2021 年度 芝浦工業大学 工学部 情報工学科

卒　　業　　論　　文

主張と根拠のクラスタを用いた

多様な主張を提示するニュース推薦手法の提案

学籍番号　　**AL18036**

氏　　名　　片岡　凪

指導教員　　木村　昌臣

目 次

第1章　序論

1.1　研究の背景と目的

ニュース記事となる出来事は，記者によって受け取り方や伝え方が異なる．RSF（Reporters Sans Frontières）は，政治や文化などの要因によって記者の主張が制限されていることを問題視している [1]．近年の Web ニュースには世界中の記者たちの様々な主張が見られるが，言語の違いによる時間的コストなどから，多くの読者はそれらの主張の一部しか把握できない．また，ニュースサイトのアルゴリズムによって読者に合わせて読者の読む記事が選別されているため，選別されなかった記事の主張を読者が把握することは難しい [2][3].

Yang らは，ニュース読者が出来事を正確かつ迅速に把握できるように階層的クラスタリングを用いて記事に対するツイートの主張をグループ化する手法を提案した [4]．この研究では COVID-19 の話題に限定した主張の文をグループ化しているが，この手法を記事に適用した場合，多くの話題について主張のまとまりが生成されることになる．これでは，COVID-19 の飲み薬やワクチンなど異なる話題に含まれる安全性に関する共通した主張がグループ化されるため，読者が興味を持っている出来事の異なる主張の収集に時間を要してしまう.

そこで本研究では，まず世界の記事を出来事の類似度を用いてグループ化し，同じグループの記事群から主張の類似度を用いて再度グループ化することによって，類似した出来事の異なる主張の文を提示する手法を提案する．提示した主張の文に加え，文の抽出元である記事を推薦する手法についても検討した.

1.2　本論文の構成

本論文は全 7 章で構成される．第 2 章では本研究で用いる知識と技術について説明する．第 3 章では本研究と比較する関連研究を紹介し，第 4 章では関連研究を踏まえた上で提案手法について説明する．その後，第 5 章で提案手法を評価するシステムの実装方法について説明し，第 6 章で説明したシステムを用いた評価実験について述べる．最後に，第 7 で本研究の全体を踏まえたまとめと今後の課題を述べる．

第2章　本研究で用いる知識と技術

2.1　ニュース推薦システム

多くの Web ニュースサイトでは，様々なアルゴリズムを用いて読者の趣向に合わせた記事が提示されている．アルゴリズムの例として，読者が閲覧した記事に似た記事を推薦するコンテンツベースフィルタリング（Content-Based Filtering）や似た趣向を持つ他の読者の閲覧記事を推薦する協調フィルタリング（Collaborative Filtering），この 2 つや他のルールを併用したハイブリッドな手法などがある [5]．このような読者にパーソナライズした推薦アルゴリズムをニュース推薦システム（News Recommender Systems）という．アルゴリズムの評価指標には，記事の閲覧回数，閲覧記事のカテゴリ，読者の位置情報，他サイトの閲覧履歴や購買情報などがある．

2.1.1　ニュース推薦システムが生むバイアス

パーソナライズするニュース推薦システムは，読者が得る情報に偏りを生む．本節 2.1.1 では，代表的な問題としてフィルターバブル問題とエコーチェンバー問題について記す．本研究は，両問題の部分的な緩和を目指すものである．

2.1.1.1　フィルターバブル問題

Eli Pariser は 2011 年に，インターネットコンテンツの推薦アルゴリズムにパーソナライズ機能を組み込むことで生じる諸問題をフィルターバブル問題として提

唱した [2][3]．Pariser は，インターネット上で個人の趣向に合わせた限定されたコンテンツばかりが推薦されている状況を，ユーザーが泡の中に閉じ込められたような状態であると例えた．例として，Facebook でユーザーが支持する政党ばかりが推薦され，Google 検索でユーザーの居住地域と関係が薄い時事問題が推薦されるような状況が問題視されている．泡の中の偏った情報は，新しいアイディアや異なる視点を生みにくくする．市民が偏った情報ばかりに触れることで，民主主義が機能しなくなる可能性もある．

この泡の中で多様性のある情報を求めることは難しい．なぜなら，泡の中の情報は個人のキャリアや行動履歴によって決まるため，個人で推薦アルゴリズムをコントロールできないことが多いからである．同じ理由で，個人が泡の外の情報を特定できないことも大きな問題である．

2.1.1.2　エコーチェンバー問題

SNS 上で価値観の似た者同士が交流し，共感し合うことにより，偏った意見や思想が反響室のように増幅されて影響力をもつ問題をエコーチェンバー問題という [3] [6]．Conover らや笹原和俊らは，アメリカの選挙期間中の Twitter ユーザーが自身の支持する政党に関するツイートを多くリツイートしており，エコーチェンバー問題が生じていることを確認した [7][8]．

エコーチェンバーの中では，SNS ユーザーは自身と異なる価値観や考え方を持つユーザーと交流する機会を失い，偽の情報を訂正する情報を得にくくなってしまう問題もある [8]．近年のニュースは読者でコメントを交わす SNS のような性質を有しており [9]，エコーチェンバーによる情報の偏りを生むと考えられる．

2.2 機械学習の基礎

機械学習は，コンピュータにデータを学習させるコンピュータープログラミングに関する科学技術である [10]．ここでいう「学習」は，コンピュータに与えたタスクについて，その評価指標が向上するようなコンピュータープログラミムを実行することを指す．機械学習を用いることで，既知のアルゴリズムが無い問題を解決したり，データの予想外な傾向を発見したりすることができる．また，僅かに異なる複数のデータごとにアルゴリズムを用意せずとも実行できる強みをもつ．以下では，本研究に関連する機械学習に関する知識について記す．

2.2.1 教師あり学習

教師あり学習は，データにラベルと呼ばれる出力の答えの情報を付与する機械学習である [10]．ラベルは人間の関与の基に設定されることが多い．

本研究で行うクラス分類は教師あり学習のひとつである．クラス分類ではクラスのラベルを基にデータの特徴とクラスの関係を学習し，新規にモデルに入力されたデータがどのクラスに属するかの確率を出力する．

2.2.2 教師なし学習

教師なし学習は，データにラベルを付与しない機械学習である [10]．人間が関与した答えを用いずにデータの傾向を分析する．

本研究で行うクラスタリングは教師なし学習のひとつである．クラスタリングでは，データの特徴を基に似たデータ同士をクラスに振り分ける．データの何を特徴とし，何を基に似たデータと判断するかが重要で，目的によって工夫する必要がある．

2.3　ニューラルネットワーク

図 2.1 に示すニューラルネットワークは，動物の脳細胞からヒントを得た機械学習の基本的なモデルである [10]．ニューラルネットワークは式 (2.2) に示すような複数の入力から 1 つ以上の出力を得る関数や，この関数をいくつか合成させた式 (2.1) に示すような関数で表される．図 2.1 の個々の円をノード，左端のモデルの入力を受け取るノード群を入力層，右端のモデルの出力を担うノード群を出力層と呼ぶ．また，入力層と出力層の間に位置する縦 1 列のノード群を中間層（隠れ層）という．中間層のうち，前後の層の全てのノードと接続する層を全結合層という．

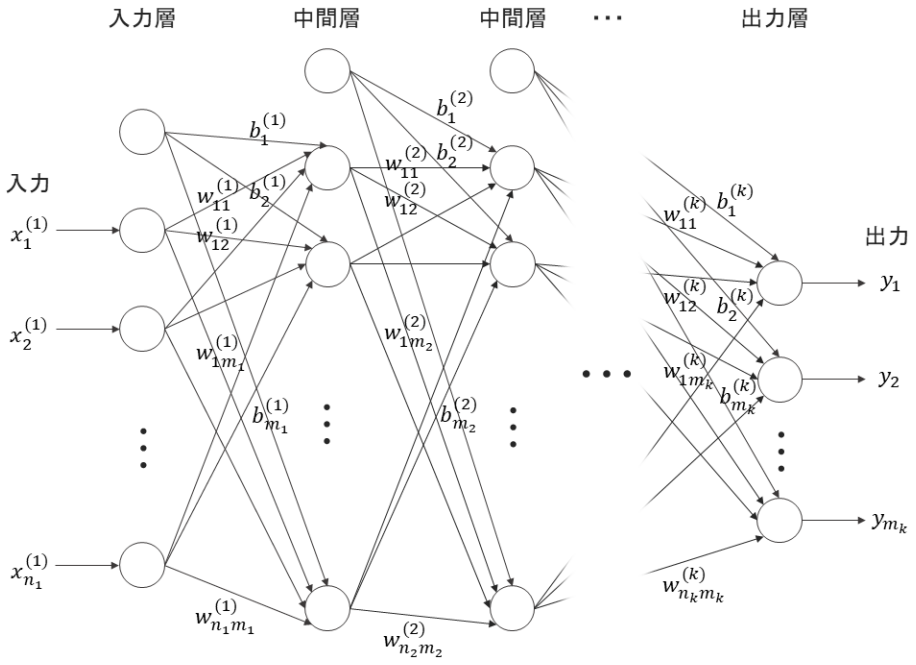

図 2.1: ニューラルネットワークのモデルアーキテクチャ

$$\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{m_k} \end{bmatrix}$$

$$= \left(\boldsymbol{h}_{\boldsymbol{W}^{(k)},\, \boldsymbol{b}^{(k)}} \circ \boldsymbol{h}_{\boldsymbol{W}^{(k-1)},\, \boldsymbol{b}^{(k-1)}} \circ \cdots \circ \boldsymbol{h}_{\boldsymbol{W}^{(1)},\, \boldsymbol{b}^{(1)}}\right)\left(\boldsymbol{x}^{(1)}\right) \tag{2.1}$$

$$\boldsymbol{h}_{\boldsymbol{W}^{(k)},\, \boldsymbol{b}^{(k)}}\left(\boldsymbol{x}^{(k)}\right) = \boldsymbol{\phi}^{(k)}(\boldsymbol{x}^{(k)}\boldsymbol{W}^{(k)} + \boldsymbol{b}^{(k)}) \tag{2.2}$$

$$\boldsymbol{x}^{(k)} = \begin{bmatrix} x_1^{(k)} & x_2^{(k)} & \cdots & x_{n_k}^{(k)} \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{W}^{(k)} = \begin{bmatrix} w_{11}^{(k)} & w_{12}^{(k)} & \cdots & w_{1m_k}^{(k)} \\ w_{21}^{(k)} & w_{22}^{(k)} & \cdots & w_{2m_k}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_k1}^{(k)} & w_{n_k2}^{(k)} & \cdots & w_{n_km_k}^{(k)} \end{bmatrix} \tag{2.4}$$

$$\boldsymbol{b}^{(k)} = \begin{bmatrix} b_1^{(k)} & b_2^{(k)} & \cdots & b_{m_k}^{(k)} \end{bmatrix} \tag{2.5}$$

$\boldsymbol{x}$ は数値化したデータの特徴を示す特徴量，$\boldsymbol{W}$ はノード間の関係の強さを示す接続重み，$\boldsymbol{b}$ は一般に 1 を要素とするバイアスニューロンと呼ばれるものである．

ニューラルネットワークは脳細胞と似たように，個々のノード間の関係の強さを調節して機能する．具体的には，式 (2.6) に示すように次節 2.3.1 で説明する損失関数 L を利用して次の学習時の接続重みを更新する．

$$w_{i,j}^{(next\ step)} = w_{i,j} - \eta \frac{\partial L}{\partial w_{i,j}} \tag{2.6}$$

$$\eta = Const. \tag{2.7}$$

種々のタスクに特化した機械学習モデルは，このニューラルネットワークの各ノードの接続方法や追加の関数を工夫して作成される．

2.3.1 損失関数

損失関数は，機械学習モデルの出力層の出力と正解の出力との誤差を表す関数であり，モデルの学習ごとの性能指標として用いられる [10]．本研究のクラス分類では，2 クラスの分類によく用いられる損失関数として式 (2.8) に示すバイナリクロスエントロピー（BCE）を用いた．

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\{\ y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\ \} \tag{2.8}$$

$$y_i = \begin{cases} 0 \\ 1 \end{cases} \tag{2.9}$$

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_m \end{bmatrix} = h_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{x}^\top\theta\right) \tag{2.10}$$

$$\sigma(t) = \frac{1}{1+\exp(-t)} \tag{2.11}$$

y_i は入力する特徴 $\boldsymbol{x}$ の個々の要素のラベルであり，$\hat{\boldsymbol{y}}$ は特徴 $\boldsymbol{x}$ の個々の要素についてモデルが予測したラベルが 1 となる確率である．バイナリクロスエントロピー $L(\theta)$ は個々の y_i と $\hat{y}_i$ の差が大きいほど大きくなり，この差が小さいほど 0 に漸近する．式 (2.10) の θ は特徴 $\boldsymbol{x}$ のどの要素を重視するかを設定する重み係数である．式 (2.11) の $\sigma(t)$ はシグモイド関数と呼ばれる連続関数であり，実数 t を入力したとき図 2.2 のように確率の範囲 $(0,1)$ で実数を出力する．シグモイド関数を用いることで，$(0,1)$ の範囲にないモデルの出力を $(0,1)$ の範囲に落とし込むことができる．

ニューラルネットワークの重みの更新は損失関数を基に行われ，勾配降下法や誤差逆伝播法を用いて効率よく計算される．

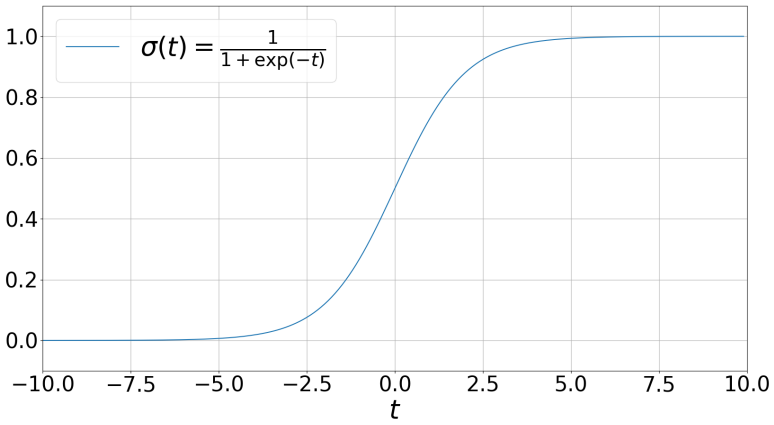

図 2.2: シグモイド関数

2.4 機械学習の文章データへの応用

日々の気温や株価，Web サイトのアクセス数のように，時刻ごとに計測されるデータを時系列データという [10]．上記の例では，過去のデータの傾向から未来のデータを予測することができる．特に，予測したい時刻 t のデータの直前のデータは，時刻 t のデータに類似していたり時刻 t 周辺のデータの変化率の情報を含んでいたりするため，予測に大きく役立つ．

時刻を文章内の単語の順番に対応させると，文章データも時系列データと見なすことができる．文章内の任意の単語は，その前後の単語の意味や文法の傾向から予測することができる．特に，予測したい単語の直前・直後の単語は，単語の予測の大きなヒントとなる．

本節 2.4 では，文章をはじめとした時系列データを分析するための機械学習の応用例について記す．

2.4.1 単語埋め込みと文埋め込み

データのカテゴリを表現する訓練可能な密なベクトルを埋め込みという [10]．特に，単語や文を表現しようとする埋め込みをそれぞれ単語埋め込み，文埋め込みと呼ぶ．

単語埋め込みはデフォルトでは無作為な数値で初期化される．初期化した単語埋め込みは，意味が似た単語の埋め込み同士は距離を小さくするなど，何らかのアルゴリズムで表現を学習していく．学習した表現が適切であるほど，単語埋め込みを利用した自然言語処理の機械学習タスクはより精度が増す．埋め込みの次元数は，目的やモデルの構造によって調節する必要がある．

