特 別 研 究 報 告 題 目

Sentence-BERTを用いたネット小説の文体評価

Evaluation of Web Novel Style using Sentence-BERT

指 導 教 員　　出口　利憲　教授

岐 阜 工 業 高 等 専 門 学 校　　専攻科　先端融合開発専攻

2022Y22 瀬尾　慎介

令 和 6 年（ 2 0 2 4 年 ）　　1 月 3 1 日 提 出

Abstract

In this study, we propose a method for evaluating the Web novel style by machine learning. If the writing style of novels can be quantified, it will be possible to evaluate the novels based on the text, and excellent novels will be evaluated more highly.

This method uses Sentence-BERT to train a model specialized for the writing style of Web novels. In addition, we verified whether it is possible to extract similarities and features between novels by comparing vectors of the models.

As a result of the experiments, the model produced extreme relative-distance embedding vectors. Furthermore, the usefulness of the model was confirmed in the task of discriminating authors. However, it became difficult to discriminate distinctive writing styles.

目 次

第1章　序論

インターネットには、ネット小説と呼ばれる文芸作品が存在する。インターネット上で公開されている小説・エッセイ等の文学作品のことを指し、プロから素人まで様々な作者がサイト上で作品を公開している。中には有名なネット小説がその人気の高さから書籍化され、アニメや映画にまでなるケースもある。

一方で、近年話題になった技術として、「ChatGPT」と呼ばれる生成 AI がある。これはネット上の膨大なデータを学習させた人工知能チャットボットで、様々な質問に対して人間が入力したような自然な回答を返すことができる。この技術をきっかけに自然言語処理や AI の技術は飛躍的に向上し、ChatGPT 等の AI を活用したサービスも世間に大きく普及した。中には高精度な画像生成 AI や作曲 AI といった芸術系の AI も作られており、AI の可能性が非常に広がったといえる。

さて、上で説明した通り、ネット小説はそのハードルの低さから膨大な量の作品が存在する。そのため読者は人気ランキングやジャンル・タグ検索によって作品を探すことが多い。しかし、これらの検索手法では小説本文を考慮することが難しく、優れた内容の作品が注目されづらい問題がある。内容が面白くてもマイナージャンルの作品や新人作者の作品はランキングや検索で注目されづらい。これによって優れた作品であっても評価がされず、膨大な作品に埋もれてしまっている現状がある。

このため、本研究では機械学習によってネット小説の内容を評価する手法を提案・検証する。小説本文の特徴を評価・数値化できれば内容を考慮した評価が可能となり、優れた作品が高く評価されやすくなると考える。提案手法では Sentence-BERT を用いて文章のベクトル化を行い、出力したベクトルを比較することで小説同士の類似度や特徴抽出が可能かどうかを検証する。Sentence-BERT は文章内の単語間の関係性も考慮した学習を行うため、文章の内容を考慮したベクトルの生成が期待できる。

第2章　基本知識

2.1　自然言語処理

2.1.1　自然言語処理

自然言語処理とは、人間がコミュニケーションで使用する英語や日本語といった自然言語をコンピュータが処理する技術のことである。文章の構文・構成などを分析する基礎技術から、ChatGPT のような言語で受け答えが可能なシステム等の自然言語を扱う技術全般を指す。自然言語には意味や構文に曖昧性があるため、コンピュータが文章の意味を正確に理解することは困難だった。しかし、コンピュータの性能向上や機械学習の研究の進展により、コンピュータによる自然言語の理解は大きく進んだ。

2.1.2　形態素解析

形態素解析とは、文法的な情報の注釈がない自然言語のテキストデータを、文法や単語の品詞情報などに基づいて形態素（言語で意味を持つ最小単位）に分解し、それぞれの品詞等を判別する技術である。形態素解析を用いることで単語ごとの出現頻度や単語間の相関関係が計算可能となり、テキストデータからコンピュータで解析可能な形式への変換が可能となる。形態素に分解する際、英語などの単語間が区切られている言語は単語の接辞や変化を調べるだけでいいが、日本語のような区切られていない言語は分解が困難である。そのため、単語分割のために膨大な辞書データが必要となる。形態素解析を行うツールは形態素解析器と呼ばれる。

2.2　テキストの埋め込み表現

2.2.1　埋め込み表現

埋め込み表現とは、ある要素を実ベクトル空間上へ投射することでベクトルで表現する手法である。主に単語をベクトルで表現した際に用いられる用語で、分散表現といわれる場合もある。

テキストをコンピュータで分析する際に、単語や文章の特徴を何らかの方法で数値化しなければコンピュータで処理することができない。そのため、単語の出現頻度や他単語との共起関係、機械学習で学習させた結果の出力ベクトル等の数値をその単語を表すベクトルの一要素として割り当てる。これによって単語をベクトルという数値表現で表

すことができる。

単語をベクトルに変換する場合、one-hot 表現が最も単純である。これは全単語数の要素を持つベクトルに対して、単語に対応する要素が 1、それ以外が 0 となる数値表現である。シンプルな表現だが、ベクトル同士の比較で意味を抽出することができない。そのため、分散表現と呼ばれる低次元ベクトルが主に使われる。分散表現手法では機械学習や数学的な次元削減手法によって one-hot 表現より低次元なベクトルで表現しており、単語間の類似度計算が可能である。

2.2.2 テキスト（単語）のトークン化

テキストを機械学習で処理する場合、機械学習では数値しか扱えないためテキストデータをそのまま入力に使用することはできない。そのため、機械学習する際にはテキストをトークンと呼ばれる単位に分割し、それぞれにトークン ID を割り振ることで入力可能な形式に変換する。テキストの最小単位は文字だが、文字単位で分割すると失われる情報が多いことから、多くの場合でトークン＝単語として処理する。そのため、トークン単位の分割には形態素解析器が用いられる。また、機械学習によって各トークンのベクトルが得られるようにすることで、トークンに対応した単語の埋め込み表現が得られる。

2.3 機械学習

2.3.1 機械学習

機械学習とは、大量に用意された学習データからパターンを学習し、特定のタスクを解くためのモデルを構築するプログラムを指す。学習データにある複数の変数を用いて特定の計算を行い、計算結果を分析したり出力と比較したりして計算の最適化を行う。例を挙げると、明日の売り上げを予測するタスクを解くために、過去の売上データを学習データとして前日のデータから翌日のデータを予測するプログラムがそうである。

機械学習のタスクは大きく分けて以下の 3 つに分類される。

- 教師あり学習　　… 入力と出力の関係性を学習する
- 教師なし学習　　… 入力のみからパターンを学習する
- 強化学習　　… より多い報酬が得られるパターンを学習する

上の例で挙げたプログラムは入力と出力があるため、教師あり学習である。

2.3.2　ニューラルネットワーク [1)]

ニューラルネットワークとは、人間の脳の細胞であるニューロンを模倣した機械学習手法である。ニューロンは樹状突起から他細胞からの情報を受け取り、情報処理した結果を軸索を通して他細胞へ伝達する。この情報を伝達する細胞の結合部分をシナプスと呼ぶが、ここの結合強度が外的刺激で変化することで伝達される情報も変化する。このシナプスの結合強度の変化が脳の学習メカニズムであるといわれている。

この構造を計算ユニットとして模倣したものを Figure.2.1 に示す。左側にある「Weights（重み）」がニューロンでの結合強度を表し、「inputs（入力）」と重みから計算された出力を「activation function（活性化関数）」に通した値を出力する。学習する際には、学習データの出力と計算ユニットの出力の誤差が小さくなるように繰り返し重みを調整して最適化する。この計算ユニットをいくつも接続したものをニューラルネットワークと呼び、いくつもの重みを調整することで複雑な学習が可能となる。

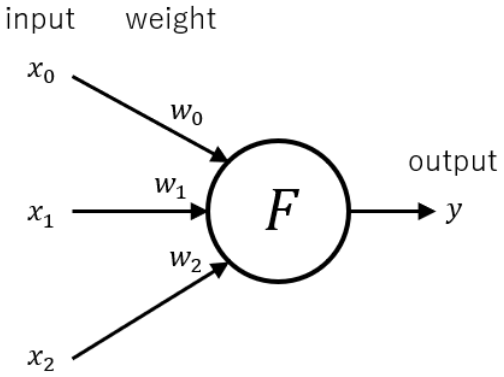

Figure 2.1　Neural network unit.

