令和 5 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

日本語の NLP タスクに対して有効な Data Augmentation 手法

所属 情報工学専攻

著者 高萩恭介 (22NM730L)

指導教員 新納浩幸教授

令和 6 年 1 月 29 日 (月)

令和 5 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

日本語の NLP タスクに対して有効な Data Augmentation 手法

著者

高萩恭介 (22NM730L)

指導教員

新納浩幸教授

論文要旨

Data Augmentation は，教師あり学習におけるモデルの性能を改善させるために，訓練データを水増しする手法である．Data Augmentation は，Computer Vision の分野において広く研究・利用されているが，自然言語処理においては未発展であるといえる．本論文では，我々がこれまでに考案した日本語の自然言語処理タスクに用いることができる Data Augmentation の手法を二つ取り上げる．一つは，文に含まれる単語を，BERT の Masked Language Modeling を用いて別の単語に置換する手法である．もう一つは，文の係り受け関係が崩れないように文節の順序をシャッフルする手法である．これら 2 つの手法の概要や変換方法について示した後，各手法がどのようなタスクで効果を発揮するのかについて述べる．

Master's Thesis in Scholastic 2023, Major in Computer and Information Sciences,
Graduate School of Science and Engineering, Ibaraki University

Effective Data Augmentation Methods for Japanese NLP Tasks

Author

Kyosuke Takahagi (22NM730L)

Adviser

Prof. Hiroyuki Shinnou

Abstract

Data Augmentation is a technique for augmenting training data to improve model performance in supervised learning. It has been widely studied and used in the field of Computer Vision, but it is still underdeveloped in natural language processing. In this paper, we focus on two data augmentation methods that can be used for Japanese natural language processing tasks. One is to replace a word in a sentence with another word using Masked Language Model of a different BERT from BERT used in analysis and inference. The other is to shuffle the order of phrases so that the dependency relations of sentences are not broken. After providing an overview of each method and its conversion method, we describe what tasks each method is effective for.

目次

表目次

図目次

第 1 章

序論

Data Augmentation (以下，DA) は，機械学習における訓練データの数を増やすための手法であり，モデルを学習する際に，そのモデルの汎化性能を向上させるために利用される．一般的には，既存の訓練データに何らかの変換を施したデータを生成することによって，訓練データを水増しする．

DA においてデータを変換する際には，モデルの学習に悪影響を与えない自然なデータを生成する必要がある．また，教師あり学習において DA を用いるときは，ラベル付きデータのラベルは変換せずにデータのみに変換を施すのが一般的である．そのため，変換後のデータは，元のラベル付きデータのラベルと一貫性を保っている必要がある．

しかし，自然言語処理で扱われるテキストデータは，画像データと比較して複雑な構造を持つため，データに変換を施すことで不自然なデータが生成されたり，ラベルとの一貫性が損なわれたりする可能性が高い．そのため，自然言語処理の分野において，DA を用いてモデルの汎化性能を向上させることは困難であるとされている．ただし，自然言語処理においても，いくつかの効果的な DA の手法が考案されている．

我々はこれまでに，事前学習済みモデルである BERT (Bidirectional Encoder Representations from Transformers) [1] の Masked Language Modeling を用いて，文に含まれる単語を別の単語に置換する手法 [2] と，文の係り受け関係が崩れないように文節の順序をシャッフルする手法 [3] の二つを提案した．また，いくつかの日本語の自然言語処理タスクを解く際に，これらの手法を用いて訓練データセットを拡張することで，モデルの性能が改善することを示した．本論文では，これらの DA の手法における変換方法や手法の効果を検証した実験の結果についてまとめる．また，これらの研究から得られた結果について改めて議論する．

本論文は，本節を含めて 9 節から構成される．2 節では，自然言語処理における DA の研究に対する概況について述べた後，代表的な手法や本研究の提案手法に関連する手法について概観する．3 節では，本研究で提案する 2 つの手法の詳細と，各手法におけるデータ変換の手順について示す．4 節では，提案手法を評価するために本論文で用いられるデータセット・ベンチマークについての概要を示す．5 節では，本論文で利用される事前学習済みの言語モデルについて，その概要を示す．6 節では，2 つの提案手法を評価するために行われる実験の設定について示す．7 節では，6 節で示した設定で行った実験の結果について示す．8 節では，実験で得られた結果をもとに，いくつかの観点から考察を行う．9 節では，本論文で示した研究についての内容とその成果について総括した後に，今後の研究の展望について述べる．

第 2 章

関連研究

2.1 自然言語処理における Data Augmentation

DA は，画像処理分野で広く研究されており，画像の反転や切り抜きなどの幾何学的な変換や色空間の変換，画像の混合など，様々な手法が存在する [4].

一方で，自然言語処理の分野では，画像処理で行われた研究に対して，二次的に研究が行われることが多く，画像処理と比較して DA に関する研究の数もあまり多くない．これは，言語がもつ離散的な性質や複雑な意味・構文構造によって，ラベルを保持したままテキストデータの変換を行うのが難しいことが原因であると考えられている．しかし，自然言語処理においても，ラベルを保持したまま変換できる効果的な手法はいくつか提案されている [5] [6].

2.1.1 EDA

EDA (Easy Data Augmentation) とは，Wei らによって提案された言語モデルや外部データを必要としないシンプルな DA の手法である [7]．具体的には，以下の 4 つの手法を用いて文を修正する．

Synonym replacement（同義語置換）

文中の単語をランダムに選択し，選んだ単語をランダムな同義語に置き換える．これを n 回繰り返す．ただしストップワードは除く．

Random deletion(ランダム削除)

文中の各単語についてそれぞれ確率 p で削除する．

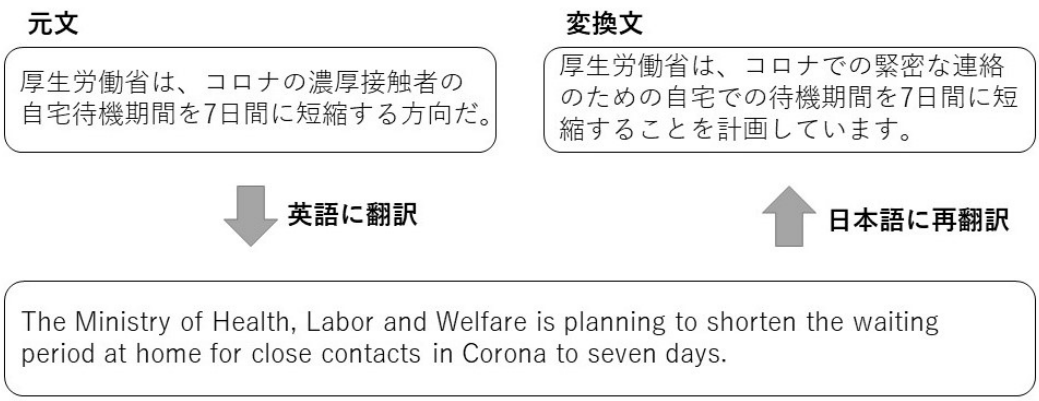

図 2.1: Google 翻訳を利用した逆翻訳の例

Random swap（ランダム入れ替え）

文中の 2 つの単語をランダムに選択し．選んだ単語同士の位置を入れ替える．これを n 回繰り返す．

Random insertion（ランダム挿入）

文中のランダムな単語の同義語をランダムに選び，文中のランダムな位置に挿入する．これを n 回繰り返す．ただしストップワードは除く．

上記の手法において，ある単語に対する同義語の取得には Wordnet という英語の概念辞書が用いられる．また，手法の説明にあるストップワードとは，何らかの処理を行う上で処理対象外となる単語のことである．一般的には，頻繁に登場する単語や存在の有無が文意に大きな影響を与えない単語などがストップワードとなる．

2.1.2　逆翻訳

逆翻訳とは，文を別の言語に翻訳した後，元の言語に翻訳し直すことである（図 2.1）．この逆翻訳は，機械翻訳を中心に多くのタスクで有効な DA の手法であり，広く研究が行われている．[8] [9].