2.4.2 テキストの前処理

単語埋め込みのように，自然言語処理ではテキストをコンピュータが解釈しやすい数値に変換して処理することが多い．このとき，コンピュータはテキストを文字の羅列としか見ていないため，Apple と apple を別の単語と見なし，違う数値に変換してしまう．このような単語の表記揺れは，文章中の単語の出現頻度の情報を不正確なものにする．表記揺れによって増加した出現頻度の少ない単語が学習のノイズとなってしまうこともある．従って，テキストに何らかの処理をする前に，大文字を小文字に変換するなどの表記揺れを解消する処理を行う必要がある．URL やメールアドレスといったユニークな文字の羅列は，出現頻度が極めて少ないノイズとなり得るため，事前に除去しておくと良い．

2.4.3 Attention

Bahdanau らが 2014 年に提案した Attention は，時系列データの機械学習を行う際，出力データと関わりが強い入力データに効率よく比重を与える仕組みである [10][11]．モデルに Attention を組み込むことで無駄なデータの学習が減り，モ

デルが過去に学習した内容を忘却しにくくなる．これにより，30 語以上の長い文章の機械翻訳タスクの精度が大幅に向上する．

図 2.3 に機械翻訳モデルに組み込まれた Attention 機構を示す．

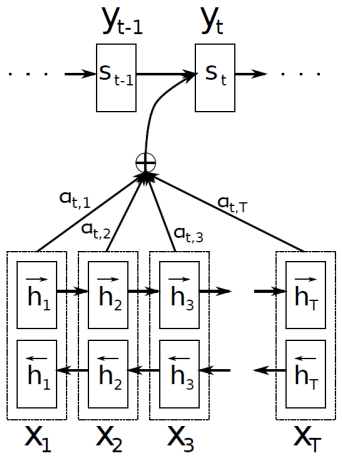

図 2.3: 入力単語群 $(x_1, x_2, ..., x_T)$ を基に t 番目の出力語 y_t を生成する Attention のモデルアーキテクチャ[1]

直前に翻訳した単語 y_{t-1} と過去に翻訳した単語群の情報 s_{t-1} に，Attention の式 (2.12) を考慮して次の単語 y_t を予測している．$\vec{h_i}$ は文脈を加味した単語の意味情報をもつベクトルであり，BiGRU と呼ばれるニューラルネットワークを用いて生成されている．

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.12}$$

$$\alpha_{ij} = \underset{j}{\mathrm{softmax}}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.13}$$

$$e_{ij} = a(s_{i-1}, h_j) = v_a^\top \tanh(W_a s_{i-1} + U_a h_j) \tag{2.14}$$

式 (2.13) はソフトマックス関数と呼ばれる関数に式 (2.14) を代入したものであり $\sum_j \alpha_{ij} = 1$ となる意味ベクトル h_j の重み係数を表す．式 (2.14) は，過去に翻訳

[1] [11] より引用

した単語群の情報 s_{t-1} と入力単語の意味ベクトル h_j を基に，どの単語に比重を置くかを決定するニューラルネットワークである．式 (2.14) の $v_a^\top$, W_a, U_a はニューラルネットワークの重みである．

2.4.4 Transformer

Vaswani らが 2017 年に提案した Transformer は，Attention を組み込んだ翻訳タスクに用いられる機械学習モデルで，2017 年の最先端のモデルと比べて数分の 1 のコストで最大級の精度を有する [10][12]．時系列データを扱う従来の機械学習には RNN（Recurrent Neural Network）が多く組み込まれていたが，このモデルは単語を逐次的に処理する仕組みになっており，並列演算が難しい．一方で Transformer は，RNN を組み込まずに並列演算が可能な Attention を多く組み込んで時系列データを処理するため，処理コストが低い．

Transformer は，文章を文意を表すベクトルに変換するエンコーダ部分と，そのベクトルを用いて目的の文章を生成するデコーダ部分に分かれるエンコーダ・デコーダモデルの一種である．図 2.4 に示す Transformer のうち，左半分がエンコーダ，右半分がデコーダである．エンコーダの入力は翻訳前の文章であり，デコーダの入力はエンコーダの出力と翻訳後の文章である．デコーダの出力は翻訳中の文章の次の単語の予測確率である．なお，デコーダに入力される翻訳後の文章は，モデルの学習の際には全文が渡され，予測の際には翻訳中の単語群が渡される．

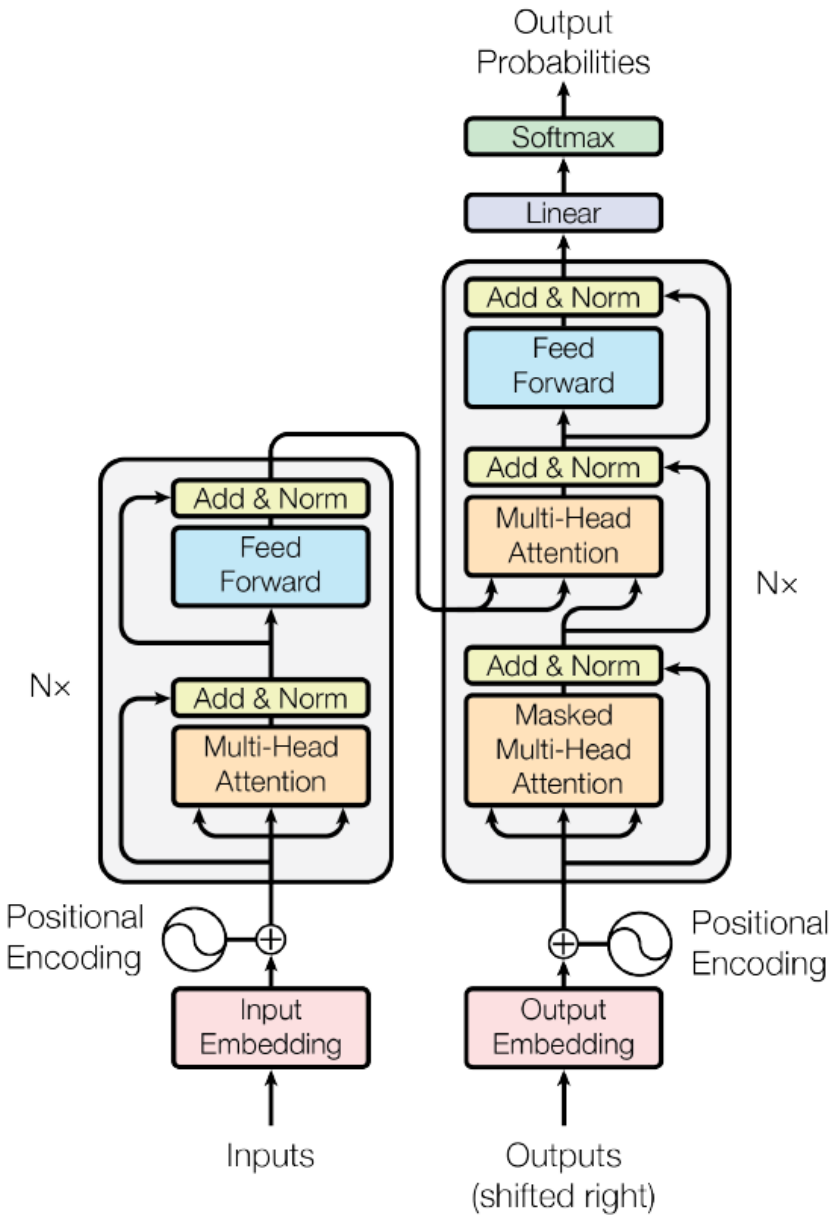

図 2.4: Transformer のモデルアーキテクチャ[2]

Transformer の学習コストを低減する所以となる Attention は，図 2.5 のように工夫されて組み込まれている．

[2][12] より引用

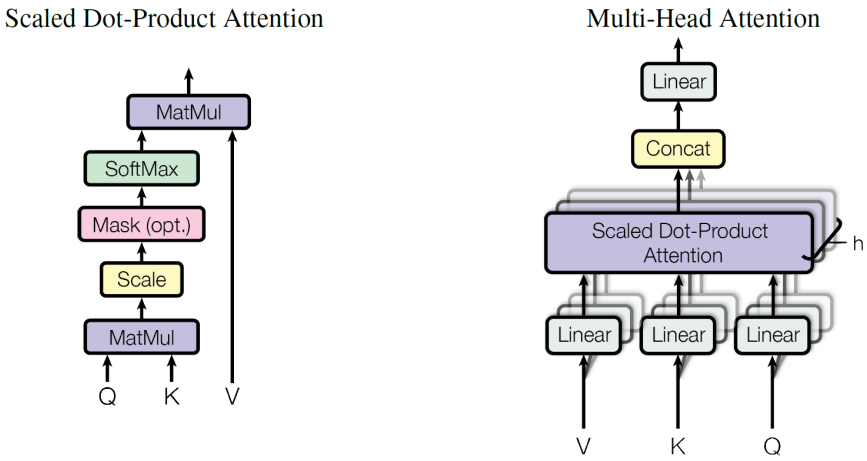

図 2.5: Transformer を構成する Multi-Head Attention（右）とそれを構成する Scaled Dot-Product Attention（左）のモデルアーキテクチャ[3]

Scaled Dot-Product Attention は式 (2.15) で表される．Q（Queue）は入力を意味し，式 (2.15) は Q と K（Key）との類似度を基にした重みづけによって出力 V（Value）を調節するような効果をもつ．

Multi-Head Attention は式 (2.16) で表される．式 (2.16) を構成する式 2.15) の Q, K, V は同値であり，Transformer のデコーダへの入力文を Output Embedding と Positional Encoding で加工したもの（X とする）を表す．これらの Q, K, V に W_i^Q, W_i^K, W_i^V を作用させて役割を変えている．QW_i^Q は X のどの部分を処理するかを表し，KW_i^K は X の注目の仕方を表し，VW_i^V は出力の様子を調整する役割を担う．

$$\text{Attention}\ (Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.15}$$

$$\text{MultiHead}\ (Q, K, V) = \ \text{Concat}\ (\ \text{head}_{\ 1}, \ldots,\ \text{head}_{\ \text{h}})\, W^O \tag{2.16}$$

$$\text{head}_{\ \text{i}} = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.17}$$

[3][12] より引用

2.4.5 BERT

Liu らが 2019 年に提案した BERT(Bidirectional Encoder Representations from Transformers) は，文の意味や文脈を加味した自然言語処理を行うための機械学習モデルである [10][13]．研究者が大規模な計算資源を用いて自然言語の特徴を学習した BERT モデルを公開しており，学習されたモデルに用途に合わせた追加の学習を行うことで，小規模な計算資源で様々な自然言語タスクを高精度で行うことができる．このような前段階での学習を事前学習，その後の用途に合わせた学習を転移学習と呼ぶ．

図 2.6 に中間層が 1 つの BERT のモデルアーキテクチャを示す．入力と出力はともに次元が等しいベクトルである．中間層と出力層には Transformer のエンコーダ部分が用いており，文から単語の意味を効果的に学習することができる．また，個々の Transformer エンコーダが前の層の全てのノードから値を受け取っており，文を前から後ろに読む文脈情報と後ろから前に読む文脈情報の両方を加味した学習が可能である．BERT モデルは，中間層の数や個々の Transformer の数，Transformer 内の Attention の数などを調整し，種々の自然言語処理タスクの性能を上げている．

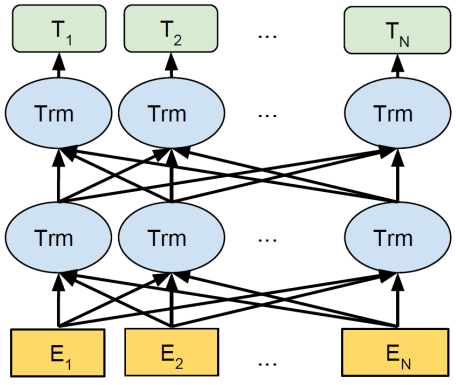

図 2.6: 中間層が 1 つの BERT のモデルアーキテクチャ[4]．E は 3 万単語の語彙をもつ WordPiece embeddings を用いた単語などの埋め込み，Trm は Transformer エンコーダ，T は出力されるベクトルを表す．

[4][13] の図より一部抜粋

図 2.7 に BERT の事前学習と転移学習の手順を示す．事前学習で入力するベクトルの要素は，分類のための出力の次元調整に用いる定ベクトル [CLS]，入力する 2 文の単語埋め込み，2 文の分割位置を示す定ベクトル [SEP] で構成される．図中の左の事前学習では，文法や単語の意味の意味を理解するための MLM（Masked Language Model）としての学習と，文意と文脈を理解するための NSP（Next Sentence Prediction）としての学習を同時に行う．

MLM としての学習では，まずモデルに入力する 2 文の単語の 15%のうち，80%を文字列 [MASK] に置き換え，10%を無作為に抽出した単語に置き換える．残りの 10%は何も置き換えない．この置き換えにより，事前学習でしか入力しない文字列 [MASK] について，事前学習と転移学習とのミスマッチを低減することができる．その後，出力されたベクトルの入力でマスクされた位置と同じ位置の要素を使用し，マスク前の単語がどの単語であったかを予測する単語ごとの確率を計算して出力する．出力した確率とマスクした正解の単語を基に損失関数を計算し，モデルの重みの更新を行う．

NSP としての学習では，入力文の 50%を文章中で連続する 2 文，残りの 50%を文章中で連続しない無作為に抽出された 2 文として入力する．入力する 2 文には，文章中で連続した 2 文であるかそうでないかのラベル付けがなされている．その後，出力されたベクトルの [CLS] と同じ位置の要素 C とラベルを使用してどちらの 2 文であったかの損失関数を計算し，モデルの重みを更新する．

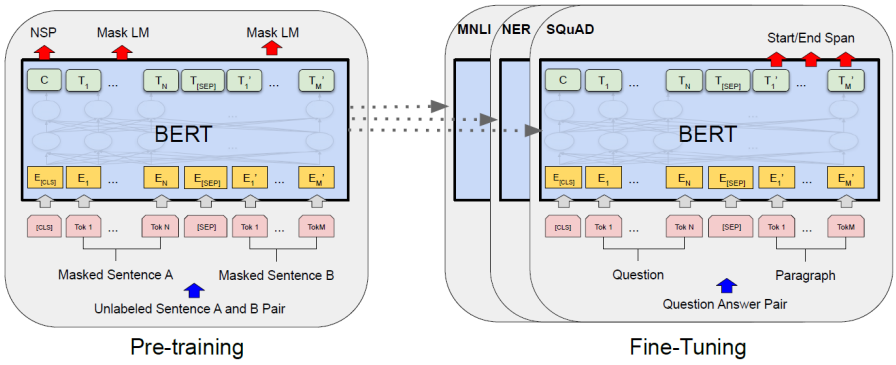

図 2.7: BERT の事前学習と転移学習の手順[5]

事前学習の後は, MNLI（Multi-Genre Natural Language Inference), NER（Named Entity Recognition)，SQuAD（The Stanford Question Answering Dataset）などの行いたい自然言語処理タスクに合わせて入力と使用する出力の要素を変える．図 2.8 に 1 文のクラス分類のための BERT の転移学習のモデルアーキテクチャを示す．この学習では文字列 [CLS] と分類したい 1 文を入力し，[CLS] に対応する出力 $\boldsymbol{C}$ と正解のラベルを基に損失関数の計算を行っている．