2.3.3　ディープラーニング（深層学習）

ディープラーニングとは、階層構造の学習モデルによって特徴の関係性を学習する学習手法である。一般的にはニューラルネットワークが 4 層以上重なったモデルを指すが、他にも CNN や GAN、Transformer といった様々なネットワークモデルが存在する。

多層のニューラルネットワークの構造を Figure.2.2 に示す。図のように、並列に繋げた計算ユニットを重ねたような構造である。最適化には誤差逆伝播法が用いられ、出力と学習データの誤差を基に各ユニットの重みを出力側から順に修正する。複雑なネットワーク構造によって高い学習性能を発揮するが、膨大な計算量に加えて勾配消失問題、過学習等の問題点も存在する。しかし、コンピュータの性能向上やインターネットの普

及による学習データの流通、そして研究の進歩によってこれらの問題も対策され、十分な学習が可能となった。

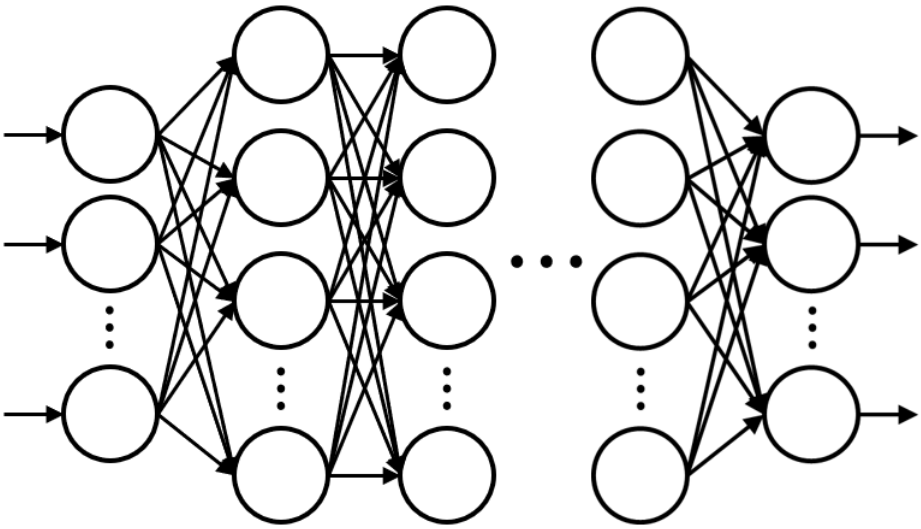

Figure 2.2　Deep neural network.

第3章　実験で使用した技術

3.1　Sentence-BERT

3.1.1　Transformer[2),3)]

Transformer とは 2017 年に Google の研究者が発表した、Attention 機構を用いた深層学習モデルである。主に自然言語処理を目的としたモデルに使用されており、後述する BERT の他にも、ChatGPT をはじめとする GPT シリーズにも使用されている。Attention 機構とは系列データの要素同士の関係性を計算する手法で、これを何層も積み重ねた構造をしている。従来の系列データの学習モデルは先頭データから順に処理するため、大量な計算量が必要だったり離れたデータ同士の関係性を考慮しづらい欠点があった。しかし Attention 機構は系列データを並列処理できるため、従来の逐次処理していたモデルより高速に計算できる。また、データ同士の距離に関係なく関係性を学習するため、より複雑な関係性を学習可能になった。

Transformer の構造を Figure.3.1 に示す。左側をエンコーダー、右側をデコーダーとした構造で、灰色ブロックが Multi-Head Attention 機構を中心とした処理である。この処理では一度に複数パターンの方向性を処理可能で、これを複数回繰り返すことで複雑な関係性を学習する。

3.1.2　BERT[4)]

BERT とは自然言語処理の学習モデルの一つで、Transformer ベースの機械学習手法である。Transformer のエンコーダー部分のみを使用したモデルで、文章をトークンで分割したトークン列を入力すると埋め込み表現を出力する。従来のモデルは単方向の関係性しか学習できなかったが、Transformer を使用することで前後の双方向で学習が可能となった。

BERT の特徴として、事前学習として「Masked Language Model（MLM）」と呼ばれるタスクを行っている点がある。これはランダムなトークンを「[Mask]」というトークンに変更し、その位置の正しい単語を予測するタスクである。このタスクは [Mask] トークンの位置の単語を予測するために前後のトークンを考慮する必要があるため、Transformer の特徴である広い範囲の関係性の考慮が有効に機能する。また、このタスクはラベルを設定する必要がないため、大量の学習データが用意可能である。これらの利点によって、

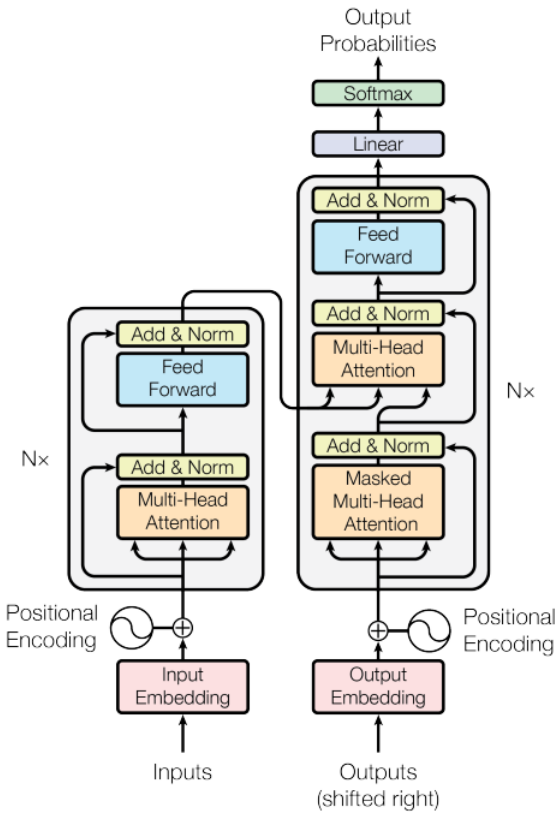

Figure 3.1　Transformer.

BERT は大量の文章データで事前学習することで高い文章理解を、そしてそのモデルを転移学習・ファインチューニングすることで高い汎用性を持つ。

3.1.3　Sentence-BERT[5)]

Sentence-BERT とは、BERT を埋め込み表現生成に特化したモデルへファインチューニングする手法である。BERT は多くのタスクで高い性能を出すが、埋め込み表現そのものをデータ分析に使用する場合には従来のモデルより性能が低いという欠点がある。例えば複数文書の類似度の計算をするタスクの場合、通常は各文書を数値表現である埋め込み表現に変換して類似度を計算する。しかし、BERT で処理する場合だと埋め込み表現を用いた計算は望ましくない。よって、二つの文書を結合した系列データを入力して類似度を予測するモデルを使用する方が高い精度を期待できる。しかし、この方法は対象の文書の数が多ければ多いほど文書同士の組み合わせが増大するため、モデルの計算回数が増えてしまう。モデルの計算は時間がかかるため、全ての組み合わせの類似度を計算するためには膨大な時間がかかってしまう。しかし、文書を埋め込み表現に変換した後に埋め込み表現同士の類似度を計算すれば、モデルの計算回数は文書数と等しく

なり計算時間が少なくて済む。そのため、BERT の高い文章理解を埋め込み表現に利用できるようにするために Sentence-BERT は提案された。

Sentence-BERT の構造を Figure.3.2 に示す。学習データには二つの文章データ Sentence A, Sentence B と、その関係性を示すラベルを使用する。まず、各文章データを BERT に入力し、出力された各トークンの平均 pooling である埋め込み表現 u, v を取得する。それらを比較した結果と学習データのラベルの誤差から BERT の最適化を行うことで、最適な埋め込み表現 u,v を出力できるモデルを学習する。最後の u,v の比較方法には二種類あり、Figure.3.2 の左側は u,v の差分を softmax で分類する手法、右側は cos 類似度を計算する手法である。本研究では右側の手法を使用して学習を行った。

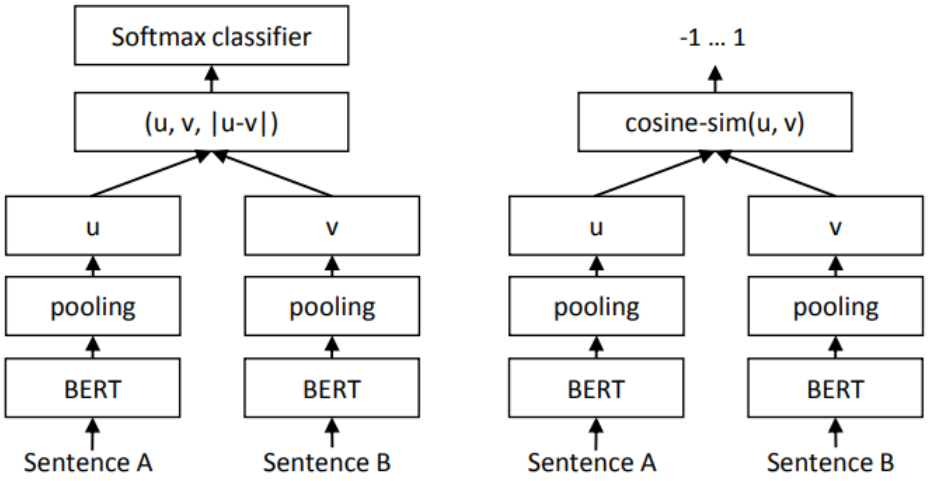

Figure 3.2　Sentence BERT.