2.1.3　Mixup

Mixup は，訓練データのペアに対してラベルとデータをそれぞれ線形補完し，新たな訓練データを作成する DA の手法である [10]．Mixup は Computer Vision の分野で提案された手法であるが，Gun らはそれを応用し，単語の埋め込み表現を混合する手

アルゴリズム1: Data Augmentation approach

入力: 訓練データセットD_{train}

事前学習済みモデル $G \in \{AE, AR, Seq2Seq\}$

1. Fine-tune G using D_{train} to obtain G_{tuned}
2. $D_{sybthetic} \leftarrow \{\}$
3. foreach $\{x_i, y_i\} \in D_{train}$ do
4. 　Synthesize s examples $\{\hat{x}_i, \hat{y}_i\}_p^1$ using G_{tuned}
5. 　$D_{synthetic} \leftarrow D_{synthetic} \cup \{\hat{x}_i, \hat{y}_i\}_p^1$
6. end

図 2.2: 統一的な Data Augmentation アプローチ

法（wordMixup）と文の埋め込み表現を混合する手法（senMixup）の 2 つを提案している [11].

2.1.4　事前学習済みモデルを使った Data Augmentation

Kumar らは transformer をベースとする事前学習済みモデル（GPT-2/BERT/BART）を用いた DA の手法を提案している [12]．そして，この研究では，異なる 3 つの事前学習済みモデルを使った DA の性能について平等に比較を行うために，図 2.2 のような統一的な DA アプローチが提案されている．また，事前学習済みモデルに対して，ファインチューニングを行う際にクラスラベルをテキスト列に前置することで，DA に効果的な条件を与える方法が示されている．

2.2　事前学習済みモデル

2.2.1　BERT

概要

BERT [1] は Bidirectional Encoder Representations from Transformers の略で, 2018 年に Google が発表した言語モデルである．BERT は当時，多くの自然言語処理タスクにおいて，最先端のモデルの大きく上回るスコアを記録した．BERT は，Attention という方法を用いることで，文脈に応じた処理を行うことができる．また，学習は，事前学習とファインチューニングの 2 段階で行われている．

アーキテクチャ

BERT は，トークン列をベクトル化したものを入力として受け取り，入力列の各トークンに対応するベクトルを出力するモデルである．BERT は図 2.3 のようにいくつかの層から構成される．それぞれの層は各トークンに対応するベクトルを出力し，次の層はそれを受けて，新たに各トークンに対応するベクトルを出力する．

BERT の各層には， Transformer というモデルの Encoder（Transformer Encoder）が用いられている．Transformer Encoder は，おもに Multi-Head Attention と Feed-forward Network という要素で構成されている．そして各層では，Attention(注意機構) という方法を用いて，各トークンの情報を処理するときに他のトークン情報を直接参照する処理を行う．このとき，それぞれのトークンの情報にどの程度注意を払うかは，それぞれのトークンに応じて適応的に決定する．

BERT は，Attention を用いることで，離れた位置にあるトークンの情報も適切に取り入れることができる．そのため，より深く文脈を考慮したトークンの分散表現を獲得することが可能である．また，層内でのそれぞれのトークンに対する出力は独立に計算できる．つまり，並列化による高い計算効率が実現可能である．

事前学習

BERT は事前学習を行うことで，汎用的な言語のパターンを学習することができる．事前学習には大量のラベルなしデータが利用され，そのタスクには以下の 2 つが用いられる．

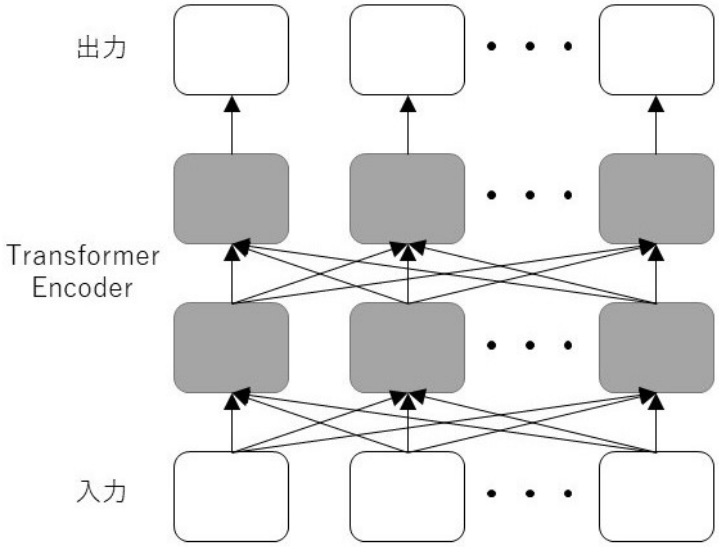

図 2.3: BERT のアーキテクチャ

Masked Language Modeling

文中の一部の単語を隠し，隠した単語が何であるかを予測するタスク．

Next Sentence Prediction

入力された 2 つの文が，連続したものであるか否かを予測するタスク．

ファインチューニング

ファインチューニングとは，ある特定のタスクにおけるラベル付きデータを用いて，BERT がそのタスクに特化するように学習を行うことである．BERT でなんらかのタスクを解く際には，そのタスクに応じて BERT に分類器などを接続するなどして，そのタスクに特化したモデルを作る．つまり，自然言語処理タスクにおいて，BERT は特徴抽出器のような働きをする．ファインチューニングは次の手順で行う．

1. モデルのパラメータの初期値を設定する．
   - BERT のパラメータの初期値…事前学習で得たパラメータ
   - BERT に接続された分類器のパラメータの初期値…ランダムな値
2. ラベル付きデータを用いて，BERT と分類器の両方のパラメータを学習する．

このように，事前学習で得られたパラメータを初期値とすることで，比較的少数の訓練データからでも高い性能のモデルを構築できる．

2.2.2　RoBERTa

RoBERTa [13] は，BERT のアーキテクチャはそのままに，事前学習に用いるデータやハイパーパラメータ等が修正されたモデルであり，BERT を大幅に超える性能をもつ．BERT からの主な変更点は以下の通りである．

- Masked Language Modeling におけるマスキングには，dynamic masking(エポック毎に異なるトークンがマスキングされる) を使用する．
- Next Sentence Prediction を行わない．
- エンコーディングには，文字ではなくバイトをサブワードとする Bite Pair Encoding（BPE）を使用する．
- より大きなバッチサイズで事前学習が行われる．
- 事前学習に用いられるデータの量を増やす．
- 事前学習のステップ数を増やす．

2.3　日本語処理用のツール

2.3.1　MeCab

MeCab(和布蕪)[*1]は，日本語用の形態素解析エンジンである．形態素解析とは，文を意味を持つ最小の単位である形態素に分割し，それぞれの形態素の品詞などを判別する処理のことである．MeCab を用いて日本語の文を形態素解析した結果を図 2.4 に示す．

MeCab による形態素解析は，様々な単語の情報が格納されたデータベースである辞書に基づいて行われる．この辞書には様々な種類があり，用途に応じて利用する辞書を選ぶことができる．

2.3.2　CaboCha

CaboCha は Support Vector Machines に基づく，高性能な日本語係り受け解析器である [14].