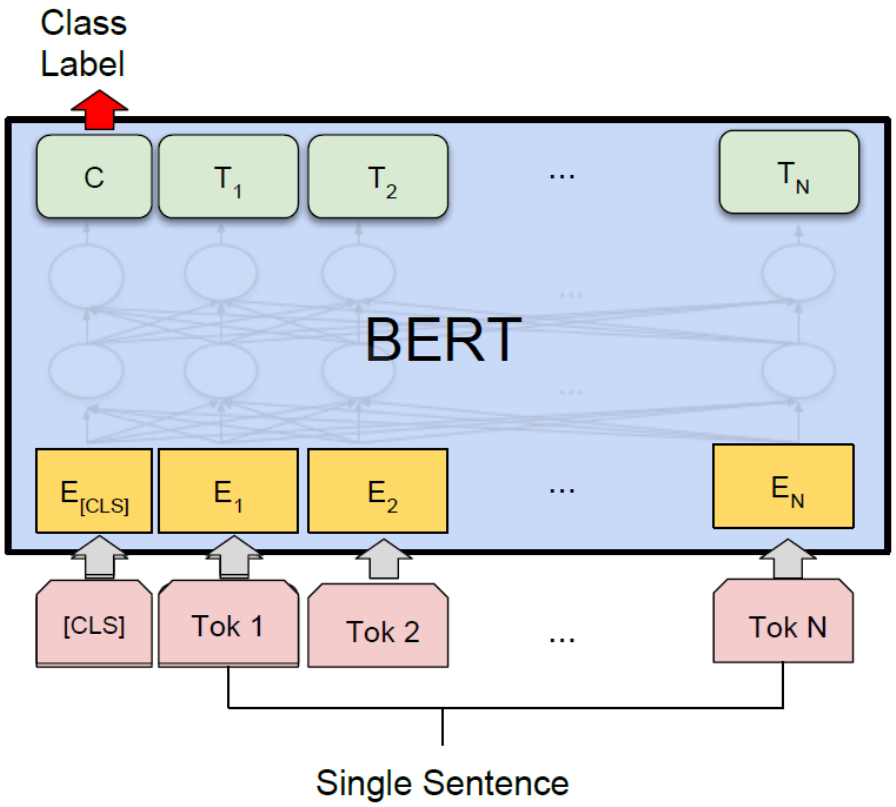

図 2.8: 1 文のクラス分類のための BERT の転移学習[6]

[5] [13] より引用
[6] [13] より引用

2.4.6 RoBERTa

Devlin らが 2019 年に提案した RoBERTa（Robustly Optimized BERT Pretraining Approach）は BERT の学習方法を改善した機械学習モデルで，多くの自然言語処理タスクのベンチマークで BERT のスコアを上回る．RoBERTa のモデルアーキテクチャは図 2.6 の BERT と同じものを使用する．MLM としての学習で BERT では 10 回学習するごとに同じマスクを使用していたため，どの学習でも異なるマスクを使用することで SQuAD（Stanford Question Answering Dataset）と SST-2（The Stanford Sentiment Treebank）のスコアを向上させた．BERT が行っていた NSP としての学習は比較実験により有効でないことが確認され，RoBERTa ではこれを行っていない．比較実験では，BERT が行った SEGMENT-PAIR の NSP の学習の代わりに SENTENCE-PAIR の NSP の学習，FULL-SENTENCES の NSP をしない学習，DOC-SENTENCES の NSP をしない学習を行い，NSP の学習を行わなくても SQuAD, SST-2, MNLI-m（The Multi-Genre Natural Language Inference Matched），RACE（Reading Comprehension）のスコアが概ね向上することを確認した．BERT では学習データとして 250 万単語の English Wikipedia と 80 万単語の BooksCorpus で計 16GB のデータを用いていたが，RoBERTa ではこれに加えて 76GB の CC-News，38GB の OpenWebText，31GB の Stories を学習した．さらに，1 回の学習で用いるデータサイズ（バッチサイズ）を 256 から 8,000 に増やすことで，BERT よりも SQuAD，MNLI-m，SST-2 のスコアが向上することを確認した．

表 2.1 に BERT と RoBERTa の GLUE（General Language Understanding Evaluation）タスクのベンチマークスコアを示す．それぞれのベンチマークでは以下のタスクを行っている．

- MNLI（The Multi-Genre Natural Language Inference）
  - 2 つの文が含意，矛盾，中立のどの関係にあるかを判定

- QNLI（Question Natural Language Inference）
  - 質問文とともに入力した文が質問に対する正しい回答であるかを判定
- QQP（Quora Question Pairs）
  - 2 つの質問文が同じ意味であるかを判定
- RTE（Recognizing Textual Entailment）
  - 2 つの文が含意関係にあるかを判定
- SST（The Stanford Sentiment Treebank）
  - 映画のレビュー文がポジティブであるかネガティブであるかを判定
- MRPC（Microsoft Research Paraphrase Corpus）
  - 2 つの文が同じ意味であるかを判定
- CoLA（The Corpus of Linguistic Acceptability）
  - 英文の文法が正しいかを判定
- STS（Semantic Textual Similarity Benchmark）
  - ニュースの 2 つの見出し文の類似度を 5 段階で評価

本研究におけるクラス分類ではSTSのスコアが重要であり, このスコアはRoBERTaの学習方法により 90.0 から 92.4 に向上している.

表 2.1: BERT と RoBERTa の GLUE タスクのベンチマークスコア. [7] 全ての結果は 24 層のアーキテクチャを使用し，5 回の実行結果の中央値を取っている.

	MNLI	**QNLI**	**QQP**	**RTE**	**SST**	**MRPC**	**CoLA**	**STS**
$\text{BERT}_{\text{LARGE}}$	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0
RoBERTa	**90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**

2.5 教師あり学習の評価

教師あり学習は，訓練データ（教師データ）と呼ばれる特徴を学習したデータとは別に，学習に用いていないテストデータと呼ばれるデータで入出力の評価を行う必要がある [10]．この評価は，モデルに未知のデータを入力したときに目的の出力を得るために必要な操作である．

訓練データで目的の出力を得られてもテストデータで目的の出力が得られないとき，モデルは過学習（過剰適合）しているといい，モデルは未知のデータに対応できていない．過学習は，モデルの学習回数が多く訓練データの特徴を学習しすぎたときや，学習の手法が適切でないときに起こり得る．一方で，訓練データでも目的の出力を得られていないとき，モデルは過少適合しているという．過少適合はモデルの学習回数が少なく訓練データの特徴を学習できていないときや，学習の手法が適切でないときに起こり得る．目的の出力が得られているかどうかは，学習ごとの損失関数の出力の大きさから確認することができる．

同じデータでテストデータの評価を行いたいときは，データを訓練データとテストデータに分割して評価する．例えば，100 件のデータの 80 件を訓練データとして学習し，学習に用いていない 20 件のデータをテストデータとして評価を行う．

2.5.1 混同行列を用いた分類器の評価

分類器の評価では，混同行列を用いたいくつかの評価指標がよく用いられる [10]．混同行列は，分類器の入力の n 種類のラベルと出力の n 種類のラベルの計 n^2 組の組み合わせについて，それぞれの数を要素とした n 次正方行列である．

[7] [14] より一部抜粋

表 2.2 に 10 件のデータを 0 か 1 かの 2 値ラベルで分類したときの混同行列の例を示す．評価指標の計算で着目するクラスを陽性クラス（positive class），その他のクラスを陰性クラス（negative class）といい，表 2.2 では 1 を陽性クラス，0 を陰性クラスとして考える．表 2.2 の 2 件のデータは正しく分類した陰性クラスであり，真陰性（TN; true negative）があるという．表 2.2 の 1 件のデータは誤って陽性クラスだと分類した陰性クラスであり，偽陽性（FP; false positive）があるという．表 2.2 の 3 件のデータは誤って陰性クラスだと分類した陽性クラスであり，偽陰性（FN; false negative）があるという．表 2.2 の 4 件のデータは正しく分類した陽性クラスであり，真陽性（TP; true positive）があるという．以降の小小節 2.5.1.1，2.5.1.2，2.5.1.3，では，これら TN，FN，FP，TP を用いて計算した評価指標である適合率，再現率，マシューズ相関係数について記す．

表 2.2: 混同行列の例

	予測は 0	予測は 1
入力は 0	2	1
入力は 1	3	4

2.5.1.1 適合率

適合率（precision）は，陽性クラスだと分類したデータのうち真陽性クラスであるデータの割合である [10]．表 2.2 の例では，ラベルが 1 だと予測データのうち実際に 1 であったデータの割合を指す．したがって，式 (2.18) のように適合率が計算される．

$$precision \ = \ \frac{TP}{TP + FP} \ = \ \frac{4}{4 + 1} \ = \ 0.8 \tag{2.18}$$

適合率は，数ある映像から子どもに観せても安心な映像を検出する分類器などで重視される．観せて安心な映像を陽性クラスの映像だとしたとき，分類器の適合率が高ければ，観せて安心な映像だと分類した映像の多くは真に観せて安心な映像であることが多い．

2.5.1.2 再現率

再現率（recall）は，陽性クラスのデータのうち真陽性クラスであるデータの割合である [10]．表 2.2 の例では，ラベルが 1 のデータのうち予測も 1 であったデータの割合を指す．したがって，式 (2.19) のように再現率が計算される．

$$recall \ = \ \frac{TP}{TP+FN} \ = \ \frac{4}{4+3} \ \sim \ 0.57 \tag{2.19}$$

再現率は，監視カメラに映る人物の映像から万引き犯の映像を検出する分類器などで重視される．万引き犯の映像を陽性クラスの映像だとしたとき，分類器の再現率が高ければ，万引き犯の映像の多くは予測も万引き犯の映像となる．

適合率と再現率は一般にトレードオフの関係にあり，分類器の目的によってそれぞれの評価指標をどれほど重要視するかを考慮する必要がある．上述の監視カメラの例では，適合率が低く万引き犯だと分類した映像のいくつかが万引き反でない人物となってしまうことよりも，再現率が高く確実に万引き犯を検出できることの方が重要である．

2.5.1.3　マシューズ相関係数

Matthewsが1975年に提案したマシューズ相関係数（MCC;Matthews Correlation Coefficient）は，混同行列を基に分類器の精度を総合的に評価する指標で，TP，TN，FP，FNの数が不均衡なときにも頑健に評価できる [15]．マシューズ相関係数は式 (2.20) のように計算される $[-1, 1]$ の範囲の実数値であり，この値が大きいほど分類器の精度が高いといえる．

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}} \tag{2.20}$$

2.6　文章の距離の算出

後述のクラスタリングを行うにあたり，クラスタリングに必要な文章の距離（非類似度）の算出法について記す．本節2.6では，小節2.6.1で文章を埋め込みに変換するSentence-BERTについて説明し，小節2.6.2で変換した埋め込みを用いて計算するコサイン距離について説明する．

2.6.1 Sentence-BERT

Reimers らが 2019 年に提案した Sentence-BERT は，文の意味や文脈を加味した文埋め込みを生成するための BERT を転移学習した機械学習モデルである [16]. 図 2.9 に Sentence-BERT のモデルアーキテクチャを示す．Sentence-BERT では転移学習済みの BERT に 1 つの文を入力し，出力したベクトル群に pooling と呼ばれる情報の抽出処理を行って文章の埋め込みを得る．転移学習では，埋め込みを応用した STS タスクなどの精度を上げるために，3 種類の工夫された学習が行われている．pooling では 3 つの手法が比較されており，BERT で出力したベクトル群の平均を埋め込みとする手法が最も STS タスクのスコアが優れていた．

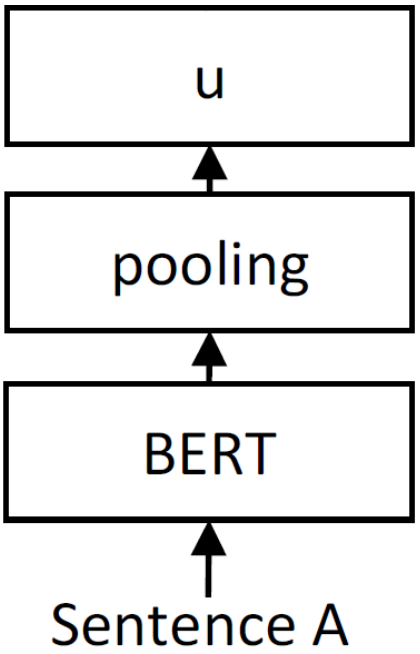

図 2.9: Sentence-BERT のモデルアーキテクチャ[8]

[8][16] の図より一部抜粋

BERT や RoBERTa で 1 万文から最も意味的類似度が高い 2 文を得たいとき，図 2.7 のように 2 文同時に入力すると $\frac{10000(10000-1)}{2} = 49995000$ 回の実行が必要であり，Reimers らの実験では 65 時間の処理を要した．そこで Reimers らは，BERT や RoBERTa に 1 文を入力して単語埋め込みを得る処理を 10000 回行い，埋め込み間の類似度を計算し，処理時間を約 5 秒に短縮した．埋め込み間の類似度には，式 (2.21) に示すコサイン類似度が用いられている．

$$\text{cos-sim}\left(\boldsymbol{u}, \boldsymbol{v}\right) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} \tag{2.21}$$

表 2.3 に SentEval ツールキットを用いた文埋め込みの比較評価を示す．表より，Sentence-BERT は多くの自然言語処理タスクにおいて 2019 年の最先端のスコアを有することがわかる．

表 2.3: SentEval ツールキットを用いた文埋め込みの比較評価[9]

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.0	72.87	81.52
Avg. fast-text embeddings	77.96	79.23	91.68	87.81	82.15	83.6	74.49	82.42
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.8	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	**90.38**	84.18	88.2	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.2**	70.14	85.10
SBERT-NLI-base	83.64	89.43	94.39	89.86	88.96	89.6	**76.00**	87.41
SBERT-NLI-large	**84.88**	**90.07**	**94.52**	90.33	**90.66**	87.4	75.94	**87.69**

[9] [16] より引用

2.6.2 コサイン距離

式 (2.22) にコサイン距離は，2 つのベクトルの距離の指標の 1 つである．2 つの 3 次元ベクトル間のコサイン距離は，2 つのベクトルがなす角の大きさに相当する．コサイン距離は，単語埋め込みや文章の埋め込みの距離の算出によく用いられる．

$$\text{cos-dist}\,(\boldsymbol{u},\boldsymbol{v}) \;=\; 1-\text{cos-sim}\,(\boldsymbol{u},\boldsymbol{v}) \;=\; 1-\frac{\boldsymbol{u}\cdot\boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} \tag{2.22}$$

2.7 階層的クラスタリング

クラスタリングは，データの集合を何らかのアルゴリズムに基づいてクラスタと呼ばれる部分集合に分割する教師なし学習である．類似したデータをクラスタにまとめることで，データの要約や傾向の分析を行うことができる．

本節 2.7 では，小節 2.7.1 でクラスタ間の距離を算出する Ward 法について説明し，小節 2.7.2 でクラスタリングの一種である階層的クラスタリングについて説明する．

2.7.1 Ward 法

階層的クラスタリングの処理には，データの部分集合であるクラスタ同士の距離（クラスタ間距離）が必要となる．クラスタ間距離の算出には単一連結法や最遠隣法など様々な手法が用いられているが，本研究では 1963 年に Ward が提案した Ward 法（最小分散法）を使用する [17]．Ward 法は異常なデータの値（外れ値）の影響を受けにくく，クラスタ間距離の算出法として広く用いられている．

Ward 法では式 (2.23) に示すように，併合後のクラスタ $C_k \cup C_c$ の分散と併合前のクラスタ C_k, C_c のそれぞれの分散の和との差 d_{kc} をクラスタ間距離とする．

$$d_{kc} = \mathrm{Var}\left(C_k \cup C_c\right) - \left(\mathrm{Var}\left(C_k\right) + \mathrm{Var}\left(C_c\right)\right) \tag{2.23}$$

2.7.2 階層的クラスタリング

階層的クラスタリング（凝集型クラスタリング）は，図 2.10 に示すようにクラスタ間距離が最も近い 2 つのクラスタを順次 1 つのクラスタに結合していく手法である．図 2.10 の右側のグラフのように，階層的クラスタリングの様子をツリー状に可視化したものをデンドログラムという．

結合前のクラスタは結合後のクラスタの部分集合となり，最終的には 1 つのクラスタとなる．従って階層的クラスタリングは，このような部分集合の階層構造をもつデータの分析に有用である．どのクラスタ間距離でクラスタの結合を止めるかによって，様々な粒度のクラスタ群を得ることができる．

