生成的言語モデルを用いた映画レビュー文からの細粒度な観点名の生成

石井　智大† 　荘司　慶行†† 　山本　岳洋††† 　大島　裕明††† 　藤田　澄男††††

Martin J. Dürst††

† 青山学院大学大学院 理工学研究科　〒 252–5258 神奈川県 相模原市 中央区 淵野辺
†† 青山学院大学 理工学部 情報テクノロジー学科　〒 252–5258 神奈川県 相模原市 中央区 淵野辺
††† 兵庫県立大学大学院 情報科学研究科　〒 651–2197 兵庫県 神戸市 西区 学園西町
†††† ヤフー株式会社　〒 102–8282 東京都 千代田区 紀尾井町 東京ガーデンテラス紀尾井町 紀尾井タワー
E-mail: †ishii@sw.it.aoyama.ac.jp, ††{shoji,duerst}@it.aoyama.ac.jp, †††t.yamamoto@sis.u-hyogo.ac.jp,
††††ohshima@ai.u-hyogo.ac.jp, †††††sufujita@yahoo-corp.jp

あらまし　本論文では，生成的言語モデルを用いて，あるレビュー文の言及する観点名（すなわち，その文が作品のどの側面に注目して書かれているか）を推定をする方法を提案する．具体的には,「宇宙空間での爆発がリアルだった。」という文から,「特撮技術」という観点名を生成する．このようレビュー文に観点名をつける場合，従来は事前に人手で決めた観点に分類するか，観点名を抽出していた．一方で，映画レビューにおける観点は多様で，あらかじめ列挙しきれないため，生成的言語モデル T5（Text-to-Text Transfer Transformer）を学習させ，文を入力すると観点名を生成するようにした．この際，少ないデータで観点名を正しく生成可能にするため，ドメインデータによる追加学習や，経由タスクによるファインチューニングを施した．こうした学習上の工夫の効果を検証するために，実際に Yahoo!映画のデータを用いて，評価実験を行った．生成された観点名の正しさ，多様さについて人手でラベル付けすることで，提案手法は少量の学習データから細粒度な観点を正しく多様に生成できることが分かった．

キーワード　情報検索，観点抽出，評判情報，T5

1　はじめに

禅宗の教えの中に，一水四見という熟語がある．道端の水たまりを見た際に，喉の乾いた人であれば「この水は飲み水としてどうか」と考えるし，見た目を気にする人であれば「この水たまりに映った自分の姿はどうか」と，鏡として考える．一方で，人間ではなく，魚に水たまりを見せた場合には「この水たまりは住みやすいか」という観点で水たまりを評価する．単純な水たまりですら，立場や，ものの見方によって，水たまりのどの側面に注目し，どう感じるかは異なるといえる．単純な水たまりひとつとっても，多様な見方が存在するのに，映画のような複雑な映像コンテンツであれば，感じ方は千差万別であるに違いない．

近年，情報通信技術の発展に伴い，映画とそれに関する感想との接し方が，大きく変化してきている．サブスクリプションサービスに代表されるような，見放題の映像配信サービスの普及に伴い，人々が日常的に接する映像コンテンツの量は増加してきている．人々は，毎日，膨大な選択肢の中から，どの映像を見るかを判断しながら生活するようになってきている．加えて，Yahoo!映画 [1]や，Filmarks[2]に代表される，映画レビューサービスも一般化してきている．

映画レビューサービスのレビューを読み，どの映画を観るかを判断することは，日常の一部になりつつある．限られた時間の中で，どの映画を視聴するかを判断するうえで，レビュー情報は判断を助ける重要な情報源の一つになりつつある．一方で，映画レビューサービスのレビュー投稿は膨大であるため，その全てに目を通すことは不可能である．加えて，老若男女，素人から映画マニアまで，多数多様なユーザがレビューを投稿するため，投稿されたレビューは投稿者によって注目している観点が異なることも多い．例えば，ある映画について,「この映画は時代考証が正確らしいので，そこに言及しているレビューを読みたい」と思った際に，実際にレビューを読んだら俳優について言及するものばかりだった，ということは，ままある．

このような問題を解決するために，評判情報分析の分野では，レビュー中に含まれる記述が，どんな観点に注目したものであるかを推定する研究も多く行われてきている．しかしながら，従来の観点分類や抽出の研究では，それぞれの記述に対し，事前に決めた観点に分類するか，記述から観点名を抽出するアプローチが一般的である．例えば，テレビやカメラといった一般的な商品分野では，あらかじめ「細部がくっきり見える」という記述は「解像度」,「持ち運びに便利」という記述は「重さ」や「サイズ」など，あらかじめ定められた観点に紐づけることが一般的であった．

こういった際に，映画に対する感想などの，個人の主観に応じて注目観点が変わる分野では，事前に観点を列挙したり，観

1：Yahoo!映画「Yahoo!映画｜新作映画やレビュー・クチコミ情報」: https://movies.yahoo.co.jp/
2：Filmarks「Filmarks｜映画情報サービス-国内最大級の映画レビュー数」: https://filmarks.com/