[*1] `http://taku910.github.io/mecab/`

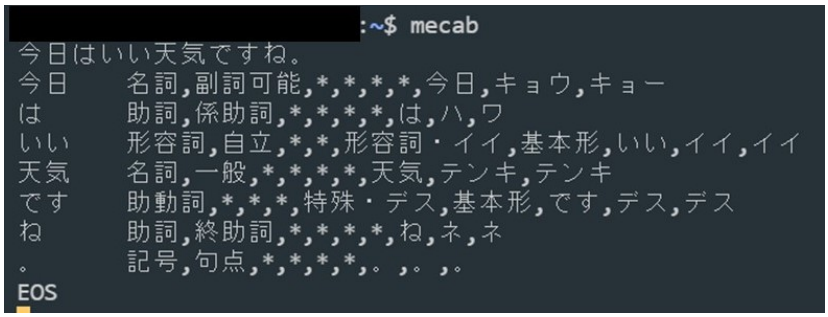

図 2.4: MeCab の実行結果の例

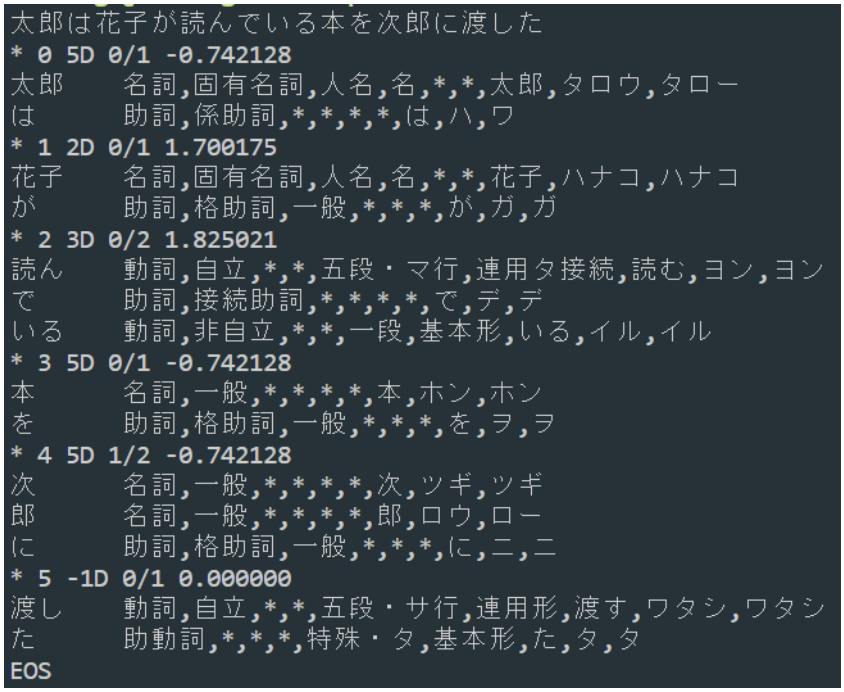

図 2.5: CaboCha による文の解析結果

CaboCha を用いることで，日本語の文を文節ごとに分割し，各文節の係り受けに関する情報を得ることが可能である．実際に例文を CaboCha を用いて解析したときの様子を図 2.5 に示す．

第 3 章

提案手法

3.1 複数の BERT を用いた Data Augmentation

BERT の事前学習で用いられる Masked Language Modeling (MLM) は，一部の単語が [MASK] というトークンで隠された文が入力として与えられ，[MASK] に入る単語を予測するためのモデルである．ここで提案する手法では，文に含まれる一部の単語を [MASK] トークンで覆い，そのトークンに入る単語を MLM で予測する．そして予測された単語で元の単語を置換することで，新たな文を生成する．

しかし，BERT を用いてなんらかのタスクを解く場合，それと同じ BERT の MLM を利用して単語置換を行っても効果はほとんどない．なぜなら，置換によって得られる単語の知識は，タスクを解くために使われる BERT に既に含まれていると考えられるからである．そこで，本手法では，タスクを解くために利用されるモデルと単語置換のために利用されるモデルについて，互いに異なるコーパスで事前訓練された BERT を用いる．このアイデアにより，タスクを解くための BERT には含まれない単語の知識を，DA を行うことで獲得できる．

本手法におけるテキストの変換は以下の手順で行う．

1. BERT の Tokenizer を用いてテキストをトークン化する．
2. テキストを構成するトークンのうち，TF-IDF が最も高い名詞のトークンを選択し，それを [MASK] に置き換える．
3. トークン列を BERT 入力用の ID 列に変換する．
4. ID 列を BERT に入力し，出力される MLM での予測結果を取得する．

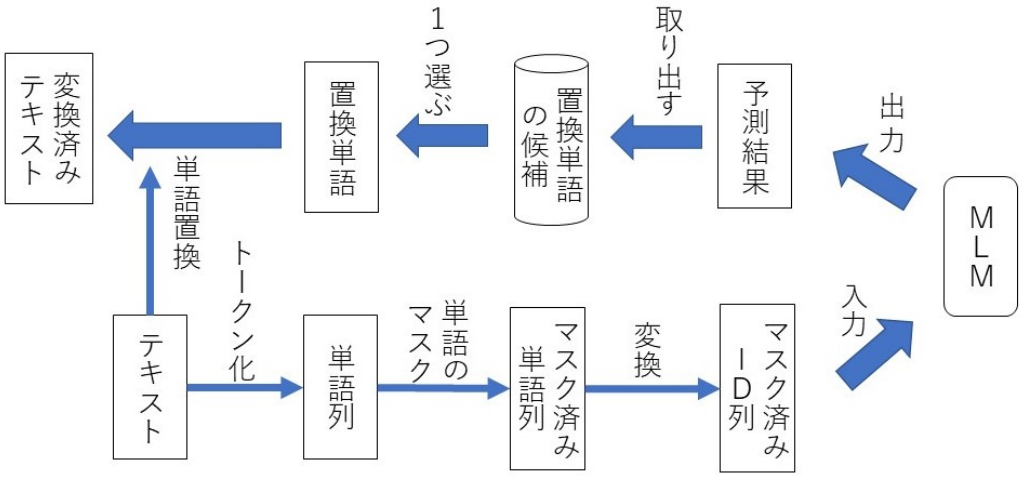

図 3.1: 複数の BERT を用いた DA における変換処理の流れ

5. 予測結果の中から，[MASK] に入ると予測される単語のうち上位 100 件を取得する．
6. 取得単語のうち，次の条件を満たす最上位の単語を選択する．
   （a）その単語は名詞である．
   （b）その単語は置き換え前の単語とは異なっている．
7. 選択した単語で [MASK] を置き換える．

上記の手順 (2) において名詞のトークンを選択するとあるが，トークンが名詞であるかの判定については，MeCab (辞書は mecab-ipadic-NEologd) に入力し，そのトークンの品詞を特定することによって実施する．また，トークンを mecab に入力したときに，万が一トークンがさらに複数の形態素に分割される場合は 1 つ目の形態素の品詞で判定を行う．さらに，拡張に用いられる BERT モデルがサブワードまで分割を行うトークナイザを利用している場合，サブワード単位で TF-IDF の計算を行うため，サブワード自体を一つのトークンとして扱う．そのため，[MASK] への置き換え対象のトークンがサブワードであった場合は，そのサブワード自体に対して名詞かどうかの判定を行う．

図 3.1 はこの変換の流れを視覚的に表したものである．

3.2　文節シャッフルによる Data Augmentation

自然言語処理における簡単な DA の手法として，文をより細かい単位に分解し，それらの順序を入れ替えるという方法が考えられる．しかし，この方法を日本語の文に適用した場合，文法的におかしい不自然な文や元の文と異なる意味を表す文等の，学習を行う上でノイズとなるような文が生成されてしまう可能性が高い．そこで提案手法では，日本語の文を文節単位に分解後，係り受け関係が崩れないようにシャッフルを行う．これにより，元の文と同じ意味を持つ自然な文を生成することができる．

