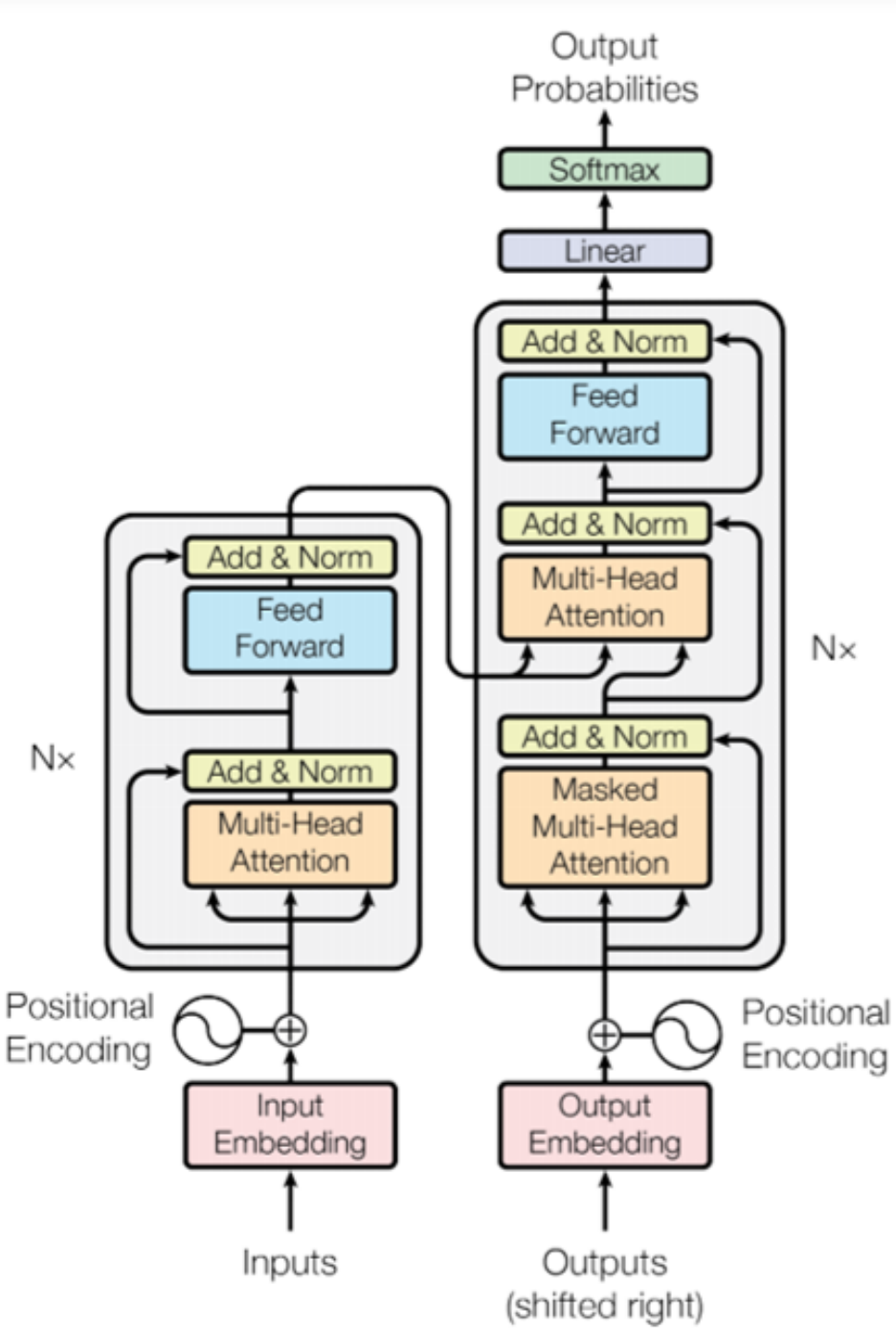

図 2.4: Transformer の構造 (元論文 [6] からの引用)

また，Transformer を利用した事前学習モデルとして BERT がある．

2.6　BERT

BERT(Bidirectional Encoder Representations from Transformers) は 2018 年に Google が作成して以降利用が注目されている高性能な事前学習済みモデル [4] であり，分類，単語予測，文脈判断などに活用ができる．構造としては，Transformer で用いられた Multi-head attention を 12 層重ねて作られている．従来のモデルでは対象単語の右または左の単語のみから文脈を学習していたのに対し，BERT では対象単語の前後の単語から文脈を判断するため，より正確に文章を処理できるようになっている．また，従来のモデルは文章分類，翻訳，感情分析など特定のタスクごとに 1 つのモデルを必要としていたため，タスクが変わればモデルを一から作り直す必要があったが，BERT は一つの言語モデルを事前学習させ，そこから目的に合わせた転移学習やファインチューニングを行い，様々な処理を行わせることができる．タスクが変わっても事前学習モデルに層を追加するだけでよいため汎用性が高いと言える．BERT の詳細を以下に記す．

2.6.1　BERT の学習

BERT は以下の二つの方法にて事前学習を行う．

(1)Masked Language Model(**単語の予測**)

文中のいくつかの単語が別の単語に置換され，本来そこにあるべき単語を予測するというものである．具体的には以下の動作が行われる．

1. 15% の確率で各単語がマスクされる．

   my dog is hairy → my dog is [MASK]

2. マスクトークンが，10% の確率でランダムで別の単語に，

   my dog is apple

   別の 10% の確率で元の単語に，

   my dog is hairy

   残りの 80% の確率でマスクのままにされる．

   my dog is [MASK]

3. マスク部分を前後の文脈から予測する．

(2)Next Sentence Prediction(2 **文の関係性の理解**)

入力として前後の関係性を持った 2 つの文があり，50 %の確率で後ろの文が関係性のない文に書き換わる．文のつながりが正しいなら IsNext，間違っているなら NotNext の判定を出す．具体例を以下に示す．

入力：　[CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

判定：　IsNext

入力:　[CLS] the man went to [MASK] store [SEP] penguin [MASK] are flight ##less birds [SEP]