3.1.4　Triplet loss

Triplet loss とは、ベクトル間の距離の最適化によって学習を行う手法である。Anchor、Positive、Negative の三つのベクトルを一組とし、Anchor を基準とした時の Positive、Negative の距離を求める。そして Anchor-Positive 間の距離を小さく、Anchor-Negative 間の距離を大きくするようにモデルの最適化を行う。本研究では Sentence-BERT の学習にこの手法を使用した。

下に Triplet loss の損失を求める式を示す。d_p は Anchor-Positive 間距離、d_n は Anchor-Negative 間距離、α は最適化を行うマージンを示すハイパーパラメータである。

$$Loss = \max\left(d_p - d_n + \alpha, 0\right) \tag{3.1}$$

3.2　埋め込み表現の評価

3.2.1　cos 類似度

cos 類似度とは、ベクトル同士がどの程度似ているのかを示す指標の一つである。ベクトル同士の類似度計算はいくつかあるが、cos 類似度はベクトル同士のなす角のコサイン値を計算することで類似度としている。コサイン値なので、ベクトル同士がまったく似ていない、つまりベクトル同士が逆方向だった場合に –1 となり、完全に同じ、つまりベクトル同士が同方向だった場合に 1 となるような値になる。ベクトル同士のなす角の大きさは内積の導出式を変形することで求めることができる。下に cos 類似度の導出式を示す。

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{3.2}$$

3.2.2　ヒートマップ

ヒートマップとは、二次元データの値を色で表現した可視化グラフである。色の濃淡や種類でデータの関係性が読み取れるため直感的に理解できる。様々な種類があるが、本研究の場合は表の値を色の濃淡で表現したような二次元グラフを指す。本研究では文章同士の類似度を表示するために使用した。ヒートマップの例を Figure.3.3 に示す。データの類似度を示す場合は縦軸と横軸のラベルの交点部分の濃淡が、そのラベル同士の類似度を指す。

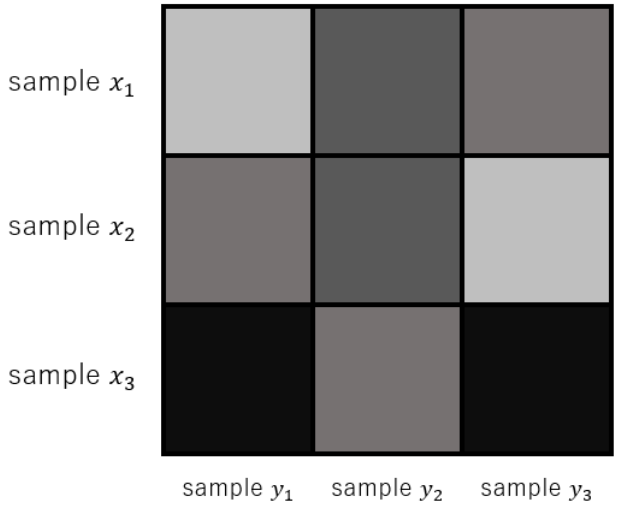

Figure 3.3　Heat Map.

3.3 Python による Sentence-BERT

3.3.1 Python による機械学習

機械学習を行う時、多くの場合で Python がプログラミング言語として使用される。その理由として機械学習ライブラリの豊富さがある。機械学習は対象データ、学習目的等に応じて様々な手法が研究されており、それらを全て独自に実装することは困難である。しかし、Python では様々な機械学習手法がライブラリとして公開されており、学習データがあれば簡単に機械学習を始めることができる。また、学習済みの高精度なモデルも多数公開されているため、これらをファインチューニングすることも可能である。代表的なライブラリとしては scikit-learn、PyTorch 等がある。

3.3.2 PyTorch

PyTorch とは、Python で公開されている深層学習ライブラリの一つである。ニューラルネットワークを直感的に構築でき、用意されたモジュールで最適化や勾配計算等が容易に行えるため、簡単に深層学習モデルの設計、学習が可能である。他のライブラリと比較して複雑なモデルが設計しやすいため、研究目的で使用されることが多い。

3.3.3 Transformers

Transformers とは、BERT や GPT シリーズなどの Transformer を用いた自然言語モデルのための Python の機械学習ライブラリである。Hugging Face Hub というサイトで公開されたモデルを使用して、分散表現・文書分類・文章生成等の自然言語処理が簡単に実行できる。また、サイトで公開されている様々な学習済みモデルや学習データセットを利用できるため、自然言語モデルの構築が簡単に行える。

3.3.4 SentenceTransformers

SentenceTransformers は、Sentence-BERT のような埋め込み表現生成モデル用の Python フレームワークである。前章の PyTorch や Transformers を利用しており、それらのライブラリの機能を活用して Sentence-BERT が容易に構築できる。

第4章　実験

4.1　実験の概要

本研究の目的は、機械学習によってネット小説の内容を評価する手法の提案・検証である。ネット小説の内容とは小説本文のことを指し、それを評価するということは小説本文の特徴を抽出する行為を指すと考える。しかし、そのためにはまず「小説本文の特徴」とは何か、定義する必要がある。文章の特徴には、ジャンルや文章構成といった広い範囲を示すものから単語出現頻度・固有表現等の小さい範囲のものまで様々な定義が可能である。更に特定のラベルを付与したデータで機械学習を行えば、より曖昧な条件で定義することもできる。

その中で本研究は、文体という曖昧な特徴に注目した。文体にははっきりした定義がないが、本研究では「作者が作文する際の特徴」であると考える。小説が注目されるためには、大前提として読みやすい文章であることが求められる。この「読みやすさ」は作者が作文する時の文章構成や表現のセンス等の技術、つまり文体で生み出される。このことから、文体を評価することができれば、その文章の優劣が評価できると考えた。

文体を抽出するには、抽出したデータによって文体が同じかどうかを判別できる必要がある。前述したように、文体は「作者が作文する際の特徴」であることから、作品内の文章同士は同じ文体であると思われる。そのため、「同小説内の文章は同じ文体で書かれている」という仮定から、同小説内の文章かどうかを判別するタスクを考えた。このタスクを機械学習することで、文章の文体に注目して計算するモデルが学習できると考える。

また、使用する機械学習モデルは埋め込み表現を出力可能なBERTモデルを選択した。文章を入力して文体の特徴を出力するモデルを学習するには、あらかじめ文体を数値化する必要がある。しかし、文体の数値化はできないため、このモデルは実現不可能である。そのため、小説の埋め込み表現の類似度から文体が類似するかどうかを判断できれば、間接的に文体を評価することになると考えた。

よって、埋め込み表現を出力するモデルとしてBERTモデルを採用したが、BERTモデルは埋め込み表現そのものによるデータ分析に向いていない。そのため、Sentence-BERTによって上記のタスクを用いたファインチューニングを行うことで、優れた埋め込み表現が出力されるネット小説の文体に特化したモデルが学習できると考えた。

以上の考察から本実験では、小説本文を用いてSentence-BERTをファインチューニングすることで、小説本文の文体を表現する埋め込み表現が生成可能なモデルを作成する。また、埋め込み表現で小説本文の文体を考慮した比較が可能かを検証する。

実験は以下の順で行う。

1. 「小説家になろう」から小説本文データを収集
2. 小説本文データから学習データを作成
3. Sentence-BERTで学習
4. モデルの評価
    - テストデータでの他手法との比較
    - 同作者作品の類似度検証
    - ブックマーク作品の類似度検証

学習用の小説本文データは小説投稿サイト「小説家になろう」からウェブスクレイピングによって収集する。収集したデータから同小説内の文章かどうかを判別するタスクを作成し、Sentence-BERTで学習を行う。その学習モデルで生成した埋め込み表現が小説本文の特徴を内包するかどうかを検証し、埋め込み表現の有用性について考察を行う。

4.2 実験準備

4.2.1 小説本文データの収集

ネット小説の本文データを用意するために、小説投稿サイト「小説家になろう」で公開されている作品に対してウェブスクレイピングを行った。手順としては以下のとおりである。

1. 評価の高い作品の作品IDを取得

   「小説家になろう」には、なろう小説APIという作品情報取得APIが提供されている。このAPIではサイトに投稿されている作品の様々なメタデータを取得することができる。これを使用して、4つの大ジャンルから250作品ずつ、合計1000作品のIDを取得した。
2. 作品IDを用いて作品ページのURLを作成

   「小説化になろう」では、各作品ページのURLに作品IDが使われている。そのため、前段階で収集した作品IDを用いて各作品ページのURLを作成した。
3. 各作品ページへウェブスクレイピング

前段階で作成したURLを用いてウェブスクレイピングを行い、各作品の本文データを収集した。全データを収集するとデータ量が膨大になるため、先頭話から順に50話分の本文データを収集し、50話以下しかない作品は収集対象外とした。これにより、収集したデータはTable.4.1のようになった。

Table 4.1　Collected Data from ”Syosetuka ni Narou”.