本手法におけるテキストの変換は以下の手順で行う．

1. (テキストが文章の場合) 文単位に分割し，各文に対して以下の処理を行う．
2. 文を文節単位に分割する．
3. 係り先が述語ではない文節 (述語は除く) をそれに続く文節と連結する．
4. 連結後の節の順番をシャッフルする (述語の位置は最後尾のままにする)．

図 3.2 は例文に対して実際に変換処理を行ったときの様子を表したものである．また，例文を係り受けの木構造で表すと図 3.3 のようになる．上記の手順 (3) を行うことで，図 3.3 からわかるように，述語を除くすべての節は述語に係るようになる．そのため，それらの節の順番を入れ替えても係り受け関係は保持される．

文の文節への分割と各文節の係り先の特定には，Support Vector Machines に基づく日本語係り受け解析器である CaboCha [14] を用いる．CaboCha を用いることで，入力として受け取った日本語の文を文節ごとに分割し，各文節の係り受けに関する情報を出力することが可能である．

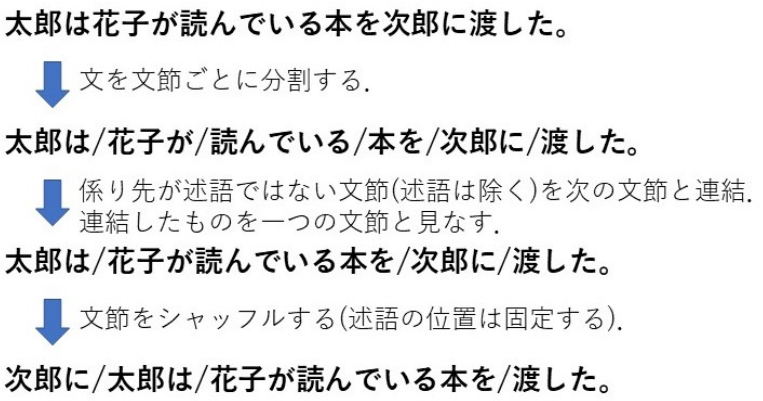

図 3.2: 文節シャッフルにおける変換処理の様子

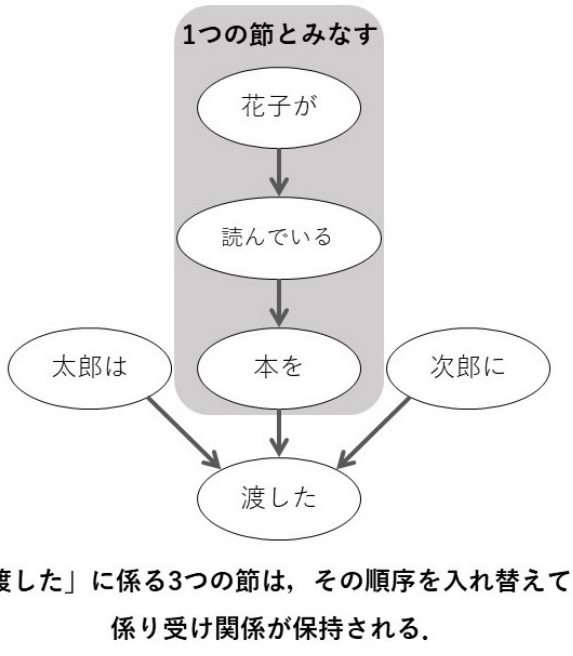

図 3.3: 例文を係り受けの木構造で表した図

第 4 章

データセット・ベンチマーク

4.1 livedoor ニュースコーパス

livedoor ニュースコーパス[*1]は，NHN Japan 株式会社が運営する「livedoor ニュース」の中からニュース記事を収集し，可能な限り HTML タグを取り除いて作成したものである．ニュース記事は合計で 7367 件存在し，各ニュース記事には下記 9 種類のカテゴリが割り振られている．

- 独女通信
- IT ライフハック
- 家電チャンネル
- livedoor HOMME
- MOVIE ENTER
- Peachy
- エスマックス
- Sports Watch
- トピックニュース

本論文では，この 9 つのカテゴリに対してそれぞれ 0 から 8 までのラベルを割り当てる．そして，ニュース記事とその記事のカテゴリに割り当てられたラベルを併せて，ラベル付きデータとする．このラベル付きデータを一定数用意して，文書分類用のデータセッ

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

表 4.1: JGLUE の構成

タスク	データセット	train	dev	test
文書分類	MARC-ja	187,528	5,654	5,639
	JCoLA	-	-	-
文ペア分類	JSTS	12,463	1,457	1,589
	JNLI	20,117	2,434	2,508
QA	JSQuAD	63,870	4,475	4,470
	JCommonsenseQA	9,012	1,126	1,126

トとして利用する．この livedoor ニュースコーパスを用いた文書分類は，複数の BERT を用いた DA と文節シャッフルによる DA の両方の評価に用いる．

4.2　JGLUE

JGLUE は，日本語の言語理解ベンチマークである [15]．このベンチマークは文書分類タスクである MARC-ja と JCoLA，文ペア分類タスクである JSTS と JNLI，QA タスクである JSQuAD と JCommonsenseQA の計 6 つのタスクから構成される (表 4.1)．今回は，この中から MARC-ja，JSTS，JCommonsenseQA の 3 つのタスクを利用し，複数の BERT を用いた DA を評価する．評価に利用しないタスクについては説明を省略する．

4.2.1　Marc-ja

MARC-ja は文書分類用データセットであり, 通信販売サイト「アマゾン」における商品レビューとそれに対する評価をまとめたコーパスである MARC (Multilingual Amazon ReviewsCorpus) [16] の日本語部分を元に構築されている．MARC-ja に含まれる各データは商品レビューと，ラベルである評価の二つからなる．タスクは，ラベルが negative と positive かを当てる二値分類となっている．評価指標には正解率 (acc) が用いられている．

4.2.2　JSTS

意味的類似度計算 (Semantic Textual Similarity, STS) とは，文ペアの意味的な類似度を推定するタスクである．JSTS は STS のデータセットであり，YJ Captions Dataset [17] を利用して構築されている．JSTS の各データは文ペアとその類似度からなる．文ペアは YJ Captions Dataset に含まれる画像に対する 2 つのキャプションである．類似度は 0 (意味が完全に異なる) 〜5 (意味が等価) の実数値となっており，その値はクラウドソーシングによって決定されたものである．評価指標には Pearson および Spearman 相関係数が用いられている．

4.2.3　JCommonsenseQA

JCommonsenseQA は，CommonsenseQA [18] という QA データセットの日本語版で，常識推論能力を評価することができる．JCommonsenseQA に含まれる各データは，問題文とそれに対する 5 つの選択肢，正解の選択肢を示すラベルから構成される．この選択肢のうち，問題文に対する正しい解答となるのは 1 つだけである．評価指標には正解率 (acc) が用いられている．

4.3　JSICK

JSICK は，英語の含意関係認識・意味的類似度判定用データセットである SICK (Sentences Involving Compositional Knowledge) [19] を，人手で日本語に翻訳し，含意関係と意味的類似度の正解ラベルを再アノテーションしたものである [20].