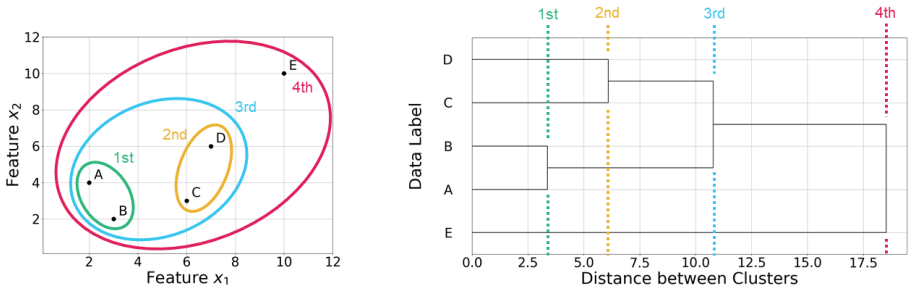

図 2.10: 階層的クラスタリング

2.7.3 シルエット係数

式 (2.24) に示すシルエット係数は，階層的クラスタリングの評価指標のひとつである [18]．全データのシルエット係数の平均 $\overline{Sil(i)}$ が大きいほど同一クラスタ内のデータがより類似し，異なるクラスタ間のデータがより類似していないといえる．

$$Sil(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))} \tag{2.24}$$

$$a(i) = \frac{1}{|C_{\text{in}}| - 1} \sum_{\boldsymbol{x}_j \in C_{\text{in}}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \quad (x_i \in C_{in},\ i \neq j) \tag{2.25}$$

$$b(i) = \frac{1}{|C_{\text{near}}|} \sum_{\boldsymbol{x}_k \in C_{\text{near}}} \|\boldsymbol{x}_i - \boldsymbol{x}_k\| = \min_{C} \left(D(i, C)\right) \tag{2.26}$$

$$D(i, C) = \frac{1}{|C|} \sum_{\boldsymbol{x}_h \in C} \|\boldsymbol{x}_i - \boldsymbol{x}_h\| \quad (x_i \in C_{in},\ C \cap C_{in} = \emptyset) \tag{2.27}$$

式 (2.25) に示す $a(i)$ はデータ $\boldsymbol{x}_i$ と $\boldsymbol{x}_i$ が属するクラスタ C_{in} 内の他の全データ $\boldsymbol{x}_j \in C_{in}$ との距離の平均である．$a(i)$ が 0 に近いほどクラスタ C_{in} 内のデータがよく類似しているといえる．

式 (2.26) に示す $b(i)$ はデータ $\boldsymbol{x}_i$ と $\boldsymbol{x}_i$ の最近傍のクラスタ C_{near} 内の他の全データ $\boldsymbol{x}_k \in C_{near}$ との距離の平均である．具体的には，$D(i, C)$ をクラスタ C_{in} 以外のあるクラスタ C についてデータ x_i とクラスタ C 内の全データ $\boldsymbol{x}_h \in C$ との距離の平均だとしたとき，$b(i)$ は任意の C における $D(i, C)$ の最小値として計算される．$b(i)$ が 1 に近いほど異なるクラスタのデータが類似していないといえる．

2.8 k-NN 分類法

k-NN 分類法（k 近傍法）は，クラスが既知のデータ群が存在するときに新規に入力したデータのクラスを推定するためのクラス分類手法である [10]．新規のデータが入力されたとき，そのデータとの距離が最も近い既存のデータを順に k 個選択し，k 個のデータのクラスのうち最も多かったクラスを新規のデータのクラスと推定する．k は事前に設定しておく必要がある．

第3章　関連研究

3.1　事前学習済みのBERTを用いたニュース推薦を行う研究

3.1.1　UNBERT: User-News Matching BERT for News Recommendation

Zhangらは，日々追加される新しい記事から読者の関心度が高い記事を推薦するため，BERTを用いたニュース推薦手法を提案した[19]．LSTUR，FIM，NAML，NRMS，NPA，DKNなどのニューラルネットワークを用いた既存のニュース推薦手法は，学習していない特徴を持つ新しい記事が入力されたときに類似した記事を推薦できない問題をもつ．この問題はコールドスタート問題と呼ばれ，ニュース推薦システムの大きな課題となっている．

そこでZhangらは，豊富な言語知識を事前学習したBERTモデルを利用し，コールドスタート問題に強いニュース推薦のための機械学習モデルとして，図3.1，3.2に示すUNBERTを提案した．Zhangらの実験では，学習済みの古い記事と未学習の新しい記事が読者が連続して読んだものであるかを分類し，既存手法と分類性能を比較した．実験の結果，図3.3に示すようにUNBERTが先に述べたどの既存手法よりも優れた性能で分類を行うことができた．

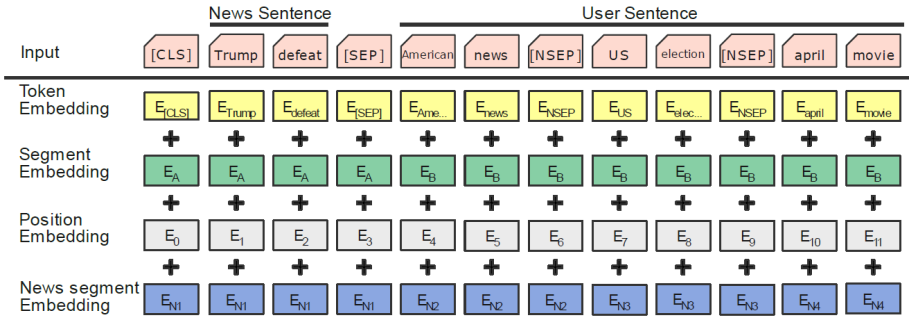

図 3.1: UNBERT の入力．[10] 新規の記事の文と読者が閲覧した複数の記事の文を文字列 [CLS]，[SEP]，[NSEP] で分割して入力する．

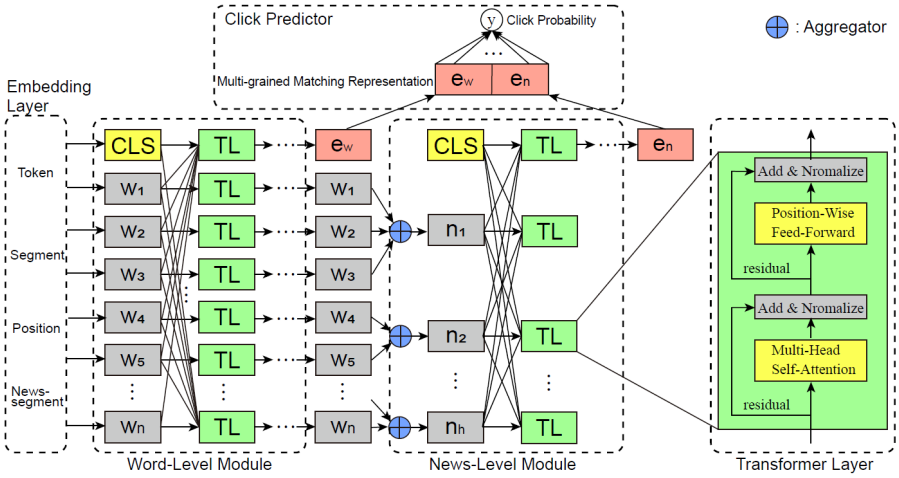

図 3.2: UNBERT のモデルアーキテクチャ．[11] 単語レベルの特徴の学習と文全体の特徴の学習で 2 つの BERT モデルを使用している．また，文字列 [CLS] に対応する位置の出力を使用し，入力した 2 種類の記事が連続して読まれたものであるか確率を予測する．

[10] [19] より引用
[11] [19] より引用

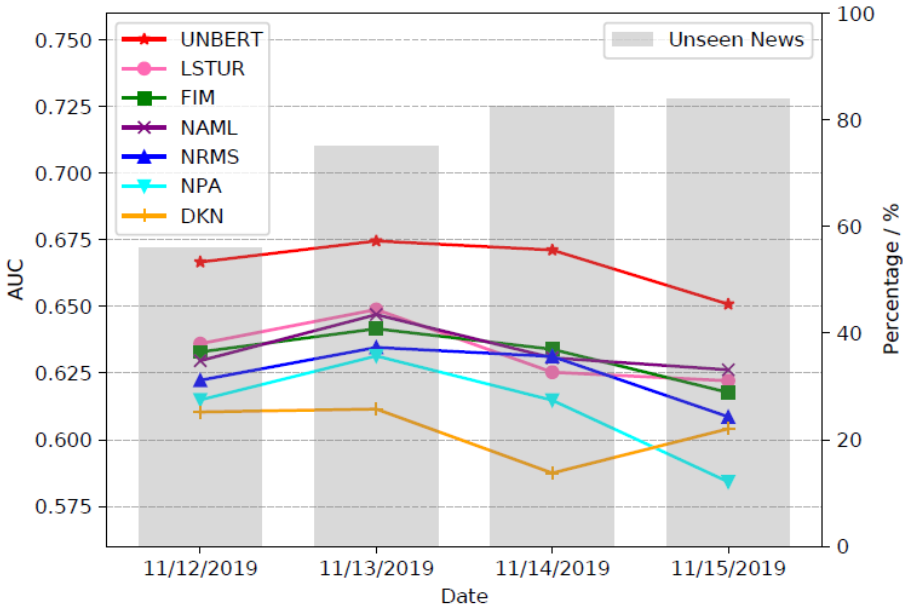

図 3.3:　各手法の新しい記事の分類性能．[12] 学習と分類のためのデータセットには MIND（Microsoft News Dataset）を使用した．各手法のモデルは 2019 年 11 月 19 日から 2019 年 11 月 11 日の記事を学習している．分類性能の評価指標には分類の適合率と再現率を用いた AUC と呼ばれる評価指標を使用した．

Zhang らの研究から，BERT が事前学習した自然言語の特徴が記事を入力とするタスクに応用可能であることがわかる．本研究では，この自然言語の特徴を記事の文の分類タスクと文埋め込みを生成するタスクに応用する．

3.2　ニュース推薦システムのバイアスを解決する研究

3.2.1　Understanding and Controlling the Filter Bubble through Interactive Visualization: A User Study

Nagulendra らは，ニュースや SNS（ソーシャルネットワークサービス）の利用者にフィルターバブルの存在の自覚を促すために，図 3.4 に示すようなフィルター

[12] [19] より引用

バブルを可視化する手法を提案した [9]．図のシステムでは，ある SNS 投稿者の投稿のうち，SNS が閲覧者に推薦している投稿のカテゴリと推薦していない投稿のカテゴリとの境界を可視化している．

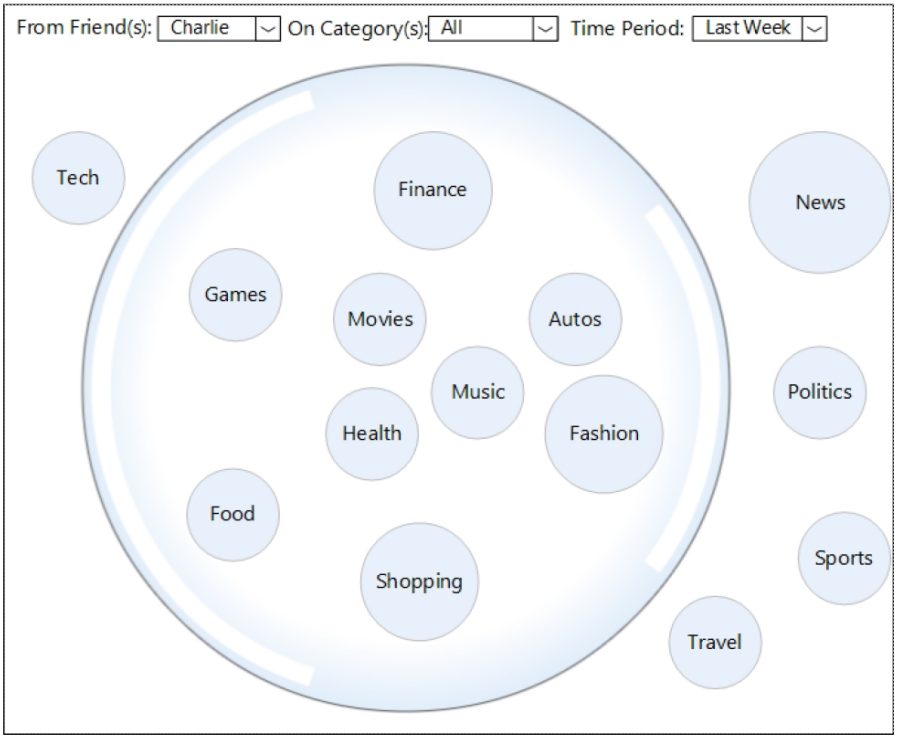

図 3.4: フィルターバブルを可視化するシステム[13]

しかしこの手法では画面上に限られた数のカテゴリしか表示できない．また，このシステム自体があるアルゴリズムに基づいて投稿を選別しており，フィルターバブルを生成してしまっている．したがって，このシステムのようにユーザインタフェースを用いて記事を推薦してしまうとフィルターバブルを生成してしまい，読者が得る情報に偏りを生んでしまうと考える．そこで本研究では，記事の推薦にユーザインタフェースを直接は使用せず，記事や主張の文をクラスタで提示することを考える．クラスタで提示することで，階層的に並ぶ主張の文の違いを読者の手で選別させることができる．

[13] [9] より引用

3.3　主張の文のクラスタリングを行う研究

3.3.1　Scalable Fact-checking with Human-in-the-Loop

Yang らは，ニュース読者が出来事を正確かつ迅速に把握できるように，階層的クラスタリングを用いて記事に対するツイートの主張のグループ化と要約を行う手法を提案した [4]．図 3.5 に提案手法の概略を示す．この研究では COVID-19 の話題に限定した主張の文をグループ化しているが，この手法を記事に適用した場合，多くの話題について主張のまとまりが生成されることになる．これでは，COVID-19 の飲み薬やワクチンなど異なる話題に含まれる安全性に関する共通した主張の文がグループ化されるため，読者が興味を持っている出来事の異なる主張の収集に時間を要してしまう．

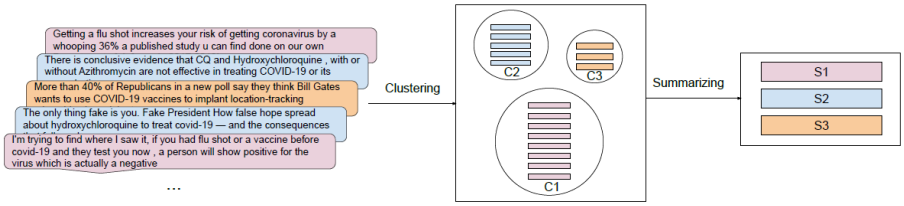

図 3.5:　記事に対するツイートの主張のグループ化と要約[14]

Yang らのシステムでは，959 件の記事に対する 28818 件のツイートを収集し，これらのツイートをクラスタ間距離 0.85 を閾値としてクラスタに分割していた．このとき，全データのシルエット係数の平均が 0.79 となる 705 個のクラスタを生成しており，記事の主張を約 74%に要約できたことになる．しかし，異なる出来事が混在した 705 件の主張のクラスタが提示されたときに，読者が興味を持っている出来事の異なる主張を収集することは難しい．

[14][4] より引用

第4章　提案手法

本研究では，記事の文章が出来事を述べる文と主張を述べる文に二分できると仮定する．図 4.1 に提案手法の概要を示す．提案手法により，より類似した出来事の異なる主張と主張に紐づく記事の推薦を行う．

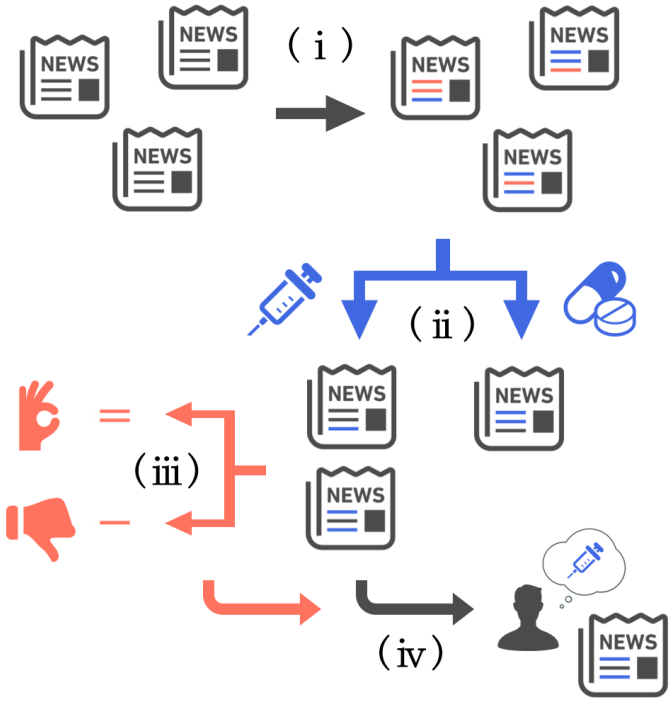

図 4.1:　提案手法の概要．(i) まず，記事の文章を出来事の文と主張の文に分類する．(ii) 次に，出来事の文章の類似度で記事をクラスタリングする．(iii) その後，主張の文の類似度で主張の文をクラスタリングする．(iv) 最後に，読者が興味を持っている記事と最も類似した出来事を扱う記事のクラスタを同定し，そのクラスタに紐づく主張の文をクラスタごとにまとめ，記事とともに読者に提示・推薦する．

第 4 章は全 4 節で構成される．節 4.1 では本研究で使用する語彙の定義を行う．節 4.2 では提案手法に必要なデータセットの選定について説明し，節 4.3 では選定したデータセットの前処理について説明する．節 4.4 では図 4.1 に示した提案手法の詳細を説明する．

4.1　使用する語彙の定義

4.1.1　文と文章

本研究では，1 つの句点で区切られる文を「文」，2 つ以上の文で構成される文の集合を「文章」として扱う．

4.1.2　出来事と主張

本研究では，出来事を「記者の解釈に依存しない事象」主張を「記者が伝えるべきだと判断した出来事の解釈」であると定義する．すなわち，記事内の文章は出来事に対して主張を述べるような構造を持つ．したがって，複数の出来事に対し，複数の主張が階層的に紐づいた構造を持つと考える．

4.2　データセットの選定

4.2.1　分類器の転移学習に用いたデータセット

提案手法の分類器は RoBERTa を転移学習して作成する．転移学習に用いるデータセットには，Wikipedia の英文を Claim を述べるか Evidence を述べるかでラベル付けした Rinott らの IBM Debater - Claims and Evidence を利用した [20]．2294 個の Claim の文は「トピックを直接サポートする一般的で簡潔な文」，4692 組の Evidence の文章は「トピックの文脈の中で Claim を直接サポートする文章」と定

義されている．ラベル付けは 5 人のアノテータによって行われており，少なくとも 3 人が同じラベルを付与した文章を収集している．

表 4.1 にラベル付けされた Claim の文と Evidence の文の例を示す．Claim の文は炭素排出量のトピックについて森林伐採に焦点を当てた簡潔な説明がなされており，具体的な森林の名前を挙げずに一般の森林の話をしている．Evidence の文ではナイジェリアの森林面積が減少していることを具体的な数字を用いて説明しており，Claim の文を直接補っている．

表 4.1: IBM Debater - Claims and Evidence でラベル付けされた Claim の文と Evidence の文の例

ラベル	IBM Debater - Claims and Evidence の文
Claim	global carbon emissions are caused by deforestation
Evidence	From 1990 to 2010 Nigeria nearly halved their amount of Forest Cover, moving from 17,234 to 9041 hectares.

表の例は，本研究で定義した主張の文と出来事の文の定義と似た性質を持つ．Claim の文は炭素排出量のトピックについて記者が伝えるべきだと判断した森林伐採の説明をしており，主張の文の定義に即している．Evidence の文は具体的な数字や森林のある地域名を用いて記者の解釈に依存しない事象を説明しており，出来事の文の定義に即している．Claim の文と Evidence の文を 100 件ずつ確認したところ，他の多くの例でも Claim が主張に，Evidence が出来事に対応していた．このことから，本研究では Claim の文を主張の文，Evidence の文を出来事の文として扱い，分類器の学習を行った．

4.2.2 分類とクラスタリングを行ったデータセット

分類とクラスタリングを行うデータセットには，Ghasiya らが収集した COVID-19 News Articles を選定した [21]．表 4.2 示すように，COVID-19 News Articles はイギリス，インド，日本，韓国の主要な新聞 8 紙の英語版ウェブサイトから約 10 万件の記事を収集している．これらの記事は，スクレイピングライブラリ Beautiful Soup を用いて収集された記事のうち文字列 COVID-19 または Coronavirus を含む記事である．このうちイギリスの記事には句点が含まれておらず，文章を文に分割できなかったため使用しなかった．

表 4.2: COVID-19 News Articles の構成

Countries	Newspapers	No. of Articles
UK	The Daily Mail	23,821
India	Hindustan Times, The Indian Express	47,342
Japan	The Japan Times, Asahi Shimbun, Mainichi Shimbun	21,039
South Korea	Korea Herald, Korea Times	10,076

COVID-19 News Articles は 3 ヶ国の英記事が含まれるため，政治や文化の違いに由来する多くの異なる主張が分析できると考えた．加えて COVID-19 に関連する出来事は記者によって多くの異なる主張が記されているため，分析に適していると考えた．また，このデータセットは 2020 年 1 月 1 日から 2020 年 12 月 1 日の 11 カ月間という短い期間で収集されているため，より類似した出来事に関する記事が得られると期待できる．Yang らの研究と同様に話題が COVID-19 に限定されてしまっているが，限定された話題でも同じ出来事の異なる主張がクラスタリングされる様子は観測できると考えた．

4.3 データの前処理

4.3.1 自然言語処理のためのテキストの前処理

IBM Debater - Claims and Evidence と COVID-19 News Articles の文章に対し，小節 2.4.2 で述べたような単語の表記揺れを少なくするいくつかの前処理を行った．具体的な処理は小節 5.3.1 で説明する．

4.3.2 省略のピリオドなどを考慮した文章の分割

記事の文を分類するにあたり，文章を文ごとに分割する必要がある．しかし，表4.3のように英語の句点に用いられるピリオドは句点以外にも様々な用法があり，人間が決めたアルゴリズムによる文章の分割が難しい [22].

表 4.3: 英語のピリオドの用例

種類	例
文頭の省略語	Prof. Kimura studies.
文中の省略語	Alphabets e.g. A, B, C.
文末の省略語	The tail of a bull, cow, ox etc.
日付	25.01.2022.
ファイル名	readme.txt
URL	http://www.wikipedia.org
IP アドレス	127.0.0.1.
メールアドレス	abc@shibaura-it.ac.jp
分類コード	A01.9

そこで，テキスト解析のための機械学習ライブラリ Stanza を用いて句点としてのピリオドの位置を同定した [23]．他のテキスト解析のツールとして UDPipe や spaCy も広く用いられているが，2018 UD Shared Task の英語の句点の同定におけるスコアは Stanza が最も優れている．

4.4 記事の出来事と主張のクラスタを用いた多様な主張を提示するニュース推薦

4.4.1 RoBERTa を用いた出来事の文と主張の文の分類

出来事と主張の文の分類のため，RoBERTa を転移学習した分類器を作成した．RoBERTa が出来事の文と主張の文の分類に適用できるよう，出来事か主張かでラベル付けされた文を含むデータセットを用いて転移学習を行った．図 2.8 に示したように入力は文字列 [CLS] と分類したい 1 文とし，[CLS] に対応する出力 C と正解のラベルを基に損失関数の計算を行う．具体的には，出力 C を入力とする 1 層の全結合層と 1 ノードの出力層を接続し，その出力とバイナリクロスエントロピーを用いて損失を計算した．

4.4.2 Sentence-BERT，コサイン距離，Ward 法を用いた記事と主張の文の階層的クラスタリング

まず，クラスタリングの前準備として「出来事と分類した文を記事ごとに結合した文章」と「主張と分類した文」のそれぞれを Sentence-BERT に入力して文埋め込みを作成した．次に，出来事の文章の埋め込みを基に記事の階層的クラスタリングを行った．最後に，それぞれ記事のクラスタについて，主張の文の埋め込みを基に階層的クラスタリングを行った．階層的クラスタリングを行うとき，文埋め込み間の距離の算出にはコサイン距離を使用し，この距離を基に Ward 法を用いてクラスタ間距離を算出した．ここで出来事の文章の埋め込みに基づいたクラスタリングを先に行ったのは，複数の出来事に対して複数の主張が階層的に紐づいた構造を持つためである．

4.4.3 k-NN 分類法を用いた主張のクラスタとクラスタに紐づく記事の推薦

まず，読者が興味を持っている記事 A を 1 つ選択し，記事内の文が出来事の文であるか主張の文であるかに分類する．この分類には小節 4.4.1 で作成した分類器を用いる．次に，出来事の文を結合した文章を Sentence-BERT に入力して文埋め込みを得る．続いて，得られた埋め込みを新規のデータ，記事のクラスタの埋め込みをクラスが既知のデータとし，k-NN 分類法を用いて記事 A がどの記事のクラスタに属するかをを同定する．最後に，同定した記事のクラスタに紐づく主張の文をクラスタごとにまとめ，記事とともに読者に提示・推薦する．

第5章　実装

本研究では研究範囲を多様な主張の提示に限定し，k-NN 分類法による記事の推薦の実験は行わないものとする．したがって主張の文に関する階層的クラスタリングまでを行うシステムを実装し，より類似した出来事の異なる主張の文を提示できているかを評価した．第 5 章では，実験に用いたシステムの実装について説明する．節 5.1 ではシステムの入出力やデータフローなどの設計指針を説明し，節 5.2 では使用したコンピュータの構成やツールについて説明し，節 5.3 では具体的なシステムの実装方法について説明する．

5.1　システムの設計指針

提案手法を実装するシステムは，入力を「読者が興味を持っている 1 件の記事」とし，出力を「入力した記事と最も類似した出来事を扱う記事のクラスタに紐づく複数の主張の文」と「出力する主張の文の抽出元である記事」とした．また，分類器の学習のために IBM Debater - Claims and Evidence を，クラスタの作成のために COVID-19 News Articles を使用した．

COVID-19 News Articles から生成した記事のクラスタ群と主張の文のクラスタ群は，クラスタごとにテキストファイルに保存した．COVID-19 News Articles の各記事には，主張の文から記事を特定するための固有の ID と記事の出来事の文章から生成した文埋め込みの情報を付与した．また，各文には文に固有の ID と抽出元の記事の ID，出来事か主張かのラベル，文埋め込みの情報を付与した．

5.2 実行環境

本システムの実行には，表 5.1 に示す構成のコンピュータと表 5.2 に示すツールを用いた．

表 5.1: 実行したコンピュータの構成

項目	詳細
OS	Ubuntu (Version 18.04)
CPU	Intel Core^{TM} i5-7640X (4.00GHz, 4 コア)
GPU	GeForce RTX 3070 (1500MHz, 8GB, CUDA Version 11.5)
メモリ	2667 MT/s 16GB × 2 枚
ストレージ	HDD 1815 Mbit/s
マザーボード	X299 チップセット

表 5.2: 使用したツール

項目	バージョン	用途
GNU Awk	4.1.4	テキストの前処理
Python	3.6.9	GNU gawk 以外のプログラムの実行
Stanza	1.3.0	文章の文への分割
pandas	1.1.5	分類器に合わせたデータの入力形式の変更
Simple Transformers	0.63.3	RoBERTA を用いた分類器の作成
PyTorch	1.8.2 + cu111	Simple Transformer の内部処理
SentenceTransformers	2.1.0	Sentence-BERT を用いた文埋め込みの作成
scikit-learn	0.24.2	混同行列を用いた評価指標の算出
LibreOffice Calc	7.2.5.2	混同行列を用いた評価指標の算出
SciPy	1.5.4	階層的クラスタリングの実行と結果の可視化

5.3 システムの実装

5.3.1 GNU Awk を用いたテキストの前処理の実装

GNU Awk は，あるパターンに従う文字列を検索し，他の文字列に置換するなどの処理を行えるプログラミング言語である．文字列のパターンは，正規表現と呼ばれる表現形式で別の文字列を用いて表すことができる．

2 つのデータセット IBM Debater - Claims and Evidence と COVID-19 News Articles について，小節 2.4.2 で述べた単語の表記揺れやノイズを少なくするため，GNU Awk と正規表現を用いて表 5.3.1 に示す処理を順に行った．

表 5.3: GNU awk を用いた前処理を行うプログラム

処理内容	ソースコード
大文字の小文字化	`$0=tolower($0)`
改行文字の除去	`gsub(/\\n\\r/," ")`
改ページ文字の除去	`gsub(/\\f/," ")`
タブ文字の除去	`gsub(/\\t\\v/," ")`
特殊な空白文字 (no break space, thin space) や絵文字などの除去	`gsub(/\\[a-zA-Z]+[0-9]+/," ")`
URL の除去	`gsub(/https?:\/\/([a-z0-9_\-]+\.)+[a-z0-9_\-]+(\/[a-z0-9_\-\.\/?%&]*)?/,"")`
画像リンクの除去	`gsub(/pic\.([a-z0-9_\-]+\.)+[a-z0-9_\-]+(\/[a-z0-9_\-\.\/?%&]*)?/,"")`

メールアドレスの除去	`gsub(/[a-z0-9_]+([\.\-\+][a-z 0-9_]+)*@[a-z0-9_]+([\.\-][a-z 0-9_]+)*\.[a-z0-9_]+([\.\-][a-z 0-9_]+)*/,"")`
英数字，空白文字，句読点以外の文字の除去	`gsub(/[^a-z0-9 \.!?]/,"")`
ピリオドの前の空白文字の除去	`gsub(/ +\./,".")`
疑問符の前の空白文字の除去	`gsub(/ +?/,"?")`
感嘆符の前の空白文字の除去	`gsub(/ +!/,"!")`
2 つ以上のピリオドを 1 つのピリオドに置換	`gsub(/\.{2,}/,".")`
2 つ以上の疑問符を 1 つの疑問符に置換	`gsub(/\?{2,}/,"?")`
2 つ以上の感嘆符を 1 つの感嘆符に置換	`gsub(/!{2,}/,"!")`
2 つ以上の空白文字を 1 つの空白文字に置換	`gsub(/ {2,}/," ")`
行頭の 1 つ以上の空白文字の除去	`gsub(/^ +/,"")`
行頭の 1 つ以上のピリオドの除去	`gsub(/^\.+/,"")`
行頭の 1 つ以上の疑問符の除去	`gsub(/^\?+/,"")`
行頭の 1 つ以上の感嘆符の除去	`gsub(/^!+/,"")`

また，この 2 つのデータセットで文の形式を合わせるため，IBM Debater - Claims and Evidence の文の末尾にピリオドを追加した．加えて，IBM Debater - Claims and Evidence の文のみに含まれていた特殊な文字列 [REF] を除去した．

5.3.2 Stanzaを用いた文章を文単位に分割する実装

Pythonライブラリである Stanzaを使用し，IBM Debater - Claims and Evidence の出来事の文章と COVID-19 News Articlesの文章の句点としてのピリオドの位置を同定した．その後，同定したピリオドの後ろに改行文字を挿入した．分割した出来事の文や主張の文には重複する文の組が多く見受けられたため，重複する文は 1 文を残して除去した．

Stanzaの英語版モデルには，ウェブログ，ニュース文，メール文，レビュー文，Yahoo! answersの16621文とそのテキスト解析結果を含むデータセット UD English EWT を学習したモデルを使用した．