ジャンル	作品数	ページ数
恋愛	201	10050
文芸	206	10300
ファンタジー	245	12250
ＳＦ	230	11500
合計	882	44100

4.2.2　本文データの前処理

前項で収集した本文データを学習させる前に、不要な情報を削減するために前処理を行う。2.1.1項に示す通り、自然言語には曖昧性があるため、最新の機械学習手法でも完全な理解は難しい。そのため、学習する前に前処理として数字や大文字小文字の違い、会話での感情表現手法等といった、文章の意味を理解するうえで不要な情報を消去することで学習精度が向上する。本研究では以下の前処理を本文データに施した。

- 改行記号の削除
- URLの削除
- 半角記号、全角記号の削除
- アルファベットの全角を半角へ変換
- 「」で囲われた会話文を削除
- 先頭から500文字以内の文章に削減
- 読点で終わるよう最後尾を調整

4.2.3　Tripletデータの作成

用意した本文データから、Triplet lossに使用するための学習データセット（以下「Tripletデータ」という。）を作成する。anchorとpositiveの組み合わせを作品の50話分の全組

み合わせである 1225 通り作成し、そこに他作品のランダムな話を negative として追加する。これを全作品で行うことで、合計 1080450 データを作成した。

4.2.4 Sentence-BERT で学習

Triplet データが用意できたため、Sentence-BERT で学習を行う。Triplet データは学習：評価：テスト＝8：1：1 の比率でランダムに分割し、学習用の 864360 データを学習に使用した。評価用は学習途中のモデル精度の変化を記録するために、テスト用は学習後のモデル検証に使用する。

BERT モデルは東北大学研究チームが公開している日本語事前学習モデル[6)] を使用し、これに平均プーリングを結合させたモデルを作成した。トークナイザーも共に公開されていた語彙サイズが 32000 のものを使用した。

4.2.5 埋め込み表現の評価について

学習した Sentence-BERT のモデル（以下「学習済みモデル」という。）で出力される埋め込み表現の有用性を評価する。その際に、他手法による埋め込み表現と比較することで学習済みモデルの特徴を考察する。

検証に使用するモデルを Table.4.2 に示す。BERT（東北大）のモデルには、学習済みモデルと同様に平均プーリング層を結合したモデルを使用する。また、MeCab+Word2vec のモデルは、MeCab で小説本文を単語で分割した後、それぞれの単語分散表現を Word2vec で出力し、その平均を取ることで小説本文の埋め込み表現とする。

Table 4.2 Evaluated Models.

モデル	説明
学習済みモデル	本研究手法のモデル
BERT（東北大）	実験で使用した東北大のモデル
MeCab + Word2vec	Word2vec で埋め込み表現を生成する

4.2.6 テストデータでの評価

4.2.4 項の学習が成功したかの検証として、テスト用 Triplet データを用いた精度検証を行う。他の手法での精度と比較することで学習によって得た特徴を考察する。

検証では、テスト用 Triplet データをモデルで埋め込み表現に変換し、Anchor を基準とした Positive、Negative の類似度を計算する。類似度計算には COS 類似度を使用し、Positive の方が類似度が高ければ正解として全データでの精度を求める。また、類似度の他に Positive と Negative の類似度の差（以下「差分」）の平均を求める。この値は Positive と Negative がどれくらい離れているかを示すため、モデルが明確に判断できているか否か分かる。これらの値を Table.4.2 のそれぞれで算出・比較する。

4.2.7　同作者の作品の類似度検証

4.1 項で文体の定義として「作者が作文する際の特徴」と述べている。この定義を基に学習タスクが作られていることから、学習済みモデルの埋め込み表現は同作者の作品の場合に類似度が高くなると考えられる。そのため、同作者の場合と異作者の場合で類似度を比較して学習済みモデルが文体を抽出できているかを検証する。また、他手法と比較することで学習済みモデルの特徴を考察する。

検証では、5 人の作者からそれぞれ 5 作品の合計 25 作品を使用する。それぞれの作品の 10 ページ分のデータを学習済みモデルで埋め込み表現に変換し、その平均ベクトルを作品の埋め込み表現とする。そして作品同士の類似度を求めてヒートマップで図示する。ヒートマップは同作者の 5 作品のものを 5 パターン、それぞれの作者から 1 作品ずつ選択した 5 作品のものを 5 パターンで、合計 10 パターンの 5×5 ヒートマップを作成する。また、このヒートマップを Table.4.2 のそれぞれで作成・比較する。

4.2.8　ブックマーク作品の類似度検証

本研究の目的は「内容が優れている作品を高く評価する埋め込み表現」を作成することであるが、「内容が優れている」と言える文章を定義することは難しい。それは、読者ごとの好みが強く反映される概念であり、絶対的なものではないと考えられるからである。そのため本検証では代わりに、その読者が面白いと感じる作品かどうかを評価できるか検証する。また、他手法と比較することで学習済みモデルの特徴を考察する。

検証では、5 人のユーザーのブックマークから 50 作品ずつの合計 250 作品を使用する。前項と同様の手法で作品の埋め込み表現を作成し、それらの類似度をヒートマップで図示する。ヒートマップは特定の 1 ユーザーのブックマーク 50 作品を基準として、同じブックマーク 50 作品で 1 パターン、その他のユーザーのブックマーク 50 作品で 4 パター

ンの、合計 5 パターンの 50 × 50 ヒートマップで作成する。そして同ブックマークと異ブックマークの場合を比較することで、ブックマークの特徴を抽出できているかを検証する。また、他手法と比較することで学習済みモデルの特徴を考察する。

4.3 実験結果

4.3.1 Sentence-BERT で学習

Sentence-BERT の学習には約 3 日かかった。学習時間が非常に長かった原因としては、学習サイズが膨大だった点、PC が低スペックでバッチサイズを 1 にする必要があった点等が考えられる。

また、Sentence-BERT での学習中の精度を Figure.4.1 に示す。横軸が学習のステップ数、縦軸が評価用データで算出したモデル精度を示す。グラフを見ると、ステップが進むにつれて上昇し最終的に 1 に近くなっていることから、学習は問題なく行われたと考えられる。序盤から 0.8 割後半と高い精度が出ているのは、BERT が事前学習済みのモデルであるためだと考えられる。

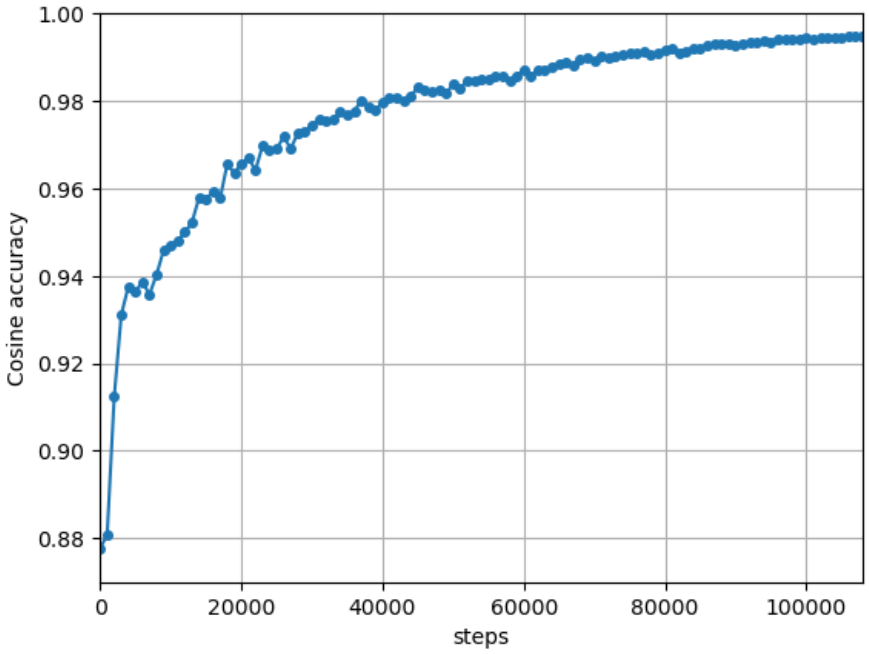

Figure 4.1 Acculacy in learning Sentence-BERT.