JSICK における含意関係の定義は元の SICK データセットの定義に準拠している．前提文 T と仮説文 H のペア (T,H) に対して，文 T が真であるとき文 H が必ず真になる場合は「含意」ラベルが，文 H が必ず偽になる場合は「矛盾」ラベルが，どちらともいえない (文 T が真であるとしても文 H の真偽はわからない) 場合は「中立」ラベルが付与されている．

JSICK は，予め訓練データとテストデータに分けられており，訓練データのうちのおよそ 10% が検証データとして利用される．JSICK に含まれるデータの例を表 4.2 に，各データの量を表 4.3 にそれぞれ示す．1 文あたりの平均単語数は 13.2 単語，語彙数は

表 4.2: JSICK に含まれるデータの例

文ペア（前提文 T，仮説文 H）	関係
T: 若い女性がギターを弾いている H: 女の子がギターを弾いている	含意
T: トラが織の外側を歩いている H: トラが織の中を歩き回っている	矛盾
T: 男性がジャガイモを切っている H: 男性がトマトを切っている	中立

表 4.3: JSICK の各データの量

	訓練	検証	テスト	合計
含意	991 (22.0%)	100 (20.0%)	1,088 (22.1%)	2,179 (22.0%)
矛盾	748 (16.6%)	75 (15.0%)	797 (16.2%)	1,620 (16.3%)
中立	2761 (61.4%)	325 (65.0%)	3,042 (61.7%)	6,128 (61.7%)
合計	4,500	500	4,927	9,927

2,432 である．含意関係ラベルは中立が多い傾向にある．

本論文では，このデータセットを用いた含意関係認識によって，文節シャッフルによる DA を評価する．

第 5 章

モデル

本論文では，いくつかの自然言語処理のタスクによって，手法の評価を行う．このとき，タスクを解くためのモデルには，事前学習済みモデルである BERT もしくは RoBERTa [13] を用いる．また，実験で実際に使用するモデルは下記 4 種類のうちのいずれかである．

- cl-tohoku/bert-base-japanese[*1] (以下，tohoku-BERT)
- cl-tohoku/bert-base-japanese-v2[*2] (以下，tohoku-BERT-v2)
- ストックマーク株式会社が公開した BERT モデル[*3] (以下，stockmark-BERT)
- rinna/japanese-roberta-base[*4] (以下，rinna-RoBERTa)

tohoku-BERT は，東北大学自然言語処理グループが公開している BERT モデルである．このモデルは，事前学習用データに日本語 Wikipedia，トークナイザには MeCab (IPA 辞書) と WordPiece が用いられている．tohoku-BERT-v2 は，tohoku-BERT から，トークン化で利用される辞書が IPA 辞書ではなく Unidic 2.1.2 辞書に変更され，事前学習に用いられるデータの数が増加している．stockmark-BERT は，ビジネスに関するドメイン向けの BERT モデルであり，事前学習用データに日本語のビジネスニュース記事，トークナイザには MeCab (NEologd) が用いられている．rinna-RoBERTa は，rinna 株式会社が公開している RoBERTa モデルである．このモデルは，事前学習用デー

[*1] https://huggingface.co/cl-tohoku/bert-base-japanese
[*2] https://huggingface.co/cl-tohoku/bert-base-japanese-v2
[*3] https://qiita.com/mkt3/items/3c1278339ff1bcc0187f
[*4] https://huggingface.co/rinna/japanese-roberta-base

タに日本語 CC-100 と日本語 Wikipedia，トークナイザには sentencepiece が用いられている．

第 6 章

実験設定

6.1 複数の BERT を用いた DA を評価する際の設定

複数の BERT を用いた DA は, livedoor ニュースコーパスを用いた文書分類と JGLUE における 3 つのタスクによってそれぞれ評価を行う．それらの詳細な設定については，6.1.1 節と 6.1.2 節にそれぞれ示す.

6.1.1 文書分類 (livedoor ニュースコーパス) による評価の設定

複数の BERT を用いた DA の評価に用いる文書分類データセットの詳細を表 6.1 に示す．訓練・検証・テストデータセットには各 270 件ずつデータが含まれている．また，それぞれのデータセットには 9 つのラベルのデータが同じ数だけ含まれている.

簡単な操作による DA は，訓練データの数が限られている場合にモデルの性能を向上させる効果が顕著に表れ，訓練データの数が十分に多い場合はその効果がわずかであると考えられている [7]．このタスクでは，訓練データ量が原因で DA による効果を確認できない可能性を考慮し，訓練データの量を少量に設定している.

DA によって訓練データを変換するときは，ラベル (ニュース記事のカテゴリ) は元のままでデータ (ニュース記事のテキスト) のみに変換を加える．この変換処理を訓練データ 270 件のうち 100 件に対して行うため，拡張後の訓練データの数は 370 件 (元のデータ 270 件 + データの変換によって生成されるデータ 100 件) に増加する.

DA における MLM を用いた単語置換には，tohoku-BERT と stockmark-BERT の 2 種類を利用する．一方で，文書分類を解くためのモデルとしても，tohoku-BERT と

表 6.1: 複数の BERT を用いた DA の評価に用いる文書分類データセット (livedoor ニュースコーパス) の詳細

ラベル	カテゴリ	訓練	検証	テスト
0	独女通信	30	30	30
1	IT ライフハック	30	30	30
2	家電チャンネル	30	30	30
3	livedoor HOMME	30	30	30
4	MOVIE ENTER	30	30	30
5	Peachy	30	30	30
6	エスマックス	30	30	30
7	Sports Watch	30	30	30
8	トピックニュース	30	30	30
	合計	270	270	270

stockmark-BERT の 2 種類を用いる．モデルのトレーニングでは，損失関数に交差エントロピー誤差を，最適化アルゴリズムに SGD (確率的勾配降下法) を採用した．またハイパーパラメータは下記のように設定した．

- 学習率：1e-3
- バッチサイズ：2
- エポック数：Early Stopping (patience=5) を用いて決定

モデルの性能については，テストデータを分類した際の正解率 (acc) で評価する (モデルは 1 種類につき 5 つ構築し，その 5 つの中で検証データに対する正解率が最も高かったものを評価に用いる)．

6.1.2　JGLUE による評価の設定

今回は，JGLUE に含まれるタスクのうち計 3 つ (MARC-ja，JSTS，JCommonsenseQA) を評価に利用する．そして，その 3 つのタスク全てにおいて，学習に用いる訓練データの数は 6.1 節と同様の理由で少量に設定する．具体的には，3 つのタスクの訓練

データをそれぞれ 100 件ずつとする．また，JGLUE における各タスクには，train (訓練) /dev (検証) /test (テスト) データの 3 つが用意されているが，論文執筆時点ではどのタスクにおいてもテストデータが公開されていない．そのため，各タスクにおいて，元の訓練データのうち実験に使わないデータの一部を検証データとして，元の検証データをテストデータとして使用することにする．各データセットの詳細は表 6.2 に示す．また，3 つのタスクそれぞれにおける DA を行う際の訓練データの変換方法を下記に示す．

- MARC-ja (文書分類) では，ラベル (negative/positive) は元のままでデータ (商品レビュー) のみに対して変換を行う．
- JSTS (意味的類似度計算) では，類似度は元のままで文ペアのうち片方にのみ変換を行う．
- JCommonsenseQA (QA) では，質問文に対する 5 つの選択肢とラベル (正解の選択肢) は元のままで，質問文のみに対して変換を行う．

いずれのタスクにおいても，訓練データ 100 件全てに対して変換を行うため，拡張後の訓練データ数は 200 件 (元のデータ 100 件 + データの変換によって生成されるデータ 100 件) に増加する．

DA における MLM を用いた単語置換には，3 つのタスク全てで stockmark-BERT を利用する．一方，タスクを解くためのモデルとしては，3 つのタスク全てで tohoku-BERT-v2 を用いる．モデルのトレーニングにおける各種設定は，全てのタスクについてバッチサイズを 8 とし，他の設定に関しては JGLUE を公開しているページ[*1]を参考にした．また，モデルは 1 種類につき 5 つ構築し，その 5 つのモデルの評価の平均値を最終的な評価に用いる．

[*1] `https://github.com/yahoojapan/JGLUE/tree/main/fine-tuning`

表 6.2: 複数の BERT を用いた DA の評価に用いる 3 つのデータセット (JGLUE を元に作成) の構成.

