修士論文

属性に対する極性判定を対象とした教師なし領域適応

LU Binghan

主指導教員 白井清昭

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和 5 年 3 月

Abstract

In recent years, there has been a lot of research focused on classifying the sentiment or polarity of reviews written by users. The task of classifying the polarity of a review (polarity classification) is a type of document classification, which is one of multi-class classification problems that predicts the polarity of a text. In general, supervised machine learning methods are frequently used for polarity classification, but there is a well-known problem that the performance of polarity classification tends to be poor when the domains of training data and test data are different.

To tackle the above problem, a technique called "domain adaptation" has been studied to reduce the difference or gap of two different domains as much as possible. Most of current domain adaptation methods on polarity classification of a user review have generally focused on a genre or medium of a text as the domain, but few studies have focused on an aspect. Aspect-based sentiment analysis is a task to classify whether an opinion expressed by a user towards an aspect of a product or service is positive or negative. However, since different labeled data is required for each aspect, a similar problem of ordinary domain adaptation may occur in aspect-based sentiment analysis, i.e., difference between training and test data of different aspects may decrease the accuracy of polarity classification.

The purpose of this study is to propose a domain adaptation method where an aspect is defined as a domain, and trains a model of polarity classification for a certain aspect that is different from one in the training data. Specifically, to automatically construct labeled data for a target domain, we combine two methods: (1) a method that automatically determines labels by using Bidirectional Encoder Representations from Transformers (BERT) trained from source domain data, and (2) a method that automatically generates labeled data for the target domain by replacing sentiment words and keywords in sentences of the source domain with those of the target domain. Here, the source and target domain refer to the domain of the training and test data, respectively. Finally, the automatically constructed labeled data is used to train a classifier that determines the polarity of the aspects in the target domain.

Labeled data of the target domain are constructed by the following two methods. The first method fine-tunes a BERT model with labeled data in the source domain, then uses it to annotate unlabeled reviews in the target domain with the polarity label. In this process, review sentences are discarded when the probability of the prediction of the label is lower than a pre-defined threshold, because the assigned polarity label may be incorrect. It enables us to construct high quality labeled data.

The second method builds labeled data of the target domain by the following

three steps: the extraction of domain specific sentiment words and keywords, the generation of review sentences of the target domain, and the filtering of the generated sentences. In this study, we call this method Cross Aspect Review Generation (CARG). For each source and target domain, we first extract the sentiment words frequently used in the reviews of that domain, i.e., aspect. Sentences in a domain corpus are split into words, then the polarity score for each word is calculated using the part-of-speech tagger and the sentiment dictionary, SentiWordNet, then words whose sentiment scores are greater than or equal to a threshold are extracted. Furthermore, "domain specific keywords", which are defined as words frequently appearing only in a certain domain, are extracted. Given a collection of reviews of different aspects, a set of reviews about one aspect is regarded as a single document, then the TF-IDF of a word is calculated. The words with the highest TF-IDF values are extracted as the domain specific keywords. Next, a new labeled sentence in the target domain is generated by replacing the sentiment words and keywords in the labeled sentence of the source domain with those in the target domain. We use the Masked Language Model (MLM) of BERT for word substitution. Finally, for each generated sentence, we measure the pseudo-log-likelihood (PLL) score, which evaluates the fluency of a sentence, then filter out the sentences whose PLL scores are low.

We fine-tune the BERT model using the dataset constructed by the above two methods to get a polarity classification model for the target domain. For fine-tuning, Focal Loss is used as a loss function to alleviate imbalance of polarity labels in the training data.

To evaluate the proposed method, we conducted experiments of domain adaptation of polarity classification on the restaurant dataset and the laptop dataset. The former consists of reviews of five aspects and the latter of four aspects. The performance of polarity classification was measured when one aspect is chosen as the source domain and another aspect as the target domain (called cross-domain setting). For comparison, we also performed polarity classification where the source and target domains are the same (called in-domain setting). The evaluation criterion is the accuracy for cross-domain, and the micro-average of the accuracy of five trials in five-fold cross-validation for in-domain. We compared the proposed method with the baseline methods and the method of previous work with respect to those evaluation criteria.

The results of the experiments showed that the proposed method outperformed the baseline methods in terms of the accuracy for 17 of the total 20 pairs of the aspects in the restaurant dataset. Compared with the previous work, the proposed method achieved the better accuracy in 14 of the total 20 aspect pairs. Comparing the systems with and without filtering of unnatural sentences, the accuracy of the

former was better for 12 of the 20 aspect pairs, indicating that the filtering was effective. Comparing the systems using the Focal Loss and the ordinary cross-entropy as the loss function, the accuracy of the former was the same or higher for 15 of the total 20 pairs. The use of Focal Loss was effective in this experiment. The average accuracy of the proposed method using both filtering and Focal Loss was 0.658, which was 0.025 and 0.013 points higher than the two baseline methods, respectively. It was the highest average accuracy among all the methods. Results of the experiments on the laptop dataset also showed that the proposed method outperformed the baseline methods. The best result among the compared methods was achieved by the proposed method that did not use Focal Loss as a loss function; its average accuracy of all combinations of the aspects was 0.801. It was better than the two baseline methods by 0.011 and 0.007 points, respectively. Compared to the restaurant dataset, however, the difference of the accuracy between the baseline and the proposed method was smaller. As for the comparison with the in-domain setting, for some pairs of the aspects, the accuracy of the proposed method in the cross-domain setting was higher than the micro-average of the accuracy in the in-domain setting. Since similar results were observed on two different datasets, the proposed method is robust in the sense that it can be applied to any genres of the review.

概要

近年，ユーザによって書かれたレビューの極性を分類する研究が盛んに行われている．レビューの極性を分類するタスク (極性判定) とは，文書分類の一種であり，テキストに含まれる感情の極性を予測する多クラス分類問題である．一般に，極性判定には教師あり機械学習の手法が用いられることが多いが，訓練データとテストデータでドメインが異なると極性判定の性能が低下する問題が知られている．

上記の問題に対する取り込みとして，異なるドメインにおけるデータの差異をなるべく小さくする「領域適応」と呼ばれる技術が研究されている．レビューの極性を分類するタスクに対する現在までの領域適応の手法では，一般にテキストのジャンルや媒体をドメインとするのが主流であるが，属性を対象とする研究は少ない．属性を対象とした極性判定とは，製品やサービスといった評価対象の属性に対してユーザが表明した意見が肯定的か否定的かを判定することを指す．しかし，属性毎にラベル付きデータを用意する必要があるため，属性に対する極性判定では異なる属性に対して訓練データとテストデータの違いが正解率の低下を招くという同様の問題が起こりうる．

本研究は，属性をドメインとみなし，ある属性に関するラベル付きデータから別の属性の極性判定のモデルを学習する領域適応の手法を提案することを目的とする．具体的には，ターゲットドメインのラベル付きデータを自動構築するために，ソースドメインのデータから学習した Bidirectional Encoder Representations from Transformers(BERT) を用いて自動ラベル付けを行う手法と，ソースドメインの文に出現する感情語や特徴語をターゲットドメインのそれに置換することで，ターゲットドメインのラベル付きデータを自動的に生成する手法を組み合わせて用いる．ここで，ソースドメインは訓練データのドメイン，ターゲットドメインはテストデータのドメインを指す．最後に，自動構築したラベル付きデータを用いて，ターゲットドメインの属性の極性を判定する分類器を学習する．

ターゲットドメインのラベル付きデータは，以下の 2 つの手法で構築する．1 つ目は，ソースドメインのラベル付きデータを用いて BERT モデルをファインチューニングし，ターゲットドメインのラベルなしデータに対してラベル付けを行う．この際，極性ラベルの予測確率が閾値より小さいレビュー文は，付与された極性ラベルが誤りである可能性があるため，除外する．これにより，高い品質を持つターゲットドメインのラベル付きデータを作成する．

2 つ目の手法は，感情語・特徴語の抽出，レビュー文の生成，生成文のフィルタリングの 3 つのステップによってターゲットドメインのラベル付きデータを構築する．本研究ではこれを Cross Aspect Review Generation(CARG) と呼ぶ．まず，ソースドメインとターゲットドメインのそれぞれについて，そのドメイン (属性) のレビューで使用される感情語を抽出する．あるドメインのレビュー文を単語に分割し，品詞タガーと感情語辞書 SentiWordNet を用いてそれぞれの単語の極性スコアを計算し，それが閾値以上の単語をドメインに固有の感情語として抽出する．また，特定のドメインだけによく使われる単語を「特徴語」と定義し，こ

れを抽出する．各属性のレビュー集合を仮想的にひとつの文書とみなし，単語の TF-IDF を計算し，この値が高い単語をドメインに固有の特徴語として抽出する．次に，ソースドメインのラベル付きデータにおける感情語もしくは特徴語をターゲットドメインの感情語もしくは特徴語に置き換えることにより，ターゲットドメインのラベル付き文を新たに生成する．単語の置き換えには BERT の Masked Language Model(MLM) を利用する．最後に，各生成文に対して，文の自然さを評価する擬似対数尤度スコアを測定し，スコアの低い文をフィルタリングする．

以上の 2 つの手法によって構築されたデータセットを用いて BERT モデルをファインチューニングし，ターゲットドメインの極性判定モデルを得る．ファインチューニングの際，Focal Loss を損失関数として使用し，極性ラベルの分布の偏りの問題に対処する．

提案手法の評価のため，レストラン・データセットとラップトップ・データセットを用いて極性判定の領域適応の実験を行った．前者は 5 つの属性，後者は 4 つの属性に関するレビューから構成される．ある属性をソースドメイン，別の属性をターゲットドメインとして，全ての属性の組について極性判定を行う実験を行った (クロスドメイン)．比較のため，ソースドメインとターゲットドメインが同じ場合の極性判定の実験も行った (インドメイン)．クロスドメインの実験では正解率を，インドメインの実験では 5 分割交差検証における 5 回の試行の正解率のマイクロ平均を評価指標とした．これらの評価指標について，提案手法とベースライン手法ならびに先行研究の手法と比較した．

実験の結果，レストラン・データセットでは，全 20 組の属性の組み合わせのうち 17 組については，ベースライン手法に比べて提案手法の正解率が上回った．先行研究との比較では，全 20 組のうち 14 組で提案手法の正解率が高かった．不自然な文をフィルタリングするシステムとしないシステムと比べて，全 20 組のうち 12 組については前者の正解率が高いことから，不自然な文をフィルタリングする手法は有効であった．Focal Loss を損失関数とするシステムと通常のクロスエントロピーを損失関数とするシステムを比較すると，全 20 組のうちの 15 組は前者の正解率が同じもしくは高かった．今回の実験では Focal Loss の導入が効果的であった．フィルタリングと Focal Loss を同時に導入した提案手法の平均正解率は 0.658 であり，2 つのベースライン手法と比べてそれぞれ 0.025，0.013 ポイント正解率を向上させ，比較した手法の中で最も高い平均正解率を示した．ラップトップ・データセットの実験では，レストラン・データセットの結果と同様に，ベースラインよりも提案手法の方が優れていることが確認された．比較した手法の中で最も結果が良かったのは，Focal Loss を損失関数としない提案手法であり，全てのドメインの組に対する平均正解率は 0.801 であった．これは 2 つのベースライン手法をそれぞれ 0.011，0.007 ポイント上回った．ただし，レストラン・データセットと比べて，ベースラインと提案手法の正解率の差は小さかった．インドメインの実験結果と比較すると，属性の組によっては提案手法によるクロスドメインでの正解率がインドメインの正解率のマイクロ平均より高いこともあった．2 つ

の異なるデータセットでおおよそ同様の結果が得られたことから，提案手法がどのようなジャンルのレビューにも適用できるという意味での汎用性を有することがわかった．

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

近年，EC サイトやレストラン・ホテルの口コミサイトなどが急速に普及しており，それに伴い，ユーザや顧客が商品・サービスに対する評価を積極的に投稿するようになっている．このような背景の下，ユーザによって書かれたレビューの極性を分類する研究が盛んに行われている．レビューの極性を分類するタスクとは，文書分類の一種であり，テキストに含まれる感情の極性を予測する多クラス分類問題である．極性のクラスとしては「肯定的」「否定的」「中立」などが用いられることが多い．このようなタスクは「極性判定」と呼ばれている．一般に，極性判定には教師あり機械学習の手法が用いられることが多いが，訓練データとテストデータでドメインが異なると極性判定の性能が低下する問題が知られている．

上記の問題に対する取り込みとして，異なるドメインにおけるデータの差異をなるべく小さくする「領域適応」と呼ばれる技術が研究されている．領域適応とは，十分なラベル付きデータが存在するソースドメイン (訓練データのドメイン) から得られた知識を，ラベル付きデータが全くあるいは少量しか存在しないターゲットドメイン (テストデータのドメイン) の分類器の学習に転移する技術である．レビューの極性を分類するタスクに対する現在までの領域適応の手法では，一般にテキストのジャンルや媒体をドメインとするのが主流である．例えば，パソコンをソースドメイン，レストランをターゲットドメインとして，パソコンに関するレビューに対して極性のラベルが付与されたデータから，レストランに関するレビューの極性を判定するモデルを学習する．

一方，極性判定の対象としては，文書 (例えばユーザレビュー全体)，文，属性などがある．属性を対象とした極性判定とは，製品やサービスといった評価対象の属性に対してユーザが表明した意見が肯定的か否定的かを判定することを指す．例えば，スマートフォンには「デザイン」「価格」「操作性」「バッテリ」などの属性があり，これらの属性に対するユーザの意見の極性を判定する．文書や文を対象とした極性判定と比べて，より詳細にユーザの意見を分析していると言える．

属性に対する極性判定でも，一般に属性毎にラベル付きデータを用意する必要があるため，訓練データとテストデータの違いが極性判定の性能を低下させるという問題が起こりうる．例えば，レストランの属性として「料理」と「価格」があり,「料理」に対するラベル付きデータしか存在しないとき，これから学習した分類器を「価格」に対する極性判定に適用しても高い正解率が得られない．この

ような訓練データとテストデータとで属性が異なる場合の属性に対する極性判定の性能が低いという問題も，解決するべき重要な課題である．解決策の例として，レビューで使用される単語は属性によって異なるため，この差異を分類モデルの学習時に考慮することが考えられる．すなわち，同じ単語でも，あるドメイン (属性) では重要だが，別のドメインではまったく重要ではない，ということがありうる．そのため，訓練データとテストデータで判定対象の属性が異なるときの極性判定の性能を向上させるため，極性判定の重要な素性となる感情語や特徴語をソースドメインからターゲットドメインに転移することが考えられる．