5.3.3 RoBERTaを用いた出来事の文と主張の文の分類器の実装

RoBERTaの事前学習モデルにはLiuらのRoBERTa baseを利用した[14]．RoBERTa baseは1層に16個の Scaled Dot-Product Attentionを含む Transformer エンコーダを 768 個持つ層を 12 層用いている．また，ニュース記事を含む約 160GB の英文をバッチサイズ 8,000 で 50 万回学習している．

RoBERTa baseの転移学習と分類の実行には，Python ライブラリである Simple Transformersを使用した．Simple Transformersの内部では同じく Python のライブラリである PyTorchが用いられている．

転移学習ではIBM Debater - Claims and Evidenceの出来事の文と主張の文のそれぞれについて8割を訓練データ，2割をテストデータに分割し，バイナリクロスエントロピーの計算を行った．このとき，データの順序に由来する不必要な学習を回避するため，文とラベルは疑似乱数を用いた無作為な順序で学習した[10]．また，出来事の文は4209文，主張の文は2169文と数が不均衡であるため，出来事の文の学習の重みに $\frac{2169}{4209}$ を乗算し，ラベルごとに均等に学習ができるようにした．学習のバッチサイズは40文とし，その他の学習のパラメータはSimple Transformers

のデフォルトの値を使用した．

学習した分類器に COVID-19 News Articles の文を入力し，分類したラベル，記事の ID，文の ID とともにテキストファイルに保存した．分類結果は Python ライブラリである scikit-learn，Simple Transformers と表計算ソフトである LibreOffice Calc を用いて混同行列として集計し，これを基に適合率，再現率，マシューズ相関係数を計算した．

5.3.4 Sentence-BERT を用いた文埋め込みを生成する実装

出来事の文は記事に記載された順序で記事ごとに結合し，Sentence-BERT に入力した．文の間には空白文字を挿入して結合した．出来事の文章を基に生成した文埋め込みは記事の ID と紐づけてテキストファイルに保存した．主張の文は 1 文ずつ Sentence-BERT に入力し，文の ID と紐づけてテキストファイルに保存した．

Sentence-BERT のモデルには Raimers らの paraphrase-MiniLM-L6-v2 を使用した [16]．paraphrase-MiniLM-L6-v2 は 1 層に 12 個の Scaled Dot-Product Attention を含む Transformer エンコーダを 384 個持つ層を 6 層用いている．MNLI と SNLI（The Stanford Natural Language Inference）の約 100 万文のうち約 10 万文の英文をバッチサイズ 16 で 1 回学習している．MNLI は話し言葉と書き言葉の様々な英文をカバーしている．

5.3.5　記事に関する階層的クラスタリングの実装

記事の ID に紐づけられた出来事の文章の文埋め込みを基に，Python ライブラリである SciPy を用いて埋め込みの距離行列を作成した．この距離行列は，i 番目の埋め込みと j 番目の埋め込みとの距離を i 行 j 列に格納した行列である．距離行列を作成することで，これらの距離を複数回参照するときにコンピュータの計算量を低減することができる．

次に，SciPy と距離行列を用いて Ward 法によるクラスタ間距離を計算し，階層的クラスタリングを行った．n 回のクラスタの結合を行う階層的クラスタリングの結果は，n 行 4 列の行列として保存される．1 列目と 2 列目には結合される 2 つのクラスタの ID，3 列目には 2 つのクラスタ間距離，4 列目には新しく生成されるクラスタの ID が保存される．

最後に，SciPy と保存された階層的クラスタリングの結果を使用してデンドログラムを作図し，データの分析を行った．

5.3.6　主張の文に関する階層的クラスタリングの実装

まず，記事の階層的クラスタリングで生成した各クラスタに対し，クラスタ内の各文の ID に紐づけられた分類結果の情報を参照し，主張の文のみを抽出した．その後，文の ID に紐づけられた文埋め込みを参照し，各クラスタについて埋め込みの距離行列の作成，階層的クラスタリングの実行，デンドログラムの作図を行った．

第6章　実験

第 6 章では，第 5 章で述べた分類器とクラスタリングの実験について説明する．節 6.1 では IBM Debater - Claims and Evidence のテストデータを用いた分類器の実験について説明し，節 6.2 では COVID-19 News Articles を用いた分類器の実験について説明する．その後，節 6.3 で記事の階層的クラスタリングの実験について説明し，節 6.4 で主張の文の階層的クラスタリングの実験について説明する．

6.1　IBM Debater - Claims and Evidence のテストデータを用いた分類器の実験

6.1.1　実験方法

分類器の最適なエポック数を決定するため，IBM Debater - Claims and Evidence の 2 割のテストデータを使用し，学習ごとのバイナリクロスエントロピーと分類結果の適合率，再現率，マシューズ相関係数を確認した．適合率，再現率，マシューズ相関係数の計算では，出来事の文と比べてデータ数が少ない主張の文のラベルを陽性クラスとした．

6.1.2　実験結果

表 6.1 にエポック数ごとの分類器の評価を示す．どのエポック数でも適合率，再現率，マシューズ相関係数の値が 0.81 と大きく，モデルが分類器として機能して

いることがわかる．バイナリクロスエントロピー，適合率，マシューズ相関係数は 1 エポックで，再現率は 3 エポックで最も優れた値となった．バイナリクロスエントロピーが単調増加しているため，学習するほどモデルが訓練データに対して過学習してしまう可能性がある．

表 6.1: エポック数ごとの分類器の評価（小数第 3 位を四捨五入）

評価指標	1 epoch	2 epochs	3 epochs	4 epochs	12 epochs	100 epochs
BCE	**0.17**	0.21	0.24	0.30	0.59	0.87
適合率	**0.90**	0.86	0.86	0.86	0.85	0.85
再現率	0.89	0.90	**0.92**	0.89	0.89	0.91
MCC	**0.84**	0.82	0.83	0.82	0.81	0.82

6.1.3 実験の考察

バイナリクロスエントロピーが単調増加しているため，多くのエポック数で学習した分類器は COVID-19 News Articles での分類精度が悪くなると考えられる．したがって，12 エポックと 100 エポックで学習したモデルは除外して考える．

一方でエポック数が少なすぎるとモデルの学習不足で過少適合となる可能性があり，同じく COVID-19 News Articles での分類精度が悪くなると考えられる．したがって，2 エポックや 4 エポックのモデルが最も精度よく COVID-19 News Articles の分類を行う可能性は捨てきれない．

1 エポックから 4 エポックにかけて適合率，再現率，マシューズ相関係数の差は最大で 0.04 と小さいため，この 4 つの学習済みモデルで COVID-19 News Articles の分類精度を検証することにした．

6.2 COVID-19 News Articlesを用いた分類器の実験

6.2.1 実験方法

IBM Debater - Claims and Evidenceを1，2，3，4エポック学習した4つの分類器について，COVID-19 News Articlesの分類結果から適合率，再現率，マシューズ相関係数を算出した．COVID-19 News Articlesの正解ラベルには，3ヶ国の記事から3件ずつ記事を抽出した計163個の文を手動でラベル付けしたものを用いた．ラベル付けを行うにあたり，IBM Debater - Claims and Evidenceの出来事の文と主張の文をそれぞれ100文ずつ分析し，ラベル付けの基準を作成した．適合率などの計算では，分類結果においても正解ラベルにおいても主張の文が出来事の文より少なかったため，主張の文のラベルを陽性クラスとした．

6.2.2 実験結果

表 6.2 に IBM Debater - Claims and Evidence の分析を基に作成したラベル付けの基準を示す．この基準に従い，COVID-19 News Articles の 163 文のラベル付けを行った．

表 6.2: COVID-19 News Articles のラベル付けの基準（E : 出来事の文，C : 主張の文）

ラベル	基準	IBM Debater - Claims and Evidence の例文
E	記者の解釈に依存しない事実を述べている	**michael martin was the first eu foreign minister to enter gaza in over a year**
	指示語を含み，別の文を補助している	in **his** book maybe one bill mckibben argues in favor of a one child policy based on **this** research
	具体的な事物の数値を含む	in **2020** total gross foreign aid to all developing countries was **76 billion**.
C	事実に対する解釈を述べている	this is the most **technologically advanced** and **safest pipeline** ever proposed.
	主語や述語の一般性が高い	**gamblers persist in gambling** even after repeated losses.

表 6.3 に COVID-19 News Articles の 163 文の分類の評価結果を示す．適合率，再現率，マシューズ相関係数の全てにおいて 3 エポックの分類器が最も優れていた．3 エポックの分類器は再現率が低いが，適合率とマシューズ相関係数が高く，分類器として機能していることがわかる．3 エポックの分類器が主張の文を出来事の文と誤分類することは 1 度もなく，適合率は 1 となっている．以降の説明では，全てこの 3 エポックの分類器を用いた実験の結果を記す．

表 6.3: COVID-19 News Articles の 163 文の分類の評価（小数第 3 位を四捨五入）

評価指標	1 epoch	2 epochs	3 epochs	4 epochs
TP	3	4	6	5
TN	146	148	148	147
FP	2	0	0	1
FN	12	11	9	10
適合率	0.60	**1.00**	**1.00**	0.83
再現率	0.20	0.27	**0.40**	0.33
MCC	0.31	0.50	**0.61**	0.50

表 6.4 に分類結果の例を示す．5 つ目の文は，一般の家族や教育者にいえる筆者の解釈が述べられているが，出来事に誤分類されていた．6 つ目の文は，一般のオンライン授業にいえる事実（them）に対する筆者（i）の解釈が述べられているが，出来事に誤分類されていた．

表 6.4: 分類器の主張の文 (C) と出来事の文 (E) の分類例

正解ラベル	分類結果	COVID-19 News Articles の文
C	C	the goal was to bolster international competitiveness.
C	C	proponents see the pandemic as a chance to break with the modern calendar.
E	E	the government is reportedly aiming to announce guidance early next month.
E	E	over 20000 people have signed an online petition calling for a shift to september school starts.
E	C	for now there is little that families and educators can do but wait to see what the abe administration has in mind.
E	C	i dont think online classes can completely replace them

6.2.3 実験の考察

分類の適合率が極めて大きいため，2019 年の最先端のモデルである RoBERTa が適切に転移学習できたと考える．一方で再現率は学習時よりも 0.52 と大幅に下がっており，分類器が訓練データに過学習している可能性がある．

再現率の改善のため，学習と分類に用いたデータセットの違いをより詳細に分析し，手法を見直す必要がある．学習に用いたデータセットでは指示語が含まれる文を出来事の文とすることが多かったのに対し，表 6.4 で出来事と誤分類した主張の文には指示語に用いられる “there” が含まれていた．しかし，誤分類した文で

は “there is” という構文で用いられ，指示語として “there” を用いていない．したがって，このような構文を加味した分類を行うことにより再現率の向上が期待できる．

また，学習に用いた IBM Debater - Claims and Evidence は自動討論システムや論証の構成を検出するシステムへの応用が想定されており，記事の文章との文の構造が根本的に異なっている可能性がある．したがって，記事をラベル付けした学習用データセットを調査もしくは作成することにより，再現率の向上が期待できる．

過学習の緩和のため，分類モデルに使用した RoBERTa よりも過学習しにくいモデルを検討するのも有効である．Carlebach らは IBM Debater - Claims Stance Dataset を学習した BERT モデルに記事由来のデータを入力して過学習を起こしてしまったが，別の機械学習モデル T5 に変更することで過学習を緩和している [24].

なお，確認した 163 文のうち主張の文は 6 文しかなかったため，より信頼できる適合率と再現率の算出のためにより多くの文での評価が必要である．

6.3 記事の階層的クラスタリングの実験

6.3.1 実験方法

階層的クラスタリングの実験には COVID-19 News Articles の 5000 件の記事を用いた．記事の ID に紐づけられた出来事の文章の文埋め込みを使用し，コサイン距離と Ward 法を用いた階層的クラスタリングを行う．

まず，クラスタを結合する度に全埋め込みのシルエット係数の平均値を算出した．これにより，クラスタを分けるクラスタ間距離ごとに，同一クラスタ内の埋め込みが類似しするか，異なるクラスタ間の埋め込みが類似しないかを確認した．

クラスタを分けるクラスタ間距離には，クラスタ間距離を大きくしたときにクラスタ数の減少が緩やかになり始める距離を選択した．この距離は，より類似しない文章がクラスタにまとめられ始めるクラスタ間距離に対応する．このクラスタ間距離を同定するため，結合するクラスタ間距離ごとのクラスタの数をグラフに表した．また，このクラスタ間距離でクラスタごとの記事の数が読者の把握できる量を超えていないか確認するため，結合するクラスタ間距離ごとのクラスタがもつ平均の記事の数をグラフに表した．

階層的クラスタリングを行った後，文字数が多いクラスタを 1 つ抽出し，クラスタ内の記事の主張の文が類似した出来事について説明しているかを目視で確認した．また，抽出したクラスタ内の主張の文のうちの，類似した出来事について述べている文の割合を算出した．

6.3.2 実験結果

図 6.1 にクラスタ数ごとのシルエット係数を示す．シルエット係数が常に 0 より大きいため，同一クラスタ内の文埋め込みが類似し，異なるクラスタ間の文埋め込みが類似していない傾向にあることがわかる．クラスタ数が 2,500 から 5,000 の範囲ではシルエット係数は大きいが，1 つのクラスタ内の記事の数が 1, 2 件であることが多く，異なる主張の文の提示には適していない．クラスタ数が 500 から 2,500 の範囲ではシルエット係数に大きな差がないため，この範囲でクラスタを分けるクラスタ間距離の設定を考える．

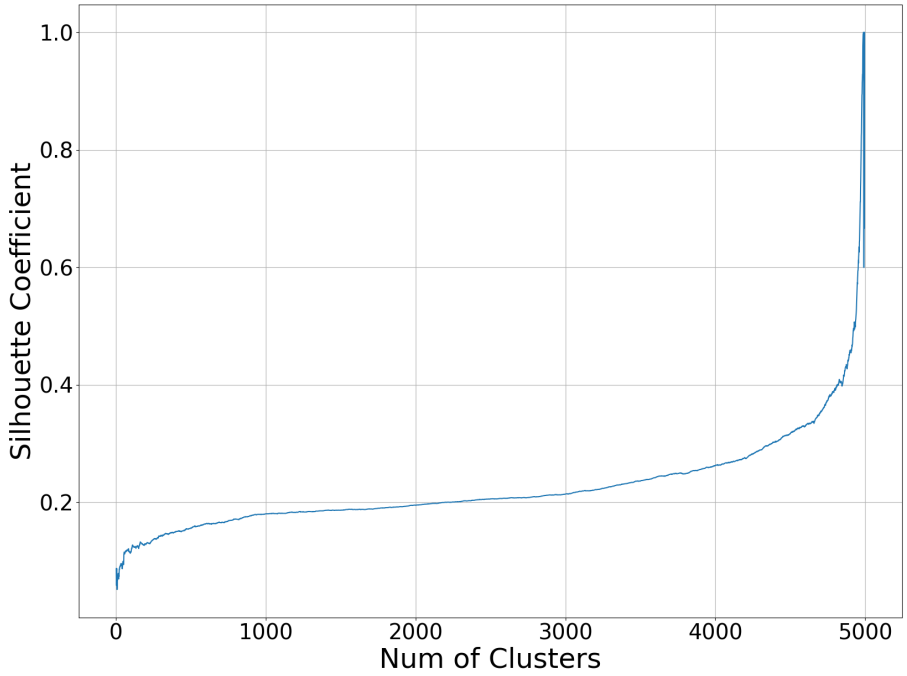

図 6.1: 記事の階層的クラスタリングにおけるクラスタ数ごとのシルエット係数

図 6.2 にクラスタを分けるクラスタ間距離ごとにまとめたクラスタ数とクラスタ内の平均記事数を示す．図 6.2 の上の図から，クラスタ間距離 0.5 の付近（クラスタ数 2000 の付近）でクラスタ間距離を大きくしたときのクラスタ数の減少が緩やかになり始めていることがわかる．クラスタ数 2000 から 500 にかけてはグラフが直線的であったため，小節 3.3.1 で述べた Yang らのシステムと比較がしやすいクラスタ間距離 0.85（クラスタ数 666）でクラスタを分割し，以降の実験を行うこととした [4]．クラスタ間距離 0.85 でのクラスタ内の平均記事数は 7.51 件であり，読者への異なる主張の提示に適当な量であると考える．