データセット	訓練	検証	テスト
MARC-ja	100	5,654	5,654
JSTS	100	1,457	1,457
JCommonsenseQA	100	1,126	1,126

6.2　文節シャッフルによる DA を評価する際の設定

文節シャッフルによる DA は，livedoor ニュースコーパスを用いた文書分類と JSICK を用いた含意関係認識によってそれぞれ評価を行う．それらの詳細な設定については，6.2.1 節と 6.2.2 節にそれぞれ示す.

6.2.1　文書分類 (livedoor ニュースコーパス) による評価の設定

文節シャッフルによる DA の評価に用いる文書分類データセットの詳細を表 6.3 に示す．訓練データセットには 90 件，検証・テストデータセットには各 900 件ずつデータが含まれている．訓練データの量が少量であるのは，6.1 節と同様の理由である．また，それぞれのデータセットには 9 つのラベルのデータが同じ数だけ含まれている.

DA によって訓練データを変換するときは，ラベル (ニュース記事のカテゴリ) は元のままでデータ (ニュース記事のテキスト) のみに変換を加える．訓練データ 90 件全てに対してこの変換を行うため，拡張後の訓練データの数は 180 件 (元のデータ 90 件 + データの変換によって生成されるデータ 90 件) に増加する.

文書分類を解くためのモデルには，tohoku-BERT を用いる．モデルのトレーニングでは，損失関数に交差エントロピー誤差を，最適化アルゴリズムに SGD (確率的勾配降下法) を採用した．またハイパーパラメータは下記のように設定した.

- 学習率：1e-3
- バッチサイズ：2
- エポック数：Early Stopping (patience=5) を用いて決定

表 6.3: 文節シャッフルによる DA の評価に用いる文書分類データセット (livedoor ニュースコーパス) の詳細

ラベル	カテゴリ	train	val	test
0	独女通信	10	100	100
1	IT ライフハック	10	100	100
2	家電チャンネル	10	100	100
3	livedoor HOMME	10	100	100
4	MOVIE ENTER	10	100	100
5	Peachy	10	100	100
6	エスマックス	10	100	100
7	Sports Watch	10	100	100
8	トピックニュース	10	100	100
	合計	90	900	900

モデルの性能については，テストデータを分類した際の正解率 (acc) で評価する (モデルは 1 種類につき 5 つ構築し，その 5 つのモデルの平均値を評価に用いる).

6.2.2　JSICK による評価の設定

文節シャッフルによる DA の評価に用いる JSICK データセットには，表 4.3 のものをそのまま利用する.

DA による訓練データの変換では，1 つのデータに対して下記の 3 通りのデータを作成する.

- ラベル・前提文 T・仮説文 H のうち，前提文 T のみを変換したデータ.
- ラベル・前提文 T・仮説文 H のうち，仮説文 H のみを変換したデータ.
- ラベル・前提文 T・仮説文 H のうち，前提文 T と仮説文 H を変換したデータ.

全ての訓練データに対して上記 3 つの拡張データを生成するため，拡張後の訓練データの数は元の 4 倍の数まで増加する.

含意関係認識を解くためのモデルには，tohoku-BERT-v2 と rinna-RoBERTa の 2 つ

を用いる．モデルのトレーニングでは，損失関数に交差エントロピー誤差を，最適化アルゴリズムに SGD (確率的勾配降下法) を採用した．またハイパーパラメータは下記のように設定した．

- 学習率：1e-3
- バッチサイズ：8
- エポック数：Early Stopping (patience=10) を用いて決定

トレーニング後のモデルの性能は，正解率 (acc) で評価する (モデルは 1 種類につき 5 つ構築し，その 5 つのモデルの平均値を評価に用いる)．このとき，テストデータ全体における正解率だけではなく，各ラベル (含意・矛盾・中立) をもつデータごとの正解率も調べる．

第 7 章

実験結果

7.1 複数の BERT を用いた DA の実験結果

7.1.1 文書分類 (livedoor ニュースコーパス) による評価結果

文書分類 (livedoor ニュースコーパス) での手法の評価は，6.1.1 節に示した設定で実施した．その実験結果を表 7.1 に示す．tohoku-BERT をモデルとして文書分類を解く場合，stockmark-BERT で訓練データを拡張したモデルの正解率は，拡張なしのモデルと比較して 0.7 ポイント高くなった．一方で，tohoku-BERT で訓練データを拡張したモデルの正解率は，拡張なしのモデルと比較して 1.5 ポイント低くなった．stockmark-BERT をモデルとして文書分類を解く場合，tohoku-BERT で訓練データを拡張したモデルの正解率は，拡張なしのモデルと比較して 3.0 ポイント高くなった．また，stockmark-BERT で訓練データを拡張したモデルの正解率は，拡張なしのモデルと比較して 2.6 ポイント高くなった．結果として，2 種類の BERT どちらをモデルとして使った場合でも，もう一方の BERT で訓練データを拡張したモデルの性能が最も高くなった．

7.1.2 JGLUE による評価結果

JGLUE に含まれる 3 つのタスク (MARC-ja，JSTS，JCommonsenseQA) での手法の評価は，6.1.2 節に示した設定で実施した．その実験結果を表 7.2 に示す．MARC-ja においては，訓練データを拡張したモデルの正解率のほうが，拡張なしのモデルと比較して 0.5 ポイント高かった．一方で JSTS においては，訓練データを拡張したモデルの Pearson/Spearman 相関係数が，拡張なしのモデルと比較してそれぞれ 0.008/0.009 低

表 7.1: 複数の BERT を用いた DA の実験結果．タスクは文書分類 (livedoor ニュース).

モデル	訓練データ	正解率
tohoku-BERT	拡張なし	0.863
	tohoku-BERT で拡張	0.848
	stockmark-BERT で拡張	**0.870**
stockmark-BERT	拡張なし	0.833
	tohoku-BERT で拡張	**0.863**
	stockmark-BERT で拡張	0.859

表 7.2: 複数の BERT を用いた DA の実験結果．タスクは JGLUE の MARC-ja，JSTS，JCommonsenseQA．モデルは tohoku-BERT-v2

タスク (評価指標)	拡張なし	拡張あり
MARC-ja (acc)	0.856±0.004	**0.861±0.011**
JSTS (Pearson/Spearman)	**0.779±0.012/0.689±0.006**	0.771±0.022/0.680±0.025
JCommonsenseQA (acc)	**0.679±0.008**	0.667±0.011

数値の表記は，平均値 ± 標準偏差.

くなった．また，JCommonsenseQA においても，訓練データを拡張したモデルの正解率が，拡張なしのモデルと比較して 1.2 ポイント低くなった．以上より，手法は文書分類・文ペア分類・QA のうち文書分類のみに有効であることが分かった.

表 7.3: 文節シャッフルによる DA の実験結果．タスクは文書分類 (livedoor ニュース)．

モデル	訓練データ	正解率
tohoku-BERT	拡張なし	0.734±0.006
	拡張あり	**0.751±0.002**

数値の表記は，平均値 ± 標準偏差．

7.2　文節シャッフルによる DA の実験結果

7.2.1　文書分類 (livedoor ニュースコーパス) による評価結果

文書分類 (livedoor ニュースコーパス) での手法の評価は，6.2.1 節に示した設定で実施した．その実験結果を表 7.3 に示す．結果としては，訓練データを拡張したモデルの正解率は，拡張なしのモデルと比べて 1.7 ポイント高くなった．

7.2.2　JSICK による評価結果

JSICK を用いた含意関係認識での手法の評価は，6.2.2 節に示した設定で実施した．その実験結果を表 7.4 に示す．含意関係認識を解くモデルとして rinna-RoBERTa を用いた場合，訓練データを拡張したモデルの正解率 (全体) は，拡張なしのモデルと比較して 0.8 ポイント高くなった．このとき，ラベル別の正解率の上昇幅を確認すると，矛盾データの正解率の上昇幅が 3.4 ポイントと最も大きかった一方で，中立データは拡張の有無によって正解率があまり変化しなかった．また，含意関係認識を解くモデルとして tohoku-BERT-v2 を用いた場合，訓練データを拡張したモデルの正解率 (全体) は，拡張なしのモデルと比較して 0.5 ポイント上昇した．また，ラベル別の正解率の上昇幅を確認すると，rinna-RoBERTa と同様に矛盾データの正解率の上昇幅が最も大きく，1.8 ポイントであった．一方で，含意データの正解率は 0.4 ポイント下がっていた．以上から，tohoku-BERT-v2 と rinna-RoBERTa どちらをモデルとして利用した場合も，手法を用いることでモデルの性能が向上することが分かった．また，ラベル別に性能を確認すると特に矛盾データを識別する性能が向上することが分かった．

表 7.4: 文節シャッフルによる DA の実験結果．タスクは含意関係認識 (JSICK)．評価指標は acc.