4.3.2　テストデータでの評価

各モデルでの結果を Figure.4.2 に示す。左側の白い棒グラフで精度を、右側の黒い棒グラフで差分を示している。

精度のグラフを見ると、学習済みのモデルが最も高精度であることが分かる。通常の BERT と MeCab+Word2vec のモデルは同程度の精度であることから、タスクに特化したモデルにファインチューニングできているといえる。学習で使用しなかったテストデータでも高精度であることから、過学習が起きていないと考えられる。また他にわかる点として、通常の BERT と MeCab+Word2vec の手法も約 0.8 とある程度高い精度が得られた点がある。通常の BERT は埋め込み表現に適しておらず、Word2vec は前世代の自然言語モデルであるため、両手法とも精度が低いと考えていたため予想外の結果だった。

一方で差分のグラフを見ると、学習済みモデルの差分が他手法と比べて非常に大きいことが分かる。このことから、学習済みモデルは他モデルより明確に Positive と Negative を判別できていることが分かる。これは Triplet Loss の学習方法が Anchor-Positive を近づけ、Anchor-Negative を遠ざけるように学習するためだと考えられる。

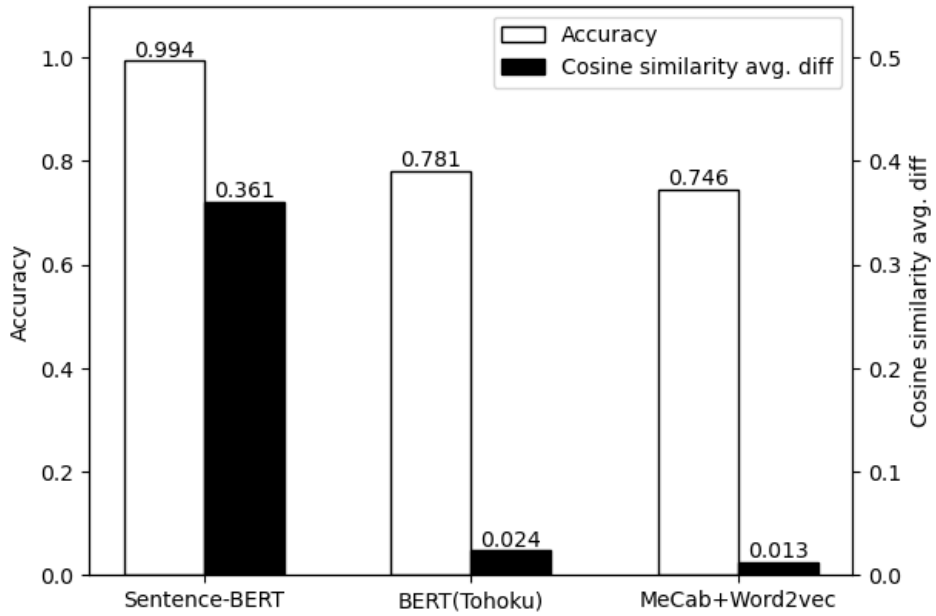

Figure 4.2　Model Comparison.

4.3.3　同作者の作品の類似度検証

学習済みモデルでの検証結果のヒートマップを Figure.4.3 に示す。左側の 5 パターンが同作者の作品同士の場合、右側が異作者の作品同士の場合のヒートマップである。右

側のバーが色と値の相関を示しており、0.2〜0.9 の範囲で濃淡を表現している。軸ラベルの番号は作品に対応しており、同番号の部分は同作品の類似度であるため 1 となっている。また、各ヒートマップの上部に類似度の平均値を算出して示している。

Figure.4.3 を見ると、左側の方が色が濃いことから、同作者の場合の方が類似度が高い場合が多いことが分かる。類似度の平均値を見ても異作者のデータが約 0.4〜0.5 に対して同作者は約 0.5〜0.7 と比較的高いことが分かる。このことから、学習済みモデルの埋め込み表現には作者の文体の特徴が含まれていると考えられる。

次に東北大の BERT モデルでの検証結果のヒートマップを Figure.4.4 に示す。ヒートマップの配置は Figure.4.3 と同様だが、濃淡の範囲が 0.9〜1.0 と狭くなっている。これはほぼ全ての類似度が 0.9 以上という高い値をとっているためである。原因は不明だが、4.3.2 項で示したように、学習済みモデル以外の差分が非常に小さかったのはこれが原因だと考えられる。

Figure.4.4 を見ると、Figure.4.3 と同様に左側の方が濃いことが分かる。平均値を見るとほぼ差がないが、ヒートマップの濃淡や類似度の値から考察すると、左側の方が類似度が高い傾向があるといえるだろう。また特徴的な点として、ヒートマップの模様が、Figure.4.3 と類似している点が挙げられる。例えば「Same Author.4」のヒートマップを見ると両図とも右下が濃くなる模様をしており、「Diff Author.5」では 1 と 5 の類似度がどれも低くなっている。この類似性は、学習済みモデルが東北大の BERT モデルをファインチューニングしているためだと考えられる。

最後に MeCab+Word2vec のモデルでの検証結果のヒートマップを Figure.4.5 に示す。ヒートマップの配置と濃淡範囲は Figure.4.4 と同様になっている。Figure.4.4 より類似度が高い傾向があり、左右のヒートマップの濃淡に差は見られない。また、他モデルでは見られた模様の類似性も見られなかった。

4.5 の特徴として、「Same Author.5」や「Diff Author.5」等で見られる低い類似度の部分がある。この部分は Figure.4.3、4.4 でも類似度が低くなっており、この部分の作品は他作品と比較しても類似性が低いことが考えられる。該当作品を調査したところ、メジャーリーグについて解説するファン作品だった。他の作品はファンタジーや SF、恋愛等のノベルであることから、低い類似度は妥当であることが考えられる。

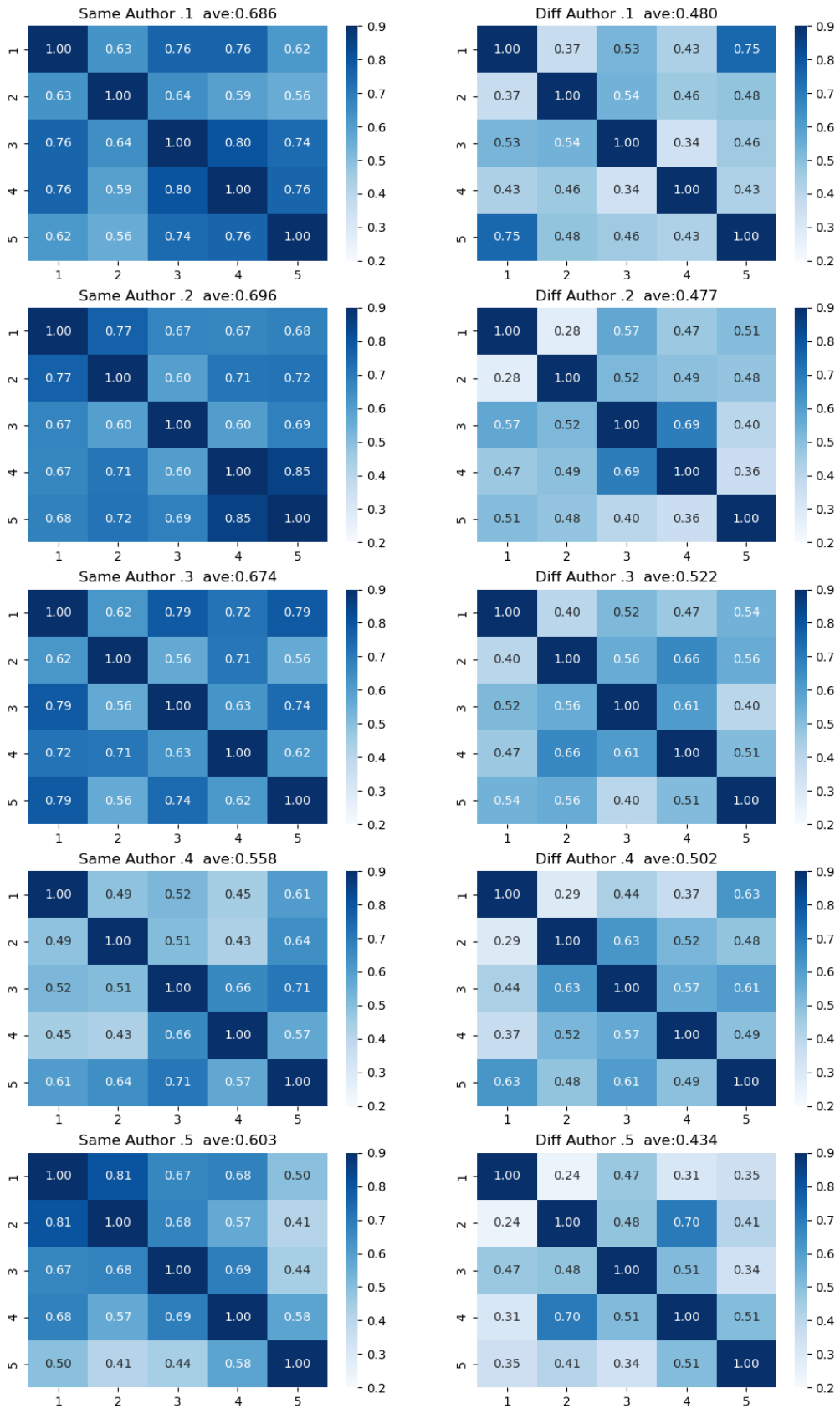

Figure 4.3 Evaluation of the Sentence-BERT on the dataset of same authors and different authors.