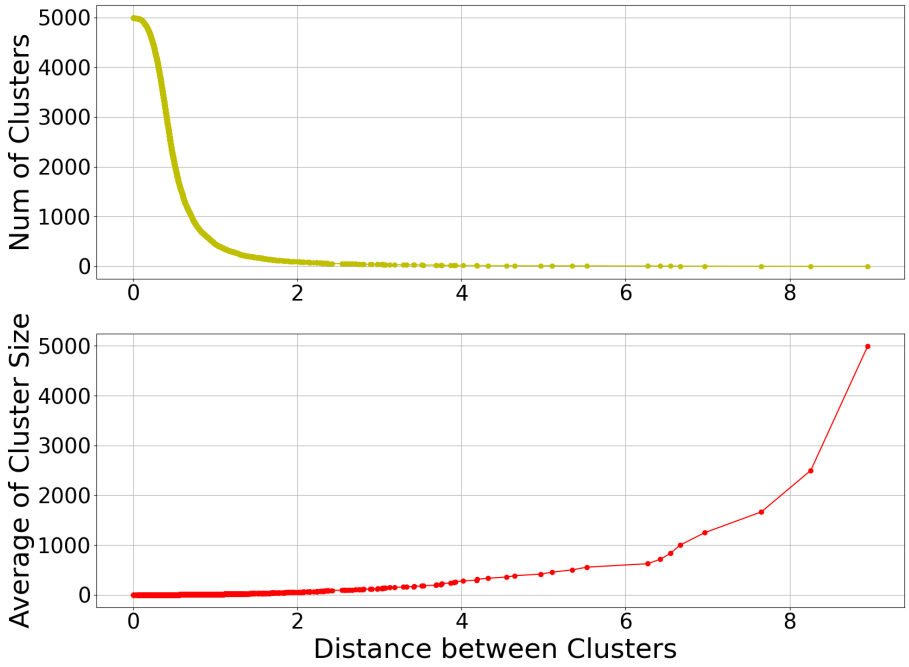

図 6.2: クラスタを分けるクラスタ間距離ごとの「クラスタ数」と「クラスタ内の平均記事数」

図 6.3 に記事の階層的クラスタリングの結果をデンドログラムで示す．クラスタ間距離 0.85 以下の最後の結合で分かれたクラスタに色を付けている．

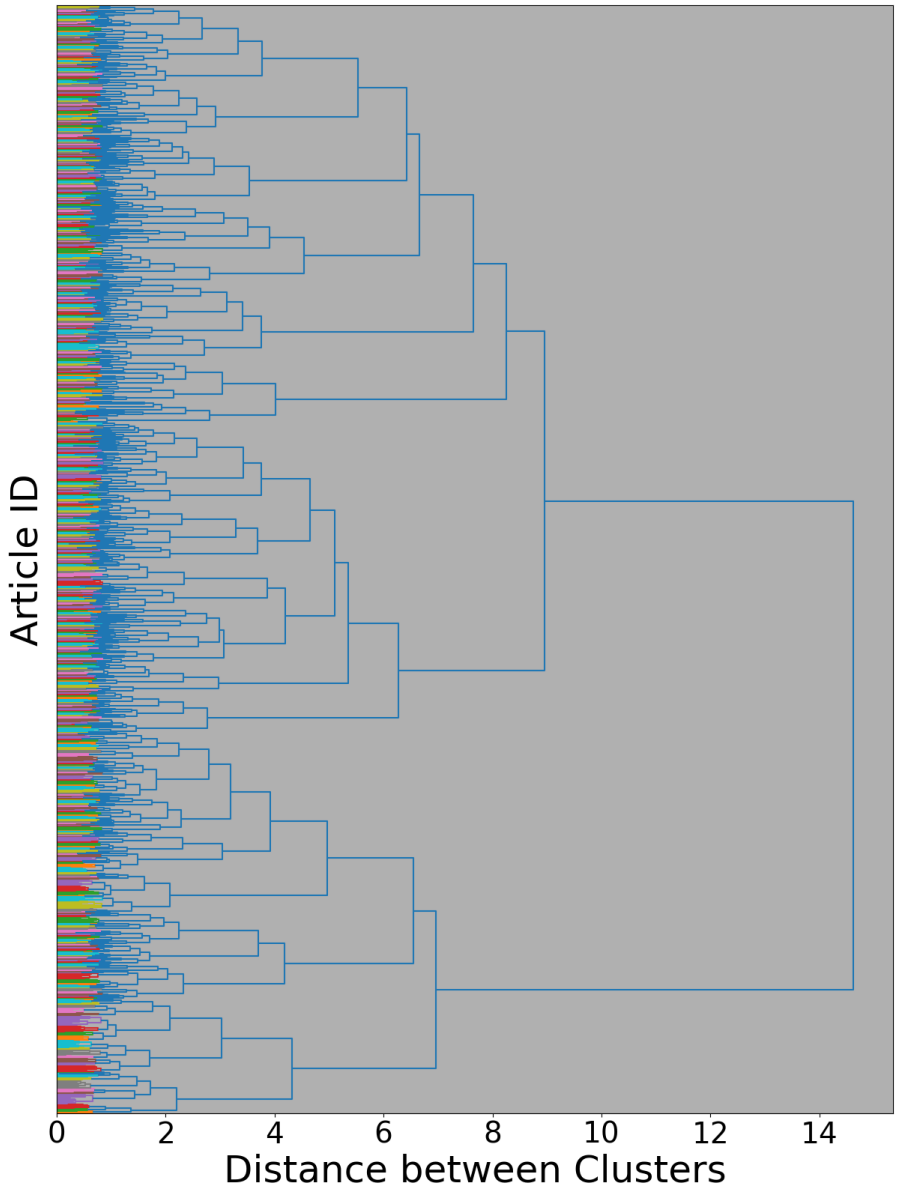

図 6.3: 記事の階層的クラスタリングの結果

色付けされたクラスタのうち，3ヶ国すべての国の記事を含む最も文字数が多いクラスタを抽出した．このクラスタには38件の記事が属しており，記事の文のうち主張の文は13文であった．表6.5にこの13文の主張の文と主張の対象とした出来事を示す．13文中10文（約77%）がウイルス感染対策に関する類似した出来事に対して主張を述べる文であった．また，残りの3文のうち2文はウイルス感染の社会の影響に関する出来事に対して主張を述べる文であり，ウイルス感染対策と出来事が類似している．

また，インドから1文，韓国から9文，日本から3文だけ主張の文が抽出されており，異なる地域から主張や記事を提示可能であると期待できる．

表 6.5: 1 つの記事のクラスタが含む主張の文と主張の対象とした出来事

国	出来事	記事のクラスタが含む主張の文
インド	仕事	it was a laborious **job**.
韓国	**感染対策**	sports event are also obligated to **keep the ceiling of 30 percent at stadiums**.
韓国	**感染対策**	south korea operates a threetier **social distancing** system.
日本	**感染対策**	chinese banks have been ordered to **disinfect** old banknotes before reissuing them to the public.
日本	**感染対策**	uncertainly remained over how best to **stem the spread of the illness**.
日本	**感染対策**	**outdoor exercise will be banned** and **wearing masks** will be mandatory.
韓国	**感染対策**	franchie cafes and dessert shops were obligated to offer **only takeout around the clock**.
韓国	**感染対策**	health authorities remain **vigilant over sporadic cluster infections** at hospitals nursing homes and riskprone facilities.
韓国	感染の影響	**travelrelated cases** continue to **outnumber local cases**.
韓国	**感染対策**	to **keep out imported infections** authorities have **imposed more stringent measures on people** arriving from countries deemed highrisk.
韓国	**感染対策**	cities and provinces that agree to accommodate passengers who require a twoweek **quarantine** will be paid government incentives.
韓国	**感染対策**	**try not to eat in restaurants** as much as possible.
韓国	感染の影響	the citys **hospitals** are facing **an overcrowding crisis**.

6.3.3 実験の考察

記事の階層的クラスタリングを行った結果，あるクラスタの主張の文の約 77%が類似した出来事に対する主張を述べており，提案手法のシステムで類似した出来事のグループ化が可能であることがわかった．また，表 6.5 で類似した出来事を対象としていると判断した文では “social distancing”，“stem the spread of the illness”，“qurantine” などの異なる単語や動詞句が判断基準となっており，Sentence-BERT によって文の意味や文脈を加味したグループ化ができていることがわかる．

小節 3.3.1 で述べた Yang らのシステムではシルエット係数の平均値が 0.79 であり，提案手法のシステムの平均値（約 0.17）よりも高い値をとっている [4]．Yang らのシステムでシルエット係数が高い理由は，内容が個々の記事に依存するツイート群を収集しているからであると考える．つまり，ある 1 つの記事に対するツイート群で類似したクラスタが形成される傾向にあり，また異なる 2 つの記事に依存した 2 つのクラスタはそれぞれ類似しないツイート群で形成される傾向にあるため，シルエット係数が高くなる傾向にあると考えられる．

提案手法のシステムでシルエット係数が低い理由は，出来事の文章の文埋め込みが 384 次元であるようにシステムが考慮する特徴が多いことに起因すると考える．出来事の文章の多くの単語が類似した出来事に関する語彙であったとしても，一部の単語が別の出来事に関する語彙であれば埋め込み同士の距離は離れてしまい，シルエット係数が高くなるようなクラスタリングが難しい．提案手法では出来事の文章を結合して文埋め込みを作成したが，システムが考慮する特徴が減るように，出来事の文章を要約した 1 文を抽出もしくは生成して文埋め込みを作成することでシルエット係数の平均値が向上する可能性がある．

抽出した記事のクラスタでは 10 件のウイルス感染対策に関する文とその他 3 件の文を出来事として区別できていないため，クラスタ間距離の算出方法やクラスタを分けるクラスタ間距離の設定方法などに改善が必要である．

6.4　主張の文の階層的クラスタリングの実験

6.4.1　実験方法

主張の文の階層的クラスタリングの実験には，表 6.5 の 13 件の主張の文を用いる．これらの文は小節 6.3.2 で述べたように，3 ヶ国すべての国の記事を含む最も文字数が多い記事のクラスタが有する主張の文である．文の ID に紐づけられた主張の文の文埋め込みを使用し，コサイン距離と Ward 法を用いた階層的クラスタリングを行う．

記事の階層的クラスタリングと同様に全埋め込みのシルエット係数の平均値を算出し，クラスタ間距離ごとのクラスタの数とクラスタがもつ主張の文の数の平均値をグラフに表した．その後，クラスタを分けるクラスタ間距離としてクラスタ数の減少が緩やかになり始める距離を選択した．

階層的クラスタリングを行った後，それぞれのクラスタに属する文章を目視で確認し，クラスタ内の主張の文が類似しているかを分析した．また，異なるクラスタ間で文が類似しないシステムになっているかを分析した．

6.4.2 実験結果

図 6.4 にクラスタ数ごとのシルエット係数を示す．シルエット係数が常に 0 より大きいため，同一クラスタ内の文埋め込みが類似し，異なるクラスタ間の文埋め込みが類似していない傾向にあることがわかる．クラスタ数が 9 から 12 の範囲ではシルエット係数は大きいが，1 つのクラスタ内の主張の文の数が 1, 2 件であることが多く，類似した主張の文をグループ化できていない可能性がある．クラスタ数が 2 から 8 の範囲ではシルエット係数に大きな差がないため，この範囲でクラスタを分けるクラスタ間距離の設定を考える．

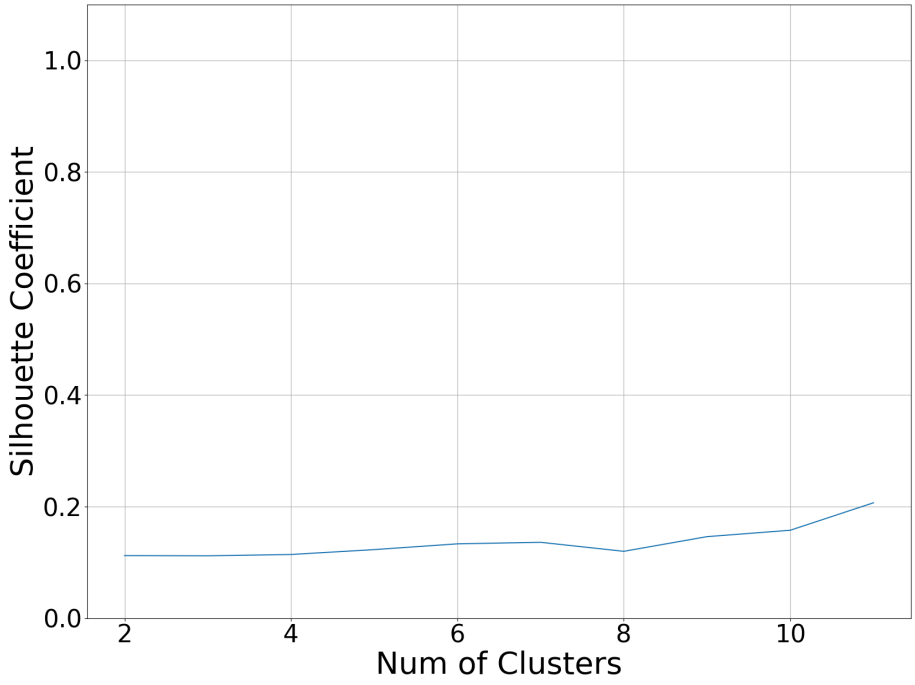

図 6.4: 主張の文の階層的クラスタリングにおけるクラスタ数ごとのシルエット係数

図 6.5 にクラスタを分けるクラスタ間距離ごとにまとめたクラスタ数とクラスタ内の主張の文の数の平均値を示す．図 6.5 の上の図から，クラスタ間距離 0.85 の付近（クラスタ数 2000 の付近）でクラスタ間距離を大きくしたときのクラスタ数の減少が緩やかになり始めていることがわかる．このクラスタ間距離は小節 3.3.1 で述べた Yang らのシステムで用いた値と一致しており，比較評価がしやすい [4]．したがって，このクラスタ間距離 0.85（クラスタ数 5）でクラスタを分割し，以降の実験を行うこととした．クラスタ間距離 0.85 でのクラスタ内の主張の文の数の平均値は 2.6 文であり，類似した主張の文をグループ化できている可能性がある．

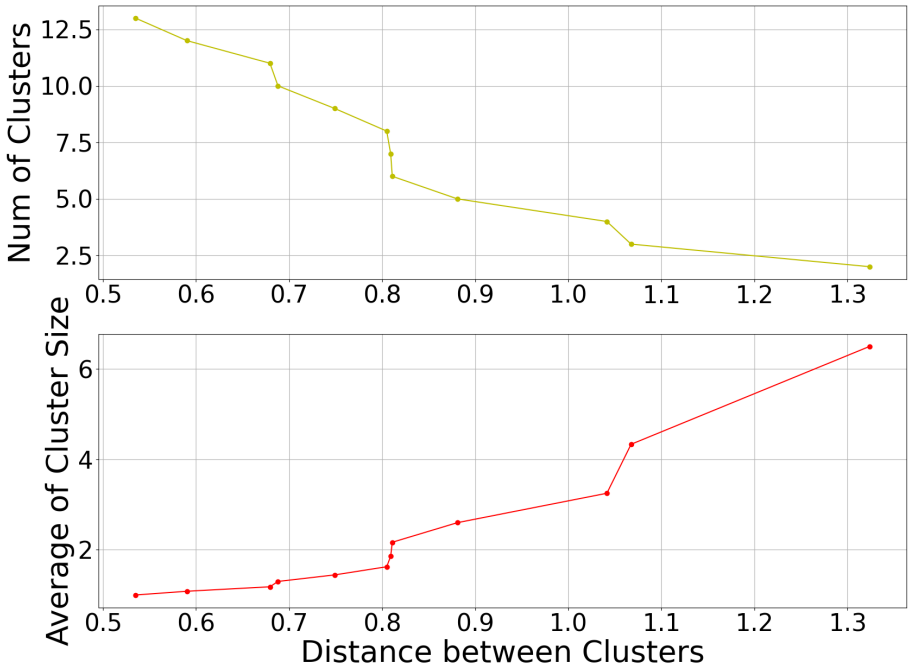

図 6.5: クラスタを分けるクラスタ間距離ごとの「クラスタ数」と「クラスタ内の主張の文の数の平均値」

図 6.6 に主張の文の階層的クラスタリングの結果をデンドログラムで示す．クラスタ間距離 0.85 以下の最後の結合で分かれたクラスタに色を付けている．

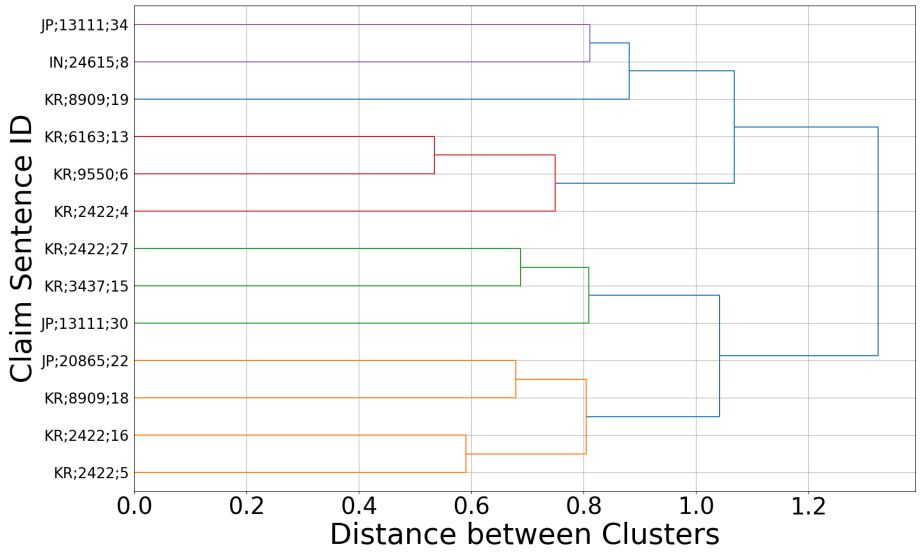

図 6.6: 記事の階層的クラスタリングの結果

クラスタ内外の主張の文の類似度を確認するため，表 6.6 に 13 文の主張の文と主張の文が対象とする事物をまとめた．

図 6.6 で緑色に色付けしたクラスタには 3 文中 2 文が飲食店での感染対策に関する類似した主張がグループ化されていた．また，図 6.6 で黄色に色付けしたクラスタには 4 文中 2 文が運動時の感染対策に関して，残りの 2 文が旅行時の感染対策に関しての主張がグループ化されていた．このことから，同一のクラスタ内で類似した主張の文が集まり，5000 件の記事の主張が要約できていることがわかる．

一方で，異なるクラスタ間で主張の概略とその対象の組が一致する文は 1 つもなく，異なるクラスタに異なる主張の文が属していることがわかる．13 文中 10 文（約 77%）が感染対策に関する類似した出来事に対して主張を述べる文であるため，提案手法で類似した出来事の異なる主張を提示できることがわかった．

表 6.6: クラスタごとの主張の文と主張の対象

所属クラスタの図 6.6 での色	主張の対象	主張の概略	主張の文
紫	-	**感染対策**が大変	uncertainly remained over how best to **stem the spread of the illness**.
紫	-	仕事が大変	it was a laborious **job**.
青	国の全域	**感染対策**をしている	**south korea** operates a threetier **social distancing** system.
赤	病院	感染が蔓延	the citys **hospitals** are facing **an overcrowding crisis**.
赤	病院など	**感染対策**をしている	health authorities remain **vigilant over sporadic cluster infections** at **hospitals nursing homes and riskprone facilities**.
赤	旅行	感染が蔓延	**travelrelated cases** continue to **outnumber local cases**.
緑	**飲食店**	**感染対策**をしている	**try not to eat in restaurants** as much as possible.
緑	**飲食店**	**感染対策**をしている	**franchie cafes** and **dessert shops** were obligated to offer **only takeout around the clock**.
緑	銀行	**感染対策**をしている	chinese **banks** have been ordered to **disinfect** old banknotes before reissuing them to the public.

黄	**運動**	**感染対策**をしている	**outdoor exercise will be banned** and **wearing masks** will be mandatory.
黄	**運動**	**感染対策**をしている	**sports event** are also obligated to **keep the ceiling of 30 percent at stadiums**.
黄	**旅行**	**感染対策**をしている	cities and provinces that agree to accommodate **passengers** who require a twoweek **quarantine** will be paid government incentives.
黄	**旅行**	**感染対策**をしている	to **keep out imported infections** authorities have **imposed more stringent measures on people arriving from countries deemed highrisk**.

6.4.3 実験の考察

表 6.6 の分析から，提案手法で類似した出来事の異なる主張を提示できることがわかった．しかし，緑色のクラスタは飲食店と銀行を主張の対象として分割できておらず，黄色のクラスタは運動と旅行を主張の対象として分割できていない．したがって，同一クラスタ内の主張の文をより類似した文のみでグループ化したいときは，クラスタ間距離の算出方法やクラスタを分けるクラスタ間距離の設定方法などに改善が必要である．

小節 3.3.1 で述べた Yang らのシステムでは 959 件の記事の主張を 705 個の「出来事が混在した異なる主張のクラスタ」（約 74%）に要約していたのに対し，提案手法のシステムでは 5000 件の記事の主張を 5 個の「類似した出来事の異なる主張のクラスタ」（0.5%）に要約できた．これにより，読者が興味を持っている出来事の異なる主張の収集に時間を要してしまう Yang らのシステムの問題は，提案手法

のシステムで改善できたと考える．

本実験での主張の文の階層的クラスタリングの評価は，1つの記事のクラスタが含む13件の主張の文で行った．類似した出来事の異なる主張を提示できるかの評価をより信頼できるものにするためには，他のより多くの記事のクラスタでの評価や入力する記事の数を増やしたときの評価，データセットを変えたときの評価などが必要である．

第7章　まとめと今後の課題

7.1　まとめ

本研究では出来事の文（Evidence の文）と主張の文（Claim の文）のクラスタを用いた多様な主張を提示するニュース推薦手法を提案した．より類似した出来事の異なる主張の文を提示するため，記事の文を出来事の文か主張の文かで分類し，出来事の文章と主張の文の文埋め込みを生成し，それぞれの埋め込みに基づいて記事の階層的クラスタリングと主張の文の階層的クラスタリングを行った．その後，読者が興味を持っている 1 つの記事に対し，k-NN 分類法を用いてその記事と類似した出来事を扱う主張の文のクラスタを提示し，主張の文に紐づく記事を推薦する手法を提案した．

出来事の文と主張の文の分類器には，事前学習された RoBERTa base モデルに 1 層の全結合層と 1 ノードの出力層を接続したモデルを使用した．このモデルの転移学習には IBM Debater - Claims and Evidence の前処理した Evidence の文と Claim の文を入力し，損失関数としてバイナリクロスエントロピーを用いた．文埋め込みの生成には Sentence-BERT の paraphrase-MiniLM-L6-v2 モデルを利用し，提案システムが文の意味と文脈を加味できるようにした．階層的クラスタリングでは文埋め込み間のコサイン距離を用いた Ward 法によるクラスタ間距離を使用した．

実験では，主張の文の階層的クラスタリングまでを行うシステムを実装し，より類似した出来事の異なる主張の文を提示できているかを評価した．分類器や階層的クラスタリングの入力には COVID-19 News Articles の 5000 件の英語の記事

を使用し，政治や文化の違いに由来する 3 ヶ国の多様な主張が分析できるようにした．

IBM Debater - Claims and Evidence を用いた分類器の実験では，適合率が 0.86〜0.90，再現率が 0.89〜0.92，マシューズ相関係数が 0.82〜0.84 である 4 種類のエポック数の精度の高い分類器を作成することができた．しかし，バイナリクロスエントロピーが 1 エポックから単調増加しており過学習となっている可能性があったため，次の実験では 4 つのモデルで比較評価を行った．

COVID-19 News Articles を用いた分類器の実験では 163 文の正解ラベルを作成し，このラベルに基づいて適合率が 1.00，再現率が 0.40，マシューズ相関係数が 0.61 となる分類を行う分類器を以降のクラスタリングの実験に採用した．

記事に関する階層的クラスタリングの実験では，クラスタ間距離 0.85 以下の最後の結合によって 5000 件の記事を 666 個のクラスタに分割した．ある記事のクラスタでは主張の文の約 77%が類似した出来事に対する主張を述べており，提案手法で類似した出来事のグループ化が可能であることがわかった．

主張の文に関する階層的クラスタリングの実験では，クラスタ間距離 0.85 以下の最後の結合により，666 個の記事のクラスタが含む 13 件の主張の文を 5 個のクラスタに分割した．ある主張の文のクラスタ内では 3 文中 2 文が類似した事物に対して主張を述べており，一方で任意の異なるクラスタ間で主張の概略とその対象の組が一致する主張の文は 1 つも存在しなかった．このことから，提案手法で類似した出来事の異なる主張の提示が可能であることがわかった．また，小節 3.3.1 で述べた Yang らのシステムでは 959 件の記事の主張を 705 個の「出来事が混在した異なる主張のクラスタ」（約 74%）に要約していたのに対し，提案手法のシステムでは 5000 件の記事の主張を 5 個の「類似した出来事の異なる主張のクラスタ」（0.5%）に要約できた．これにより，読者が興味を持っている出来事の異なる主張の収集に時間を要してしまう Yang らの手法の問題は，提案手法で改善できたと考える．

7.2 今後の課題

提案手法の分類器は記事の分類において再現率とマシューズ相関係数が小さい．したがって今後の課題として，構文を加味した分類器の作成や記事に適した学習用データセットの検討，過学習しにくいモデルの検討が必要である．

記事に関する階層的クラスタリングでは，抽出した記事のクラスタにおいて77%の主張の文と23%の主張の文とで対象とする事物が類似していなかった．したがって今後の課題として，クラスタ間距離の算出方法やクラスタを分けるクラスタ間距離の設定方法などに改善が必要である．

主張の文に関する階層的クラスタリングでは，同一クラスタ内の主張の文をより類似した文のみでグループ化するために，クラスタ間距離の算出方法やクラスタを分けるクラスタ間距離の設定方法などの改善が必要である．

全体として，より信頼性の高い評価のために，作成する正解ラベルの数や入力する記事の数を増やす必要がある．また，今回実験を行わなかったk-NN分類法を用いた記事の推薦についてもその妥当性を評価する必要がある．

謝辞

本論文の執筆には，非常に多くの方々の支えがありました．指導教員である木村先生には研究内容に関する議論をはじめとし，勉強会，論文執筆，発表資料作成でのご助言など，大変多くのご助力をいただきました．先生との研究活動を通し，社会や技術の問題点を発見し解決策を吟味する力が磨かれるなど，人間的に成長することができました．

同じ研究室の大学院生である加瀬裕也さんと疋田智也さんにも，研究活動の様々な場面で多くのご助力をいただきました．加瀬さんに研究活動の詳細なスケジュールを作成していただいたことで，円滑な研究活動を行うことができました．疋田さんに発表資料や本論文を細かく査読していただいたことで，より伝わりやすい文章を作成することができました．

同じ研究室の学部 4 年生の方々とは建設的な勉強会を行うことができ，勉強会から多くの研究のヒントを得ることができました．また，彼ら・彼女らとともに幾度も発表練習を行ったことで，より良い発表を行うことができました．

両親には本学での大学生活を金銭面，生活面で支えてもらい，また多くの悩み事を聞いてもらいました．

ご多忙な日々の中，貴重な時間を割いていただいた皆さまに心から感謝申し上げます．

参考文献

[1] RSF. 2021 World Press Freedom Index: Journalism, the vaccine against disinformation, blocked in more than 130 countries. https://rsf.org/en/2021-world-press-freedom-index-journalism-vaccine-against-disinformation-blocked-more-130-countries (2021 年 7 月 19 日参照).

[2] Eli Pariser. Beware online "filter bubbles". https://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles (2022 年 1 月 6 日参照).

[3] Axel Bruns. Filter bubble. *Internet Policy Review*, Vol. 8, No. 4, November 2019.

[4] Jing Yang, Didier Vega-Oliveros, Tais Seibt, and Anderson Rocha. Scalable Fact-checking with Human-in-the-Loop. In *2021 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, Montpellier, France, December 2021. IEEE.

[5] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems – Survey and roads ahead. *Information Processing & Management*, Vol. 54, No. 6, pp. 1203–1227, November 2018.

[6] C. Thi Nguyen. ECHO CHAMBERS AND EPISTEMIC BUBBLES. *Episteme*, Vol. 17, No. 2, pp. 141–161, June 2020. Publisher: Cambridge University Press.

[7] Michael D. Conover, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Partisan asymmetries in online political activity. *EPJ Data Science*, Vol. 1, No. 1, pp. 1–19, December 2012. Number: 1 Publisher: SpringerOpen.

[8] 笹原和俊. ウェブの功罪. 情報の科学と技術, Vol. 70, No. 6, pp. 309–314, 2020.

[9] Sayooran Nagulendra and Julita Vassileva. Understanding and controlling the filter bubble through interactive visualization: a user study. In *Proceedings of the 25th ACM conference on Hypertext and social media*, HT '14, pp. 107–115, New York, NY, USA, September 2014. Association for Computing Machinery.

[10] 長尾高弘 Aurélien Géron. scikit-learn、Keras、TensorFlow による実践機械学習 第 2 版. 株式会社オライリー・ジャパン, 2020.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016. arXiv: 1409.0473 (2022 年 1 月 7 日参照).

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. p. 11. (2021 年 7 月 24 日参照).

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805 (2021 年 7 月 23 日参照).

[14] 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:1907.11692 [cs]*, July 2019. arXiv: 1907.11692 (2021 年 7 月 25 日参照).

[15] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, Vol. 21, No. 1, p. 6, December 2020.

[16] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. August 2019.

[17] Fionn Murtagh and Pierre Legendre. Ward' s Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*, Vol. 31, No. 3, pp. 274–295, October 2014.

[18] S. Aranganayagi and K. Thangavel. Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, Vol. 2, pp. 13–17, December 2007.

[19] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. UNBERT: User-News Matching BERT for News Recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3356–3362, Montreal, Canada, August 2021. International Joint Conferences on Artificial Intelligence Organization.

[20] Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 440–450, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[21] Piyush Ghasiya and Koji Okamura. Investigating COVID-19 News Across Four Nations: A Topic Modeling and Sentiment Analysis Approach. *IEEE Access*, Vol. 9, pp. 36645–36656, 2021. Conference Name: IEEE Access.

[22] Markus Kreuzthaler and Stefan Schulz. Detection of sentence boundaries and abbreviations in clinical narratives. *BMC Medical Informatics and Decision Making*, Vol. 15, No. 2, p. S4, June 2015.

[23] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 101–108, Online, 2020. Association for Computational Linguistics.

[24] Mark Carlebach, Ria Cheruvu, Brandon Walker, Cesar Ilharco Magalhaes, and Sylvain Jaume. News Aggregation with Diverse Viewpoint Identification Using Neural Embeddings and Semantic Understanding Models. In *Proceedings of the 7th Workshop on Argument Mining*, pp. 59–66, Online, December 2020. Association for Computational Linguistics.