モデル	訓練データ	全体	含意	矛盾	中立
rinna-RoBERTa	拡張なし	0.890±0.007	0.854±0.012	0.779±0.042	0.932±0.011
	拡張あり	**0.898±0.003**	**0.860±0.030**	**0.814±0.024**	**0.934±0.013**
tohuku-BERT-v2	拡張なし	0.885±0.006	**0.853±0.040**	0.812±0.029	0.916±0.022
	拡張あり	**0.890±0.002**	0.849±0.028	**0.830±0.013**	**0.921±0.006**

数値の表記は，平均値 ± 標準偏差.

第 8 章

考察

8.1 訓練データ量と Data Augmentation による効果の関係

本論文で扱う手法のような簡易的な DA は，元の訓練データの量が少ない場合に効果的であり，量が十分な場合にはあまり効果がないと考えられている [7]．本節ではこの点を確認するために，訓練データを少量に設定して実験を行ったいくつかのタスクについて，その量を拡大して同様の実験を行う．そして，訓練データ量の大小で DA の効果に差が出るかについて調査する．

まず，複数の BERT を用いた DA を文書分類 (livedoor ニュースコーパス) で評価したケースについて，訓練データの量を元の 2 倍の量である 540 件に拡大して再実験した．このとき，拡張訓練データの数も元の 2 倍である 200 件に拡大している．なお，実験に使うモデルは tohoku-BERT のみ，訓練データは拡張なしの場合と stockmark-BERT で拡張した場合のみとする．結果としては，元の訓練データの量を増やした場合，DA を行ってもモデルの性能が向上することはなかった (表 8.1)．

次に，複数の BERT を用いた DA を JGLUE の各タスクで評価したケースについて，

表 8.1: 訓練データ量と DA による効果の関係調査．手法は複数の BERT を用いた DA．タスクは文書分類 (livedoor ニュース) ．評価指標は acc.

モデル	訓練データ量	拡張なし	拡張あり	正解率の差
tohoku-BERT	270	0.863	0.870	+0.007
	540	0.889	0.889	± 0.000

表 8.2: 訓練データ量と DA による効果の関係調査．手法は複数の BERT を用いた DA．タスクは JGLUE の MARC-ja，JSTS，JCommonsenseQA．モデルは tohoku-BERT-v2.

タスク (評価指標)	データ量	拡張なし	拡張あり	評価値の差
MARC-ja (acc)	100	0.856±0.004	0.861±0.011	+0.005
	1000	0.913±0.005	0.911±0.008	-0.002
JSTS (Pearson/Spearman)	100	0.779±0.012/0.689±0.006	0.771±0.022/0.680±0.025	-0.008/-0.009
	1000	0.848±0.004/0.800±0.008	0.833±0.006/0.780±0.011	-0.015/-0.020
JCommonsenseQA (acc)	100	0.679±0.008	0.667±0.011	-0.012
	1000	0.728±0.010	0.702±0.013	-0.026

拡張なし・拡張ありにおける数値の表記は，平均値 ± 標準偏差.

表 8.3: 訓練データ量と DA による効果の関係調査．手法は文節シャッフルによる DA．タスクは文書分類 (livedoor ニュース)．評価指標は acc.

モデル	データ量	拡張なし	拡張あり	正解率の差
tohoku-BERT	90	0.734±0.006	0.751±0.002	+0.017
	450	0.837±0.006	0.842±0.010	+0.005
	900	0.879±0.006	0.880±0.005	+0.001

拡張なし・拡張ありにおける数値の表記は，平均値 ± 標準偏差.

各タスクの訓練データの量を元の 10 倍である 1000 件に拡大して実験を行った．その結果としては，どのタスクにおいても，元の訓練データの量が大きくなることで，DA によるモデルへの影響がマイナスの方向に大きくなった (表 8.2).

最後に，文節シャッフルによる DA を文書分類 (livedoor ニュースコーパス) で評価したケースについて，訓練データの量を元の 5 倍，10 倍の量である 450 件，900 件に拡大して実験を行った．このケースでは，元の訓練データの量が大きくなるほど，DA によるモデルの性能向上の効果が小さくなるという結果になった (表 8.3).

以上の結果から，本論文で扱う 2 つの手法はどちらもタスクにおける元の訓練データの量が少量である場合に効果が高く，元の訓練データの量が大きくなるほど効果が出にくくなると考えられる.

7.2 節において，文節シャッフルによる DA を JSICK (含意関係認識) で評価した際は，データセットに含まれる矛盾のラベルをもつデータが含意・中立のラベルをもつデー

タと比較して少なかった．つまり，DA によって特に矛盾データに対して識別性能が向上していたのは，この性質が原因であったと考えられる．

8.2 複数の BERT を用いた DA における BERT 間の単語の重なり

複数の BERT を用いた DA が効果を発揮するのは，2 つの BERT で事前学習によって獲得した知識が異なるからである．本節では，ある文に含まれる一部の単語を [MASK] で隠したとき，そこに入ると予測される単語は，実験で使用した 2 つの BERT である tohoku-BERT と stockmark-BERT でどの程度異なるかを，その重なり具合から調べる．

まず，livedoor ニュースコーパスを用いた文書分類において，拡張を行った 100 件のテキストを用意する．そして，各テキストに対して，TF-IDF が高い単語を [MASK] で隠す処理を行う．その後，隠した部分の予測を，tohoku-BERT と stockmark-BERT の MLM でそれぞれ行う．最後に，それぞれの BERT が予測した上位 5 単語同士を比較し，単語の順位は考慮せずに，2 つの BERT 間で 5 単語中何単語重なるかを調査する．

この実験の結果を表 8.4 に示す．調査した 100 件において，2 つの BERT がそれぞれ予測した上位 5 単語同士は，平均で 5 個中 1.64 個，割合にすると 32.8 %重なることが分かった．

このときの単語が重なっていた分については，tohoku-BERT と stockmark-BERT で似た知識を保有していたと考えられる．一方で，単語が重ならなかった分については，それぞれの BERT で異なる知識を保有していたと考えることができる．そのため，今回の livedoor ニュースコーパスを用いた文書分類の実験では，2 つの BERT がそれぞれ異なる知識を保有しており，それらが合わさることでモデルの性能が向上したと考えられる．

また今回は，tohoku-BERT と stockmark-BERT を利用したが，これらの BERT 以外にも様々な BERT が存在する．そのため，BERT 同士の知識があまり重ならないような 2 種類の BERT のペアを選択して，今回と同じように DA を行えば，今回以上の効果が期待できる．また，DA に使う BERT の種類を 1 種類ではなく多種類に増やすことによっても，モデルの性能向上の効果がより高まることが期待できる．

表 8.4: 2 つの BERT が [MASK] に入ると予測した上位 5 単語同士は何個重なるのか．100 件のテキストで調査．

重なった数 (N/5)	0	1	2	3	4	5
該当する件数 (N/100)	13	33	34	17	3	0

8.3　文節シャッフルによる DA と類似手法の性能比較

本手法は，日本語の文に対して係り受け関係を崩さないように変換を行うことで，元の文の意味を保った自然な文を生成することができる．しかし，その特徴が含意関係認識モデルの性能を改善させる要因になっているのかは不明である．本節では，本手法と係り受け関係を考慮せずに文の構成要素の順番をシャッフルする手法を比較する．この比較によって，DA における文の変換時に，元の文の意味を保った自然な文を生成することがモデルの性能を改善する要因となっているかについて検証する．

本手法との比較に用いる手法は，係り受け関係を無視して文節単位で文をシャッフルする手法 (以下，ランダムな文節シャッフル) と単語単位で文をシャッフルする手法 (以下，単語シャッフル) の 2 つとする．ランダムな文節シャッフルでは，文を文節に分解した後に述語以外の文節の順番をシャッフルする．また，単語シャッフルでは，文を単語に分割した後に単語の順番をシャッフルする．なお，ランダムな文節シャッフルにおける文節分割には CaboCha を，単語シャッフルにおける単語分割には MeCab[*1] という形態素解析エンジンを使用する．

比較のための実験は，livedoor ニュースコーパスを用いた文書分類と，JSICK による含意関係認識で行う．このときに使用するデータセットは，6.2.1 節，6.2.2 節で述べたものと同様である．

まず，livedoor ニュースコーパスを用いた文書分類による実験結果を表 7.3 の結果と併せて表 8.5 に示す．このとき，文書分類を行うモデルには tohoku-BERT を用いる．実験結果によると，係り受けを考慮した文節シャッフルを利用したモデルの性能が最も高くなった．ランダムな文節シャッフルを適用したモデルも，DA なしのモデルと比べると性能が向上したものの，係り受けを考慮した文節シャッフルには及ばなかった．単語

[*1] https://taku910.github.io/mecab/

表 8.5: 文節シャッフルによる DA の類似手法との性能比較．タスクは文書分類 (livedoor ニュース).

モデル	訓練データ拡張に用いた手法	正解率
tohoku-BERT	拡張なし	0.734±0.006
	文節シャッフル (係り受け考慮)	**0.751±0.002**
	ランダムな文節シャッフル	0.746±0.009
	単語シャッフル	0.734±0.010

数値の表記は，平均値 ± 標準偏差.

表 8.6: 文節シャッフルによる DA の類似手法との性能比較．タスクは含意関係認識 (JSICK)．評価指標は acc．モデルは rinna-RoBERTa.

訓練データ拡張に用いた手法	全体	含意	矛盾	中立
拡張なし	0.890±0.007	0.854±0.012	0.779±0.042	0.932±0.011
文節シャッフル (係り受け考慮)	**0.898±0.003**	**0.860±0.030**	**0.814±0.024**	0.934±0.013
ランダムな文節シャッフル	0.895±0.003	0.850±0.021	0.804±0.031	**0.935±0.011**
単語シャッフル	0.888±0.005	0.831±0.049	0.809±0.032	0.930±0.013

数値の表記は，平均値 ± 標準偏差.

シャッフルを適用したモデルは，DA なしのモデルとほとんど性能の差がなかった．

次に，JSICK での実験結果を表 7.4 の結果と併せて表 8.6 に示す．このとき，含意関係認識を行うモデルには RoBERTa を用いる．結果としては，データ全体の正解率は，係り受けを考慮した文節シャッフルで拡張した場合が最も高く，単語シャッフルで拡張した場合は拡張なしよりも低くなった．また，ラベル別に正解率を見ると，含意と矛盾の正解率は係り受けを考慮した文節シャッフルが最も高く，中立の正解率のみランダムな文節シャッフルが係り受けを考慮した文節シャッフルをわずかに上回った．

以上の結果から，係り受けを考慮した文節シャッフルは，他のシャッフルの手法と比較して効果が高いことが分かった．ランダムな文節シャッフル・単語シャッフルの効果が低かった原因としては，ラベルとテキストがあっていない訓練データが係り受けを考慮した文節シャッフルと比べて多く生成され，それらのデータがノイズとなるからであると考えられる．

第 9 章

結論

本論文では，我々が考案した複数の BERT を用いた DA と，文節シャッフルによる DA という 2 つの手法についてまとめた．

タスクを解くために利用されるモデルと単語置換のために利用されるモデルについて，互いに異なるコーパスで事前訓練された BERT を用いる．

1 つ目の手法である複数の BERT を用いた DA は，BERT のような事前学習済みモデルを用いてタスクを解く場合に，そのタスクに用いるモデルとは異なるコーパスで事前学習されたモデルを用いて，単語置換による訓練データの拡張を行う手法である．我々は，この手法が文書分類に効果がある一方で，文ペア分類，QA には効果がないことを実験によって示した．

2 つ目の手法である文節シャッフルによる DA は，文の係り受け関係が崩れないように文節の順序をシャッフルすることで，訓練データを拡張する手法である．我々は，この手法が文書分類と含意関係認識に効果があることを実験によって示した．

さらに，これらの手法はその他の簡易的な DA の手法と同様に，タスクにおける訓練データが少量である場合のみモデルの性能を向上させる効果があり，十分な量がある場合はその効果を発揮しないことがわかった．

今後はこの 2 つの手法について，手法同士もしくはその他の手法と組み合わせて利用したときにその効果はどうなるのかについてや，まだ試していないタスクへの有効性なども明らかにしていきたい．

謝辞

本研究は，JSPS 科研費 JP23K11212 の助成を受けたものです．

本研究を進めるにあたり，多くのご指導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，論文執筆時にアドバイスをいただいた東京農工大学の古宮嘉那子准教授にも厚く御礼申し上げます．さらに，研究活動に取り組むにあたって，多くの知識や示唆を頂いた新納研究室の皆さまにも深く感謝の意を表します．

参考文献

[1] 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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Kyosuke Takahagi and Hiroyuki Shinnou. Data augmentation using multiple bert models. Technical report, Information Processing Society of Japan, 2021.

[3] Kyosuke Takahagi and Hiroyuki Shinnou. Data augmentation by shuffling phrases in a japanese sentence. Technical report, Information Processing Society of Japan, 2022.

[4] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, Vol. 6, No. 1, pp. 1–48, 2019.

[5] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 968–988, Online, August 2021. Association for Computational Linguistics.

[6] Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, Vol. 11, pp. 191–211, 2023.

[7] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting

performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics.

[8] Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5786–5796, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Jiaao Chen, Yuwei Wu, and Diyi Yang. Semi-supervised models via data augmentation for classifying interactive affective responses. In *AffCon@AAAI*, 2020.

[10] H Zhang, M Cisse, Y Dauphin, and D Lopez-Paz. mixup: Beyond empirical risk minimization. iclr 2018. *arXiv preprint arXiv:1710.09412*, 2017.

[11] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019.

[12] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pp. 18–26, Suzhou, China, December 2020. Association for Computational Linguistics.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[14] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[15] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thir-*

teenth Language Resources and Evaluation Conference, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[16] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual Amazon reviews corpus. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4563–4568, Online, November 2020. Association for Computational Linguistics.

[17] Takashi Miyazaki and Nobuyuki Shimizu. Cross-lingual image caption generation. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1780–1790, Berlin, Germany, August 2016. Association for Computational Linguistics.

[18] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[19] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[20] Hitomi Yanaka and Koji Mineshima. Jsick: Japanese sentences involving compositional knowledge dataset. *Proceedings of the Annual Conference of JSAI*, Vol. JSAI2021, pp. 4J3GS6f02–4J3GS6f02, 2021.