点名を抽出することが困難である．例えば，同じ映画というくくりであっても，SF 映画と恋愛映画では，それぞれのレビューに同一観点が存在するとは限らない．

そこで，本研究では，生成的言語モデルを用いて，あるレビュー文の言及する観点名（すなわち，その文が作品のどの側面に注目して書かれているか）を生成する方法を提案する．事前に観点を用意する分類的なアプローチや，評価観点を抽出するアプローチでは対応できない，抽象的な観点名を，生成的言語モデルを用いて生成する．具体的には，「宇宙空間での爆発がリアルだった.」という文には，「科学考証」という単語は含まれないし，「科学考証」という単語は細粒度すぎるので，事前に人手で用意することが困難である．このような，細粒度で具体的な観点名を，近年主流になりつつある生成的な言語モデルを用いて，網羅的に扱えるようにする．

このような機能を実現するために，本研究では事前学習された大規模言語モデルである T5（Text-to-Text Transfer Transformer）[1] を用いて，レビューが言及する観点名を生成する方法を提案する．そのために，まず，レビュー文が言及する観点名をクラウドソーシングを用いて，収集し，データセットを構築した．次に，映画レビューを用いて，T5 の事前学習を通して，映画レビューをモデルに学習させた．具体的には，「この映画は最高です．北野武らしいバイオレンス！國村隼の昔ながらのヤクザらしい演技が素敵です。終盤は同じような [MASK][MASK] の連続でちょっと飽きたかな。」というレビューを T5 に入力すると MASK に入る単語は「アクション」「シーン」を出力する．次に，クラウドソーシングでラベル付けしたレビュー文を用いて，2 文を T5 に入力してその文の観点が同一かを判定する，文を T5 に入力してその文に観点が含まれるかを判定するという 2 つのタスクで，実際にモデルに観点が含まれる文と観点が同じ文をどういう文かを覚えこませた．具体的には，「悪い男を演じるときのブラピの目力は神！」と「タランティーノはこういう時にふざけるから肩透かしになる。」という 2 つの文を T5 に入力すると，2 つの文の言及する観点は等しくないと出力する．また，「墓暴きのシーンは日本だと火葬なので、矛盾している」という文を T5 に入力すると，この文は何らかの観点を含んでいると出力する．こうすることで，少ないデータで正しい観点名を生成できるようにした．最後にクラウドソーシングでラベル付けしたレビュー文を用いて，レビュー文を T5 に入力してその文の観点名を生成するというタスクで，実際にモデルにタスクを覚えこませた．具体的には「このころの山手線は、確かに銀色に緑のラインだった！」という文を T5 に入力すると，「時代考証」という観点名を出力する．

このように学習させた言語モデルを使うことで，レビュー文中に含まれる観点名を生成可能になる．生成的言語モデル T5 は大量の学習データを必要とする．しかし，レビュー文と観点名のペアのデータセットは存在しない．また，レビュー文と観点名のペアのデータセットを作成することは難しい．少ないデータで観点名を正しく生成可能にするため，ドメインデータによる追加学習や，経由タスクによるファインチューニングを施した．こうした学習上の工夫の効果を検証するために，実際

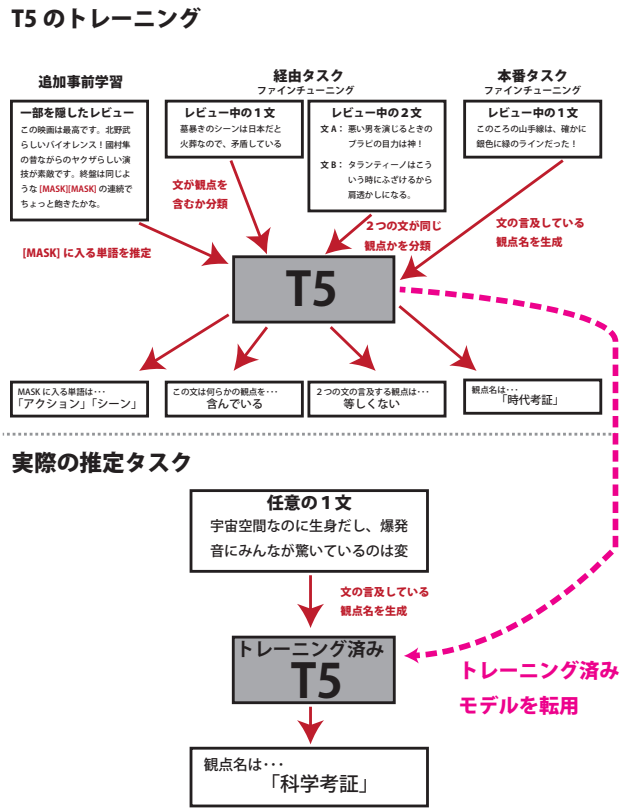

図 1　提案手法の概要図．追加事前学習，経由タスクによるファインチューニング，本番タスクによるファインチューニングを通して鍛えたモデルを用いて，実際に任意の文に対して観点名を生成する．