1.2 目的

本研究は，属性をドメインとみなし，ある属性に関する (ソースドメインの) ラベル付きデータから別の属性の (ターゲットドメインの) 極性判定のモデルを学習する領域適応の手法を提案することを目的とする．ターゲットドメインのラベル付きデータを自動構築するために，ソースドメインのデータから学習した Bidirectional Encoder Representations from Transformers(BERT)[7] を用いて自動ラベル付けを行う手法と，ソースドメインの文に出現する感情語や特徴語をターゲットドメインのそれに置換することで，ターゲットドメインのラベル付きデータを自動的に生成する手法を組み合わせて用いる．最後に，自動構築したラベル付きデータを用いて，ターゲットドメインの属性を極性を判定するモデル (分類器) を学習する．

さらに，提出手法の有効性を実験により評価する．属性に対する極性判定の 2 つのデータセットを用いて，提出手法とベースラインによる極性判定の正解率を比較する．

1.3 本文の構成

本論文の構成は以下の 5 章から構成される．第 2 章では，本論文の関連研究を紹介する．第 3 章では，提案手法の詳細を述べる．第 4 章では，提案手法の評価実験と結果について報告する．最後に，第 5 章では，本論文のまとめと今後の課題について述べる．

第2章　関連研究

本章では，本論文に関連する研究について述べる．2.1 節では，極性判定に関する研究を紹介する．2.2 節では，極性判定を対象とした領域適応に関する研究を紹介する．2.3 節では，本研究で利用する言語モデルについて紹介する．2.4 節では，文の自然さを測る研究を紹介する．最後に，2.5 節では，本研究と先行研究の違いについて論じる．

2.1　極性判定に関する研究

2.1.1　極性辞書を利用した手法

極性判定に関する初期の研究では，極性辞書が用いられることが多い．極性辞書とは，肯定的または否定的な意味を持つ単語を収録し，かつ各単語の極性の種類や強さの情報を含む辞書である．

Turney は PMI-IR(Pointwise Mutual Information and Information Retrieval) というアルゴリズムを考案し，これを用いた句を対象とした極性判定の手法を提案した [17]．PMI(自己相互情報量) とは，一般には 2 つの事象の関連性の強さを測る指標であるが，ここでは句とその句の中に出現しかつ極性を持つ単語との共起関係を測定するために用いられる．ある句の極性スコア (Semantic Orientaton Score) は，肯定の極性を持つ単語と否定の極性を持つ単語の PMI の差を計算することで算出される．例えば，極性を判定したい句に対し，その句が肯定的極性を持つ単語 “excellent” と否定的極性を持つ単語 “poor” を含むとき，句と肯定的単語 “excellent” との共起関係の方が強ければ，句の極性スコアは正の値となる．逆に，句と否定的単語 “poor” との共起関係の方が強ければ，句の極性スコアは負の値となる．

PMI-IR を用いた句の極性判定の手法は以下の 3 つのステップから構成される．ステップ 1 では，入力文の各単語に対して品詞付けを行う．ステップ 2 では，あらかじめ定義された主観表現抽出ルールによって極性判定の対象となる句を抽出する．ステップ 3 では，句 a と，その句の中の単語 b の共起関係 PMI(a,b) を計算し，句の極性を判定する．PMI(a,b) を算出するためには大規模なコーパスが必要だが，Turney は検索エンジンに「a NEAR b」というクエリ (NEAR は 2 つのキーワードが近くに出現することを表す) を与えたときのヒット数などを用いて PMI を近似的に算出している．“Automobiles” と “Banks” の 2 つのドメインのデータセット

を用いた評価実験の結果，句の極性判定の正解率はそれぞれ 84.00%と 80.00%となり，提案手法の有効性を確認した．

Turney は，同じ論文で，文中の評価表現の比率に基づいて文章の極性を判定する手法を提案している [17]．前述の PMI によって句の極性スコアを算出する手法を利用して，文書に含まれかつ肯定的または否定的な極性を持つ評価表現 (句) を抽出する．そして，抽出された全ての評価表現の極性値の平均値を求める．この値が正のときは文書全体の極性をポジティブ (肯定) と，負のときはネガティブ (否定) と判定する．

2.1.2 機械学習に基づく手法

近年の極性判定の手法は機械学習に基づく手法が主流であり，このような研究の動向を調査したサーベイ論文 [8] も存在する．本節はその関連研究をいくつか紹介する．

Pang らは教師あり機械学習の手法と機械学習の素性の最適な組み合わせを実験的に求めた [12]．機械学習の手法として，単純ベイズ分類器，最大エントロピー分類器，Support Vector Machine(SVM) の 3 つの教師あり機械学習手法を用いた．一方，素性については，uni-gram や bi-gram などを用いた．公開されたラベル付きの映画レビューデータを利用して極性判定モデルを学習した．評価実験の結果，3 つの機械学習アルゴリズムの中では，全般的には SVM の正解率が最も高いことを示した．最高の正解率が得られた組み合わせは，単語 uni-gram を素性として SVM を学習したときで，その正解率は 82.9%であった．

Wang らは，属性に対する極性判定のタスクに対し，Attention 機構に基づいたモデルを提案している [18]．著者は “Staffs are not that friendly, but the tasete covers all.” という例文を挙げ，料理の味に対しては顧客の評価は肯定的だが，サービスに対しては否定的である，と述べている．すなわち，同じ文でも属性によって極性が異なることを指摘している．以上の考察を踏まえ，Wang らは Attention 機構を組み込んだ Long Short-Term Memory(LSTM) を提案し，異なる属性に対して LSTM モデルが文の異なる部分に注目する (Attention をかける) ようにした．Attention を計算する際に属性の情報を考慮するために，以下の 2 つのアプローチを考案した．1 つ目は，入力文の各単語の埋め込み表現に属性の埋め込み表現を付加する．2 つ目は，Attention の重みを算出する際，属性の埋め込み表現と隠れ層の出力を連結してまとめて算出する．属性に対する極性判定のデータセットとして，SemEval 2014 Task4 のデータセットを使用し，評価実験を行った．レストランに関するレビューを対象にその極性を 3 値 (肯定，否定，中立) または 2 値 (肯定，否定) に分類したときの正解率は，ベースラインモデルと比べて，提案モデルの性能が最も高いことを示した．

青嶋と中川は BERT モデルを経済テキストのセンチメント分析 (感情分析) に用いている [21]．まず，日本語版の Wikipedia データセットに対して適切な前処理を

行い，大規模なコーパスを作成し，それを BERT モデルの事前学習に用いる．さらに，経済テキストに書かれている景気に対する評価を判定するために，BERT をファインチューニングする．実験には景気ウォッチャー調査の公開データセット用いた．このデータセットには，景気に敏感な人 (景気ウォッチャー) に対して行ったアンケート調査の回答テキストに対し，景気ウォッチャーが表明している景気に対する判断 (5 段階評価で「良い」,「やや良い」,「変わらない」,「やや悪い」,「悪い」) が付与されている．このデータをランダムに 6 割，2 割，2 割に分割し，それぞれ訓練データ，検証データ，評価データとして，BERT モデルのファインチューニングと評価を行った．実験の結果，BERT モデルの使用によって,「良い」の再現率を除き，他の全てのラベルにおいて精度，再現率，F 値が先行研究と比べて改善したことを示した．経済テキストを対象とした感情分析に対する BERT モデルの有効性が確認された．

2.2 極性判定を対象とした領域適応に関する研究

本節では，ジャンルに対する極性判定を対象とした領域適応に関する研究を紹介する．このような先行研究は多数存在している．

Blitzer らは，Structural Correspondence Learning(SCL) という領域適応の手法を提案している [9]．ソースドメインとターゲットドメインという 2 つの異なる分野のテキストデータがあると仮定する．ソースドメインのデータのみ正解のラベルが付与されている状況において，ターゲットドメインのデータの極性判定の正解率を向上させることを目的とする．SCL の手法では，ソースドメインとターゲットドメインのラベル付けされていないデータを用いて，両ドメインに共通する特徴語を発見する．つまり，複数のドメインにわたって頻繁に出現しかつ意味的に類似している特徴語を発見する．Blitzer らは，その共通する特徴語を “pivot” と呼び，それを抽出する．そして，pivot とそれ以外の素性との相関関係によって pivot ではない素性を選択する．次に，ターゲットドメインの極性判定の分類モデルを以下のように学習する．まず，2 つのドメインで頻出する pivot を抽出する．次に，2 つのドメインのラベルなしデータにおける各 pivot の出現を予測する線形 pivot 予測器を学習することで，pivot 特徴と他の全ての特徴との相関を分類モデルに学習させる．

Books，DVD，Electronic，Kitchen の 4 つのドメインのレビューとその極性ラベルが付与されたデータセットを用いて評価実験を行った．上記の SCL と，pivot 選択のときに頻度だけでなく相互情報量 (Mutual Infomation,MI) も利用する SCL-MI という手法を比較した．ソースドメインとターゲットドメインが同じであるインドメインの実験では, 4 つのドメインのそれぞれについて，極性判定の正解率は 80.4%，82.4%，84.4%，87.7%であった．異なるドメインを対象にしたクロスドメインの実験では，Book をソースドメイン，Electronic をターゲットドメインとする組以外の組み合わせにおいて，SCL-MI は SCL より高い正解率を示した．

Rietzler らは，ソースドメインのラベル付きデータで BERT をファインチューニングする前に，ターゲットドメインのラベルなしテキストを用いて BERT の事前学習を再度行う (以下,「再事前学習」と呼ぶ) ことで領域適応を行う手法を提案した [14]．また，BERT モデルの再事前学習の際の学習ステップ数が極性判定の性能に与える影響を分析した．ここでの学習ステップ数とは，事前学習データの文の数×エポック数と定義され，事前学習に用いる文の総数 (延べ数) を表す．実験には SemEval 2014 のデータセットを用いた．このデータセットは，ラップトップパソコンとレストランの 2 つのドメインについて，属性の極性が付与されたデータである．再事前学習に使うデータセットによって BERT-ADA Lapt，BERT-ADA Rest，BERT-ADA Joint の 3 つのモデルを構築した．ここで，Lapt，Rest，Joint は，ラップトップ，レストラン，両方のレビューを再事前学習に用いたことを表す．

評価実験の結果，BERT モデルの再事前学習は，極性判定タスクの性能に大きな影響を与えることがわかった．学習ステップ数が結果に与える影響について，レストランドメインでは学習ステップ数が 250 万程度の段階でも性能が向上し始めるのに対し，ラップトップドメインでは，正解率の大幅な向上を得るためには最低でも 1,000 万の学習ステップ数が必要であった．レストランデータが訓練データからラップトップに適応する場合，BERT-ADA Lapt モデルの正解率は 77.92%で最も高い正解率を示した．ソースドメインがラップトップでターゲットドメインがレストランのとき，BERT-ADA Rest モデルの正解率は 83.68%となり，領域適応を行わない XLNet と BERT によるベースラインと比べて 1.27 と 3.61 ポイント向上した．BERT-ADA Joint については，ラップトップとレストランの両方を合わせて訓練データとした場合，BERT-ADA Lapt や BERT-ADA Rest と比べて高い正解率が得られた．

白らは，感情分析の教師なし領域適応の手法として，BERT の単語埋め込むの利用を試みた [16]．BERT によって出力される単語埋め込みの平均ベクトルにより文書の特徴ベクトルを構築し，これを素性とする分類器を学習するが，BERT の最上位層の単語埋め込み表現ではなく，1 つ下の層の単語埋め込み表現を利用して特徴ベクトルを構築する手法を提案した．これは，BERT の最上位層よりも 1 つ下の層の埋め込み表現の方がドメインに特化しない単語の汎用的な意味が反映されているという考えに基づく．まず，文書 d に対して Bag of Words モデルと TF-IDF から文書の特徴ベクトル v_b を作る．次に，d を単語ごとに分割し，その単語列を BERT に入力し，単語埋め込み表現列を得る．最上位層の単語埋め込み表現列から作られる平均ベクトルを v_{-1} とする．また最上位層の 1 つ下の層の単語埋め込み表現列から作られる平均ベクトルを v_{-2} とする．v_b と v_{-2} を連結したベクトル $[v_b; v_{-2}]$ を文書 d の特徴ベクトルとし，これを入力とする 3 層のニューラルネットワークを極性判定の分類器として学習する．また，$[v_b; v_{-1}]$ を特徴ベクトルとする標準的な手法をベースラインとし，これと比較する．

評価実験では，Amazon のレビュー文書のデータセットを使用した．このデータセットは様々なジャンルのレビューを含むが，そのうち Books(B)，DVD(D)，

Music(M) の 3 つのジャンルをドメインとした．領域適応の組み合わせとしては B → D，D → M，M → B，B → M，M → D，D → B の 6 通りがある．ここで矢印の左はソースドメイン，右はターゲットドメインを表わす (ソースドメイン→ターゲットドメイン)．実験の結果，M → B，B → M，D → B の 3 つ (全 6 通りの組み合わせの半数にあたる) の組み合わせについて，提案手法の正解率はベースライン手法を上回った．さらに，最上位層より 1 つ下の階層よりもさらに下位の層の単語埋め込み表現列を用いて学習された分類器の性能を評価した．その結果，M → B の組み合わせでは上から 5 番目の階層の単語埋め込みを用いたときに最高の正解率が得られた．しかし，6 通りのドメインの組み合わせの平均の正解率では，BERT の最上位層の単語埋め込みを用いたときが正解率が最も高く，下の階層ほど正解率が低下する傾向が確認された．

Xi らは，Category Attention Network(CAN) というモデルを考案し，CAN と Convolutional Neural Network(CNN) を統合したモデル (CAN-CNN) を提案した [19]．ソースドメインとターゲットドメインに共通して使われる感情語とターゲットドメインに固有の感情語を統一的なカテゴリ属性語と見なし，それらを自動的に抽出することで領域適応の性能を向上させる．CAN は Category Memory Module(CMM)，Dynamic Matching(DM)，Category Attention Layer(CA) の 3 つのモジュールから構成されている．