判定:　NotNext

2.6.2　BERT の性能評価

モデルの精度を測る指標である以下のベンチマークタスク 11 種で既存のモデルを超え最高精度を達成している．

GLUE　9 種類の言語の理解に関するタスク うち 8 種類で達成

SQuAD　質疑応答タスク 陳述文から質問文の解答を抽出

CoNL　固有表現抽出タスク 単語に人物，組織，位置などをタグ付ける

SWAG　入力文に続く文を 4 つの候補の中から選ぶ

ここまで述べたように，BERT は自然言語処理において利用価値の高いモデルであり，本研究では Mix-Up 手法を利用することで，訓練データ構築のコストを下げたうえで，BERT を用いた文書分類タスクのさらなる精度向上を試みる．

2.7　画像分野でのデータ拡張 (Data Augmentation)

2.7.1　画像の加工

既存の訓練データ画像に対して，回転，平行移動，拡大縮小，色彩の変更，ノイズ [マスク] の挿入などの加工を施し，新たな訓練データとして追加するといった手法がある．回転を例にとると，元の画像を時計回りに 45 度ずつ回転させたものを各画像に対して作っていけば一周するまでに 7 枚の新たな画像を作成でき，結果的に訓練データ数を 8 倍にできる．この場合，画像の正解ラベルは元のままである．

2.7.2　GAN による生成

GAN(Generative Adversarial Network： 敵対的生成ネットワーク) は生成モデルの一種であり，元画像そっくりの偽物の画像の作成を行うことができる．それにより出来上がった偽画像を新たな訓練データとして使うことでデータ拡張となる．GAN では，生成器ネットワークと識別器ネットワークが競い合うようにして学習を行う．前者は一般に Generator(贋作者) と呼ばれ，後者は一般に Discriminator(鑑定者) と呼ばれる．Generator は Discriminator を騙せるように偽画像の完成度を高め，Discriminator は Generator に騙されないように識別能力を高めていくといった具体に学習を進める．GAN の構造を図 2.5 に示す．

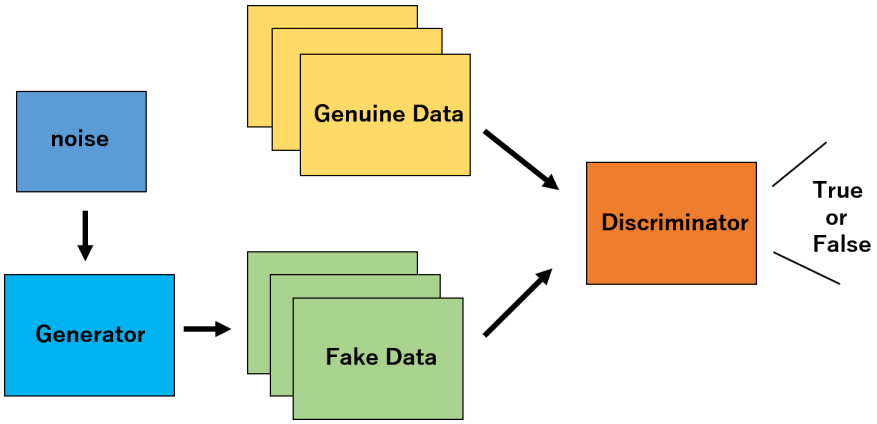

図 2.5: GAN の構造

2.7.3　Mix-Up 手法

Mix-Up とは，2017 年に Hongyi Zhang により発表された画像分野でのデータ拡張手法である [2]．画像データに関しては式 2.1，ラベルに関しては式 2.2 によってデータ拡張を行う．

$$x = \lambda x_i + (1 - \lambda)x_j \tag{2.1}$$

$$y = \lambda y_i + (1 - \lambda)y_j \tag{2.2}$$

x は画像データのベクトル，y はラベルの one-hot ベクトルである． λ は混合の割合を表わしている．これまでのデータ拡張と異なり，拡張して得られるデータのラベルが，元のラベル同士の混合ラベルとして新たに作られるのがこの手法の特徴である．単純な動作ながら効果を得られるため注目すべきデータ拡張手法と言え，本実験で利用した．

2.8　nlp でのデータ拡張 (Data Augmentation)

2.8.1　字句置換

テキスト中の字句をなんらかの指標に基づき，他の字句に置き換えることによって，新たな訓練データを作成する手法がある．指標の例としては，同義語，ベクトル表現の近いもの，省略形などがある．同義語であれば，辞書に基づき「永久」を「永遠」に置換するといった具合である．ベクトル表現の近いものへ置換する方法では，主に Word2Vec など，単語の埋め込み表現を獲得するツールを利用する．同義語の時とは違い， cat が lion に置換されるなどといった場合が考えられる．省略形というのは，主に英語で書かれた文章中において She is と She's を置換するといったものである．しかし注意すべき点として，She is ならば She's が成り立つのに対して，She's ならば She is とは限らない．というのは She has の省略形である場合があるからである．よって，このような複数の可能性がある曖昧な省略形を元に戻すのは許可しないといった工夫が必要である．

2.8.2　ノイズの追加

機械学習にてモデルを作る際，汎化性能を意識する必要がある．訓練データやテストデータでの評価値が高くても，他のデータでは性能を発揮できないモデルは良いモデル

とは言えない．また，扱うデータに人的不備があるなどの不測の事態に対しても，ある程度耐えられるモデルを作ることは重要である．ノイズや外的な影響を受けないことをロバスト (頑強，堅牢) 性があるといい，モデルがロバストになることを目的としたものが，ノイズの追加によるデータ拡張である．具体的には，単語に対しての意図的にスペルミスを加える方法や，文章の順番をシャッフルするなどの方法がある．スペルミスの追加方法として，アルファベットを qwerty キーボードの配置上近いアルファベットに置換するといったものがある．

2.8.3　逆翻訳

テキストを他言語にいったん翻訳してから，再び元の言語に翻訳し直すことを逆翻訳と呼ぶ．機械翻訳を利用し逆翻訳を行い，元のテキストと異なるテキストになった場合に新たな訓練データとして加えるといったデータ拡張手法がある．例えば，Google 翻訳[*1]を利用し，「明日は晴れるといいね」を英語に翻訳すると，”I hope it will be fine tomorrow” になり，再び日本語に翻訳し直すと「明日晴れるといいのですが」になる．この場合，元の文と逆翻訳後の文が異なっているので新たな文を獲得できたことになる．

2.8.4　テキストへの Mix-Up

2019 年，Hongyu Guo により，Mix-Up 手法を nlp に用いた研究が行われた [3]．Mix の方法は前節の画像分野での方法と同じく，式 2.1 と式 2.2 によって行っている．nlp の場合は，x が単語の埋め込み表現または文の埋め込み表現である．