に Yahoo!映画のデータを用いて，評価実験を行った．実際の実験では，最初に，観点名が正しいかを評価した．次に，被験者に観点名が細粒度かを評価してもらい，評価を行った．最後に，学習データにない観点名を生成できたかを評価した．これらの実験を通して，実際に提案手法が細粒度な観点名を正しく多様に生成できるかを明らかにした．

2　関 連 研 究

本研究は，オンラインレビューを用いて，その文が言及する観点を推定する．そのために，生成的言語モデルを用いて，レビュー文が言及する観点を生成している．文が言及する観点名を生成することは要約に近い．そのため，関連する研究として，オンラインレビューを用いたアプリケーションについて 2.1 節，aspect extraction について 2.2 節で，言語モデルを用いた要約について 2.3 節でそれぞれ説明し，本研究の位置づけを示す．

2.1　オンラインレビューを用いたアプリケーション

オンラインレビューを分析することで，ユーザの感情を推定する試みは古くから行われてきている．Xu ら [2] はレビューをユーザの質問に答えるために利用できる知識のソースに変える手法を提案している．データセットを aspect-based sentiment analysis のための一般的なベンチマークに基づいて構築する．BERT（Bidirectional Encoder Representations from

Transformers）のファインチューニングの性能を高めるために，post-training アプローチを模索している．レビューをユーザの質問に答えるために利用できる知識のソースに変えることが本研究と異なる．

Rietzler ら [3] は，aspect の感情を分類する手法を提案している．ドメイン固有の BERT のファインチューニングと，タスク固有のファインチューニングを使用している．aspect の感情を分類することが本研究と異なる．

2.2 Aspect extraction

映画レビューをはじめとした投稿レビューは，ユーザがそのアイテムに対する意見や感想を自由に記述するものである．そのため，ユーザごとに観点，観点ごとの評価，用いる表現などあらゆる基準が異なっている．そこで，各レビューにどのような意見や感想が書かれているかを分類する研究が行われている．

Singh ら [4] は，映画レビューに対する感情分類手法を提案している．SentiWordNet を用い，語の品詞やある語の周辺に存在する語に着目しながら感情表現やその極性について分類している．

Jo ら [5] は，投稿レビューから様々な観点と，観点に対する感情の組み合わせを自動的に発見する手法を提案している．観点に対する感情を発見することが本研究と異なる．

Peng ら [6] は aspect sentiment triplet extraction に対処するためのフレームワークを提案している．Aspect sentiment triplet extraction は aspect が何であるか，その感情極性がどのようになるか，なぜそのような極性を持つのかを示す triplet を抽出するタスクである．感情極性がどのようになるか，なぜそのような極性を持つのかを抽出することが本研究と異なる．

Karimi ら [7] は BERT Adversarial Training（BAT）と呼ばれるアーキテクチャを提案している．BAT は敵対的学習を，post-training BERT に適用したアーキテクチャである．敵対的学習と post-training を適用していることが本研究と異なる．

He ら [8] は首尾一貫した aspect を発見することを目的としたニューラルアプローチを提案している．Transformer を用いないことが本研究と異なる．

Xu ら [9] は 2 種類の事前学習済み埋め込みを用いた，シンプルな CNN モデルを提案している．CNN を用いていることが本研究と異なる．

Angelidis ら [10] はオンライン商品レビューから意見を要約するためのニューラルフレームワークを提案している．aspect extraction と，感情予測器の 2 つの弱教師付きコンポーネントを組み合わせていることが本研究と異なる．

本研究は，映画レビューサイトに投稿されたレビュー内の文が言及する観点を生成するという研究である．このようなレビューから言及している観点を抽出する例はいくつか存在する．しかし，本研究が対象としている映画レビューは映画により観点が異なり，無数に存在するので，あらかじめ観点を用意することができない．

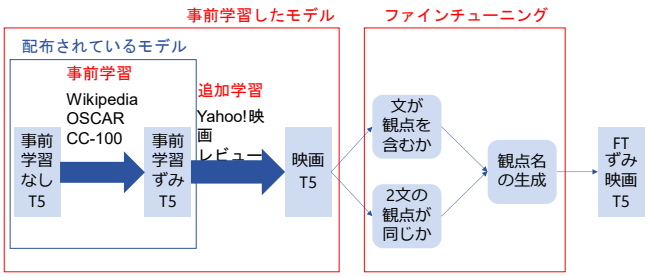

図 2　提案手法の手順流れ図

2.3 言語モデルを用いた要約

本研究では，言語モデルを用いてレビュー文から観点名を生成する．レビュー文から観点名を生成することは要約に近い．このように言語モデルを用いて要約する研究が行われている．

Liu ら [11] は抽出型要約のための BERT の単純な改良型である BERTSUM を提案している．抽出型要約と BERT を用いていることが本研究と異なる．

Lewis ら [12] は sequence-to-sequence モデルを事前学習するためのノイズ除去オートエンコーダ BART（Bidirectional and Auto-Regressive Transformers）を提案している．