第 1 の Category Memory Module では，ソースドメインのラベル付きデータから規則によってカテゴリ属性語を抽出する．また，CMM を domain-aware CMM に改良して，ターゲットドメインにおけるカテゴリ属性語を抽出する．カテゴリ属性語の候補の単語ベクトルをランダムに初期化し，モデル学習時にカテゴリ属性語の重みの分布損失を最小化することで単語ベクトルを学習・更新する．第 2 の Dynamic Matching は，単語埋め込み空間において，単語間の類似度を単語ベクトルのコサイン類似度で測り，入力文の各単語と最も類似した上位 k 位のカテゴリ属性語を動的にマッチングさせる．第 3 の Category Attention Layer では，入力文の特徴ベクトルを得る際，DM によってマッチングされたカテゴリ属性語を入力文に相互作用させて，カテゴリ (ドメイン) の特徴を反映させるようにベクトルを拡張する．極性判定の分類器として，最終のベクトルを入力とした CNN モデルを学習する．

評価実験では，Customer Review，Amazon Fine Foods，Movie Review の 3 つのドメインのレビューから構成される極性判定のデータセットを使用した．6 つのドメインの組み合わせについて，提案手法ならびに複数のベースライン手法の極性判定の正解率を比較した．その結果，提案手法の CAN-CNN は，複数のドメインの組について，ベースライン手法よりも高い正解率が得られた．

Yu らは，Cross-Domain Review Generation(CDRG) と呼ばれる新しい領域適応の手法を提案した [20]．この手法では，ソースドメインでのラベル付きレビューからターゲットドメインのラベル付きレビューを自動的に生成する．まず，ソースドメインに固有の属性語もしくは感情語をマスクする (特殊トークン [MASK]

に置換する) ことにより，ソースドメインのレビュー文をドメインに依存しないレビュー文に変換する．次に，言語モデルを用いて，そのドメインに依存しないレビュー文の [MASK] にターゲットドメインの固有の属性語もしくは感情語を埋めることで，ターゲットドメインのレビュー文に変換する．最後に，End-to-End Aspect-Based Sentiment Analysis(E2E-ABSA) と Aspect Extration(AE) タスクにおいて，Independent Training と Merge Training の 2 つの方式で領域適応の実験を行う．Independent Training では，自動生成されたターゲットドメインのレビュー文のみを訓練データとする．Merge Training では，ソースドメインのラベル付きデータと生成されたターゲットドメインのレビュー文の両方を訓練データとする．

評価実験では，Laptop(L), Restaurant(R), Device(D)，Service(S) の 4 つのドメインのレビューから構成されるデータセットを用いた．領域適応の実験設定としてこれら 4 つのドメインの組み合わせを作る際，Laptop と Device は非常に似ているため，実験対象から除いた．他のドメインの組み合わせにおいて，Independent Training と Merge Training によって分類器を学習した．先行研究との公正的な比較のため，評価指標として Micro-F1 スコアを用いた．実験の結果，Independent Training では，すなわち CDRG による生成文だけで BERT のファインチューニングを行ったモデルの Mirco-F1 スコアは，既存のベースラインを上回ることがわかった．これは，ドメインに固有の属性語・感情語を置換することでダーゲットドメインのレビュー文を生成する CDRG のアプローチの有効性を実証するものである．また，MergeTraining，すなわち生成されたターゲットドメインのレビューとソースドメインのラベル付きレビューをマージすることで，Micro-F1 スコアが更に向上することを示した．

2.3 BERT

BERT(Bidirectional Encoder Representations from Transformers)[7] は，自然言語処理の様々なタスクに適用できる汎用的な大規模言語モデルである．本節では，BERT の概要を紹介する．BERT の学習は，事前学習とファインチューニングという 2 つの手続きから構成される．

BERT の事前学習とは，事前に大量のラベルなしテキストを用いて特定のタスクに依存しない汎用的な言語モデルを学習することである．ここでの汎用的な言語モデルとは，文または文の組に対し，その意味を表す抽象表現を与えるモデルを指す．

BERT の事前学習には，Masked Language Model と Next Sentence Prediction の 2 つのタスクがある．Masked Language Model では，入力文の一部を [MASK] という特殊なトークンに置き換える．そして，置き換えた文を BERT に入力し，[MASK] の位置に出現するべき単語を予測するというタスクを解くことで汎用言語モデルを学習する．つまり，マスクされた単語を周りの単語から予測するという穴埋めのタスクである．一方，Next Sentence Prediction タスクでは，2 つの文

が与えられたとき，それらが連続して出現するか否かを判定するタスクを解くことで汎用言語モデルを学習する．具体的には，2 つの入力文の間を [SEP] という特殊トークンで連結し，さらに文頭に [CLS] という分類用の特殊トークンを付与した上で，BERT の分類器に入力する．2 つの文が連続した場合は IsNext を出力し，そうでなければ NotNext を出力するようなモデルを学習する．このタスクにより，BERT は 2 つの文の関係性を学習できる．

BERT のファインチューニングとは，特定のタスクの少量のラベル付きデータを用いて，そのタスクを解くための分類器を学習することである．例えば，属性に対する極性判定のタスクを解きたいとき，そのラベル付きデータを用意し，事前学習によって得られた汎用言語モデルを極性判定に適したモデルに変換する．事前学習済みの BERT のパラメータをタスクに適した方向に調整することで，そのタスクに対する性能を向上させることができる．BERT は，文または文の組の抽象表現を得るための階層と，抽象表現を素性として特定のタスクに対する出力を得る分類器の階層から構成されるが，ファインチューニングではその両方のパラメタが更新される．

2.4 文の自然さを測る研究

機械翻訳，自動要約，対話システムにおける応答生成など，システムが文を自動生成する研究は数多く行われている．自動生成された文がどれだけ自然な文であるかを測ることは，自然言語処理における重要な要素技術である．3.3.4 項で述べるように，本研究ではダーゲットドメインのラベル付きレビュー文を自動生成し，その品質を評価する．そこで，本節では文の自然さを測る先行研究を簡単に紹介する．

文の自然さを測る従来の研究では，式 (2.1) に示す単語の n-gram モデルが利用されてきた．

$$P(s) = P(w_i \cdots w_m) = \sum_{i=1}^{m} \log P(w_i | w_{i-n+1} \cdots w_{i-1}) \tag{2.1}$$

$P(s)$ は文 s のスコアで，s は m 個の単語列として表される．単語 w_i の生成確率は，その直前に出現した $n-1$ 個の単語列 $w_{i-n+1} \cdots w_{i-1}$ を条件とする条件付き確率で計算する．すなわち，今まで出力した単語から次の単語を予測する条件付き確率の対数尤度の和を文のスコアとし，文の自然さを測定する．

Julian らは，BERT[7] や RoBERTa[11] のような Masked Language Model(MLM) モデルで文章の自然さを計算するための PLL スコア (pseudo-log-likelihood scores) を提案した [15]．PLL スコアは，式 (2.2) に示すように，文中の各単語 w_t を BERT の MLM で予測した確率の対数尤度の和である．

$$PLL = \log P_{MLM}(W) = \sum_{t=1}^{|W|} \log P_{MLM}(w_t|W_{\setminus t}) \tag{2.2}$$

文献 [15] の評価実験では，PLL スコアが，自己回帰モデルによる言語モデルのスコアと同程度，あるいはそれよりも高い確率で，言語学的に正しい文であるかを判断できる指標になることを示している．

2.5 本研究の特色

2.2 節で述べたように，極性判定を対象とした領域適応に関する先行研究では，レビューのジャンルをドメインとした領域適応の手法が提案されていたが，本研究では，属性をドメインとした教師なし領域適応の手法を提案する．ここで教師なし領域適用とは，ターゲットドメインのラベル付きデータが存在しないことを仮定することを意味する．Yu らによる CDRG[20] を拡張し，ターゲットドメインの属性に対するラベル付きデータを自動的に生成する．また，ターゲットドメインのラベルなしデータに対し，ソースドメインのラベル付きデータからファインチューニングされた BERT を用いて極性判定を行い，擬似ラベルを付与することでターゲットドメインのラベル付きデータを構築する手法も併用する．

第3章　提案手法

3.1　概要

本章では，属性に対する極性判定のタスクにおいて，属性をドメインとした領域適応の手法を提案する．提案手法の概要を図 3.1 に示す．ラベル付きデータが存在する属性をソースドメイン，存在しない属性をターゲットドメインとする．図中の (S),(T) はそれぞれソースドメイン，ターゲットドメインのデータを表す．

ターゲットドメインのラベル付きデータを 2 つの手法で自動生成する．1 つは自動ラベル付けによる手法 (図 3.1 (A)) である．ソースドメインのラベル付きデータを用いて BERT モデルをファインチューニングする．このモデルを BERT(S) と記す．それを用いてターゲットドメインのラベルなしデータに対してラベル付けを行う．詳細は 3.2 節で述べる．

もう 1 つは，CDRG[20] を拡張し，ソースドメインのレビュー文から新たにターゲットドメインのラベル付きデータを自動生成する手法 (図 3.1 (B)) である．本研究ではこれを Cross-Aspect Review Generation (CARG) と呼ぶ，まず，ソースとターゲットのデータに対して感情語・特徴語を重要語として抽出し，感情語・特徴語のリストを作成する．次に，大量のラベルなしデータを使って BERT の MLM を再学習する．最後に，ソースドメインのラベル付きの文に含まれる感情語・特徴語を [MASK] に置き換え，BERT の Masked Language Model(MLM) を用いて [MASK] にターゲットドメインの感情語・特徴語を埋めることで，ターゲットドメインのデータを自動的に生成する．その詳細を 3.3 節で述べる．

最後に，上記 2 つの手法で作成したターゲットドメインの属性のラベル付きデータを訓練データとし，BERT によりターゲットドメインの極性判定モデルを学習する．詳細は 3.4 節で述べる．

3.2　自動ラベル付けによる訓練データ構築

本節では，BERT を用いた自動ラベル付けによってターゲットドメインの訓練データを構築する手続きについて述べる．

まず，ソースドメインのラベル付き訓練データを用いて，事前学習済みの BERT をファインチューニングし，極性判定の分類器を得る．この分類器は図 3.1 の BERT(S) に相当する．BERT のファインチューニングとは，2.3 節で述べたように，事前学

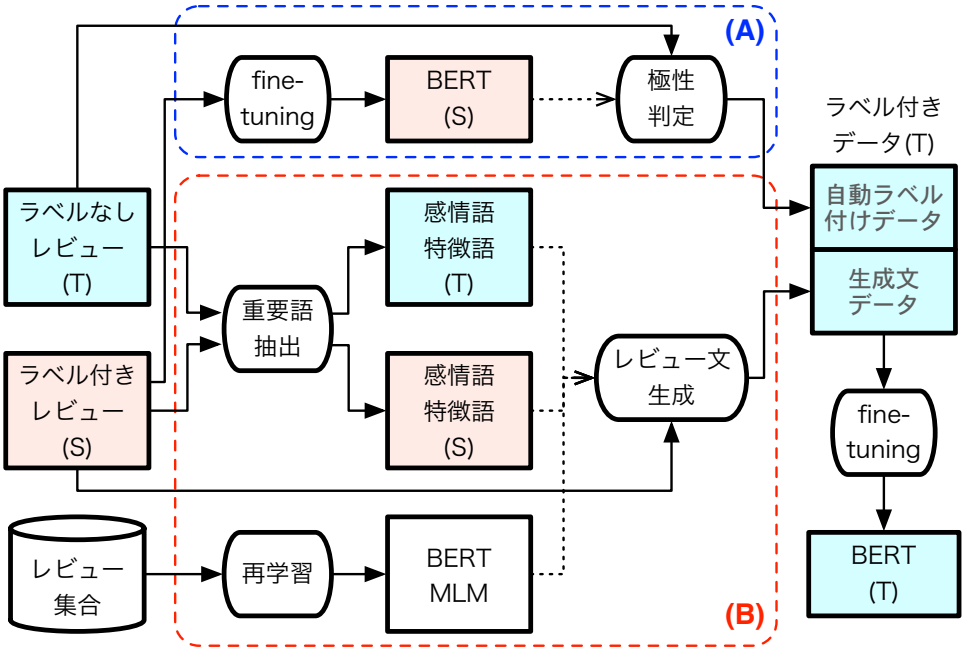

図 3.1: 提案手法の概要

習した汎用言語モデルを特定のタスク (ここでは極性判定) を解くためのモデルに調整する手続きである．ここで，事前学習済みの BERT モデルとして bert-base-cased[2] を使用する．これは言語処理タスクにおいて広く使用されている事前学習済みモデルの一つである．

次に，ファインチューニング済みの BERT(S) を用いてターゲットドメインの文に対して極性判定を行う．また，それぞれの文に対して，BERT による極性の予測確率も算出する．そして，予測された極性の予測確率が T_p 以上のものに対して，それに対応する極性のラベルを付与する．一方で，極性の予測確率が T_p より小さい文に対してはラベルを付与せず，コーパスから除外する．BERT(S) はソースドメインのラベル付きデータを用いて学習されているので，ターゲットドメインに対する極性判定の正解率はそれほど高くないと考えられる．すなわち，しばしば極性の判定を誤ることが予想される．予測確率が低い文を除外することで，より高精度な自動ラベル付きレビュー文を得ることを狙う．以下，BERT(S) によって自動的に付与されたラベルを「擬似ラベル」と呼ぶ．

自動ラベル付けの例を表 3.1 に示す．この表は，4.1 節で後述するレストラン・データセットのレビュー文に対して，どのような擬似ラベルを付与するか，あるいは付与しないのかを示している．「ソースドメイン」「ターゲットドメイン」はそれぞれに該当する属性を表し，前者のラベル付きデータから BERT モデルを学習し，後者のレビュー文の極性ラベルを決定する．「予測確率」の列は 4 つの極性クラスのそれぞれに対する予測確率を示しており，太字は一番高い予測確率を表す．「正解」の列は，該当するレビュー文の実際のラベルである．「擬似ラベル」の列は，

本手法によって割り当てられた極性ラベル，もしくは極性ラベルを割り当てないことを表す．擬似ラベルを付与する予測確率の閾値 T_p は 0.8 に設定されているとする．

表 3.1: ターゲットドメインのレビューに対する自動ラベル付けの例

ドメイン		レビュー文	予測確率				正解	疑似ラベル (T_p=0.8)
ソース	ターゲット		negative	neutral	conflict	positive		
service	food	The sauce is delicious and the crust is perfect.	0.0062	0.0516	0.0042	**0.9380**	positive	postive
service	price	It's true, this place is not cheap.	**0.5107**	0.0144	0.4697	0.0052	positive	付与しない
service	price	Could be pricey without a prix fixe meal.	**0.9904**	0.0012	0.0078	0.0006	negative	negative
service	ambience	People dress in suits or evening gowns as well as shirts jeans.	0.0019	0.0031	0.0017	**0.9933**	neutral	positive
food	service	Again, the waitress was awesome.	0.0002	0.0011	0.0005	**0.9982**	positive	positive
food	price	You'll pay at least double at any other Italian restaurant in the city, and most still don't compare.	0.0317	**0.7004**	0.1307	0.1372	positive	付与しない