(例) 6:4 の割合で Mix-Up する場合

文書 1 のベクトル

[0.2, -0.3, 0.5, ⋯]

文書 2 のベクトル

[0.4, 0.1, -0.5, ⋯]

新たな文書のベクトル

$[0.2, -0.3, 0.5, \cdots] \times 0.6 + [0.4, 0.1, -0.5, \cdots] \times 0.4 = [0.28, 0.22, 0.1, \cdots]$

[*1] https://translate.google.co.jp/?hl=ja&tab=wT

第 3 章

BERT における文書分類への Mix-Up 手法の適用方法

先行研究での Mix-Up 手法を BERT での文書分類にも採用するとなると問題が生じてしまう．先行研究の手法では，NN の学習以前に文書の特徴ベクトルを作っておく必要があるが，BERT を利用した文書分類では NN の学習プロセスで文書の特徴ベクトルを得るため，先行研究とは順序が異なる．仮に先行研究の手法を採用するとなると，BERT による特徴ベクトルの計算部分は学習の外でやることになり，分類精度が期待できない．(図 3.1 参照)

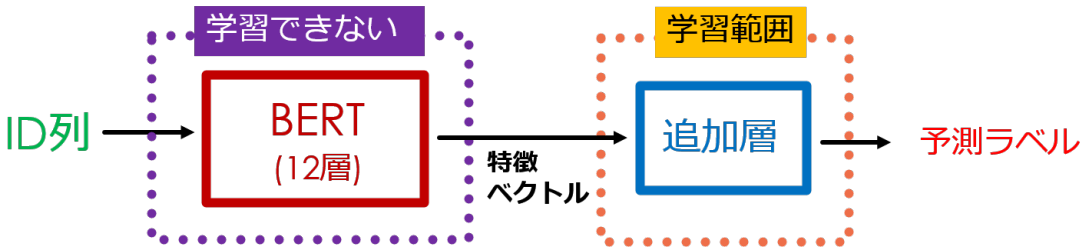

図 3.1: 先行研究手法を BERT で採用する場合の学習範囲

この問題を解決するための手法が 2021 年に Kikuta により提案された [7]．今回の実験はその手法に従い行った．手法の説明を以下に示す．

3.1　データの Mix 方法

2 つの文書において，BERT に入力する際の ID 列同士を連結するという手法である．連結させる文書は混合の割合を ID 列長の長さに反映させている．具体的には，BERT の入力最大列長は 512 のため，5:5 で混合する際は ID 列長を 256:256 に，7:3 で混合する際は 358:154 で連結させた．ID”2”は文の先頭を表すもので，ID 列の先頭に付与し，ID”3”は 2 文書の接続点を表わすもので，2 つ目 (後続) の ID 列の先頭に付与している．以下に具体例を示す，この手法であれば文書の特徴ベクトルを得る BERT 部分の学習が可能である (図 3.2 参照)．

(例)

1 つ目の ID 列

[2, 6259, 9, 12396, 14, 3596, 3]

2 つ目の ID 列

[2, 11475, 9, 3741, 5, 12098, 75, 3]

Mix-Up 後の新たな ID 列

[2, 6259, 9, 12396, 14, 3596, 3, 11475, 9, 3741, 5, 12098, 75, 3]

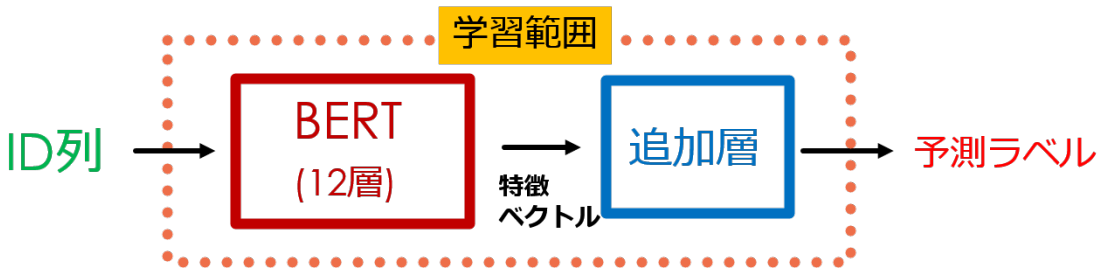

図 3.2: 本研究手法での学習範囲

3.2　ラベルの Mix 方法

各ラベルをまず 0 と 1 からなる one-hot ベクトルで表現した．Mix させる際は連結させる 2 つの文書の各割合をベクトル内で対応する箇所に反映させた．以下に具体例を示す．

(例)

ラベル 3

[0, 0, 0, 1, 0, 0, 0, 0, 0]

ラベル 6

[0, 0, 0, 0, 0, 0, 1, 0, 0]

ラベル 3 と 6 を 7:3 で Mix する場合

[0, 0, 0, 0.7, 0, 0, 0.3, 0, 0]

第 4 章

実験

4.1 概要

Mix-Up 手法を用いることで訓練データ数を増加させた状態での BERT による文書分類と，BERT を使った通常の文書分類の精度を比較し，本研究手法の効果を調べた．また，Mix-Up に用いる文書の選出方法，混合の割合，順番などを複数パターン試し，より効果的な手法は何であるか検証した．

4.2 条件

4.2.1 実行環境

実験は Google Colaboratory[*1] の GPU(Tesla T4) 環境を利用して行った．

4.2.2 使用した BERT モデル

BERT は，東北大学 乾・鈴木研究室が作成した日本語版 BERT の事前学習モデル[*2] のうちの一つである，「bert-base-japanese-whole-word-masking」を使用した．

[*1] https://colab.research.google.com/notebooks/intro.ipynb

[*2] https://github.com/cl-tohoku/bert-japanese

4.2.3　使用したコーパス

livedoor ニュースコーパス[*3] を使用し，以下の 9 ジャンルに対しての文書分類を行った.