Zhang ら [13] は PEGASUS（Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence models）を提案している．PEGASUS では，入力文書から重要な文を除去，マスクし，残りの文から 1 つの出力列を生成する．入力文書から重要な文を除去，マスクすることが本研究と異なる．

3 提 案 手 法

本節では，映画レビュー文を入力すると，その文が言及する観点名を生成する手法について述べる．本研究では，はじめに映画のレビューを大規模言語モデル（T5）に入力し，事前学習する．そして，映画レビューを用いて学習したモデルを用いて，他のタスクで学習する．最後に，クラウドソーシングでラベル付けした学習データを用いて，大規模言語モデル（T5）によってそれぞれの文の観点名を生成する．提案手法の概要を図 1 に示す．手法の要点を流れ図の形式で表したものを図 2 に示す．提案手法は，T5 で観点名を生成する前に，他のレビューをドメイン知識として利用する．また，生成の精度向上のため，レビューを用いた他のタスクで学習することを提案する．これらの詳細について，各節で説明する．

3.1 データセットの前処理

はじめに，映画レビュー文から，その文が言及する観点をラベル付けする．本手法では，実際の映画レビューサイトのレビューを句点で分割した．レビュー本文の文単位分割を行った．レビュー文が言及している観点をクラウドソーシングでラベル付けした．1 人あたり 20 文にラベル付けし，50 人がラベル付けした．これを 2 回行った．クラウドソーシングによるラベル付けに用いた Web アプリケーションのスクリーンショットを

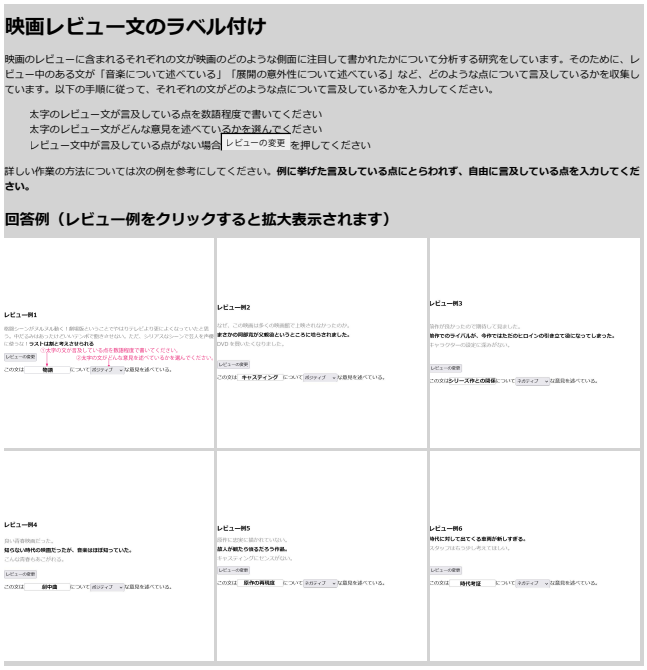

図 3　クラウドソーシングによるラベル付けに用いた Web アプリケーションのスクリーンショット．背景が灰色の部分がタスクの説明である．背景が白色の正方形は回答例の画像である．これをクリックすると図 4 のような画像が表示される．下部は実際にラベル付けするレビューである．太字の文がラベル付けする文で，灰色の文はラベル付けする文の前後の文である．

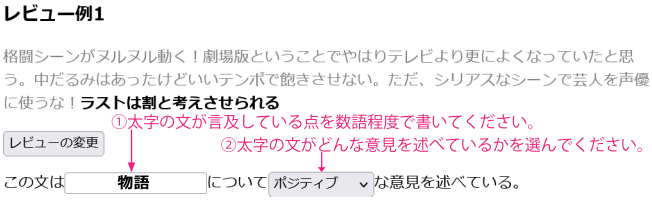

図 4　回答例．太字の文がラベル付けする文で，灰色の文はラベル付けする文の前後の文である．

図 3 に示す． 自然言語の文章を自動処理で文単位に正確に分割することは難しいため，人為的に決定したルールに基づき分割することとした．本研究では，改行文字を文の区切り文字と設定し，レビュー本文を分割した．Web アプリケーションでは観点名をラベル付けする文を太字で表示し，観点名を付けない文を灰色で表示した．また，観点名を「この文は『物語』について [ポジティブ | ネガティブ | ニュートラル] な意見を述べている。」という形式で回答させた．観点がない文や観点が分からない文は観点なしとラベル付けした．

3.2　T5 の追加事前学習

次に，映画のオンラインレビューサイトから，レビューを学習した．図 1 に示すように，Yahoo!映画などの映画レビューサイトでは，映画について投稿されたレビューを収集できる．

映画レビューを T5 に学習させるために，レビューを T5 に入力し，追加で事前学習を行った．事前学習の方法は Raffel ら [1] が提案した事前学習の方法である．本研究では，レビュー文が言及する観点名を生成するために，事前学習済みの言語モデルである T5 を用いた．手法の概要を，図 1 に示す．T5 では，事前に大規模なコーパスで学習した事前学習済みモデルに，追加学習や，ファインチューニングを施すことで，様々なタスクを解くことができる．