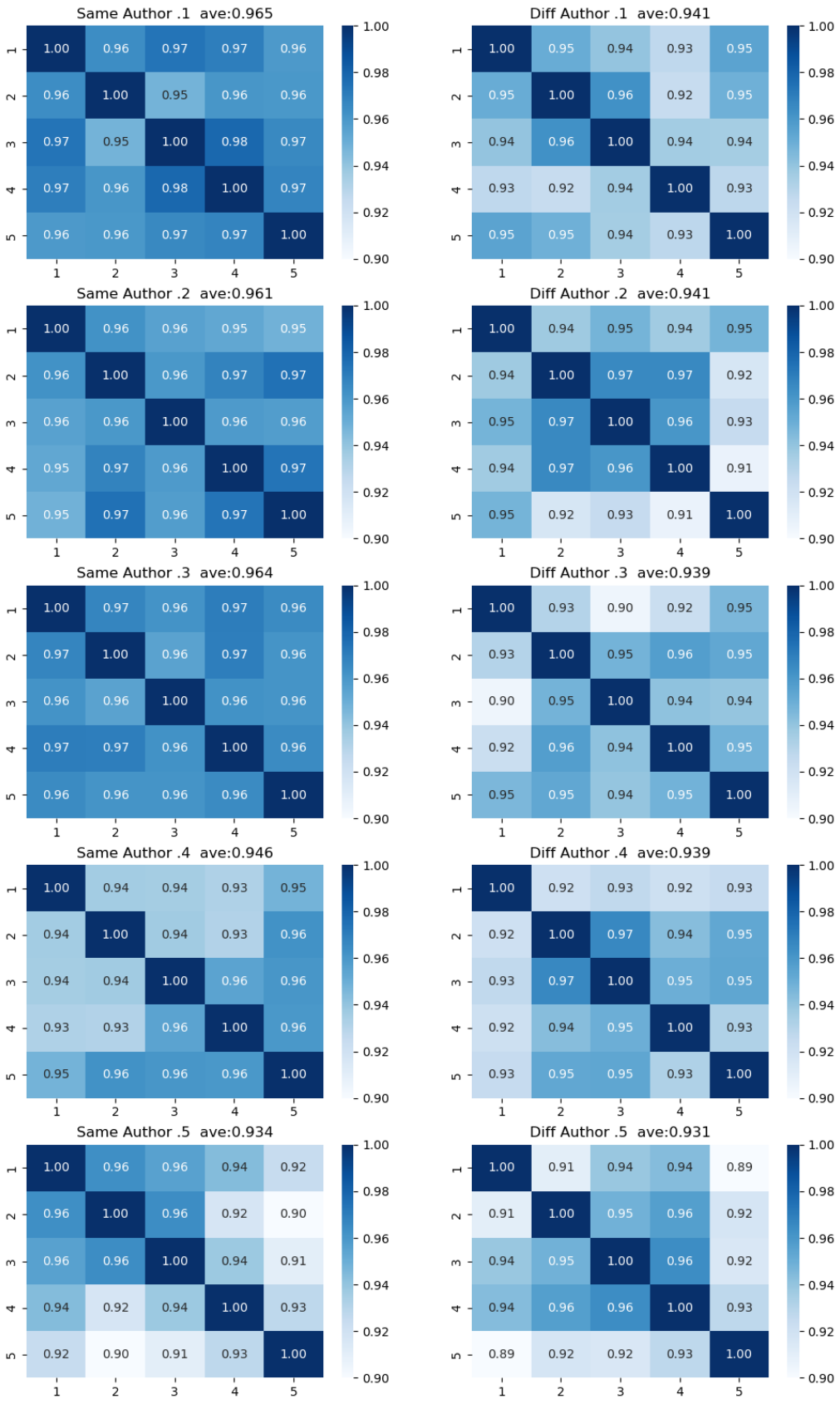

Figure 4.4 Evaluation of the BERT(Tohoku) on the dataset of same authors and different authors.

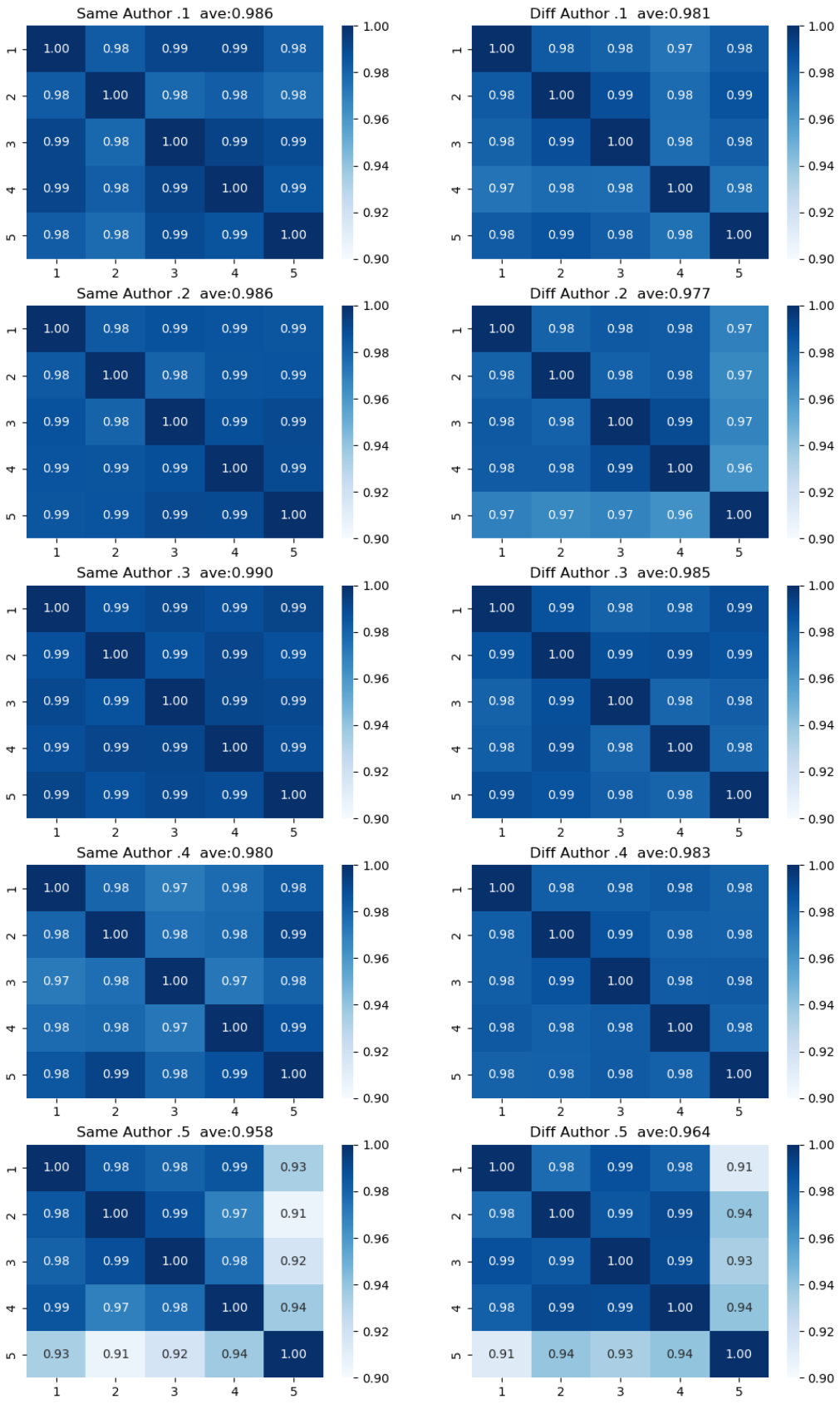

Figure 4.5 Evaluation of the MeCab+Word2vec on the dataset of same authors and different authors.