- ラベル 0：独女通信
- ラベル 1：IT ライフハック
- ラベル 2：家電チャンネル
- ラベル 3：livedoor HOMME
- ラベル 4：MOVIE ENTER
- ラベル 5：Peachy
- ラベル 6：エスマックス
- ラベル 7：Sports Watch
- ラベル 8：トピックニュース

4.3　実験手順・手法

4.3.1　データの準備

本実験では livedoor ニュースコーパスより 3870 個の記事 (本文) を抽出し，1 ラベル当たり，訓練データとして 30 個，テストデータとして 400 個を振り分けた.(表 4.1 参照)

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

表 4.1: 使用データの内訳

ラベル	train	test	sum
0	30	400	430
1	30	400	430
2	30	400	430
3	30	400	430
4	30	400	430
5	30	400	430
6	30	400	430
7	30	400	430
8	30	400	430
sum	270	3600	3870

train は訓練時に用いたデータ，test は出来上がったモデルの最終的な分類精度を確認する際に使用したテストデータである．また，以降の手順において各文書は BERT により形態素解析と ID 化を済ませた状態で扱っている．

4.3.2　訓練データの拡張

訓練データに対し Mix-Up 手法を用いてデータ拡張を行った．本実験ではベースラインと Mix-Up をそれぞれ複数パターン試すことで，提案手法の有効性の有無を調べ，今回の Mix-Up 手法における混合割合・文書の選出方法・混合させる際の順序が結果にどう影響を与えるかの検証を行った．調査したパターンを以下の表 4.2 に示す．

表 4.2: 実験パターン

No.	Mix-Up	混合割合	混合文書の選出方法	補足
1.	なし			ベースライン
2.	あり	5:5	ランダム	
3.	あり	5:5	ランダム	1. における混合の前後入れ替え
4.	あり	7:3 と 3:7	ランダム	
5.	あり	7:3 と 3:7	ランダム	3. における混合の前後入れ替え
6.	あり	3:7 と 7:3	ランダム	3. における混合割合を入れ替え
7.	あり	3:7 と 7:3	ランダム	5. における混合の前後入れ替え
8.	あり	5:5	ランダム (条件付き)	分類精度の低いラベル＋その他のラベル
9.	あり	5:5	ランダム (条件付き)	同一ラベル同士
10.	あり	5:5	ランダム (条件付き)	9. における混合の前後入れ替え
11.	あり	5:5	ランダム (条件付き)	同一ラベル同士 (分類精度の低いラベルのみ)
12.	なし			訓練データの初期数を 315 個に変更 (各ラベル 35 個)
13.	なし			訓練データの初期数を 360 個に変更 (各ラベル 40 個)

説明が必要だと思われるものについては以下に詳細を示す.

2.　乱数を用いて 270 個の文書 (訓練データ) をランダムに並び替え，一番目から順に隣り合った 2 つの文書 (とそのラベル) を Mix させた．図 4.1 に示す．この方法の場合，文書間の隙間の数だけペアが作られるため，269 個の新たなデータができる．その結果訓練データは 270 個から 539 個に拡張した．

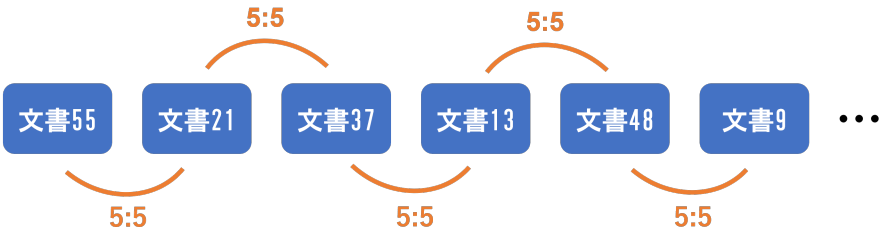

図 4.1: 実験パターン 2

4.　混合割合以外は 2. と同様．混合割合は前から順に 7:3 → 3:7 → 7:3 → 3:7 →... と 7:3 と 3:7 を交互に行った．図 4.2 に示す.

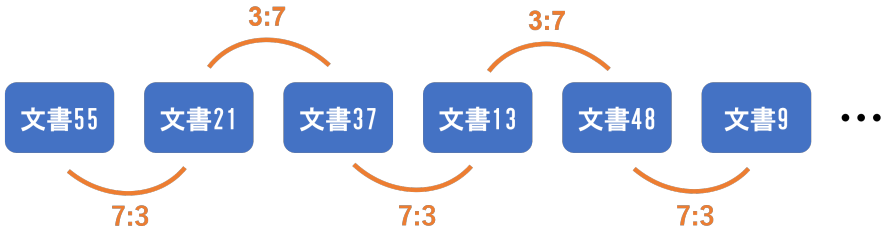

図 4.2: 実験パターン 4

6.　混合割合以外は 2. と同様．混合割合は前から順に 3:7 → 7:3 → 3:7 → 7:3... と 3:7 と 7:3 を交互に行った．図 4.3 に示す．

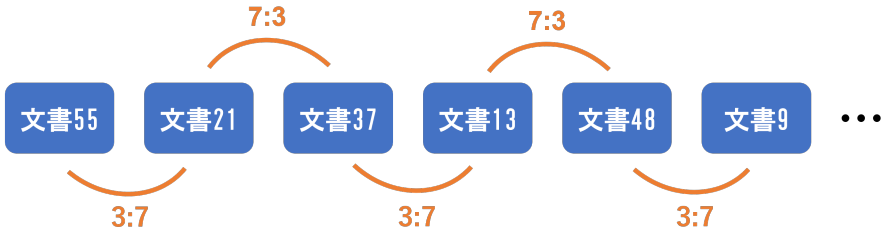

図 4.3: 実験パターン 6

8.　ラベルごとの正解率を出した際に特に正解率が低かったラベル 3 の文書を必ず選出した．具体的には，ラベル 3 の文書 30 個それぞれをラベル 3 以外の文書 10 個 (ランダム) と組み合わせ，300 個の新たなデータ個作った．その結果訓練データは 270 個から 570 個へと拡張した．図 4.4 に示す．