3.3　T5 を用いた同一観点の判定

T5 を少量のデータセットでファインチューニングすることで観点名を生成できるようにするために，別のタスクでファインチューニングする．本研究は T5 を少量のデータセットでファインチューニングすることで観点名を生成できるようにする必要がある．そのために 2 文を T5 に入力し，2 文が言及している観点が同一かを 0 か 1 を出力して判定するというタスクでファインチューニングした．このようなタスクにすることで文と文の組み合わせの数のデータを作ることができる．文 A と文 B をスペースで区切り,「同一観点の判定:」という接頭辞を付けた．例えば,「同一観点の判定: 永作博美の怪演も必見 ストーリーはアメリカ映画の王道」というように入力した．

3.4　T5 を用いた観点有無の判定

T5 を少量のデータセットでファインチューニングすることで観点名を生成できるようにする必要がある．そのためにレビュー文を T5 に入力し，その文に観点が含まれているかを 0 か 1 を出力して判定するというタスクでファインチューニングした．入力に「観点有無の判定:」という接頭辞を付けた．例えば,「観点有無の判定: 永作博美の怪演も必見」というように入力した．

3.5　T5 を用いたレビュー文ごとの観点名の生成

本研究では，レビュー文からその文が言及する観点名を生成するために，事前学習済みの言語モデルである T5 を用いた．T5 では，事前に大規模なコーパスで学習した事前学習済みモデルに，追加で事前学習や，ファインチューニングを施すことで，様々なタスクを解くことができる．今回の場合は，任意の文に対して，T5 にその文が言及する観点名を推論させる．レビュー文を T5 に入力し，観点名を出力するというタスクで T5 をファインチューニングした．文の観点名が「観点なし」以外の文を入力した．入力に「観点名の生成:」という接頭辞を付けた．例えば,「観点名の生成: 永作博美の怪演も必見」というように入力した．

4　評 価 実 験

自動評価では，自動評価指標を用いて，生成された観点名とラベル付けされた観点名の類似度を計算する．

被験者実験では，実際に被験者に，生成された観点名の正し

さ，多様さについて人手でラベル付けさせる．本節では，提案手法の有効性を示すために用いたデータセットと，そのデータセットを用いて行った評価実験の詳細について述べる．

評価実験として，用意した文について，提案手法を含めた手法で観点名の生成を行い，観点名について，被験者実験によるアンケートで文との適合度を測った．評価実験では

- 観点名が正しいか
- 観点名が細粒度か
- 未知の観点名を生成できたか

を評価する．観点名が正しいかは，生成された観点名とクラウドソーシングでラベル付けした観点名のコサイン類似度を計算することによって評価する．観点名が細粒度かは，被験者実験によるアンケートで評価する．未知の観点名を生成できたかは，生成した観点名のうち，学習データにない観点名の割合で評価する．

4.1 データセットの概要

提案手法の評価のために，ヤフー株式会社と株式会社 GYAO が運営する映画情報サイトである Yahoo!映画に投稿されたユーザーレビューデータを用いた．T5 の追加学習には 176,970 件のレビューを T5 に入力した．2 文の観点が同一かの判定タスクには 10,000 ペアを T5 に入力した．2 文の観点が同一かの判定タスクには，クラウドソーシングでラベル付けしたレビュー文の組み合わせを用いた．クラウドソーシングでラベル付けしたレビュー文の組み合わせで，観点名のコサイン類似度を Sentence-BERT を用いて計算し，0.9 以上の場合，2 文の観点が同一とラベル付けした．レビュー文に観点名が含まれるか判定するタスクには，3,759 文を T5 に入力した．観点名の生成のファインチューニングには 1,512 文を T5 に入力した．観点名が細粒度かと未知の観点名を生成できたかの評価には，学習に用いていないデータの内，観点があると分類された文のみを用いた．また，被験者実験に用いる文は学習に用いていないデータの内，観点があると分類された文の内，明らかに感想が書かれている文や，観点が分からない文を除外したデータを用いた．レビュー文に観点が含まれているかを分類し，観点が含まれるレビューのみを用いた．分類には，クラウドソーシングでラベル付けしたレビューを用いてファインチューニングした BERT を用いた．ファインチューニングした BERT のテストデータの正解率は 0.629 だった．

4.2 比 較 手 法

第 3 節で述べた手法の効果を確認するために，追加学習の有無，経由タスクの有無と順序を変更し，比較手法を用意した．提案手法と比較手法を表 1 に表す．具体的には，

追加学習のみ： レビューを用いて T5 を追加学習した後，レビュー文を入力して観点名を出力するタスクで T5 をファインチューニングした手法である．

追加学習→同一観点推定： レビューを用いて T5 を追加学習した後，2 文を入力して，その文の観点が同一かを判定した後，観点名を生成するタスクでファインチューニングした手法である．

追加学習→観点有無推定： レビューを用いて T5 を追加学習した後，レビュー文を入力して，その文に観点が含まれるかを判定した後，観点名を生成するタスクでファインチューニングした手法である．