最初の例では，food 属性のデータセットのレビュー文 "The sauce is delicious and the crust is perfect." に対し，service 属性のデータセットでファインチューニングされた BERT(S) を使って極性を判定する．それぞれの予測確率は negative:0.0062，neutral:0.0516，conflict:0.0042，positive:0.9380 である．positive クラスの予測確率が一番高く，かつ事前に設定された閾値 0.8 より大きいので，positive の疑似ラベルを付与する．このレビュー文の実際のラベルも positive なので，擬似ラベルが正しく付与されている．

2 番目の例では，price 属性のレビュー文 "It's true, this place is not cheap." に対する BERT の予測確率は negative:0.5107，neutral:0.0144，conflict:0.0144，positive:0.0052 である．最大の予測確率を持つ極性クラスは negative であるが，閾値 0.8 より小さいので，判定の信頼度が低いとみなし，疑似ラベルを付与しない．

4 番目の例では，ambience 属性のレビュー文 "People dress in suits or evening gowns as well asshirts jeans." の極性を判定し，positive の予測確率が 0.9933 と最大であるので，positive の疑似ラベルが付与されている．ただし，レビュー文の実際のラベルは neutral なので，誤った擬似ラベルが付与されたことになる．

3.3 Cross-Aspect Review Generation

本節では，Cross-Aspect Review Generation (CARG)，すなわちターゲットドメインのレビュー文を自動的に生成する手法を説明する．3.3.1 項と 3.3.2 項では，それぞれ，感情語もしくは特徴語を抽出する手法について説明する．3.3.3 項はレビュー文の生成について述べる．3.3.4 項では，生成した文の中から不自然なものを除外するフィルタリングについて述べる．

3.3.1 感情語の抽出

ソースドメインとターゲットドメインのそれぞれについて，そのドメイン (属性) のレビューで使用される感情語を抽出する．感情語とは “excellent”, “bad” など書き手の感情や評価を表す単語である．感情語の抽出には SentiWordNet[5] を用いる．同辞書は，単語の感情的なニュアンスを評価するための言語資源である．SentiWordNet においては，それぞれの単語は，肯定のスコアと否定のスコアが付与されている．肯定的なスコアは 0 から 1 の値で，1 に近いほど肯定的な極性を持つことを表す．否定的なスコアも同様に 0 から 1 の値で，1 に近いほど否定的な極性を持つ．これらのスコアを用いて，単語の感情極性を評価することができる．

単語は一般に複数の意味 (語義) を持つ．SentiWordNet では，正確には単語ではなく，語義に対して肯定的スコアと否定的スコアが定義されている．ここで，語義は品詞毎に区別されている．また，それぞれの単語について，語義は出現頻度の順にランク付けされている．

本研究では，SentiWordNet を利用し，単語のそれぞれの語義に対して割り当てられているスコアを用いて，単語の極性スコアを計算する．具体的には，単語 w の極性スコア $SS(w)$ を式 (3.1) のように定義する．

$$SS(w) = \sum_{i=1}^{n} \frac{1}{n} \cdot |pos(w,i) - neg(w,i)| \tag{3.1}$$

i は語義の出現頻度のランク，$pos(w,i)$ と $neg(w,i)$ はそれぞれ単語 w の i 番目の語義の肯定と否定のスコア，n は単語 w の語義の総数を表す．すなわち，語義毎に肯定的スコアと否定的スコアの差の絶対値を求め，語義のランクに応じて重みを与えて加算する．出現頻度のランクが高い語義ほど大きい重みを，ランクが低い語義ほど低い重みを与えることにより，単語の最も一般的な語義に対する極性スコアがより重視され，単語の極性スコアをより正確に計算できるという考えに基づく．最終的に，ドメインに固有の単語を抽出する際には，$SS(w)$ が閾値 T_s 以上の単語を抽出する．

単語の極性スコアを計算する具体的な手続きについて説明する．まず，レビュー文のそれぞれの単語の品詞を決定する．本研究では，品詞付けには NLTK[3] と呼ばれるライブラリを用いる．レビュー文に対する品詞付けは以下のように行う．

1. あるドメインのレビュー文を文字列として変数に格納する．
2. NLTK の pos tag 関数を使用して形態素解析を実行する．pos tag 関数は，文字列で表わされた文を受け取り，それを単語に分割し，各単語に対して品詞をタグ付けした結果を返す．
3. pos tag 関数の解析結果を適切にフォーマットし，品詞付きの文を出力する．

次に，以下の手続きにしたがって，SentiWordNet を利用して単語の極性スコアを計算する．

1. NLTK の SentiWordNet の API[4] を利用し，SentiWordNet のデータベースを読み込む．

2. SentiWordNet のデータベースには，それぞれの単語に対して品詞に応じた複数の語義が登録されている．品詞がタグ付けされた文中の各単語について，SentiWordNet を検索し，単語が属している品詞に対する全ての語義の極性スコアを得る．すなわち，各語義に対応する極性スコア (Positive, Negative の差の絶対値) を計算する．

3. 得られた各語義の極性スコアに，その語義の出現頻度のランクの逆数を重みとして掛ける．そして，全ての語義について重み付きの極性スコアを加算し，単語の極性スコアを得る．

表 3.2 は，SentiWordNet における単語 “great” の品詞，語義，および極性スコアを示している．“great” は名詞と形容詞の 2 つの品詞を持つ．名詞には 1 つの語義があり，形容詞には 3 つの語義がある．語義の出現頻度の順位は，n.01 → s.01 → s.02 → s.03 である．式 (3.1) に従い，名詞 “great” の極性スコアは式 (3.2)，形容詞 “great” の極性スコアは式 (3.3) のように計算される．

$$SS(great.n) = \frac{1}{1} \cdot |0 - 0| = 0 \tag{3.2}$$

$$SS(great.a) = \frac{1}{1} \cdot |0 - 0| + \frac{1}{2} \cdot |0.750 - 0| + \frac{1}{3} \cdot |0.250 - 0.125| = 0.417 \tag{3.3}$$

表 3.2: SentiWordNet における単語 “great” の情報

単語	品詞	語義	肯定的スコア	否定的スコア
great	名詞	n.01	0.000	0.000
	形容詞	s.01	0.000	0.000
	形容詞	s.02	0.750	0.000
	形容詞	s.03	0.250	0.125

表 3.3 は，ドメインが service という属性のとき，そのドメインに固有の感情語として抽出された単語の例である．式 (3.1) によって計算される各単語の極性スコアも掲載している．ここでは閾値 T_s を 0 と設定し，それ以上の極性スコアを持つ単語を抽出している．

3.3.2 特徴語の抽出

特定のドメインだけによく使われる単語を「特徴語」と定義する．例えばドメイン (属性) が food のときには “dinner” や “dessert” などが，service のときは “staff”

表 3.3: service ドメインに固有の感情語の例

単語	品詞	極性スコア
bring	動詞	0.0173
understand	動詞	0.1752
friendly	形容詞	0.1900
happy	形容詞	0.6950
attentive	形容詞	0.3333
obviously	副詞	0.5000
couple	名詞	0.0912

や “waiter” などがそのドメインの特徴語となる．ソースドメインとターゲットドメインのそれぞれについて，そのドメインの特徴語を抽出する．

いくつかの属性について，その属性について言及されたレビュー集合があると仮定する．各属性のレビュー集合を仮想的にひとつの文書とみなし，属性 a における単語 w の TF-IDF を式 (3.4) にしたがって計算する．

$$\text{TF-IDF}(w, a) = tf_{w,a} \cdot idf_w = tf_{w,a} \cdot \log \frac{N}{df_w} \tag{3.4}$$

ここで $tf_{w,a}$ はドメイン a のコーパスにおける単語 w の出現頻度，df_w は w の文書頻度 (w が出現するドメインの数)，N はドメインの総数である．あるドメイン a のレビュー集合のみに出現し，かつ TF-IDF(a, w) の上位 T_d 件の単語をドメインの特徴語として抽出する．

ドメインの特徴語の抽出例を表 3.4 に示す．これは，4.1 節で後述するレストランのレビューのデータセットにおいて，service，food，price，ambience，anecdotes の 5 つの属性のそれぞれに対し，TF-IDF の値が高い単語を抜粋したものである．それぞれの属性と関連が深い単語が特徴語として抽出されたことが確認できる．

3.3.3 レビュー文の生成

ソースドメインのラベル付きデータにおける感情語もしくは特徴語をターゲットドメインの感情語もしくは特徴語に置き換えることにより，ターゲットドメインのラベル付き文を新たに生成する．単語の置き換えには BERT の Masked Language Model(MLM) を利用する．

Masked Language Model の利用と再事前学習

2.3 節で述べたように，BERT の事前学習は Masked Language Model タスクと Next Sentence Prediction タスクの 2 つを解くことで行われる．BERT は本来は

表 3.4: 抽出されたドメインの特徴語の例

属性名	service	food	price	ambience	anecdotes
特徴語	service	fresh	price	decor	trip
	attentive	chicken	reasonable	music	favorite
	friendly	delicious	pricey	atmosphere	friend
	server	sauce	inexpensive	romantic	recommended
	slow	food	cheap	cozy	disappointed
	manager	tasty	steal	outdoor	stumbled
	minutes	beef	dollars	cramped	neighborhood
	prompt	pizza	bargain	loud	highly
	helpful	steak	cost	seating	anniversary

文または文の組に対する分類問題を解くために使われるモデルであるが，Masked Language Model タスクを解くために使われることもある．すなわち，[MASK] というトークンを含む文を入力し，[MASK] に当てはまる単語を予測するモデルとして使われる．ここでは，[MASK] に該当する単語を予測するモデルそのものを Masked Language Model あるいは MLM と呼ぶ．ターゲットドメインのラベル付き文を生成する際には，ソースドメインのラベル付き文の感情語や特徴語を [MASK] に置き換え，BERT の MLM により [MASK] に該当するターゲットドメインの感情語や特徴語を予測させる．

公開されている事前学習済み BERT モデルは，様々な種類のテキストから構成される大規模コーパスから学習されたモデルである．これを MLM として使用する場合，ドメインに依存しない一般的な単語が予測される．一方，マスクの穴埋めによってドメイン固有のレビュー文を生成するためには，ドメインに関連した単語を予測する MLM を用いる方が望ましい．そこで，本研究では，ドメインコーパスを用いて BERT の MLM の事前学習を再度行うことで，そのドメインに特化した MLM を獲得する．具体的には，まず，あるドメインのラベルなしコーパスを用意する．人手によるアノテーションを必要としないので，大規模なコーパスを比較的容易に準備できる．次に，公開済みの事前学習済み BERT を初期のパラメタとして，このラベルなしコーパスを用いて Masked Language Model タスクによる事前学習を行う．具体的には，文中の 15%の単語をランダムに選択して [MASK] に置換し，元の単語を予測させるタスクを解くことで BERT のパラメタを更新する．以下，この手続きを BERT の MLM の再事前学習と呼ぶ．これは図 3.1 の「レビュー集合」→「再学習」→「BERT MLM」という処理の流れに該当する．

文生成の詳細

ターゲットドメインのラベル付き文を自動生成する手法の大まかな処理の流れは以下の通りである．

1. ラベル付きのソースドメインの文について，その中にソースドメインの感情語と特徴語が出現していれば，それを [MASK] に置換する．
2. [MASK] に対して左から順に以下の処理を実行する．
   (a) MLM によって [MASK] に埋めるべき単語を予測する．BERT の MLM による予測確率が高い上位 T_k 件の単語集合を得る．[MASK] をその単語に置換し，T_k 個の新しい文を得る．
   (b) 文の数が組み合わせ的に増大することを避けるため，2 個目以降の [MASK] を置換するときは，$T_k \times T_k$ 個の文のうちスコアが高い T_k 個の文を選択し，次の [MASK] の処理に移る．ここでの文のスコアは，複数箇所の [MASK] を置換した単語の MLM による予測確率の和とする．ただし，この処理は最後の [MASK] のときには行わず，最終的に最大で $T_k \times T_k$ 個の文を得る．
3. $T_k \times T_k$ 個の文のうち，[MASK] に埋められた全ての単語がターゲットドメインの感情語・特徴語のリストに含まれている文を選択し，ターゲットドメインの文とする．上記の条件を満たさない場合，その文は除外する．
4. 生成された生成文に対し，元のソースドメインのレビュー文の極性ラベルと同じラベルを付与する．

次に，レビュー文生成の詳細を Algorithm 1 に示す．入力はソースドメインのラベル付きレビュー文の集合 R^s，ソースおよびターゲットドメインの感情語の集合 P^s および P^t，同じく特徴語の集合 K^s および K^t である．R^s は (s_k, l_k) の集合であり，s_k は文，l_k はその極性ラベルを表す．出力はターゲットドメインのラベル付きレビュー文の集合 R^t である．

Algorithm 1 Cross Aspect Review Generation

Input: $R^s = \{(s_k, l_k)\},\ P^s,\ P^t,\ K^s,\ K^t$

Output: R^t

$R^t \leftarrow \emptyset$
for $(s_k, l_k) \in R^s$ **do**
  $ms \leftarrow$ replace all words included in P^s and K^s with [MASK] in s_k
  $R_{posit} \leftarrow GetMaskPosition(ms, [MASK])$
  $R_{new} \leftarrow \{ms\}$
  for $i = 1$ to n **do**
    $R'_{new} \leftarrow \emptyset$
    for $s_j \in R_{new}$ **do**
      $PW \leftarrow Unmask(s_j, [\text{MASK}]_i)$
      add all sentences where $[\text{MASK}]_i$ is replaced with $w \in PW$ to R'_{new}
    end for
    if $i \neq n$ **then**
      $R_{new} \leftarrow Select(R'_{new}, T_k)$
    else
      $R_{new} \leftarrow R'_{new}$
    end if
  end for
  $R_{select} \leftarrow \emptyset$
  for $r_{new} \in R_{new}$ **do**
    validity-check-flag $\leftarrow True$
    for $r_{posit} \in R_{posit}$ **do**
      $r_w \leftarrow GetPositionWord(r_{new}, r_{posit})$
      if $r_w \notin K^s \cup K^t$ **then**
        validity-check-flag $\leftarrow False$
      end if
    end for
    if *validity-check-flag* $= True$ **then**
      $R_{select} \leftarrow R_{select} \cup \{r_{new}\}$
    end if
  end for
  for $r_{new} \in R_{select}$ **do**
    $R^t \leftarrow R^t \cup \{\ (r_{new}, l_k)\ \}$
  end for
end for
Function $Unmask(s, [MASK], T_k)$
  $PW \leftarrow PredictByMLM(s, [MASK])$
  return $Select(PW, T_k)$

R^s のそれぞれの文 s_k について，それに含まれる感情語や特徴語を [MASK] に置き換えた文 ms を生成する (3 行目)．関数 $GetMaskPosition$ は，置き換えた文の [MASK] の文中における位置のリストを返す (4 行目)．ms に含まれるそれぞれの [MASK] について 6〜17 行目の処理を繰り返す (n は [MASK] の数)．関数 $Unmask$ は，文 s 内の [MASK] に埋めるべき単語を MLM によって予測し，MLM による予測確率が高い上位 T_k 件の単語集合を返す関数である．[MASK] を関数 $Unmask$ で得られた単語に置き換え，T_k 件の文を生成する (10 行目)．この操作を全ての [MASK] について繰り返すが，生成される文の数が組み合わせ的に増大するため，ひとつの [MASK] の処理が終わるたびにスコアが上位 T_k 件の文を選別する (13 行目)．ここでの文のスコアは複数箇所の [MASK] を置換した単語の MLM による予測確率の和である．ただし，最後の [MASK] を置換したときはこの操作を行わず，$T_k \times T_k$ 個の文を得る (15 行目)．次に，置き換えられた単語がターゲットドメインの感情語または特徴語であるかをチェックする．関数 $GetPositionWord$ によって，生成された文 r_{new} のマスクの位置 r_{posit} に出現する単語 r_w を得る (22 行目)．r_w がターゲットドメインの感情語の集合 K^s，特徴語の集合 K^t に含まれるかをチェックし，含まれないときは validity-check-flag を偽に設定する (23〜25 行目)．[MASK] に埋められた全ての単語がこのチェックにパスしたとき，そのレビュー文を R_{select} に追加する (28 行目)．そうでなければ，そのレビュー文は削除 (無視) される．以上の操作で，[MASK] をターゲットドメインの感情語または特徴語に置き換えた文集合 R_{select} を獲得する．最後に，自動生成した文に元の文の極性ラベル l_k を付与し，これを R_t に加える (32 行目)．

上記の文生成手法は CDRG[20] を元にしているが，CDRG では属性語を置換するのに対し，本研究では特徴語を置換する点が異なる．また，CDRG では 1 つの文からターゲットドメインの文を 1 つ生成するのに対し，本研究では最大で $T_k \times T_k$ 個の文を生成する点が異なる．パラメタ T_k によって自動生成する文の数を制御できる．

文生成の例

CARG によってターゲット文のラベル付き文を生成する例を表 3.5 に示す．Step の番号は 18 ページに記載した箇条書きの番号に対応する．表中の文について，下線は MLM によって予測され [MASK] に埋められた単語，太字は埋められた単語がターゲットドメインの感情語・特徴語であることを表す．

まず，ソースドメインが属性 service のとき，そのドメインのレビュー文 “The service was attentive and her suggestions of menu items was right on the mark” が与えられたものとする．Step1 では，この文中に出現しているソースドメインの感情語と特徴語を [MASK] に置換する．Step2(a) では，左から 1 番目の [MASK] に対し，MLM によって上位 T_k 件の単語を予測し、その単語を埋めて T_k 件の文を生成する．ここでは，“staff”，“service”，“food” などの単語が埋められている．Step2(b)

は，左から 2 番目の [MASK] に対して上位 T_k 件の単語を埋め，合計 $T_k \times T_k$ 個の文を生成する．“friendly”，“good” などの単語が埋められている．そして，$T_k \times T_k$ 個の文の中からスコアが大きい上位 T_k 件の文を選択する．Step3 は，Step2 の操作を繰り返して全ての [MASK] が埋められた状態を表す．この時点では $T_k \times T_k$ 個の文を残している．これらの文のうち，置換された単語の全てがターゲットドメインの感情語や特徴語である場合，その文をターゲットドメインの文とする．そうでなければ，その文は削除する．表 3.5 の例では，✓が付与された top6 の生成文 “The food was good and her choice of menu items was right on the menu” は上記の条件を満たす一方，✗が付与されている top1〜top5 の文は条件を満たさない．最後に，Step4 では，Step3 で得られた生成文にソースドメインのレビュー文の極性ラベルを付与する．

表 3.5: CARG による文生成の例

	service → food
ソースドメイン レビュー文	The service was attentive and her suggestions of menu items was right on the mark
Step1	The [MASK] was [MASK] and her [MASK] of menu items was [MASK] on the [MASK]
Step2(a)	top1:The staff was [MASK] and her [MASK] of menu items was [MASK] on the [MASK] top2:The service was [MASK] and her [MASK] of menu items was [MASK] on the [MASK] top3:he food was [MASK] and her [MASK] of menu items was [MASK] on the [MASK] ····· topk: ·····
Step2(b)	top1:The staff was friendly and her [MASK] of menu items was [MASK] on the [MASK] top2:The service was good and her [MASK] of menu items was [MASK] on the [MASK] top3:The food was good and her [MASK] of menu items was [MASK] on the [MASK] ····· topk: ·····
Step3	top1:The staff was friendly and her **choice** of menu items was **great** on the **menu**(✗) top2:The service was **good** and her **choice** of menu items was **great** on the **menu**(✗) top3:The **food** was **good** and her **choice** of menu items was **great** on the inside(✗) top4:The staff was friendly and her **choice** of menu items was **great** on the inside(✗) top5:The service was **good** and her **choice** of menu items was **great** on the inside(✗) top6:The **food** was **good** and her **choice** of menu items was **right** on the **menu**(✓) ····· topk: ·····
Step4	top6:The **food** was **good** and her **choice** of menu items was **right** on the **menu**(✓)(positive) ····· topn: ····· (polarity)

3.3.4 生成文のフィルタリング

3.3.3 項の手続きで生成された文には不自然なものも多く含まれる．そこで，生成された文の自然さを擬似対数尤度スコア PLL(pseudo-log-likelihood score)[15] によって測る．PLL の定義を式 (3.5) に再掲する．PLL は，生成された文中の各単語 w_t を BERT の MLM で予測した確率の対数尤度の和である．自動生成されたレ

ビュー文のうち，$\log P_{(MLM)}(W)$ が閾値 T_f より小さい文を削除する．この手続きを生成文のフィルタリングと呼ぶ.

$$\log P_{MLM}(W) = \sum_{t=1}^{|W|} \log P_{MLM}(w_t|W_{\backslash t}) \tag{3.5}$$

図 3.3.4 は “Food is delicious” という文の PLL スコアを計算する手順を示している．文中の各単語をマスクした文を入力とし，BERT の MLM によって各単語を予測したときの条件付き確率を得る．それぞれの確率の対数を取り，その合計を算出する．これが最終の PLL スコアとなる．

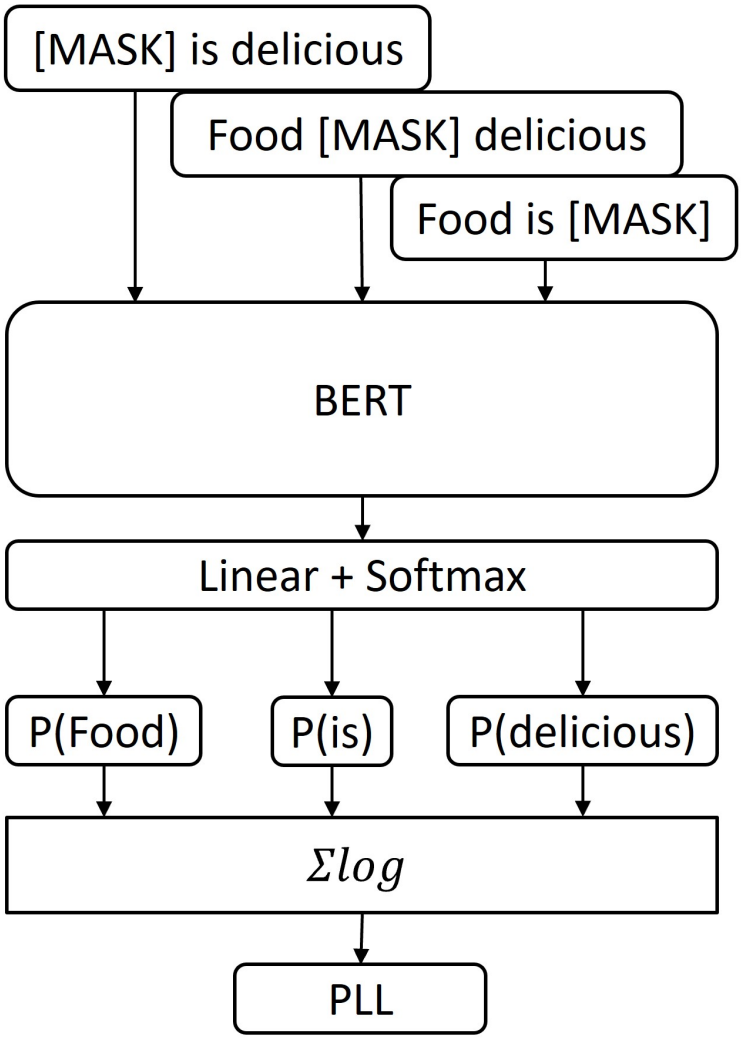

図 3.2: 文 “Food is delicious” に対する PLL の計算

3.4 極性判定モデルの学習

3.4.1 BERT による極性判定ための分類器の学習

3.2 項と 3.3 項で生成したラベル付きレビューを用いて BERT モデルの分類器をファインチューニングし，ターゲットドメインの極性判定モデルを得る．これは図 3.1 の BERT(T) に該当する．これを用いてターゲットドメインのレビューに対する極性を判定する．

BERT による極性判定のための分類器のアーキテクチャを図 3.3 に示す．最初の層では，入力レビュー文に対して単語埋め込みベクトルを得る．Transformer と呼ばれるユニットを重ねたいくつかの中間層を経て，文頭の [CLS] トークンの埋め込みを得る．最後に，最終層として，シグモイド関数を用いた全結合層をつなげ，出力を得る．ここでの出力は，極性ラベルと同じ数のノードであり，それぞれの極性ラベルの予測確率が出力され，最大の予測確率を持つ極性ラベルが出力ラベルとして選択される．BERT のファインチューニングでは，最終の出力ベクトルに対し，Softmax による Cross Entropy 関数によって損失を求める．損失を最小化するように BERT モデルのパラメタを更新する．

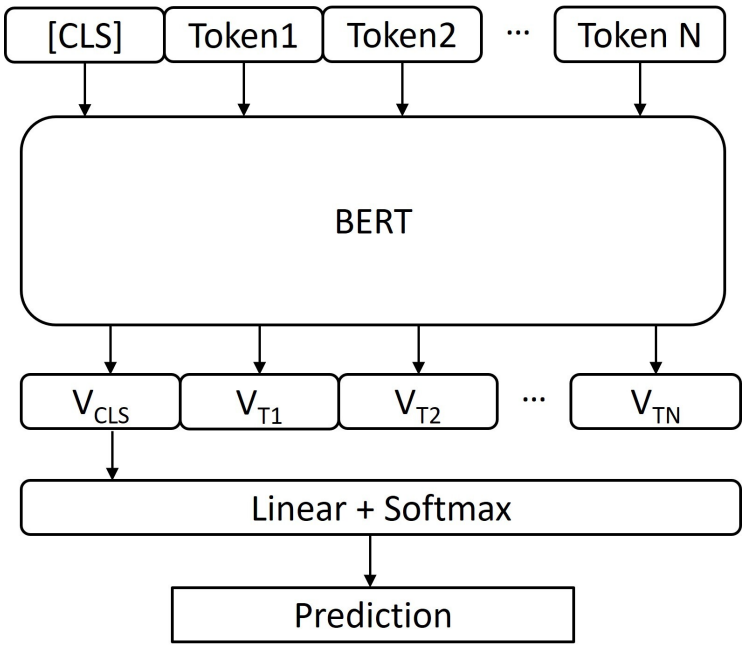

図 3.3: BERT による極性判定の分類器

3.4.2 Focal Loss の適用

4.1 節で後述するように，本論文の実験で用いるデータセットには，極性ラベルの分布に偏りが見られる．この問題に対処するため，本研究では Focal Loss[10] を損失関数として使用する．二値分類のときの Focal Loss の定義を式 (3.6) に示す．

$$FL(p_t) = -(1 - p_t)^{\gamma} \cdot \log(p_t) \tag{3.6}$$

ここで，p_t はあるクラスに対する予測確率，γ はハイパーパラメタである．

データセットの中で大きな割合を占めるデータのクラスは，その予測確率 p_t は大きくなる傾向がある．通常の cross entropy の式に $(1 - p_t)^{\gamma}$ を加えることにより，多数の極性クラスの損失が小さく見積られる．逆に少数の極性クラスについては損失が大きく見積もられ，その分類エラーが大きく評価される．

第4章　評価

本章では，提案手法の評価実験について述べる．まず，4.1 節で評価実験に用いるデータセットを紹介し，4.2 節で実験の設定について説明する．4.3 節では，インドメインとクロスドメインの評価実験の結果を報告し，提案手法の有効性について考察する．

4.1　データセット

評価実験には 2 つのデータセットを使用する．4.1.1 項と 4.1.2 項でそれぞれのデータセットの詳細を述べる．また，4.1.3 項では BERT の MLM を再学習するためのコーパスについて紹介する．

4.1.1　レストラン・データセット

SemEval-2014 Task 4 Aspect Based Sentiment Analysis のレストランのデータセット [13] を評価実験に用いる．このデータセットは，レストランに関するレビュー文に対し，属性のカテゴリとそれに対する極性クラスが付与されている．属性は “service”, “food”, “price”, “ambience”, “anecdotes” の 5 つであり，本実験ではこれをドメインとする．極性クラスは “positive”, “negative”, “conflict”, “neutral” の 4 つである．データセットの詳細を表 4.1 に示す．全ての属性について，positive のレビューが一番多い．一方，conflict や neutral は，anecdotes を除いて，データ数が少ない．全体的に，positive のレビューが特に多く，極性クラスの分布に不均衡が見られる．

表 4.1: レストランデータセットの詳細

	service	food	price	ambience	anecdotes
positive	127	542	49	127	472
negative	55	138	46	55	167
conflict	33	54	7	33	24
neutral	18	62	6	18	326
total	336	796	108	233	989

4.1.2 ラップトップ・データセット

Laptop ACOS(Aspect-Category-Opinion-Sentiment) のデータセット [6] を評価実験に用いる．このデータセットは，Amazon に投稿されたラップトップパソコンに関するレビューに対し，評価対象の属性と極性が付与されている．極性カテゴリは “postive”, “negative”, “neutral” の 3 つである．一方，付与されている属性は，“keyboard”，“monitor”，“price” など，パソコンの属性を表す具体的な単語である．本実験では，データセットに付与されている属性を “general”, “design”, “performance”, “quality” の 4 つに人手で分類し，これを属性のドメインとみなした．データセットの詳細を表 4.2 に示す．このデータも neutral のレビューが極端に少ないという極性クラスの偏りが見られる．

表 4.2: ラップトップ・データセットの詳細

	general	quality	performance	design
positive	194	285	156	85
negative	468	114	152	139
neutral	22	15	11	31
total	684	414	319	255

4.1.3 BERT の MLM の再学習用コーパス

レストラン・データセットについては，CARG を用いる際，BERT の MLM を再学習するために 3 万件の Yelp のレストランレビュー [1] を使用した．このコーパスには，全てのレビューで極性のラベルは付与されていない．公開されている BERT の事前学習モデルとして bert-base-cased[2] を用い，これを初期のパラメタとして，Yelp のレビュー集合を用いて Masked Language Model タスクによる事前学習を行った．

ラップトップ・データセットについては，MLM の再学習はせずに，事前学習済みの bert-base-cased を MLM としてそのまま用いた．

4.2 実験設定

4.2.1 実験手順

レストラン・データセットとラップトップ・データセットにおいて，一方の属性をソースドメイン，もう一方の属性をターゲットドメインとし，5 つまたは 4 つの属性の全ての組み合わせについて，極性判定の領域適応の実験を行う．ソース

ドメインとターゲットドメインが異なるため，この実験設定を「クロスドメイン」と呼ぶ．

比較のため，ソースドメインとターゲットドメインが同じ場合の極性判定の正解率を測る．この実験設定を「インドメイン」と呼ぶ．インドメインの設定では，訓練データとテストデータとでドメインが異なるために極性判定の正解率が低下することはない．つまり，インドメインでの極性判定の正解率は領域適応の上限とみなせる．

インドメインの実験では 5 分割交差検証を行う．図 4.1 は 5 分割交差検証の手続きを示している．まず，あるドメイン (属性) のデータをランダムに 5 つに分割する．1 回目の試行では 1 番目のデータ集合をテストデータ，残りのデータ集合を訓練データとする．2 回目の試行では 2 番目のデータ集合をテストデータ，残りのデータ集合を訓練データとする同様に，3 回目から 5 回目も同様の手続きを行い，テストデータと訓練データを変えながら極性判定モデルの学習と評価を行う．5 回の試行で得られた評価指標のマイクロ平均で分類モデルの性能を測る．

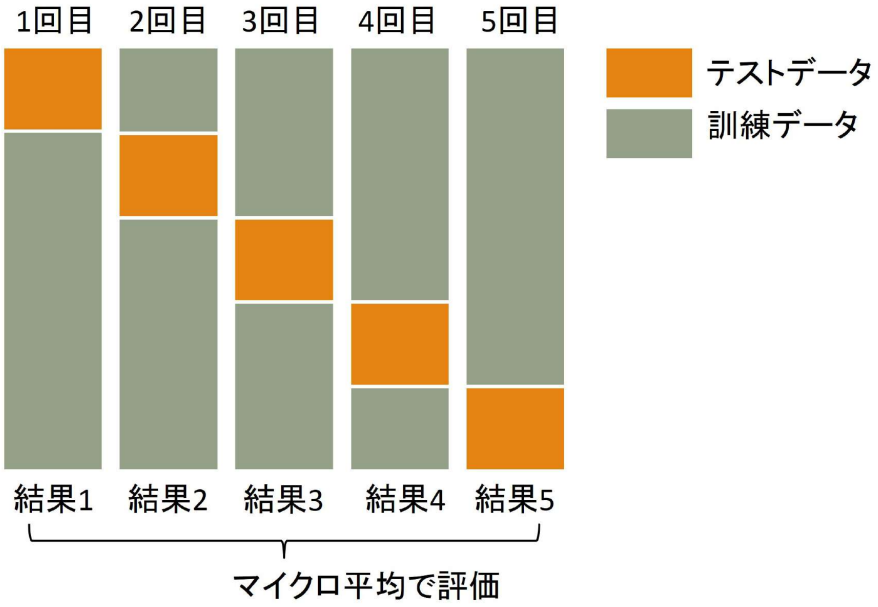

図 4.1: 5 分割交差検証

4.2.2 比較手法

本実験では，本研究の提案手法を，2 つのベースライン手法ならびに先行研究の手法と比較する．また，提案手法を構成するモジュールの有効性を比較するため，提案手法としては 3 つのモデルを比較する．以下に詳細を述べる．

ベースライン 1 (BL1) ソースドメインのラベル付きデータを用いて BERT をファインチューニングする手法．領域適用をしない手法である．

ベースライン 2 (BL2) 3.2 節で述べた自動ラベル付けによって得られたターゲットドメインの擬似ラベル付きデータを用いて BERT をファインチューニングする手法.

CDRG 先行研究の Yu ら [20] の手法に準ずる手法によって生成されたレビュー文を訓練データとして BERT をファインチューニングする手法．具体的には，ソースドメインの 1 つのレビュー文からターゲットドメインのレビュー文を 1 つだけ生成する手法.

CARG1 BL2 の擬似ラベル文と CARG によって生成されたレビュー文を訓練データとして BERT をファインチューニングする手法.

CARG2 CARG1 に加え，3.3.4 項で述べた生成文のフィリタリングを行ったうえで BERT をファインチューニングする手法.

CARG3 CARG2 に加え，3.4 項で述べた Focal Loss を適用して BERT をファインチューニングする手法.

一方，インドメインの実験では，訓練データを用いて BERT をファインチューニングすることで極性判定の分類器を得る．BERT の事前学習モデルとして bert-base-cased を用いる．

4.2.3 評価指標

本項では，領域適応モデルの評価指標について述べる．

ソースドメインとターゲットドメインが異なるクロスドメインの実験では，正解率 (Accuracy) を評価指標として使用する．正解率とは，分類タスクの評価に用いられる指標のひとつで，予測された極性クラスと正解の極性クラスの一致率と定義される．多クラスの分類問題における正解率の定義を式 (4.1) に示す.

$$Accuracy = \frac{\sum_{i=1}^{N} c_{ii}}{\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}}, N = 3 \text{ or } 4 \tag{4.1}$$

c_{ij} は，混同行列において，i 番目の分類クラスが正解であるデータに対して分類モデルが j 番目の分類クラスを予測した回数を表す．分子の c_{ii} は，分類モデルの予測が正解であったデータの数を表す．本実験では，分類クラスの数 N は，レストラン・データセットの場合は 4，ラップトップ・データセットの場合は 3 となる．インドメインの実験でも正解率を評価指標とする．ただし，5 分割交差検証を行うため，正確には 5 回の試行の正解率のマイクロ平均を評価指標とする．

4.2.4 パラメタの設定

3 章で述べたように，提案手法には様々なパラメタが存在する．本実験では，パラメタを表 4.3 のように設定する．これらのパラメタは直観により決定している．ただし，T_p と T_s についてはパラメタを変えて実験を行い，パラメタが極性判定の正解率に与える影響を調査する．

表 4.3: パラメタの設定

パラメタ	値
自動ラベル付けにおける予測確率の閾値 T_p(レストラン)	$\{0.7, 0.8\}$
自動ラベル付けにおける予測確率の閾値 T_p(ラップトップ)	$\{0.5, 0.7, 0.8\}$
感情語抽出の極性スコアの閾値 T_s(レストラン)	$\{0, 0.3\}$
感情語抽出の極性スコアの閾値 T_s(ラップトップ)	0
特徴語抽出の際の件数 T_d	100
CARG における文生成数 T_k	100
生成文のフィルタリングの PLL の閾値 T_f	-30
Focal Loss のハイパーパラメタ γ	2

4.3 実験結果と考察

本節では評価実験の結果を報告し，それに対する考察を述べる．4.3.1 項では，インドメインの実験結果を報告する．4.3.2 項では，クロスドメインの実験結果を報告する．4.3.3 項では，パラメタが極性判定に与える影響の調査について述べる．最後に，4.3.4 項では，CARG による生成文について考察し，エラー分析の結果を述べる．

4.3.1 インドメインの実験結果

レストラン・データセットを用いたときのインドメインの実験結果を表 4.4 に，ラップトップ・データセットを用いたときのインドメインの実験結果を表 4.5 に示す．これらの表は，5 分割交差検証における 5 回の試行のそれぞれの正解率，ならびにそのマイクロ平均を示している．

レストラン・データセットの結果を見ると，属性が price や ambience のとき，5 回の試行によって正解率にばらつきがあることがわかった．price の 2 回目試行の結果が 0.455，ambience の 4 回目と 5 回目試行の結果がそれぞれ 0.565 や 0.543 であり，他の試行と比べて正解率がかなり低かった．これらの結果がマイクロ平均

を下げることになった．一方，属性が service，food，anecdotes のときは，5 回の試行の正解率に大きな差は見られなかった．

ラップトップ・データセットの結果を見ると，属性が general のとき，各試行の正解率はほぼ同じであった．一方，他の属性では，正解率が低くなる試行がいくつかあった．全体的には，ラップトップ・データセットの個々の試行の結果はレストラン・データセットよりも安定しており，マイクロ平均も比較的高かった．

表 4.4: レストランデータセットのインドメインの実験結果

	service	food	price	ambience	ancdotes
1 回目	0.765	0.719	0.727	0.723	0.732
2 回目	0.806	0.805	0.455	0.745	0.768
3 回目	0.821	0.805	0.636	0.809	0.742
4 回目	0.851	0.755	0.619	0.565	0.758
5 回目	0.731	0.755	0.619	0.543	0.797
マイクロ平均	0.795	0.768	0.611	0.678	0.759

表 4.5: ラップトップデータセットのインドメインの実験結果

	general	quality	performance	design
1 回目	0.854	0.843	0.828	0.843
2 回目	0.854	0.855	0.781	0.725
3 回目	0.876	0.940	0.875	0.804
4 回目	0.854	0.880	0.938	0.725
5 回目	0.912	0.817	0.810	0.686
マイクロ平均	0.870	0.867	0.846	0.757

4.3.2 クロスドメインの実験結果

レストラン・データセットを用いたときのクロスドメインの実験結果を表 4.6 に示す．IN-D はインドメインの実験結果であり，表 4.4 に示した正解率のマイクロ平均の再掲である．太字は，インドメインの結果を除いて，それぞれのドメイン適応の組において正解率が最も高いことを表す．

ベースライン手法 (BL1,BL2) と提案手法 (CARG1~3) を比較すると，20 のドメインの組のうち 17 組について，提案手法の方が優れている．このことから，ターゲットドメインのラベル付きデータを自動生成する CARG の有効性が確認できる．CARG1 と CARG2 を比較すると，全 20 組のうち 12 組について CARG2 の正解率が高いことから，自動生成された文のうち生成確率の低い文をフィルタリングする

手法はある程度有効であると言える．CARG2 と CARG3 を比較すると，全 20 組のうち 15 組については CARG3 の正解率が同じもしくは高い．今回の実験では Focal Loss の導入が効果的であった．全ての組の平均正解率で比較すると，CARG3 の平均正解率は 0.658 であり，BL1，BL2 と比べてそれぞれ 0.025，0.013 ポイント正解率を向上させ，比較した手法の中で最も高い平均正解率を示している．CARG1 や CARG2 の平均正解率も BL1，BL2 より高いことから，CARG の有効性が確認できた．

ドメインの組に注目して実験結果を考察する．ソースドメインが service，price，ambience の場合，全ての組において提案手法 (CARG1，CARG2，CARG3 のいずれか) の正解率は BL1，BL2 より高い．しかし，ソースドメインが food のとき，ターゲットドメインが anecdotes の場合では，提案手法よりも BL1 と BL2 の方が正解率が高い．ソースドメインが anecdotes のとき，ターゲットドメインが service，ambience の場合でも提案手法は BL1 と BL2 を下回る．この結果から，anecdotes は，ソースドメインとターゲットドメインに関わらず，比較的に適応しにくい属性であることがわかる．

CARG1，CARG2，CARG3 の平均正解率は，先行研究の手法 CDRG に比べ，それぞれ 0.002，0.007，0.010 ポイント高かった．この原因として，CARG は CDRG に比べてより多くのターゲットドメインのレビュー文を生成することが考えられる．

CARG と BL1，BL2 に有意差あるかどうかを確認するために，マクネマー検定を行った．表 4.6 において，*は BL1 と比べて，+は BL2 と比べて，$p<0.05$ で有意差があることを表す．その結果，全 20 組の過半数の CARG モデルはベースラインと有意差があることが確認できた．

表 4.6: レストラン・データセットにおける領域適応の実験結果

(source)	S				F				P				Am				An				平均
(target)	F	P	Am	An	S	P	Am	An	S	F	Am	An	S	F	P	An	S	F	P	Am	
BL1	.776	.667	.687	.533	.765	.574	.674	**.607**	.679	.697	.472	.421	.771	.758	.620	.526	**.577**	.707	.537	**.614**	.633
BL2	.795	.676	.687	.537	.771	.620	.687	**.607**	.711	.735	.459	.446	.762	.794	.639	.540	.574	.730	.528	.597	.645
CDRG	.785	.676	**.691**	**.547**	.798	.620	.674	.602	.711	.747	.433	.453	**.804**	.784	.630	.533	.586	.730	.537	.597	.648
CARG1	.798$^{*}_{+}$	.694	.682	.540$^{*}_{+}$	.783$^{*}_{+}$	.630*	**.704***	.585	.708	.731*	.472	**.461**	.786	.789$^{*}_{+}$	.630	.531$^{*}_{+}$	.571	**.756***	.546	.597	.650
CARG2	.791$^{*}_{+}$	**.731**	**.691**	.541$^{*}_{+}$	.802*	**.657***	.691*	.590	.705	**.754**$^{*}_{+}$	.464	.433	.798$^{*}_{+}$	.794$^{*}_{+}$	**.639**	**.558**$^{*}_{+}$	.563	.727	**.574**$^{*}_{+}$	.597	.655
CARG3	**.799**$^{*}_{+}$	.713	**.691**	.542$^{*}_{+}$	**.815***	**.657***	.687*	.602	**.717**	.747$^{*}_{+}$	**.476**	.451	.795$^{*}_{+}$	**.795**$^{*}_{+}$	**.639**	.556*	.574	.731*	**.574**$^{*}_{+}$	.605	**.658**
IN-D	.768	.611	.678	.759	.795	.611	.678	.759	.795	.768	.611	.678	.795	.768	.611	.759	.795	.768	.611	.678	

S: service, F: food, P: price, Am: ambience, An: anecdotes.
∗: vs. BL1($p<0.05$)，+: vs. BL2($p<0.05$)

ラップトップ・データセットの実験結果を表 4.7 に示す．レストラン・データセットの結果と同様に，ベースラインよりも提案手法の方が優れている．ただし，統計的に有意差があるドメインの組はレストラン・データセットに比べて少ない．レストランの実験では Yelp のレビューを用いて MLM を再学習したのに対し，ラップトップでは事前学習された MLM をそのまま用いたことが原因のひとつとして

考えられる．また，CARG1 よりも CARG2 の方が正解率が高い傾向にあることが確認できるが，CARG3 は CARG2 と比べて同等もしくはやや劣る．2 つの異なるデータセットでおおよそ同様の結果が得られたことから，提案手法がどのようなジャンルのレビューにも適用できるという意味での汎用性を有することが示唆される．

最後に，クロスドメインの実験結果とインドメインの実験結果を比較する．通常，クロスドメインの極性判定の正解率はインドメインよりも劣るが，レストラン・データセットでは，多くのドメインの組について提案手法の正解率がインドメインの正解率を上回っている．特に，ターゲットドメインが price のとき，ソースドメインが anecdotes の場合を除く全ての属性について，CARG のいずれかはインドメインの正解率を上回る．表 4.1 に示したように，price のデータ数は他の属性に比べてかなり少ない．5 分割交差検証をしたときの訓練データの量が十分ではなく，そのため十分に高い正解率が得られなかったことが原因として考えられる．その他のドメインの組についてもクロスドメインの正解率がインドメインを上回ることがあるが，明確な傾向は見出せなかった，クロスドメインの結果がインドメインの結果を上回る理由については，今後精査が必要である．

ラップトップ・データセットについては，全体的にクロスドメインの結果はインドメインの結果よりも劣る．ただし，ターゲットドメインが design のときは，提案手法の正解率はインドメインの正解率 0.757 を上回る．表 4.2 によると，design のデータ量は 4 つの属性うち一番少ないため，レストラン・データセットにおける price 属性と同様に，訓練データ量が十分に多くないことが原因として考えられる．

表 4.7: ラップトップ・データセットにおける領域適応の実験結果

(source)	G			Q			P			D			平均
(target)	Q	P	D	G	P	D	G	Q	D	G	Q	P	
BL1	.807	.784	.765	.741	.837	.769	**.789**	.850	.773	.737	.821	.809	.790
BL2	.804	**.809**	.757	.731	.850	.765	.773	.862	.788	.756	.831	.803	.794
CARG1	.807	.803	**.780***	.744+	.850*	.769	.768	**.865***	.788	.770*	**.833***	.806	.799
CARG2	.812	.800	**.780***	**.746**+	**.853***	.765	.779	.862*	**.796***	**.772**$^*_+$	.829*	**.815**	**.801**
CARG3	**.816**	.800	**.780***	.744+	.846	**.773**	.779	.855	.788	.768*	.829*	**.815**	.800
IN-D	.867	.846	.757	.870	.846	.757	.870	.867	.757	.870	.867	.846	

G: general, Q: quality, P: performance, D: design

$*$: vs. BL1($p<0.05$)，+: vs. BL2($p<0.05$)

4.3.3 パラメタによる影響の調査

本項では，提案手法における 2 つのパラメタについて，それが極性判定の正解率に与える影響を調査する．

閾値 T_p の影響の調査

まず，閾値 T_p の影響について調査する．3.2 節で述べたように，ソースドメインのデータでファインチューニングした BERT を用いてターゲットドメインの文のラベル付けを行う際，極性クラスの予測確率が T_p 以上のときのみラベルを付与する．T_p を大きく設定するとターゲットドメインのラベル付き文の量は増えるが，誤って極性ラベルが付与されたデータが増える可能性がある．

レストラン・データセットを用いた実験では，T_p を 0.7 または 0.8 に設定した．T_p は自動ラベル付けによる訓練データの自動構築に関するパラメタであるので，自動ラベル付けによる訓練データのみを用いて BERT をファインチューイングするベースライン 2(BL2) の正解率を比較した．異なる閾値 T_p による BL2 の極性判定の正解率を表 4.8 に，ラベルを付与したレビュー文の数を表 4.9 に示す．T_p を 0.8 に設定した場合，T_p が 0.7 のときと比べて，ラベルを付与された文の数は当然減少するが，その差はわずかである．一方，正解率を比較すると，全 20 組のうち，T_p が 0.8 のときの結果が優れているのは 14 組であり，T_p が 0.8 のときの平均正解率も 0.7 の平均正解率より高い．したがって，T_p は 0.8 に設定した方が適切である．このため，4.3.2 項で述べたレストラン・データセットを用いた実験では，閾値 T_p を 0.8 に設定した．

表 4.8: 閾値 T_p を変えたときの極性判定の正解率 (レストラン・データセット)

(source)	S				F				P				Am				An				平均
(target)	F	P	Am	An	S	P	Am	An	S	F	Am	An	S	F	P	An	S	F	P	Am	
BL2(T_p=0.7)	.786	.657	**.700**	.532	.762	.611	.665	.601	.702	.667	.455	.437	**.807**	.780	.630	**.547**	**.577**	.720	**.537**	**.614**	.639
BL2(T_p=0.8)	**.795**	**.676**	.687	**.537**	**.771**	**.620**	**687**	**.607**	**.711**	**.735**	**.459**	**.446**	.762	**.794**	**.639**	.540	.574	**.730**	.528	.597	**.645**

S: service, F: food, P: price, Am: ambience, An: anecdotes.

表 4.9: 閾値 T_p を変えたときのラベル付き文の数 (レストラン・データセット)

(source)	S				F				P				Am				An			
(target)	F	P	Am	An	S	P	Am	An	S	F	Am	An	S	F	P	An	S	F	P	Am
BL2(T_p=0.7)	738	96	214	866	308	93	213	885	265	617	169	701	248	654	79	674	291	704	97	203
BL2(T_p=0.8)	690	91	202	818	284	89	204	818	216	534	123	549	210	583	60	552	276	669	92	189

S: service, F: food, P: price, Am: ambience, An: anecdotes.

ラップトップ・データセットを用いた実験では，T_p を 0.5，0.7，0.8 のいずれかに設定し，ベースライン 2 の正解率を比較した．異なる閾値 T_p による BL2 の極性判定の正解率を表 4.10 に，ラベルを付与したレビュー文の数を表 4.11 に示す．T_p を高く設定するほど生成される文の数は減少するが，ソースドメインが general もしくは performance の場合，T_p を 0.7 もしくは 0.8 に設定しても，T_p=0.5 のときと比べて，文の数に大きな差はない．一方，ソースドメインが design のときに T_p を 0.7 または 0.8，quality のときに T_p を 0.8 に設定した場合， T_p が 0.5 のときと比較

すると，ラベルを付与された文の数が少なく，正解率が低下する要因となる．実際，表 4.10 の結果を見ると，上記の状況で正解率の大きな低下が確認できる．また，正解率の比較では，全 12 組のうち，T_p が 0.5 のときの結果が最も優れているのは 9 組であり，T_p が 0.5 のときの平均正解率も 0.7 または 0.8 のときの平均正解率より高い．したがって，T_p は 0.5 に設定した方が適切である．このため，4.3.2 項で述べたラップトップ・ドメインを用いた実験では，閾値 T_p を 0.5 に設定している．

表 4.10: 閾値 T_p を変えたときの極性判定の正解率 (ラップトップ・データセット)

(source)	G			Q			P			D			
(target)	Q	P	D	G	P	D	G	Q	D	G	Q	P	平均
BL2(T_p=0.5)	.804	**.809**	.757	**.731**	**.850**	**.765**	.773	**.862**	**.788**	**.756**	**.831**	**.803**	**.794**
BL2(T_p=0.7)	**.814**	.781	**.761**	.725	.784	.690	**.794**	.853	.761	.684	.275	.476	.700
BL2(T_p=0.8)	.775	.755	.686	.396	.489	.416	.791	.845	.769	.684	.275	.476	.613

G: General, Q: Quality, P: Performance, D: Design.

表 4.11: 閾値 T_p を変えたときのラベル付き文の数 (ラップトップ・データセット)

(source)	G			Q			P			D		
(target)	Q	P	D	G	P	D	G	Q	D	G	Q	P
BL2(T_p=0.5)	406	317	254	660	317	249	680	408	252	566	371	277
BL2(T_p=0.7)	355	297	221	501	245	188	597	377	222	252	83	110
BL2(T_p=0.8)	265	232	182	289	153	103	530	351	208	210	74	93

G: General, Q: Quality, P: Performance, D: Design.

閾値 T_s の影響の調査

次に，CARG のパラメタ T_s が極性判定の正解率に及ぼす影響について調査する．3.3.1 項で述べたように，ドメインに固有の感情語を抽出する際，単語の極性スコアが閾値 T_s 以上の場合に感情語として抽出する．T_s を低く設定するほど，多くの感情語を抽出する．T_s を 0 または 0.3 に設定し，CARG1，CARG2，CARG3 のレストラン・データセットにおける正解率を比較した．実験結果を表 4.12 に示す．なお，ラップトップ・ドメインについては同様の調査はしていない．

まず，平均正解率を比較する．表 4.12 の結果，CARG1 と CARG3 は，T_s を 0 に設定した方が 0.3 に設定したときよりも平均正解率が高い．その差は，CARG1 で 0.04，CARG3 で 0.01 ポイントである．一方，CARG2 では，T_s が 0 のときと 0.3 のときで平均正解率は等しく，ともに 0.655 となっている．このことから，感情語抽出の閾値 T_s を 0 に設定した方が良い結果が得られることがわかった．これは，T_s を 0 にするとより多くのターゲットドメインの感情語が抽出され，これに

伴ない MLM によるマスクの穴埋めで埋められる単語の候補数も増え，結果として自動生成される文の数が増える，すなわち訓練データの量が増えるためと考えられる．

CARG1 について，個々のソース・ターゲットドメインの属性の組み合わせを見ると，全 20 組のうち 11 組について，T_s を 0 に設定した方が 0.3 と設定したときよりも正解率が高い．一方，5 組では T_s=0.3 の方が優れ，4 組では正解率が同じである．CARG1 では，T_s を 0 を設定した場合が 0.3 より優れている組が多いことがわかった．

CARG2 については，T_s=0 と T_s=0.3 の場合を比較すると，平均正解率は同じだが，T_s=0 の方が正解率が高いドメインの組は 11 組，T_s=0.3 の方が正解率が高いのは 8 組，正解率が同じなのは 1 組であった．平均正解率は同じでも，T_s=0 の方が良いケースが T_s=0.3 の方が良いケースよりも 3 組多い．CARG1 と同様に，閾値 T_s を 0 に設定した方がわずかに良いことがわかる．

CARG3 については，T_s=0 の方が正解率が高いのは 11 組，T_s=0.3 の方が正解率が高いのは 6 組，同じなのは 3 組であった．T_s=0 のケースが正解率が高い組の数が多い．平均正解率の比較でも T_s=0 の方が T_s=0.3 よりも 0.01 ポイント高い．CARG3 についても，T_s の値は 0.3 よりも 0 と設定する方が良いと言える．

表 4.12: 閾値 T_s を変えたときの極性判定の正解率 (レストラン・データセット)

(source)	S				F				P				Am				An				平均
(target)	F	P	Am	An	S	P	Am	An	S	F	Am	An	S	F	P	An	S	F	P	Am	
CARG1(T_s=0)	.798	.694	.682	.540	.783	.630	**.704**	.585	.708	.731	.472	**.461**	.786	.789	.630	.531	.571	**.756**	.546	.597	.650
CARG1(T_s=0.3)	.793	.685	.682	**.543**	.783	.639	.682	.598	.708	.740	**.481**	.458	.786	.783	.620	.530	.565	.735	.528	.588	.646
CARG2(T_s=0)	.791	**.731**	.691	.541	.802	**.657**	.691	.590	.705	**.754**	.464	.433	.798	.794	.639	**.558**	.563	.727	.574	.597	.655
CARG2(T_s=0.3)	.795	.704	**.695**	.538	.798	.648	.682	**.605**	**.720**	.735	.455	.433	.795	.781	**.667**	.537	.571	.751	**.611**	.571	.655
CARG3(T_s=0)	**.799**	.713	.691	.542	**.815**	**.657**	.687	.602	.717	.747	.476	.451	.795	**.795**	.639	.556	.574	.731	.574	**.605**	**.658**
CARG3(T_s=0.3)	.791	.704	.691	.541	.792	**.657**	.691	.601	**.720**	.739	.472	.442	**.801**	.774	.639	.541	**.589**	.747	.602	.597	.657

S: service, F: food, P: price, Am: ambience, An: anecdotes.

4.3.4 CARG による生成文の考察

本節では，CARG によって生成されたターゲットドメインの文の数，ならびにターゲットドメインの文として適切なものが生成されたかを考察する．

表 4.13 は，レストラン・データセットにおける全てのドメインの組について，CARG によって生成された文の数を示している．提案手法として CARG1 と CARG2 を使用し，感情語抽出における単語の極性スコアの閾値 T_s を 0 または 0.3 に設定している．なお，CARG2 と CARG3 の違いは，Focal Loss 関数の使用の有無なので，生成される文の数は同じである．また，先行研究の CDRG によって生成された文の数も示す．

不自然な文を削除するフィルタリングを行わない CARG1 の方が CARG2 よりも多くの文を生成することが確認できる．また，T_s を 0 に設定した方が 0.3 と設定したときと比べて，ドメイン固有の感情語をより多く抽出し，それを使って [MASK] を埋めて文を生成するため，生成文の数が多くなっていることも確認できる．CARG1 で T_s=0 のとき，およそ 1000 件から 8000 件といった十分な量の文が生成されているが，CARG2 で T_s= 0.3 のときは 100 件から 1000 件程度まで減少する．

ソースドメインが food や anecdotes の場合，他の属性の場合よりも多くのレビュー文が生成された．これは，表 4.6 に示したように，レストラン・データセットにおいて，food や anecdotes のレビュー数が多いためである．

CDRG について，ソースドメインが anecdotes，ターゲットドメインが service, food，ambience の組み合わせを除いて，生成されたレビュー文の数は一番少ない．これは，CDRG では，1 つの文から生成されるターゲットドメインの文はたかだか 1 つであるためである．

表 4.13: CARG によって生成されたレビュー文の数 (レストラン・データセット)

Source Domain	Service				Food				Price				Ambience				Anecdotes			
Target Domain	F	P	Am	An	S	P	AM	An	S	F	Am	An	S	F	P	An	S	F	P	Am
CARG1 T_s=0	1626	535	709	1723	5049	2019	2932	5668	802	1131	533	1143	1432	1451	393	1396	7437	8390	2505	5997
CARG1 T_s=0.3	1053	234	497	941	2623	1087	1854	3674	402	509	308	525	783	948	244	877	2662	3029	1005	2214
CARG2 T_s=0	822	299	491	1163	2076	1146	1470	3694	439	327	339	667	997	557	259	869	3045	2390	730	4343
CARG2 T_s=0.3	248	103	351	510	914	599	1041	1177	179	304	142	325	375	410	98	484	562	612	455	465
CDRG	65	75	92	196	524	388	456	608	80	112	84	124	116	132	61	128	648	768	303	604

S: service, F: food, P: price, Am: ambience, An: anecdotes.

表 4.14 は，ラップトップ・データセットにおける全てのドメインの組について，CARG によって生成された文の数を示している．この実験では閾値 T_s は 0 と設定している．レストラン・データセットの結果と同様に，フィルタリングを行わない CARG1 の方が CARG2 よりも生成されるレビュー文の数が多いことが確認できる．CARG1 ではおよそ 500 件から 3700 件，CARG2 ではおよそ 150 件から 1200 件の文が生成できている．ラップトップ・データセットでは属性が general であるレビュー文の数が多いため，ソースドメインが general のときに最も多くの文が生成されている．

CARG によって生成された文の例を表 4.15 と表 4.16 に示す．これらの表では，「ソース」のレビュー文を元に「ターゲット」のレビュー文が生成されたことを表す．下線は置換された単語を表す．ソースドメインのレビュー文の後ろの括弧はデータセットに付与された正解のラベルである．CARG では，生成したターゲットドメインの文には元の文と同じ極性ラベルが付与されるが，ターゲットドメインの文の後の括弧はその極性ラベルを表す．さらに，付与された極性ラベルが正しい場合は ✓を，正しくない場合は ✗を表示している．

表 4.14: CARG によって生成されたレビュー文の数 (ラップトップ・データセット)

Source Domain	General			Quality			Performance			Design		
Target Domain	Q	P	D	G	P	D	G	Q	D	G	Q	P
CARG1 T_s=0	2205	3762	3222	1167	927	771	1353	1197	522	531	459	987
CARG2 T_s=0	777	1257	1176	273	225	231	696	615	156	153	240	564

G: General, Q: Quality, P: Performance, D: Design.

表 4.15 は CARG によって適切なターゲットドメインの文が生成された例を示している．属性が service であるときのレビュー文 “The service was attentive and her suggestions of menu items was right on the mark.” に対して，評価語が “good” や “great” など food の評価に良く使われるものに置換され，また “service” が “food” に置換されており，ターゲットドメイン (food) のレビューらしい文が生成されている．また，food 属性のレビュー文 “The wine list is extensive and impressive.” について，“wine” と “list” といった food に関連する単語が “customer” と “service” に置換され，“extensive” と “impressive” といった food 属性に言及する文によく使われる形容詞が servicee 属性に言及する文によく使われる形容詞 “good” と “friendly” に置換され，service 属性らしい文が生成されている．同様に，属性が ambience の文 “The decor is really blah and not at all hip or happening.” については，“decor”，“hip”，“happening” といった単語が food に関連する単語 “pizza”，“salty”，“spicy” に置換された．

一方，CARG によって生成されたレビュー文の全てが適切なわけではなく，不自然な文やドメインに関連しない文，または自動付与された極性ラベルが真の極性と一致しない文も多数存在している．このようなレビュー文は極性判定モデルの性能に悪影響を与える可能性がある．表 4.16 はそのような不適切な文の例を示している．属性が food であるレビュー文 “The dinner menu is diverse and top-notch as well.” に対して，food 属性に言及した文によく使われる単語 “menu” を service の特徴語 “table” に置換して文を生成した．しかし，文全体の意味を見ると，service 属性に関する文とはみなせない．2 番目の例では，属性が service であるレビュー文 “All my co-workers were amazed at how small the dish was.” に対して，food 属性によく使われる単語 “small” や “dish” が，“good” や “service” といった単語に置換されて文を生成した．元の文では small という比較的否定的な意味を持つ単語があるため，文の極性として negative のラベルが付与されている．しかし，“smal” が “good” という肯定的な意味を持つ単語に置換されたため，文の極性ラベルとして positive を付与するべきであるが，元の文と同じ negative という誤ったラベルが付与されている．

以上の考察から，CARG による文生成手法には以下のような改善の余地がある．

1. [MASK] に埋めるべき単語を予測するとき，単語の極性を考慮せず，単に

BERT の MLM を適用している．しかし，置換された単語が元の単語の極性と異なる場合，結果として文全体の極性が変化する可能性がある．したがって，[MASK] をターゲットドメインの感情語に置換する際，元の単語との極性の一致を考慮する必要がある．

2. 同様に [MASK] に埋めるべき単語としてターゲットドメインとの関連性が低い単語が選ばれることがある．これは，ドメインのレビュー文の数が少ないときに，ドメインに固有の特徴語を適切に抽出できていないことが原因と考えられる．対応策として，[MASK] に埋められる単語がドメインに属する確率を計算する機能を追加することが考えられる．つまり，[MASK] に埋められる単語がどのドメインに属するのかを判定する機能を追加する．

3. 上の考察では述べていないが，生成されたレビュー文の全体を見ると，ほぼ同じ意味を持つ文が重複して生成されていることが確認されている．また，置き換えられた特徴語に多様性がない (ごく限られた特徴語のみに置換されている) ことも確認されている．これは，ビームサーチアルゴリズムによって MLM による予測確率が低い単語への置き換えを切り捨てていることと，MLM によって予測する単語の数が閾値 T_k で制限されている (4.2.4 項で示したように今回の実験では最大 100 件) ことが原因だと思われる．アルゴリズムを改良し，より多くの単語を用いて [MASK] を置き換えて文を生成することにより，この問題が解決できる可能性がある．

表 4.15: CARG によって生成された適切なターゲットドメインの文の例

ドメイン	レビュー文
service (ソース)	The service was attentive and her suggestions of menu items was right on the mark. (positive)
food (ターゲット)	the food was good and her choice of menu items was great on the menu. (positive✓)
food (ソース)	The wine list is extensive and impressive. (positive)
service (ターゲット)	The customer service is good and friendly. (positive✓)
ambience (ソース)	The decor is really blah and not at all hip or happening. (negative)
food (ターゲット)	The pizza is really blah and not at all salty or spicy. (negative✓)

表 4.16: CARG によって生成された不適切なターゲットドメインの文の例

ドメイン	レビュー文
food (ソース)	The dinner menu is diverse and top-notch as well. (positive✓)
service (ターゲット)	The dinner table is diverse and top-notch as well. (positive✓, ドメインに関連しない)
service (ソース)	All my co-workers were amazed at how small the dish was. (negative✓)
food (ターゲット)	All my co-workers were amazed at how good the service was. (negative✗)

第5章　おわりに

5.1　まとめ

本論文は，属性に対する極性判定において，異なる属性のラベル付きデータを利用して極性判定の分類器を学習する領域適応の手法を提案した．問題設定として，ソースドメインの属性に関するレビュー文については極性のラベルが付与されたデータが存在するが，ターゲットドメインの属性に関するレビュー文についてはそのようなデータは存在しないと仮定した．ターゲットドメインの極性判定の分類器を機械学習するために，そのラベル付きデータを自動構築した．

ターゲットドメインのラベル付きデータを自動構築するため，以下の2つの手法を用いた．1つ目の手法は，自動ラベル付けによる訓練データの自動生成であった．ソースドメインのラベル付きデータを用いて極性判定のモデルを BERT によって学習し，これを用いて，ターゲットドメインにおけるラベルなしデータセットに対して極性のラベルを自動的に付与した．この際，自動判定された極性ラベルは誤りを含む可能性があるため，BERT による判定の信頼度の高い事例のみにラベルを付与した．

2つ目の手法は，Cross-Aspect Review Generation による訓練データの生成であった．まず，ソースドメインとターゲットドメインのそれぞれについて，ドメインに固有の感情語と特徴語を抽出した．次に，ソースドメインのラベル付きレビュー文について，ソースドメインに固有の感情語と特徴語を特殊トークン [MASK] に置き換えた．さらに，BERT の Masked Language Model によって [MASK] に当てはまる単語を予測し，その予測確率が大きく，かつターゲットドメインに固有の感情語もしくは特徴語に該当するものに置き換えて，ターゲットドメインのレビュー文の候補を複数生成した．最後に，PLL を用いて生成されたレビュー文の自然さを測り，自然ではない文の候補を除外した．

以上の2つの手法によって構築されたデータセットを用いて BERT モデルのファインチューニングを行い，極性判定ための分類器を学習した．学習の際，訓練データにおける極性ラベルの分布に偏りがあるという問題に対処するために，Focal Loss を適用した．

提案手法の評価実験を行った．5つの属性に関するレビューから構成されるレストラン・データセットと，4つの属性に関するレビューから構成されるラップトップ・データセットを実験に用いた．それぞれのデータセットについて，ソースドメインとターゲットドメインのそれぞれに属性を割り当て，全ての属性の組み合

わせによる領域適応の実験を行った．ベースライン手法と提案手法について，正解率を評価指標として，極性判定の性能を比較した．レストラン・データセットとラップトップ・データセットの両方について，ベースラインより提案手法の方が優れていることを確認した．提案手法の最高の正解率は，全てのドメインの組の平均正解率で，レストラン・データセットで 0.658，ラップトップ・データセットで 0.801 となり，いくつかのベースラインをおよそ 0.005 から 0.020 ポイント上回った．2 つの異なるデータセットでおおよそ同様の結果が得られたことから，提案手法がどのようなジャンルのレビューにも適用できるという意味での汎用性を有することを確認した．

さらに，CARG によって生成されたレビュー文に対して，誤り分析を行った．その結果，ドメインに関連しないレビュー文，不正確なラベルが付与された文，意味的に重複する文が生成されていることが確認され，それらがモデルの性能に悪影響を与える可能性があることがわかった．

5.2 今後の課題

評価実験では，全ての属性の組に対して提案手法がベースラインを上回ったわけではなく，極性判定の正解率が向上しなかった属性の組も存在する．その主な原因は，提案手法によって有効的なレビュー文が常に生成できるわけではないことが挙げられる．

そこで，今後の課題として，上記の問題の原因を精査し，任意のドメインの組において自動生成する文の品質を向上させる手法を探究したい．例えば，属性に関連する語として特徴語を抽出しているが，属性の特徴を明示的に示す単語だけでなく暗黙的に示す単語も抽出することで，特徴語のリストを拡張することを検討する．また，[MASK] の置換によってレビュー文を生成する際に，置換される単語がドメインに属する確率を計算する手法を導入し，よりドメインに関連する単語を含む文を生成できるようにする．生成されたレビュー文の極性を付与する方法も改良が必要である．現在の提案手法では，ソースドメインのデータセットで元の文に付与されている極性ラベルをそのまま自動生成したターゲットドメインの文の極性ラベルとしているが，両者が一致しないことがある．つまり，置換される単語と元の単語の極性が一致しないことなどにより，極性が反転することが確認されている．よって，極性ラベルの正しさを検証する機構を導入する．最後に，現在の手法ではレビュー文を独立に生成しているため，似ている文が複数生成されることも多いが，類似した文が重複して訓練データに含まれることは望ましくないため，レビューの生成アルゴリズムを改良して，意味的に重複しない文を生成する．

謝辞

本研究を進めるにあたり，数多くのご指導およびご助言いただきました主指導教員である白井清昭准教授に深謝いたします．また，副指導教員である池田心教授，白井研究室のメンバーに心より感謝いたします．

参考文献

[1] Yelp dataset. https://www.yelp.com/dataset, (2022年12月閲覧).

[2] bert-base-cased(huggingface). https://huggingface.co/bert-base-cased, (2023年1月閲覧).

[3] nltk-pos-tagging. https://www.nltk.org/book/ch05.html, (2023年1月閲覧).

[4] Sentiwordnet. https://www.nltk.org/howto/sentiwordnet.html, (2023年1月閲覧).

[5] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pp. 2200–2204, 2010.

[6] Hongjie Cai, Rui Xia, and Jianfei Yu. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 340–350, 2021.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[8] 平賀一昭. 商品レビューを対象とした極性判定ならびに属性抽出に関する研究動向の調査. 課題研究報告書, 北陸先端科学技術大学院大学, 9 2018.

[9] Blitzer John, Dredze Mark, and Pereira Fernando. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, p. 440–447, 2007.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv:1708.02002*, 2017.

[11] 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 e-prints*, p. arXiv:1907.11692, July 2019.

[12] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 79–86. Association for Computational Linguistics, July 2002.

[13] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. *Association for Computational Linguistics*, Vol. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35, 2014.

[14] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of the 12th Conference on Language Resources and Evaluation(LREC 2020)*, pp. 4933–4941, 2020.

[15] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2699–2712, 2020.

[16] 白静, 田中裕隆, 曹類, 馬ブン, 新納浩幸. BERT の下位階層の単語埋め込み表現列を用いた感情分析の教師なし領域適応. 情報処理学会研究報告, Vol. 2019-NL-240, No. 17, pp. 1–6, 2019.

[17] Peter D Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424, 2002.

[18] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, Texas, November 2016. Association for Computational Linguistics.

[19] Dongbo Xi, Fuzhen Zhuang, Ganbin Zhou, Xiaohu Cheng, Fen Lin, and Qing He. Domain adaptation with category attention network for deep sentiment analysis. In *WWW '20: The Web Conference 2020 Taipei Taiwan*, pp. 3133–3139, 2020.

[20] Jianfei Yu, Chenggong Gong, and Rui Xia. Cross-domain review generation for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics*, pp. 4767–4777, 2021.

[21] 青嶋智久, 中川慧. 日本語 BERT モデルを用いた経済テキストデータのセンチメント分析. 人工知能学会全国大会論文集, Vol. JSAI2019, pp. 4Rin127–4Rin127, 2019.