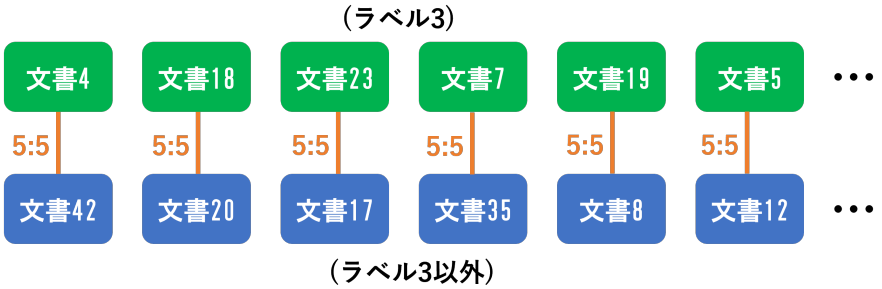

図 4.4: 実験パターン 8

9.　同一ラベルの文書同士で Mix させた．具体的には，ラベルごとに訓練データを分け (1 ラベルあたり 30 個)，30 個の訓練データをランダムに並び替え，隣り合ったもの同士で Mix させた．30 番目の文書は 1 番目の文書と Mix させ

ている．これにより，270(30 個× 9 ラベル) 個の新たなデータができ，訓練データは 270 個から 540 個へと拡張した．同一ラベル同士の Mix のため，混合後もラベルは 1 要素が 1.0 で残りの 8 要素が 0.0 の one-hot ベクトルである．図 4.5 に示す．

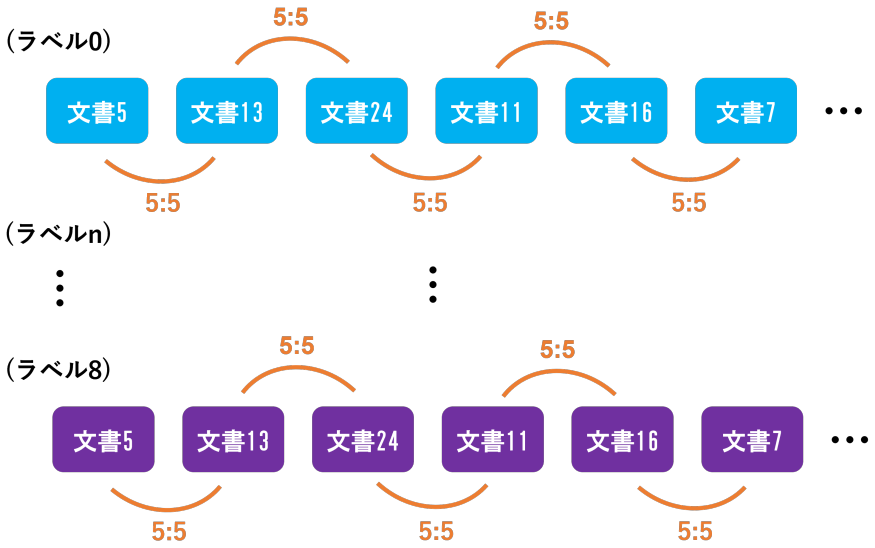

図 4.5: 実験パターン 9

11.　ラベルごとの正解率を出した際に特に正解率が低かったラベル 3 の文書同士のみを Mix させた．具体的にはラベル 3 の文書をランダムに並び替え，隣り合ったもの同士で Mix させるという動作を 9 ループ行った．これにより，270(30 個× 9 ループ) 個の新たなデータ (ラベル 3 のみ) ができ，訓練データは 270 個から 540 個へと拡張した．同一ラベル同士の Mix のため，混合後もラベルは 1 要素が 1.0 で残りの 8 要素が 0.0 の one-hot ベクトルである．図 4.6 に示す．

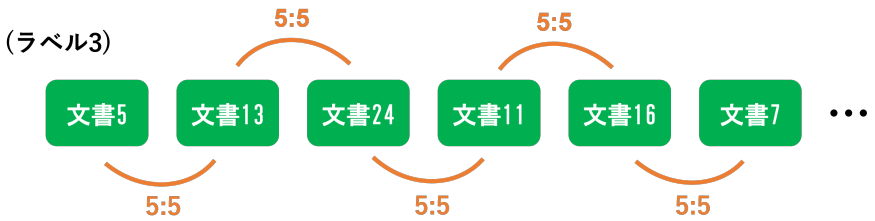

図 4.6: 実験パターン 11

12-13.　Mix-Up により精度が向上したモデルと同程度の精度を出すのに必要と

なる訓練データの初期数はいくらかを検証するために行った.

なお，実験は 1 パターン当たり 10 回繰り返しており，2.-7. までの実験における乱数は周回数ごとに同じものを用いている．(例：2. の 3 回目の実験における文書のランダムな並びと，4. の 3 回目の実験における文書のランダムな並びは同じ．)

4.3.3　作成した分類器

分類器として使ったニューラルネットワークのモデルは BERT の直後に全結合層を一層追加したものを用いた．長さ 512 以下の ID 列を BERT に入力し，768 次元の文書の特徴ベクトルを出力として得る．それを全結合層に入力し，9 次元の各ラベルに対する予測値を出力として得るという流れである．細かな設定を以下に示す.

損失関数

交差エントロピー：今回，ラベルは one-hot 表現になっており，nn.CrossEntropyLoss() に直接入力することはできないため, nn.LogSoftmax() を利用し，交差エントロピーの定義 (式 4.1) 通りに計算して損失を求めた．本実験ではバッチを使っているため，式 4.1 のバッチ平均 (式 4.2) が損失値となる.

$$E = -\sum_{k} t_k \log y_k \tag{4.1}$$

$$E = -\frac{1}{B}\sum_{b}\sum_{k} t_k \log y_k \tag{4.2}$$

t_k は正解値，y_k は予測値，B はバッチサイズである.

最適化関数

確率的勾配降下法 (SGD)：学習率は分類精度と学習効率の観点から 0.01 を採用した.

訓練データのバッチサイズ

バッチサイズは実行環境である Google Colaboratory にて可能な値における最大値であった 10 を採用した.

エポック数

訓練時における損失関数の出力を考慮 (図 4.7 参照) した結果，10 エポックで

十分な学習ができていると判断した．

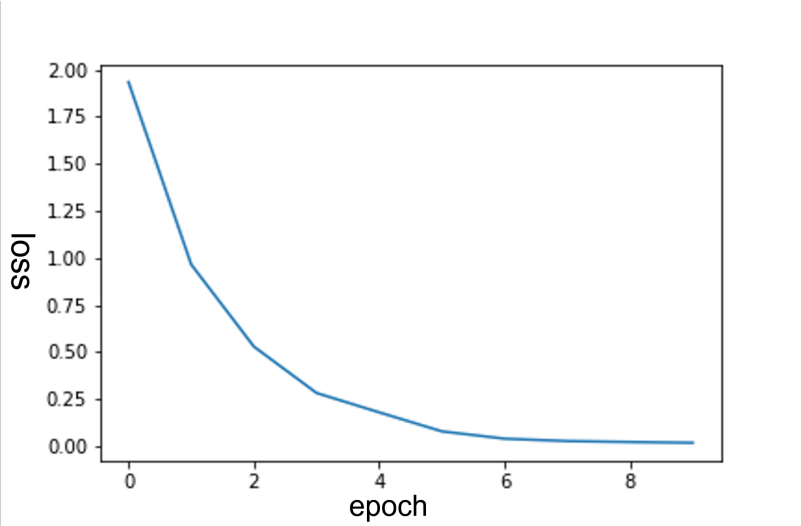

図 4.7: 各エポックごとの損失関数の出力 (Mix-Up なし)

4.4　実験結果

13 種類のパターンにおいて各 10 回ずつ実験を行い，箱ひげ図にまとめた．図 4.8 に示す．また，平均値の比較を表 4.3 に示す．※小数点第 5 位を四捨五入している．

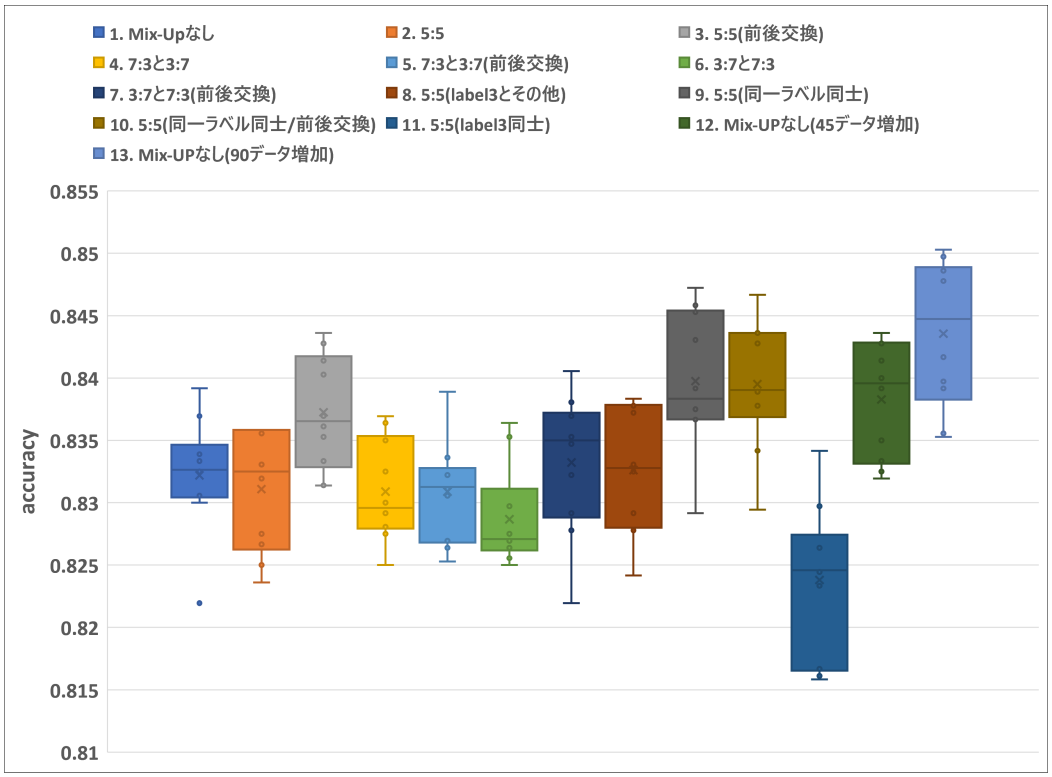

図 4.8: 実験結果

表 4.3: 平均値の比較

No.	平均値	No.	平均値
1.	0.8322	8.	0.8326
2.	0.8311	9.	0.8397
3.	0.8372	10.	0.8395
4.	0.8309	11.	0.8238
5.	0.8309	12.	0.8383
6.	0.8287	13.	0.8436
7.	0.8332		

次に，13 種類の実験パターンにおける各ラベルごとの正解数 (最大値 400) を図 4.9-図 4.21 に示す．※各パターンごとに 10 回実験しており，10 回分の平均値を記載している．

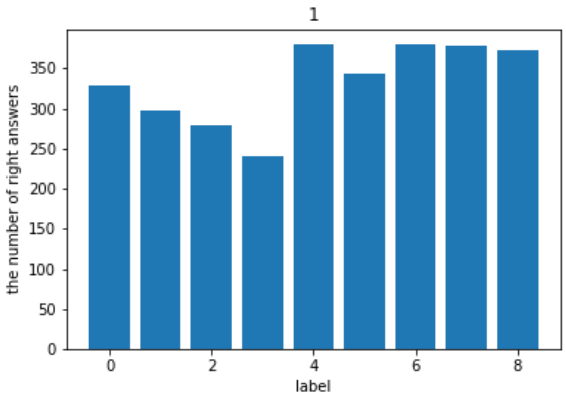

図 4.9: 1. Mix-Up なし

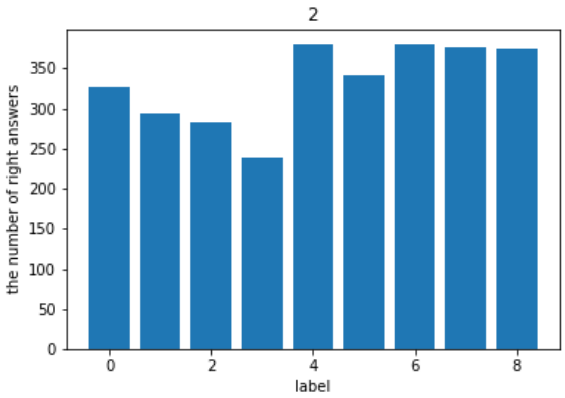

図 4.10: 2. 5:5

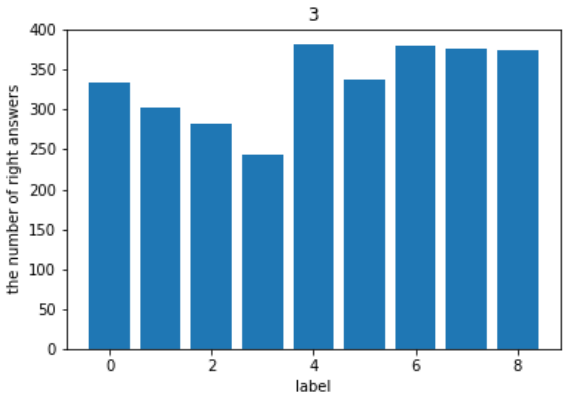

図 4.11: 3. 5:5(前後交換)

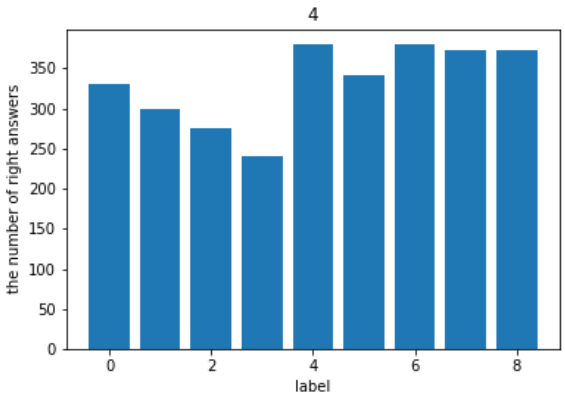

図 4.12: 4. 7:3 と 3:7

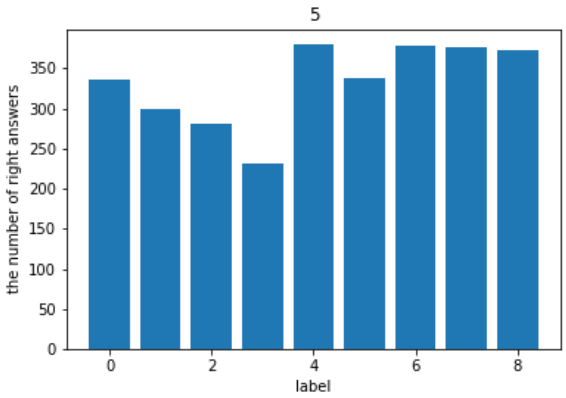

図 4.13: 5. 7:3 と 3:7(前後交換)

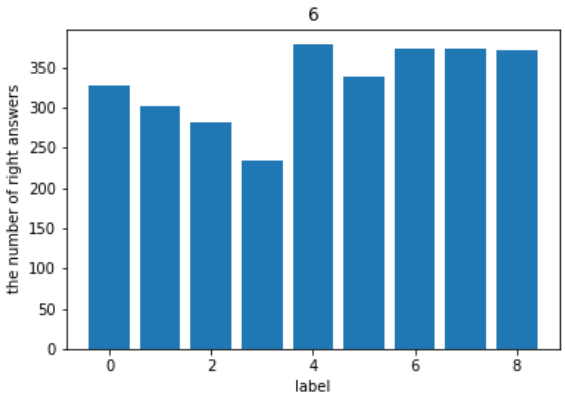

図 4.14: 6. 3:7 と 7:3

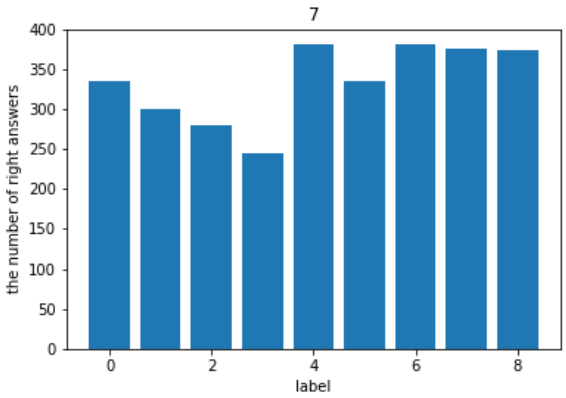

図 4.15: 7. 3:7 と 7:3(前後交換)

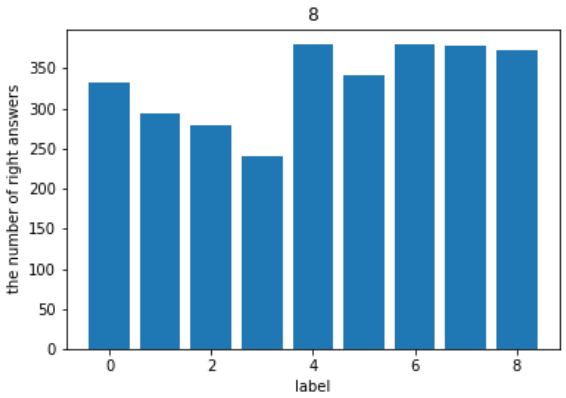

図 4.16: 8. 5:5(label3 とその他)

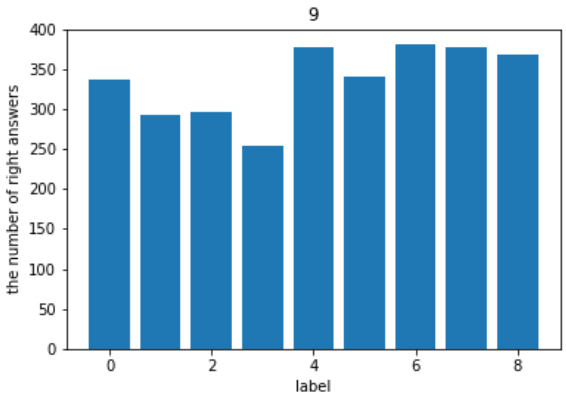

図 4.17: 9. 5:5(同一ラベル同士)

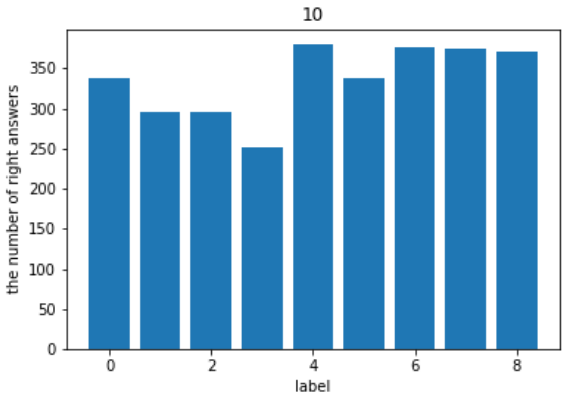

図 4.18: 10. 5:5(同一ラベル同士/前後交換)

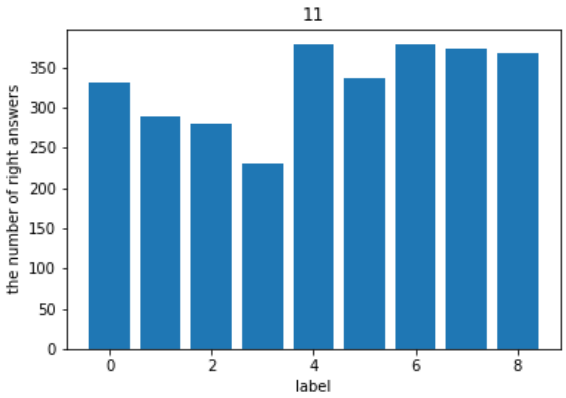

図 4.19: 11. 5:5(label3 同士)

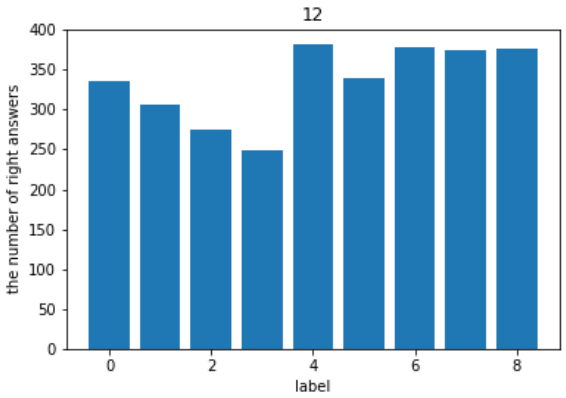

図 4.20: 12. Mix-UP なし (45 データ増加)

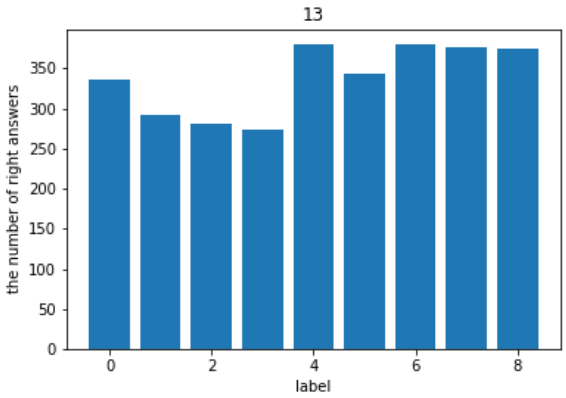

図 4.21: 13. Mix-UP なし (90 データ増加)

第 5 章

考察

実験パターン 2. に比べ 3. が，また，6. に比べ 7. の分類精度が上がっていることから，Mix させる 2 文書の前後の順番によって結果が大きく変わることもあると言える．そして，3. と 12. の結果から，訓練データの初期数が 270 個 (各ラベル 30 個) の状態から，5:5 でランダムに Mix した際には，訓練データを 45 個 (各ラベル 5 個) 増やした場合と同程度の分類精度が出ることがあると証明された．混合割合を 5:5 から 7:3(3:7) に変更して試したが，5:5 にしたものと比べて 7:3(3:7) にしたものは，混合割合の順番や連結する際の前後の順番によっては分類精度が安定せず，5:5 のものと比べ精度の低いモデルが多く作成されている．これは 7:3 のうちの 3 の側の文書の情報が不足していることが原因ではないかと考えられる．よって，この実験手法における混合割合は 2 文書の情報を過不足なく供給できる 5:5 が適切だと言える．

今回，Mix させる文書を同じラベル同士で試したものがベースラインを越え，最も分類精度が高かった．この手法の効果は 9.，10.，12.，13. の結果から，訓練データを 45 個 (各ラベル 5 個) 増やした場合の精度以上かつ，90 個 (各ラベル 10 個) 増やした場合の精度未満とわかる．異なるラベルの文書を Mix した文書は分類難易度が高い曖昧な文書となるため，そうした文書を正しく分類する学習を行ったモデルは分類精度が高くなると予想していたが，その予想に反する結果となった．このことから，曖昧なデータを訓練データに追加するよりも，1 つのラベルに定まった明確なデータを新たなデータとして追加する方が効果的と言える．

また，各ラベルごとの正解率に注目し，特に正解率の低かったラベル (ラベル 3) のデータを増やす実験 (8.，11.) も行った．予想では，ラベル 3 の正解率が上昇し，それに伴い全体の正解率も上昇すると考えていたが，実際にはラベル 3 の正解率に上昇は見られず，

全体の正解率も上昇しなかった (図 4.16，図 4.19)．特に，ラベル 3 同士の Mix のみを行った 11. の結果がベースラインを大きく下回った (図 4.8) ことから，特定ラベルに特化せず，バランスよく全てのラベルのデータを拡張することが効果的と分かった．

第 6 章

結論

本論文では BERT を用いた文書分類タスクに Mix-Up 手法を適用することを試みた．具体的には BERT は 2 文の入力が可能であるため，2 文書の ID 列を連結して入力し，one-hot ベクトルを利用して複数クラスの出力を可能にしたものを教師データとした．livedoor ニュースコーパスを用いた実験で，２文書を混合させる際の選出方法・混合方法を複数パターンで試し，それぞれについての有効性の検証を行った．実験の結果，1. 同じ 2 文書の組み合わせ方でも前後の順序によって結果が変わること，2. 混合の割合は 7:3(3:7) よりも 5:5 が適していること，3. 同一ラベルの文書同士を組み合わせた際に分類精度が向上すること，4. 分類精度の低いラベルのデータに特化してデータ拡張を行っても精度の上昇は見られないこと，の 4 点が分かった．

謝辞

最後に，本論文を作成するにあたり指導・助言を頂いた，指導教官の新納浩幸教授に心より感謝申し上げます．また，普段の勉強会にて指摘・助言を頂いた新納研究室の皆さんに感謝致します．

参考文献

[1] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, Vol. 6, No. 1, p. 60, 2019.

[2] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017.

[3] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study, 2019.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers forlanguage understanding, 2018.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[7] Kikuta Naoki and Hiroyuki Shinnou. Application of mix-up method in document classi?cation task using bert, 2021.