4.3.4　ブックマーク作品の類似度検証

学習済みモデルでの検証結果のヒートマップを Figure.4.6 に示す。最上部のヒートマップが同ブックマークの場合、それ以下の 4 パターンが異ブックマークの場合である。また、各ヒートマップの上部に類似度の平均値を算出して示している。

ヒートマップを見比べると特に違いは見られず、同ブックマークの場合に類似度が高い傾向はないことが分かる。平均を見ると同ブックマークが最も高いが、User1.1-5 も同程度であることから重要性は低いと考えられる。また、ヒートマップの模様を見るとチェック柄のように交互に濃淡が入れ替わっていることが分かる。よって、ブックマーク作品に対する文体の特徴抽出では明確な成果が得られなかった。

次に東北大の BERT モデルでの検証結果のヒートマップを Figure.4.7 に示す。学習済みモデルでは類似度が適度に分散していたが、東北大の BERT モデルでは一部作品に対する類似度が極端に低くなった。そのため、低い類似度に濃度範囲を合わせたヒートマップを左側に、高い類似度に合わせたヒートマップを右側に示した。ヒートマップ上部には類似度の平均値と共に濃度範囲を示している。

左側のヒートマップをみると、Figure.4.6 と違い全体的に濃い色になっており、ほぼ全ての類似度が高いことが分かる。これは 4.3.3 項の 4.4 と同様の結果である。しかし、一部作品に対する類似度は極端に低く、ヒートマップに線状の模様ができている。対象の作品をサイトで調査するとどの作品もノベル形式で書かれておらず、掲示板のような形式の文章や会話のみで構成されている文章等の特殊な文体だった。このことから、BERT モデルは特殊な文体の特徴に対しての感度が高いことがわかる。

また右側のヒートマップを見ると、Figure.4.6 と同様に同ブックマークと異ブックマークの場合の違いは見られず、チェック柄の模様が見られるのみだった。よって、学習済みモデルと同様に、ブックマーク作品に対する文体の特徴抽出では明確な成果が得られなかった。

最後に MeCab+Word2vec のモデルでの検証結果のヒートマップを Figure.4.8 に示す。BERT モデルと同様の結果だったため、Figure.4.7 と同じ構成で示している。しかし MeCab +Word2vec のモデルでは平均の類似度が BERT モデルより高くなっており、右側の濃度範囲がより狭くなっている。

左側のヒートマップを見ると、全体的に高い類似度と一部作品のみ極端に低い類似度というFigure.4.7と同じ結果だとわかる。類似度が低い作品もBERTモデルと共通だったため、BERTモデルと同様に特徴的な文体に対する感度が高いと考えられる。右側のヒートマップも他モデルと同様の結果となっており、明確な成果は得られなかった。

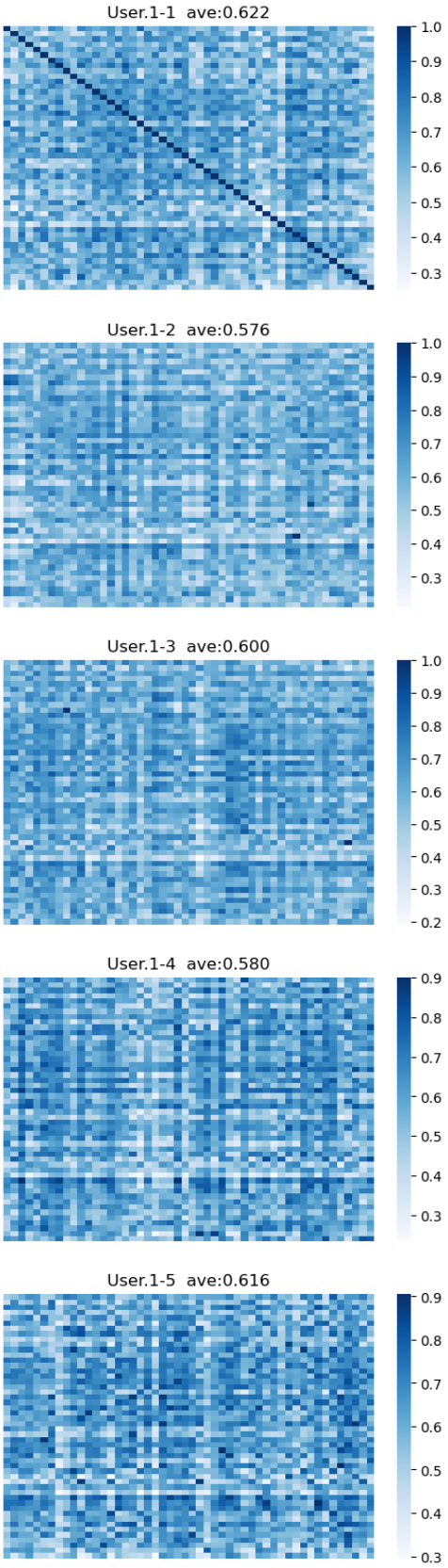

Figure 4.6 Evaluation of the Sentence-BERT on the dataset of bookmark.

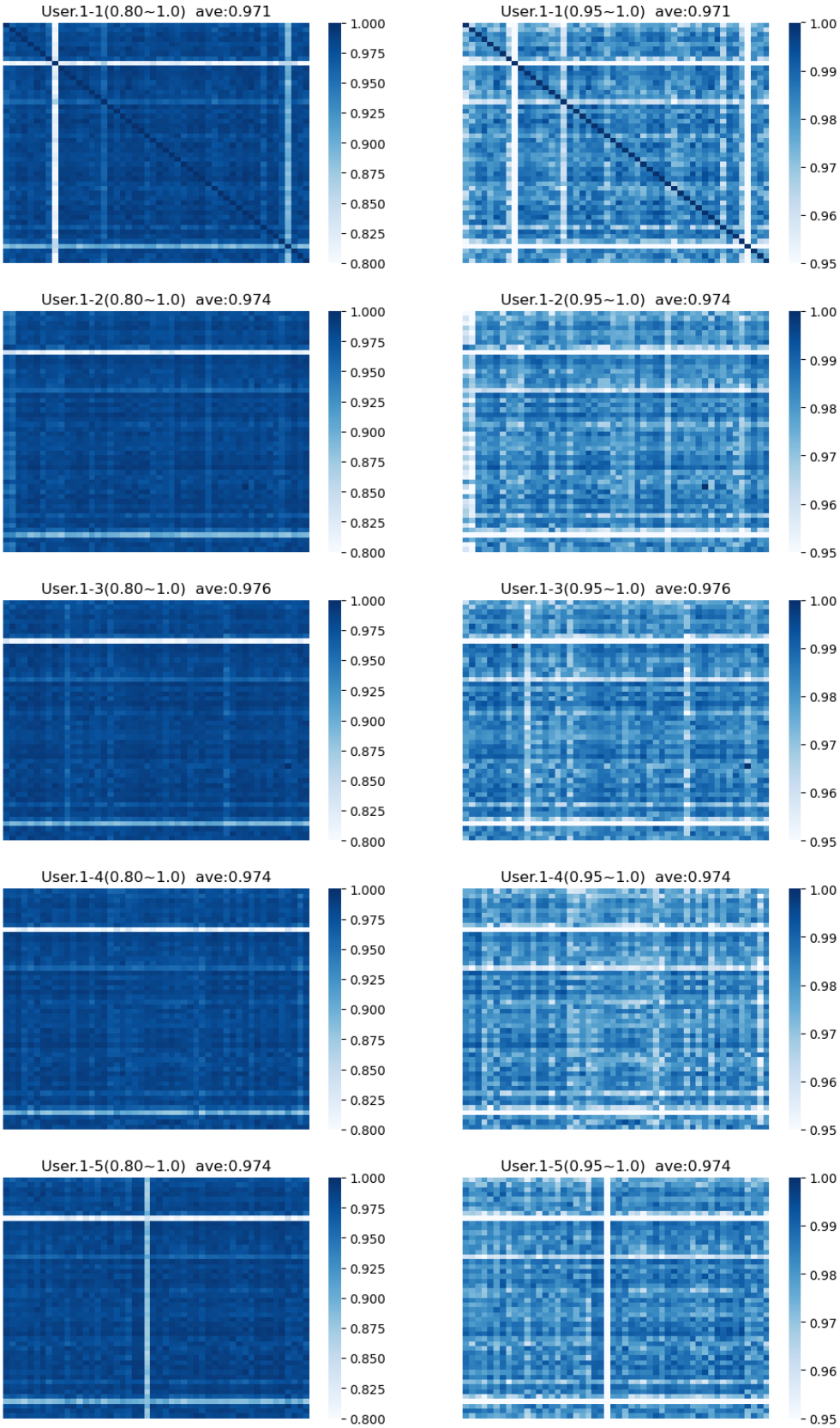

Figure 4.7 Evaluation of the BERT(Tohoku) on the dataset of bookmark.