追加学習→同一観点推定→観点有無推定： レビューを用いて T5 を追加学習した後，2 文を入力して，その文の観点が同一かを判定した後，レビュー文を入力して，その文に観点が含まれるかを判定した後，観点名を生成するタスクでファインチューニングした手法である．

追加学習→観点有無推定→同一観点推定： レビューを用いて T5 を追加学習した後，レビュー文を入力して，その文に観点が含まれるかを判定した後，2 文を入力して，その文の観点が同一かを判定した後，観点名を生成するタスクでファインチューニングした手法である．

追加学習なし： レビュー文を入力して観点名を出力するタスクで T5 をファインチューニングした手法である．

同一観点推定： 2 文を入力して，その文の観点が同一かを判定した後，観点名を生成するタスクでファインチューニングした手法である．

観点有無推定： レビュー文を入力して，その文に観点が含まれるかを判定した後，観点名を生成するタスクでファインチューニングした手法である．

同一観点推定→観点有無推定： 2 文を入力して，その文の観点が同一かを判定した後，レビュー文を入力して，その文に観点が含まれるかを判定した後，観点名を生成するタスクでファインチューニングした手法である．

観点有無推定→同一観点推定： レビュー文を入力して，その文に観点が含まれるかを判定した後，2 文を入力して，その文の観点が同一かを判定した後，観点名を生成するタスクでファインチューニングした手法である．

の 10 手法を比較する．

4.3 実　　装

T5 は Python で書かれた Transformer ベースのモデルのライブラリである Hugging Face Transformers[3]による実装を用いた．T5 は日本語の事前学習済みモデル [4]を用いた．追加学習のパラメータは，最大シーケンス長を 512，バッチサイズを 16，optimizer を Adafactor，学習率を 0.005，Weight decay を 0.001，Warmup steps を 2,000 に設定した．

ファインチューニングのパラメータは Hugging Face Transformers のデフォルト値を用いた．ただし，学習率を 0.0003 に設定した．それ以外のパラメータは Hugging Face Transformers のデフォルト値を用いた．Hugging Face Transformers では，生成的言語モデルの generate メソッドに文を与えることで，そのモデル上で観点名を生成できる．観点名が正しいか

3：Hugging Face「Hugging Face Transformers」：
https://huggingface.co/docs/transformers/index

4：Hugging Face「sonoisa/t5-base-japanese」：
https://huggingface.co/sonoisa/t5-base-japanese

表 1　提案手法と比較手法．同一性推定とは 2 文を T5 に入力し，その文の観点が同一かを判定する．観点有無推定とは，文を T5 に入力し，その文に観点が含まれているかを判定する．観点名生成とはレビュー文を T5 に入力し，その文が言及する観点名を生成する．

手法名	追加学習の有無	経由タスク	本番タスク	生成観点名数	自動評価（平均類似度）	未知観点名数
追加学習のみ		なし	観点名生成	84	0.673	31
学習あり同一のみ		同一性推定		79	0.678	23
学習あり有無のみ	あり	観点有無推定		**92**	0.677	**35**
学習あり同一先（提案手法 A）		同一性推定→観点有無推定		11	0.676	1
学習あり有無先（提案手法 B）		観点有無推定→同一性推定		56	**0.687**	11
追加学習なし		なし		76	0.678	25
同一推定のみ		同一性推定		74	0.678	23
有無推定のみ	なし	観点有無推定		74	0.675	28
同一先		同一性推定→観点有無推定		77	0.676	28
有無先		観点有無推定→同一性推定		69	0.679	27

表 2　提案手法と比較手法の BERTScore の結果．それぞれの手法で生成された観点名とラベル付けされた観点名を BERTScore に入力し，適合率と再現率と F_1 スコアを計算した．それぞれの観点名に適合率と再現率と F_1 スコアが計算されるため，それを手法ごとに平均した．

手法名	適合率	再現率	F_1
追加学習のみ	0.800	0.793	0.796
学習あり同一のみ	0.800	**0.794**	0.796
学習あり有無のみ	0.799	0.791	0.795
学習あり同一先（提案手法 A）	0.798	0.786	0.790
学習あり有無先（提案手法 B）	**0.802**	0.793	**0.797**
追加学習なし	0.800	0.791	0.795
同一推定のみ	0.794	0.789	0.791
有無推定のみ	0.795	0.789	0.791
同一先	0.797	0.791	0.794
有無先	0.798	0.792	0.795

の評価には Python で書かれた SentenceTransformers[5]を用いた．SentenceTransformer は多言語の事前学習済みモデル [6]を用いた．

観点名が細粒度かと未知の観点名を生成できたかの評価に用いるデータの分類には，日本語の事前学習済みモデル [7]を用いた．

4.4　各手法による観点名の生成

評価実験を行うために，実際に観点名を生成するための文を選定した．文を BERT に入力し，文に観点名が含まれるかを判定し，観点が含まれる文を選定した．観点が含まれると判定された文を各手法のモデルに入力し，観点名を生成した．

5：SentenceTransformers「SentenceTransformers Documentation」：
https://www.sbert.net/

6：Hugging Face「sentence-transformers/distiluse-base-multilingual-cased-v2」：
https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

7：Hugging Face「cl-tohoku/bert-base-japanese-whole-word-masking」：
https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

4.5　自動評価による評価実験

各手法を用いて得られた観点名の評価のために，自動評価による実験を行った．生成された観点名とクラウドソーシングでラベル付けした観点名を SentenceTransformer に入力し，観点名のコサイン類似度を計算した．クラウドソーシングでラベル付けされた 505 文を各手法のモデルに入力した．クラウドソーシングでラベル付けされた文それぞれに，生成された観点名とラベル付けされた観点名の類似度があるため，それを平均した．

4.6　被験者実験による評価実験

各手法を用いて得られた観点名の評価のために，被験者実験による被験者実験を行った．

被験者には 1 文と，その文を入力した結果得られた観点名を 1 つを提示し，

- 観点名は観点として成立しているか
- 観点名は正しいか
- 観点名は十分に細粒度か

を回答させた．観点名が観点として成立していれば，観点名は正しいかと観点名は十分に細粒度かという項目に回答させた．観点名が観点として成立しているかは，成立していないと成立しているの 2 段階で回答させた．観点名が正しいかは合っていないことを表す 1 からぴったりと合っていることを表す 5 までの，5 段階で回答させた．観点名が十分に細かいかは普遍的で荒いを表す 1 から十分に細かいことを表す 5 までの，5 段階で回答させた．各手法で生成した観点名を並べて，どの手法の観点名が合っているか，間違っていれば，何という観点名を付けるかを回答させた．

4.7　未知の観点名を生成できたかの評価実験

各手法を用いて得られた観点名の評価のために，未知の観点名を生成できたかの評価を行った．

4.8　実 験 結 果

本節では，評価実験結果について述べる．

4.8.1　自動評価による評価実験

自動評価による評価実験の結果を表 1 と表 2 に示す．自動評

表 3　被験者実験の結果．成立とは，生成された観点名が観点として成立しているかである．合っているかとは，観点名が文が言及している観点として合っているかである．細粒度かとは，観点名が十分に細かいかである．

手法	成立	正しいか	細粒度か
追加学習のみ	**1.945**	**3.500**	1.982
学習あり同一のみ	1.941	3.314	2.031
学習あり有無のみ	1.924	3.384	**2.037**
学習あり同一先（提案手法 A）	1.937	3.198	1.680
学習あり有無先（提案手法 B）	**1.945**	3.353	1.929
追加学習なし	1.928	3.455	1.901
同一推定のみ	1.941	3.348	1.920
有無推定のみ	1.937	3.383	1.995
同一先	1.941	3.238	1.933
有無先	1.937	3.329	1.919

表 4　レビュー文に含まれない観点名が生成される文の例（レビュー文は，解説のために人手で作成したもの）

レビュー文	観点名	手法
北野武監督らしさが出た、独特の展開だったと思います！	演出	学習あり同一観点 学習あり同一先 学習あり有無のみ 学習あり有無先 追加学習なし 同一推定のみ 同一先 有無推定のみ 有無先

表 5　文中に含まれる観点名が抽出的に生成されるレビュー文の例（レビュー文は，解説のために人手で作成したもの）．

レビュー文	観点名	手法
知らない時代ですが、音楽はほぼ知っている曲ばかりでした	曲	学習あり同一観点

価（平均類似度）と適合率と F_1 スコアは提案手法 B が一番高かった．自動評価（平均類似度）と BERTScore はどちらも手法ごとの差があまりなかった．表 1 から，2 文の観点が同一かというタスクでファインチューニングすると生成される観点名の数が減ることが分かる．

4.8.2　被験者実験による評価実験

被験者実験による評価実験の結果を表 3 に示す．表 3 から追加学習のみと提案手法 B が，一番，観点名が観点として成立していることが分かる．また，追加学習のみが一番，観点名が正しかった．また，学習あり有無のみが一番，観点名が細粒度だった．

4.8.3　未知の観点名を生成できたかの評価実験

未知の観点名を生成できたかの評価実験の結果を表 1 に示す．表 1 から，生成した観点名の数が一番多かったのは，学習あり有無のみだった．また，生成された観点名の内，学習データにない観点名が一番多かったのも学習あり有無のみだった．

5　考　　察

本節では，評価実験の結果について考察する．まず，自動評価による評価実験の結果について考察する．表 1 と表 2 から提案手法 B の自動評価（平均類似度）と BERTScore が一番高いことが分かる．このことから，追加学習をした後，観点有無推定をした後，同一観点推定をすることがラベル付けしたデータとの類似度を上げる手法として有効であると考えられる．また，表 1 と表 2 から手法ごとの自動評価（平均類似度）と BERTScore の差があまりないことが分かる．このことから，追加学習と中間タスクによるファインチューニングはラベル付けされた観点名との類似度を上げることに大きく影響しないと考えられる．また，このことから，観点名の生成タスクでファインチューニングすることがラベル付けされた観点名との類似度を上げることに大きく影響すると考えられる．表 6 に，観点名の生成に失敗するレビュー文の例を示す（レビュー文は説明のために作成した架空のもので，実データではない）．一部の生成された観点名で,「C」などの一文字の観点名が生成される場合があった．これは，モデルが観点の有無を推定するタスクで学習しており，そのタスクは入力されたレビュー文に観点が存在するかしないかを 0 か 1 を出力するタスクで，一文字を出力することを学習したからと考えられる．このように比較手法は観点名の生成に失敗することもあるが，おおむね観点名の生成に成功する．

表 7 に，全ての手法が同じ観点名を出力するレビュー文の例を示す．表 7 から，レビュー文中に学習時に使われていた観点名が含まれていた際に，モデルはその観点名を生成結果として出力する傾向がみられた．表 1 から，生成された観点名は多くが，学習データに含まれることが分かる．このことから，レビュー文から観点名を生成すると，生成された観点名は学習データに含まれることが多いと考えられる．表 1 から，2 文の言及する観点が同一か判定するタスクでファインチューニングすると，生成される観点名の数が減ることが分かる．これは，2 文の言及する観点が同一か判定するというタスクが難しいと考えられる．また，今回，2 文をスペースで区切って，T5 に入力したが，レビュー文によっては文にスペースが含まれることがあるなどの理由で 2 文の区切りが分かりにくいと考えられる．

次に，被験者実験の結果について考察する．表 3 から，全ての手法で，観点名が観点として成立しているかは 1.9 以上である．このことから，全ての手法が生成した観点名の多くは観点として成立していることが分かる．表 3 から，提案手法 A が一番，粗粒度であることが分かる．

次に，学習データにない観点名を生成できたかの実験の結果について考察する．表 5 から，生成された観点名はレビュー文から抽出したような観点名が多いことが分かる．これは検証のために生成した例である．しかし，表 4 から，生成された観点名はレビュー文から抽出したものではない観点名もある．これは検証のために生成した例である．

表 1 と表 2 と表 3 から全ての手法で差があまりなかった．このことから，細粒度な観点名を生成するには，レビュー文を入力して観点名を生成するタスクでファインチューニングすることが一番有効であると考えられる．

また，今回，学習と評価に用いたデータは，レビューを文単

表 6　観点名の生成に失敗するレビュー文の例（レビュー文は，解説のために人手で作成したもの）.

レビュー文	観点名	手法名
國村隼の昔ながらのヤクザらしい演技が素敵です。	演技演技…	学習あり同一先

表 7　全ての手法が同じ観点名を生成するレビュー文の例（レビュー文は，解説のために人手で作成したもの）.

レビュー文	観点名
シナリオが結構面白かった	シナリオ
キャスティングはこれしか考えられない	キャスティング

位に区切るために，レビューを句点で区切ったが，実際のレビューには句点が存在しないこともあった．これが，レビュー文と観点名の関係を学習しづらくしていると考えられる．

6　まとめと今後の課題

本章では，本論文のまとめと今後の課題について述べる．

本研究では，映画レビュー文から文が言及する観点名を生成する手法を提案した．本研究の貢献は，レビュー文の観点の推定に生成的言語モデルが使えることを示したこと，および，少ないデータで観点名を生成できるようにするために，追加学習と経由タスクによるファインチューニングが有効であることを示したことの 2 点である．T5 で文から観点名を生成する際，追加学習と中間タスクでファインチューニングすることで，生成精度の向上を図った．Yahoo!映画の実際の作品ユーザーレビューに対し，提案手法と，ベースライン手法を適用し，被験者実験で比較することで，提案手法を評価した．評価実験の結果，レビュー文から観点名を生成するタスクでファインチューニングすることが，有効であることが分かった．

今後の課題として，データのクレンジングの強化と，応用的な評価を考えている．具体的には，生成された観点名を用いて，レビュー文を検索するシステムを構築し，それを評価する．さらに，学習データの量による生成観点名の質についても評価する予定である．

謝　　辞

本研究の一部は JSPS 科研費 21H03775，21H03774，22H03905 による助成，ならびに 2022 年度国立情報学研究所共同研究 22S1001 の助成を受けたものです．ここに記して謝意を表します．

文　　献

[1] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

[2] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, jun 2019.

[3] 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 Twelfth Language Resources and Evaluation Conference*, pp. 4933–4941, Marseille, France, May 2020. European Language Resources Association.

[4] V. K. Singh, R. Piryani, A. Uddin, and P. Waila. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pp. 712–717, 2013.

[5] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, p. 815–824, New York, NY, USA, 2011.

[6] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 05, pp. 8600–8607, Apr. 2020.

[7] Akbar Karimi, Leonardo Rossi, and Andrea Prati. Adversarial training for aspect-based sentiment analysis with bert. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8797–8803, 2021.

[8] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[9] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2018.

[10] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3675–3686, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[11] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, p. arXiv:1903.10318, 2019.

[12] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[13] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.