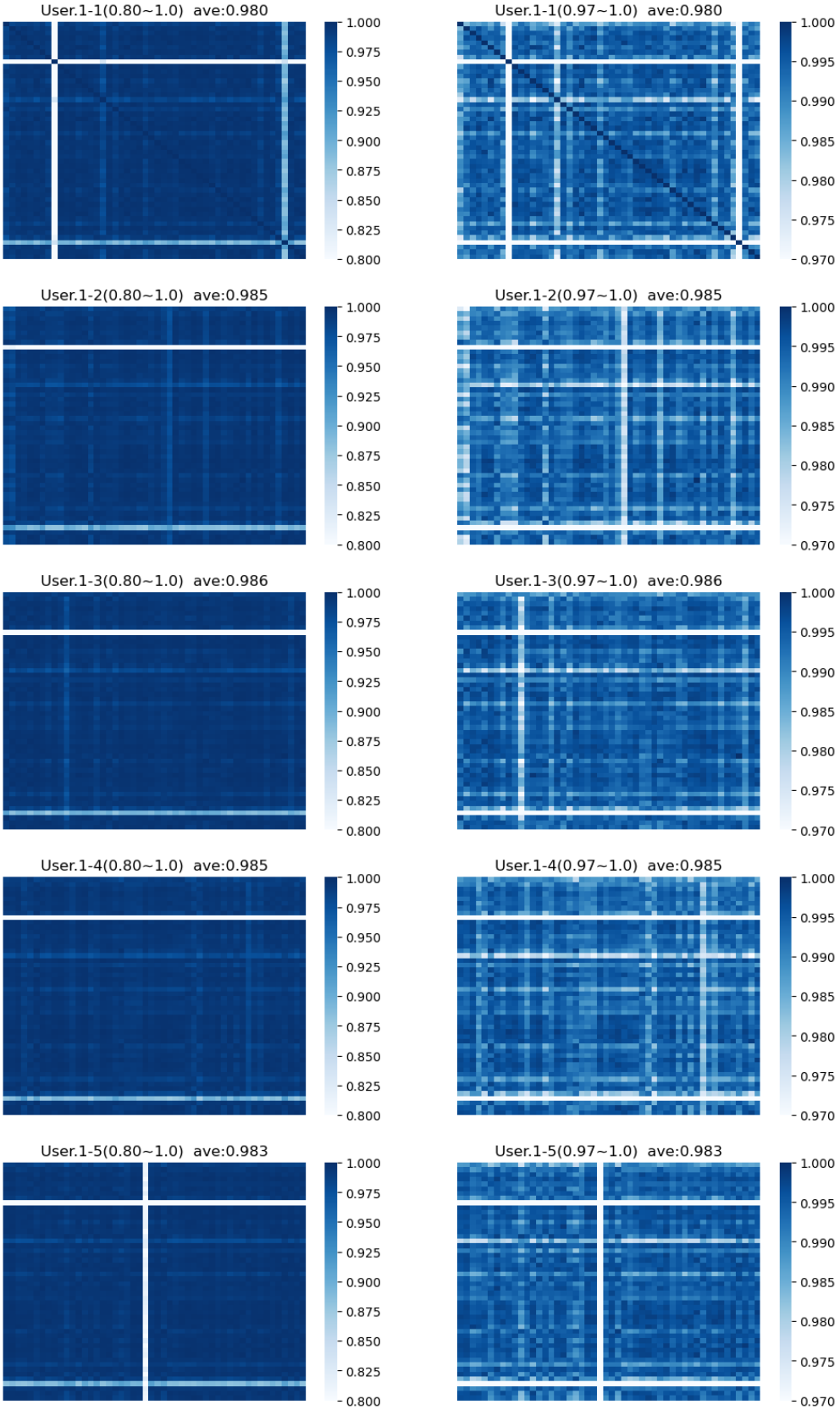

Figure 4.8 Evaluation of the MeCab+Word2vec on the dataset of bookmark.

4.4 考察

4.4.1 Sentence-BERT の影響

本実験では、東北大の BERT モデルに対して Sentence-BERT によるファインチューニングを行った。それによってモデルにどのような影響があったか、学習済みモデルと BERT モデルの検証結果の比較によって考察する。

まず 4.3.3 項、4.3.4 項で示された相違点として、文章の類似度の分布が異なる点がある。BERT モデルでの類似度は平均で 0.9 以上あり、非常に高い値で固まっている。一方で学習済みモデルでの類似度は 0.2〜0.8 と広い範囲に分散している。この違いの原因は、学習手法で使用した Triplet Loss だと考えられる。Triplet Loss は正解の組み合わせを近づけて不正解の組み合わせを遠ざけることで精度を向上させるが、Figure.4.2 で分かるように BERT モデルの状態でも 0.8 と高い精度がある。そのため Triplet Loss がモデルに与えた影響としては精度向上より、Anchor に対する相対距離を極端にして正解と不正解の分類を明確にする変化が大きかったと考えられる。事実、Figure.4.2 で示した差分では学習済みモデルが最も高い値だったことからも Triplet Loss の影響がみられる。また、4.3.3 項で示しているように、各ヒートマップには模様の類似性があった。これは Triplet Loss では相対距離を極端にしているだけで埋め込み表現同士の関係性が変化していないためであると考えられる。

以上のことから、学習済みモデルは Sentence-BERT の学習手法である Triplet Loss によって埋め込み表現の相対距離が極端になるような変化が加えられていることが考えられる。この変化によって狭い範囲に密集していた埋め込み表現が分散し、埋め込み表現の分析に適したモデルとなってるといえる。

4.4.2 学習済みモデルの有用性

検証によって学習済みモデルの特徴や他モデルとの違いが考察できた。そこから学習済みモデルの有用性について考察する。

4.3.3 項で学習済みモデルは同作者の場合に比較的高い類似度を算出できたことから、学習済みモデルの埋め込み表現は作者の文体を抽出できていると考えられる。他モデルでは類似度が全体的に高すぎることから同作者と異作者の判別が困難だったため、学習済みモデルは他モデルより優れていたといえる。また、学習済みモデルは他モデルより埋め込み表現の類似度の分布が広い特徴があった。そのため類似度を基にした各作品の

関係性が分かりやすく、分析に適した埋め込み表現だといえる。

一方で 4.3.4 項では、各作品の類似度をヒートマップにしてもブックマーク内の作品の関係性を見ることができなかった。しかし、「同ブックマークの作品は類似している」という意見はあくまで私の仮説であり、実際は関係性はない可能性もある。他モデルでも同様に特徴が見られなかったことから、アプローチが間違っている可能性も考えられる。

また、気になった点として 4.3.4 項で見られた極端に類似度が低い作品がある。この作品は BERT モデルと MeCab+Word2vec のモデルでヒートマップを作った際に見られたが、学習済みモデルのヒートマップからはわからなかった。原因としては前項で示したように、Triplet Loss で相対距離を極端にしたことで、学習以前に極端に離れていた埋め込み表現が目立たなくなってしまったことが原因だと考えられる。これによって特徴的な文体が抽出しずらくなっている可能性があり、Sentence-BERT による学習の弊害といえる。

第5章　結論

本研究の目的は、ネット小説の文体を評価可能な埋め込み表現を生成する機械学習モデルの作成である。そのために、Sentence-BERT を用いて埋め込み表現の生成に適したモデルを作成し、その有用性を検証した。

モデルの作成では、東北大学の BERT モデルを Sentence-BERT と Triplet Loss を用いてファインチューニングする方法を使用した。学習データには小説投稿サイト「小説家になろう」からウェブスクレイピングで収集した作品の本文データを使用し、ネット小説に特化したモデルを作成した。

また、モデルで生成した埋め込み表現の性能確認のため、同作者の作品の類以度と同ブックマークの作品の類以度でヒートマップを作成し、作品の関係性を読み取れるか検証した。その結果、学習済みモデルは BERT モデルによる埋め込み表現の相対距離を極端にしたモデルで、埋め込み表現同士の差分が他手法より大きい特徴があることが分かった。また、この特徴により作者の文体の判別が類似度から可能だと分かった。一方で、ブックマーク作品での検証では作品同士に関係性は見られず、有用性は確認できなかった。また、他モデルでは可能だった特徴的な文体の作品の判別が、学習済みモデルだと困難になるデメリットが見つかった。

以上の結果から、学習済みモデルは BERT モデルより埋め込み表現に適しており、作者の文体を判別可能という有用性があることが分かった。これが発展すれば、お気に入りの作者と類似した書き方をする作者を検索可能なシステムができると考える。一方でブックマークから作品の関係性は判別不能だという点や、学習前の BERT モデルでは可能だった特徴的な文体の判別が困難になっていた点など課題点も見つかった。これを解決するためには Sentence-BERT や Triplet Loss のハイパーパラメータの調整や、モデル同士を組み合わせたシステムで弱点を補う方法等が考えられる。また、今回は同小説内の文章かどうかを判別するタスクで学習したが、同作者の作品・文章かを判別するタスクや、同ブックマーク内の作品・文章かを判別するタスク等で学習することで異なる結果となると思われる。そのため、今後の展望として他手法でのアプローチも挑戦する必要があると考える。

謝辞

最後に、本研究を進めるにあたり、ご多忙中にも関わらず多大なご指導をしていただきました出口利憲先生、また、共に勉学に励んだ同研究室のメンバーに厚く御礼申し上げます。

参考文献

1) ニューラルネットワーク - Wikipedia.
https://ja.wikipedia.org/wiki/ニューラルネットワーク

2) Transformer (機械学習モデル) - Wikipedia.
https://ja.wikipedia.org/wiki/Transformer_(機械学習モデル)

3) Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762. 2017

4) Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805. 2018.

5) Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-networks. arXiv:1908.10084. 2019

6) Tohoku NLP Group. “cl-tohoku/bert-base-japanese-whole-word-masking”. Hagging Face. 2021.
https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking