人工知能学会 インタラクティブ
情報アクセスと可視化マイニング研究会(第30回)
SIG-AM-30-09

大規模 Web ニュース記事群における論争発見と論点整理

Discovery and Organization of Controversial Topics in Large-Scale Web News Articles

藤原拓己 [1*]　吉岡真治 [1]
Takumi Fujiwara[1]　Masaharu Yoshioka[1]

[1] 北海道大学大学院情報科学院
[1] Graduate School of Information Science and Technology, Hokkaido University

Abstract: In recent years, a huge number of Web news articles have been distributed by various media around the world. On the other hand, there is a background of increasing use of news aggregation services such as Google News, which can deliver articles personalized for individual interests, and to compensate for their shortcomings, It is desirable to provide means for users to comprehend topics from a bird's-eye view. The purpose of this study is to present information that promotes users' understanding of current affairs by discovering and organizing controversies that exist in articles on a specific topic.

1　はじめに

インターネットの発達によって全世界的に Web ニュース記事の配信が盛んであるが，同じ出来事を報じたものであってもニュースサイトごとに，話題を否定または賛成的どのような立場から発信するのかが異なる場合がある．一方でユーザーの立場においては，これらの話題に関するすべてのニュースサイトを閲覧することが困難であるため，Google ニュースや SmartNews などのニュースアグリゲーションサイトやサービスなどが提供する記事のみを閲覧する場合が増えている．しかしそのようなサービスは，近年フィルターバブルやエコーチェンバーという名称で指摘されるように，各ユーザーの興味に類する記事の提供しやすいといった性質があるため，それとは対照的に話題を俯瞰的に捉える手段を提供することができれば，世相を理解する有用な方法となる．

この問題に対し，我々は，世界中の様々なサイトから発信されるニュース記事群から，特定の話題に関する記事を抽出し，その賛否の分布からニュースサイトのスタンスを分析する研究が行なってきた [1, 2]. 本研究ではこれらの研究を発展させ，これらのニュースサイトで扱うサブトピックを中心として，どのような〈論争〉が存在するのかを整理することで俯瞰的な理解につながる情報を提供することを目的とする．

論争という語は一般にある話題について対立する意見や主張による争いが生じている状況を意味するが，本研究においては特に，特定の話題をより細かく分割したサブトピックにおいて賛否が対立するような意見が量的に均衡して存在しているようなものについて〈論争〉が生じているものとして扱う．

本研究では，これまでの研究 [1, 2] と同様に，全世界で配信されるニュースのメタデータを配信しているサイトである The GDELT Project (以下，GDELT と略す) [1]のデータを利用し，特定の話題に関する大規模 Web ニュース記事群を収集する. これらの特定の話題に関する記事群からサブトピックを構築し, 賛否を示す意見の偏りなどを分析することによって,〈論争〉が起きているサブトピックの発見に加えて，その内部に存在する代表的な意見 (論争文) を提示する方法を提案する．

また，GDELT の記事データからサブトピックの抽出と分析を行なった事前検討を行なった際に，反対または賛成の意見が大部分のサブトピックが構成されることが確認された．その中身について確認したところ，その記事群には同一性のある記事が多数存在し，その影響を軽減する必要がある．そのため本研究では冗長性の除去という観点で，記事群の前処理方法についても提案を行なう．

*連絡先：北海道大学大学院情報科学院
〒 060-0814 北海道札幌市北区北 14 条西 9 丁目
E-mail: tfujiwara@kb.ist.hokudai.ac.jp

[1]https://www.gdeltproject.org/

2 関連研究

SNS 上における論争の説明を試みる研究 [3] が存在する．これでは Twitter を対象として，論争の生じている話題についてその論争を説明するような最もよい要約ツイートを発見することを目的としている．その際，ツイート群を対立するようなスタンスを基準に二分する必要があり，そのために特定の話題を反映したようなハッシュタグをもつツイート群に対して，ツイート間のリツイート関係によって作成されるグラフから分割を求める手法 [4] が用いられており，本稿で扱う〈論争〉とはその性質において差異がある．

ニュース記事に限定しない Web 文書を対象とした関連研究として，ユーザーが入力した任意のクエリに対して，Web 上の文書情報を対象にその情報発信者や賛否意見などの情報を提供するために，WISDOM[5] というシステムや，クエリに関係するトピックについての文書群から，そこに含まれる言論を意味的に整理することで，上記の情報にくわえて多様な情報をユーザーに提供する言論マップ生成 [6] の手法が提案されている．これらは，Web 上の大規模な文書情報を対象とし，ある話題についての俯瞰的な理解につながる情報をユーザー提供する手法という意味において本研究と類似性があるが，広範に論争のあるようなサブトピックを探すことは行なっていない．

3 ニュース記事群からの論争発見と論点整理

3.1 概要

本研究では Web ニュース記事の収集のために，世界中のニュース記事からその記事中に現れる固有名詞などのメタデータを付与して公開されている GDELT の GKG というデータベースを参照することによって行なう．また，本文の抽出には，Newspaper3k[2]を用いた．さらにそこに存在する記事本文をトピックごとのクラスターへ分割する際には Top2Vec[7] を用いる．Top2Vec では文書の分散表現を得る手法 [8] の実装である Doc2vec[3]などによって入力文書とそこに含まれる語の分散表現を作成し，UMAP[9] によって次元削除を行なったのち，DBSCAN[10] や DENCLUE[11] といったクラスタリング手法を性能改善したものである HDBSCAN[12] を適用し，分散表現の密集箇所を検出することでトピックの発見を行なう．各トピッククラスターの重心をトピックベクトルとして，これと任意の文書のコサイン類似度がその文書のトピックに対するスコアとして算出される．また Top2Vec ではトピッククラスター同士を統合することで任意の個数にそのクラスター数を削減する階層的トピックの機能を提供する．

そして,〈論争〉の発生状況を考慮するために，記事文書に賛否情報を付与する必要があるが，これには畳み込みニューラルネットワークを用いた手法 [13] を用いている Python の自然言語処理パッケージ Stanza の Sentiment Analysis 機能を用いる．

これらの要素を用いて，論争発見と論点整理に有用なクラスターを構築することで,〈論争〉の内容を具体化する．

3.2 記事群に対する前処理

先にも述べたように,GDELT から取得される記事群には，同一性のある記事が複数存在するという問題がある．本章では，そのような記事の取り扱いとサブトピック分析を行なう際の関連キーワード記事群の選択や，収集した記事から意見性の高い文章を選択するための手法について紹介する．

3.2.1 ニュースアグリゲーションサイトの削除

ニュースサイトのうち，様々なスタンスをもつサイトから記事を寄せ集めたものであるニュースアグリゲーションサイトは，記事に意見性が存在しないと考えられるため，収集された記事から取り除いた．また，“freerepublic.com” などといったフォーラムサイトも同様のものとみなした (表 1) ．

GKG から収集された記事群からこれらのサイト発行のものを取り除いた記事群を分析対象記事群 A とした．

表 1: アグリゲーションサイトとみなしたサイト

サイト名	
au.finance.yahoo.com	au.news.yahoo.com
au.tv.yahoo.com	ca.finance.yahoo.com
ca.news.yahoo.com	ca.sports.yahoo.com
ca.style.yahoo.com	finance.yahoo.com
freerepublic.com	in.finance.yahoo.com
in.news.yahoo.com	news.yahoo.com
nz.finance.yahoo.com	nz.news.yahoo.com
ph.news.yahoo.com	sg.finance.yahoo.com
sg.news.yahoo.com	uk.finance.yahoo.com
uk.news.yahoo.com	uk.style.yahoo.com
www.yahoo.com	www.msn.com

[2]https://newspaper.readthedocs.io/
[3]https://radimrehurek.com/gensim/models/doc2vec.html

3.2.2 記事群の冗長性削減

ニュースサイトには通信社や系列メディアが存在し，すなわちニュース記事には配信元・配信先で同一性のある記事が配信される場合がある．このような親子関係のモデルが生じているときに，本研究では配信元のサイトを Provider Site と呼ぶこととし，記事群から，そのようなサイトの抽出を試みた．

Provider Site が求められたならば，重複している記事群が存在する場合に，Provider Site の発行した記事の 1 つを意見を発した代表記事と定めることで，記事の重複性排除にくわえて，サイトごとの意見を分析する際に，サイト別記事群のサイズを大きくすることによってより確実な分析が可能となるほか，同一のサイト内同一記事によって生じる重複性も取り除くことができる．分析対象記事群 A について，記事同士の同一性に注目したグラフ $G_A = (A, E_A)$ を作成した．記事同士が同一性をもつ条件として，タイトルまたは本文 (の一部) が等しいということに注目した．本文については，ニュース記事上に掲載されている記事本文中には記事の内容を示す文字列に加えて，それぞれの記事に固有な広告や警告文が含まれる場合があるため，以下のように記事 a_i $(\in A)$ を部分文字列に分割した文字列集合 L_{a_i} を考慮した．ただし Printable は入力された文字列について英文印字可能な文字[4]を同じ順で抜き出した文字列を返す関数で，PSplit は，ピリオド (“.”) で区切った文字列集合を返すものである．

$$L_{a_i} = \{\mathrm{Printable}(t) \mid t \in \mathrm{PSplit}(a_{i,\mathrm{text}})\}. \quad (1)$$

記事同士の比較対象としての部分文字列 $l_{a_i,j}$ $(\in L_{a_i})$ を十分大きいものに限定するため，および G_A 上の辺 E_A 作成時の計算量を削減する目的で部分文字列から一定の文字列長 t_{len} をもつものを L'_{a_i} として採用した．

$$L'_{a_i} = \{l_{a_i,j} \in L_{a_i} \mid |l_{a_i,j}| \geq t_{\mathrm{len}}\}. \quad (2)$$

ここで，t_{len} は $l_{a_i,j}$ の平均長の整数部分とした．

$$t_{\mathrm{len}} = \left\lfloor \frac{\sum_i \sum_j |l_{a_i,j}|}{\sum_i |L_{a_i}|} \right\rfloor \ (l_{a_i,j} \in L_{a_i}). \quad (3)$$

E_A は無向辺であり，

$$P_{\mathrm{iden}}(a_i, a_j) \colon a_{i,\mathrm{title}} = a_{j,\mathrm{title}} \vee |L'_{a_i} \cap L'_{a_j}| \geq t_{\mathrm{elem}} \quad (4)$$

$$E_A = \left\{(a_i, a_j) \in A^2 \;\middle|\; i < j \wedge P_{\mathrm{iden}}(a_i, a_j)\right\} \quad (5)$$

[4] ここでは，英アルファベット大文字・小文字，アラビア数字および記号 (`!"#$%&'()*+,-./:;<=>?@[\]^_‘{|}~`)，空白とした．

つまり，相異なる記事同士を比較したときに，そのタイトル文字列が等しい，または L'_{a_i} と L'_{a_j} とで部分文字列が t_{elem} 個以上共通する[5]場合に辺が定義される．このように，グラフ上で同じ辺を共有している場合にそれらの記事同士に同一性があるものとする．ただし t_{elem} は以下のようにした．

$$t_{\mathrm{elem}} = \max\left\{\min\left\{|L'_{a_i}|, |L'_{a_j}|\right\}, 1\right\}. \quad (6)$$

Provider Site は複数のオリジナル記事を作成し，配信先のサイトはそれらから選択的に記事を受け取り，掲載していると考えられる．つまり，記事同一性グラフ G_A 上で考えた場合，Provider Site の方がより複数の連結成分の記事の発行元としてみられるものとして G_A から Provider Site を抽出する方法を考案した (Algorithm 1)．これを行なうためにまず，G_A を連結成分ごとに分割した部分グラフ集合 $C_A = \{G_1, G_2, \ldots, G_n\}$ について，G_i に含まれるサイト群

$$S_i = \{s \mid a_j \in G_i,\ s = a_{j,\mathrm{site}}\} \quad (7)$$

を考える必要がある．

さらに，G_i $(\in C_A)$ にただ 1 つ対応する記事 α_i $(\in G_i)$ を代表記事とする．代表記事群

$$A_{\mathrm{repr}} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \quad (8)$$

を決定するために，得られた Provider Site 列を S_{pro} を用いて Algorithm 2 を実行した．これでは各連結成分 G_i が $|G_i| > 1$ の場合に，Provider Site 列に存在していて，より列の先頭の位置にあるサイトが発行元となるような記事を代表記事 α_i として採用する．なお Algorithm 1 より，$|G_i| > 1$ のとき $a_{j,\mathrm{site}} \in S_{\mathrm{pro}}$ $(a_j \in G_i)$ となるような記事 a_j が存在することが保証される．

Algorithm 1 Provider Site 列を抽出するアルゴリズム

```
1: procedure CREATE-PROVIDER-SITE-RANKING(C_A)
2:     Initialize h_freq ▷ A hash of a site and its frequency
3:     for all s ∈ S do
4:         C_s ← {G_j ∈ C_A | s ∈ S_j}
5:     end for
6:     for all G_i ∈ C_A do
7:         if |G_i| > 1 then
8:             s_max ← SELECT-RANDOM-ONE(argmax_t |C_t|)
9:             h_freq(s_max) ← h_freq(s_max) + 1
10:        end if
11:    end for
12:    S_pro ← GET-KEYS(h_freq)
13:    SORT-DESC-BY-FREQ(S_pro, h_freq)
       return S_pro
14: end procedure
```

[5] 文字列比較の際は計算量削減のため，実装において C++標準ライブラリの `std::hash<std::string>` 構造体で作成したハッシュ値による比較を行なった．

Algorithm 2 代表サイト群 A_{repr} を決定するアルゴリズム

```
procedure EXTRACT-REPR-ARTICLES(C_A, S_pro)
    Initialize A_repr = ∅
    for all G_i ∈ C_A do
        if |G_i| = 1 then
            A_repr ← A_repr ∪ {a_j} (a_j ∈ G_i)
            continue
        end if
        for all s ∈ S_pro do
            for all a_j ∈ G_i do
                if a_{j,site} = s then
                    A_repr ← A_repr ∪ {a_j}
                    go to 15
                end if
            end for
        end for
    end for
    return A_repr
end procedure
```

3.3 論争発見と論点整理

Top2Vec は，代表的な話題を見つけることができる一方，話題に関する記事数のばらつきが大きく，話題が小さくなりすぎると，反対または賛成の意見がほとんどのサブトピックなどが構成されることがあった．本研究ではこれらのサブトピックを賛否の異なる類似したサブトピックを併合していくことで，共通の話題に対する賛否が存在するようなサブトピックの作成を行なうことで,〈論争〉の内容を具体化する方法を提案する．

3.3.1 入力対象記事文書の決定

Top2Vec への入力対象とする記事群 A_{inp} を以下のように定めた．

$$A_{\text{inp}} = A_{\text{repr}} \cap A_{\text{kwd}}. \tag{9}$$

ただし，A_{kwd} は任意のキーワードに関連する記事群であり，A に含まれる記事であって，記事の参照元の各 GKG のデータにおいてそのフィールドのいずれかにキーワードが部分文字列として含まれる場合に，その記事は A_{kwd} に含まれる．

そして，文章の文単位への分割には高速動作を期待し，Python の自然言語処理ライブラリである spaCy の Tokenizer の機能を用いた．また，各文には Stanza の Sentiment Analysis 機能により，賛否情報を示す 0, 1, 2 のいずれかの値を付与した．

ここで，分割された文について意味のあるものを扱うため，文の構成トークン数が 10 個以上，かつ文に動詞が含まれるようなものを入力対象とした．各トークンへは Stanza によって分割するとともに POS タグを付与したが，このうち “VERB” (動詞) および “AUX” (助動詞) のタグをここでは動詞と定義して扱った．

3.3.2 Top2Vec によるトピックモデリング結果の利用

記事 a_i ($\in A_{\text{inp}}$) に含まれる，3.3.1 で得られた文を含む文書集合 $\mathcal{D}_{a_i} = \{D_{a_i,1}, D_{a_i,2}, \ldots, D_{a_i,n}\}$ について，すべての入力文集合 $\mathcal{D}_{\text{inp}} = \bigcup_i \mathcal{D}_{a_i}$ を Top2Vec に入力する[6]ことで，文あるいはそこに含まれる単語による分散表現を得て，さらにそれをトピックごとに分割するクラスタリングを行なった．ただし，$D_{a_i,j}$ は文 $d_{a_i,j}$ とその感情値 $c_{a_i,j}$，サイト名 $s_{a_i,j}$ を含む列 $(d_{a_i,j}, c_{a_i,j}, s_{a_i,j})$ である．

Top2Vec で $\mathcal{D}_{\text{inp}}$ を対象にトピックモデリングを行ない，トピックを示すクラスターを含む，下位トピッククラスター群 $\mathcal{T}_{\text{lower}} = \{T_i, T_j, \ldots\}$ ($T_i = \{D_{a_k,l}, D_{a_m,n}, \ldots\}$) を作成した．さらに Top2Vec のトピック削減機能により，$|\mathcal{T}_{\text{upper}}|$ を任意の値に定めたうえで $|\mathcal{T}_{\text{upper}}| < |\mathcal{T}_{\text{lower}}|$ となるような上位トピッククラスター群 $\mathcal{T}_{\text{upper}} = \{T_k, T_l, \ldots\}$ を得た．ここで，$\mathcal{T}_{\text{upper}}$ の T_i が $\mathcal{T}_{\text{lower}}$ の T_j を包含する場合，T_j は T_i のサブトピックであるとみなし，サブトピック関係 R_{sub} を定義した．

$$R_{\text{sub}} = \{(T_i, T_j) \in \mathcal{T}_{\text{upper}} \times \mathcal{T}_{\text{lower}} \mid T_i \supseteq T_j\}. \tag{10}$$

3.3.3 上位トピッククラスターに関する代表的な賛否文の発見

論争文の発見を行なう前に，任意の上位トピッククラスター関して賛否いずれかを代表するような文の発見を行なった．

T_i のサブトピッククラスター群 $\mathcal{T}_i = \{T_j \mid (T_i, T_j) \in R_{\text{sub}}\}$ について, ブトピッククラスターに存在している文書をサイト s ごとに分割した集合

$$T_{i,s} = \left\{ D_j \in \bigcup_{T \in \mathcal{T}_i} T \;\middle|\; s \in D_j \right\} \tag{11}$$

を用いて，否定派サイト群 $S_{i,\text{neg}}$ および賛成よりサイト群 $S_{i,\text{pos}}$ を得た．

$$S_{i,\text{neg}} = \{s \mid n_{\text{neg}}(T_{i,s}) > n_{\text{pos}}(T_{i,s})\} \tag{12}$$

$$S_{i,\text{pos}} = \{s \mid n_{\text{pos}}(T_{i,s}) > n_{\text{neg}}(T_{i,s})\}. \tag{13}$$

これより，否定的文書群および賛成的文書群を求めた．否定的文書群 $\mathcal{D}_{i,\text{neg}}$ は，

$$\mathcal{D}_{i,\text{neg}} = \left\{ D_j \in \bigcup_{T \in \mathcal{T}_i} T \;\middle|\; s, c \in D_j,\ s \in S_{i,\text{neg}} \wedge c = 0 \right\} \tag{14}$$

[6] Top2Vec (Ver. 1.0.26) の実装では，文書として文字列が対象のため，実装上，$\left(c_{a_i,j}, s_{a_i,j}\right)$ が対応づけられた $d_{a_i,j}$ を入力した．

とし，賛成的文書群 $\mathcal{D}_{i,\mathrm{pos}}$ は (14) 式において条件を $c = 2$ としたものとして定義した．

文書 D に対する文書ベクトル v_D を用いて，$T_{i,\mathrm{neg}}$ に属する文書ベクトルの重心を上位トピック T_i ($\in \mathcal{T}_{\mathrm{upper}}$) に対する否定的文書群重心 $v_{i,\mathrm{neg}}$ とし，

$$v_{i,\mathrm{neg}} = \frac{\sum_{D \in \mathcal{D}_{i,\mathrm{neg}}} v_D}{|\mathcal{D}_{i,\mathrm{neg}}|} \tag{15}$$

賛成的文書群重心も同様に定義した．ただし文書ベクトルは次元削減前の次元空間のものである．

否定的文書群重心 $v_{i,\mathrm{neg}}$ に対してそのベクトルが最もコサイン類似度の高くなるように文書群 $\mathcal{D}_{i,\mathrm{neg}}$ を並べたものを上位トピック T_i ($\in \mathcal{T}_{\mathrm{upper}}$) に対する否定的代表文群の一覧とした (ただし一覧のうち，文書に同じサイトまたは文が重複して現れる場合はその類似度が最も大きいものを残し，重複分を削減した)．賛成的代表文の一覧も同様にして求めた．

3.3.4 賛否の均衡を考慮したトピッククラスターの再構成

T_i ($\in \mathcal{T}_{\mathrm{upper}}$) のサブトピッククラスター群においてクラスター同士の併合を行なうことで，賛否文数の比率がなるべく均等となるようなクラスターの作成方法を考案した (Algorithm 3)．クラスター T の賛否特性として行 8 にあるように評価値 $p(T)$ を定義した．$p(T)$ は賛成的な文書数 $n_{\mathrm{neg}}(T)$ に比して否定的な文書数 $n_{\mathrm{pos}}(T)$ がどの程度存在するのかという極性を文書数 $|T|$ によって正規化したものである．

$$p(T) = \frac{n_{\mathrm{neg}}(T) - n_{\mathrm{pos}}(T)}{|T|}. \tag{16}$$

クラスターの統合を繰り返し行なうと，極性の大きく偏った 1 つのクラスターへと肥大化する事象が生じることを確認したため，統合されたクラスターのサイズが t と $\mathcal{T}_{\mathrm{sub}}$ によって定まる閾値以上となった場合にこれを取り除くようにした (行 14, 15)．

3.3.5 サイトごとの賛否状況を考慮した論争の検出

Web ニュース記事の配信元であるニュースサイトは，記事本文中の文書の内容の発言者であると考えられるため，サイトごとの否定または賛成派グループを手がかりとして論争の検出を行なう．

T_i($\in \mathcal{T}_{\mathrm{upper}}$) の内部でのように再構成されたサブトピッククラスター群 $\mathcal{U}_i = \{U_{i,1}, U_{i,2}, \ldots, U_{i,n}\}$ について，$U_{i,j}$ 内で文書の発行元であるサイトが s であるような文書集合

$$V_{i,j,s} = \{D_{a_k,l} \in U_{i,j} \mid s \in D_{a_k,l}\} \tag{17}$$

を作成し，

$$\mathcal{V}_{i,j,\mathrm{neg}} = \{V_{i,j,s} \mid n_{\mathrm{neg}}(V_{i,j,s}) > n_{\mathrm{pos}}(V_{i,j,s})\} \tag{18}$$

$$\mathcal{V}_{i,j,\mathrm{pos}} = \{V_{i,j,s} \mid n_{\mathrm{pos}}(V_{i,j,s}) > n_{\mathrm{neg}}(V_{i,j,s})\} \tag{19}$$

として，比 $|\mathcal{V}_{i,j,\mathrm{neg}}|/|\mathcal{V}_{i,j,\mathrm{pos}}|$ に，$U_{i,j}$ における賛否それぞれの文数の比 $n_{\mathrm{pos}}(U_{i,j})/n_{\mathrm{neg}}(U_{i,j})$ を乗算することで補正した値である

$$r(U_{i,j}) = \frac{n_{\mathrm{pos}}(U_{i,j}) \cdot |\mathcal{V}_{i,j,\mathrm{neg}}|}{n_{\mathrm{neg}}(U_{i,j}) \cdot |\mathcal{V}_{i,j,\mathrm{pos}}|} \tag{20}$$

を定義した．さらに

$$r'(U_{i,j}) = \begin{cases} r(U_{i,j}) & (r(U_{i,j}) \leq 1) \\ 1/r(U_{i,j}) & (r(U_{i,j}) > 1) \end{cases} \tag{21}$$

としたときに以下が真，偽となる場合にそれぞれ $U_{i,j}$ において論争が生じている，またはこれが生じていないと定義した．ただし $U_{i,j}$ に対して $r'(U_{i,j})$ が定義されない場合は $P_{\mathrm{cont}}(U_{i,j})$ を偽とする．

$$P_{\mathrm{cont}}(U_{i,j})\colon r'(U_{i,j}) \geq r_{\mathrm{cont}}. \tag{22}$$

3.3.6 論争の起こっているクラスターからの論争文の発見

$P_{\mathrm{cont}}(U_{i,j})$ が真となる $U_{i,j}$ について，論争文の発見を行なった．$U_{i,j}$ から，否定ラベルが付与された文書群をそれらスコアの降順に並べたものを否定的な論争文列とした．賛成的な論争文列も同様にして定義した．ただし，分析対象記事群 A の冗長性削減されない分も考慮し，それぞれの論争文列において，文書はスコアの最も高いものを残すように重複分を削除した．

Algorithm 3 サブトピッククラスター再構成アルゴリズム

1: **procedure** REMODEL-SUBTOPIC-CLUSTERS-IN-TOPIC(T_i, t)
2: Initialize $\mathcal{T}_{\mathrm{popped}} = \emptyset$
3: $\mathcal{T}_{\mathrm{sub}} \leftarrow \{T_j \mid (T_i, T_j) \in R_{\mathrm{sub}}\}$
4: $\mathcal{T}_{\mathrm{cur}} \leftarrow \mathcal{T}_{\mathrm{sub}}$ ▷ Create a copy not a reference
5: **repeat**
6: **for all** $T \in \mathcal{T}_{\mathrm{cur}}$ **do**
7: $(n_{\mathrm{neg}}(T), n_{\mathrm{pos}}(T)) \leftarrow$ COUNT-SENTIMENT-LABELS(T)
8: $p(T) \leftarrow (n_{\mathrm{neg}}(T) - n_{\mathrm{pos}}(T))/|T|$
9: **end for**
10: $T_{\min} \leftarrow$ SELECT-RANDOM-ONE $\left(\underset{T \in \mathcal{T}_{\mathrm{cur}}}{\operatorname{argmin}} p(T)\right)$
11: $T_{\max} \leftarrow$ SELECT-RANDOM-ONE $\left(\underset{T \in \mathcal{T}_{\mathrm{cur}}}{\operatorname{argmax}} p(T)\right)$
12: $T_{\mathrm{merged}} \leftarrow T_{\min} \cup T_{\max}$
13: $\mathcal{T}_{\mathrm{cur}} \leftarrow \mathcal{T}_{\mathrm{cur}} \setminus \{T_{\min}, T_{\max}\}$
14: **if** $|T_{\mathrm{merged}}| \geq \left(\sum_{U \in \mathcal{T}_{\mathrm{sub}}} |U|\right)/t$ **then**
15: $\mathcal{T}_{\mathrm{popped}} \leftarrow \mathcal{T}_{\mathrm{popped}} \cup \{T_{\mathrm{merged}}\}$
16: **else**
17: $\mathcal{T}_{\mathrm{cur}} \leftarrow \mathcal{T}_{\mathrm{cur}} \cup \{T_{\mathrm{merged}}\}$
18: **end if**
19: **until** $|\mathcal{T}_{\mathrm{cur}}| \leq 1$
return $\mathcal{T}_{\mathrm{popped}} \cup \mathcal{T}_{\mathrm{cur}}$
20: **end procedure**

4 実験

3.2 および 3.3 の手法を用いて，具体的な条件設定のもとで実験を行なった．

4.1 記事群に対する前処理

収集する記事については，“Biden”, “Putin”, “Ukrine” または “Zelensky” のいずれかをキーワードとして含む英文で記述されたもので，2022 年 1 月 1 日から 4 月 30 日の間に GKG のデータベースにおいて集計されたレコードから取得されるものを対象とした．それらよりニュースアグリゲーションサイトの発行した記事を取り除いた分析対象記事群 A の個数 $|A|$ は 477,689 であった．

また，A から抽出された Provider Site 列は初項から，$(\text{site}, h_{\text{freq}}(\text{site})) = (\text{www.europesun.com}, 2{,}100)$, (www.breitbart.com, 1,025), (www.newsbusters.org, 904), (apnews.com, 875), (www.cnsnews.com, 818), ... となった (参考として h_{freq} の値も示した)．

4.2 論争発見と論点整理

4.2.1 分析対象記事の決定

2022 年の出来事として 2 月に起きたロシアによるウクライナ侵攻がある．このことに際してウクライナ側の対応が注目され，当時のウクライナ第 6 代目大統領のウォロディミル・ゼレンスキーの発言や振る舞いがメディアに取り上げられた．実験においては，この出来事およびゼレンスキーについての記事群を対象とすることを目的として，キーワードとして “Zelensky” を含む記事群 A_{kwd} をもとに入力記事群 A_{inp} を決定した．各記事数 $|A_{\text{repr}}|$ $|A_{\text{kwd}}|$ $|A_{\text{inp}}|$ はそれぞれ 224,357, 13,913, 12,191 件で，入力記事文書群の個数 $|\mathcal{D}_{\text{inp}}|$ は 568,718 件であった．

4.2.2 下位および上位トピッククラスターの取得

Top2Vec に $\mathcal{D}_{\text{inp}}$ を入力し，下位トピッククラスター群 $\mathcal{T}_{\text{lower}}$ を得た (表 2)．要素数 $|\mathcal{T}_{\text{lower}}|$ は 6,639 であった．さらに，$|\mathcal{T}_{\text{upper}}|$ が 100 となるように Top2Vec の階層的トピック削減を行ない $\mathcal{T}_{\text{upper}}$ を得た (表 3)．Top2Vec は Ver. 1.0.26 の実装[7]を用いて，また実行の際，各種パラメータはデフォルトとした．

[7]https://github.com/ddangelov/Top2Vec/releases/tag/1.0.26

4.2.3 サブトピッククラスターの取得

T_4, T_9 ($\in \mathcal{T}_{\text{upper}}$) のサブトピックを実験対象としたため，それぞれについてのサブトピッククラスターを求めた (T_4 についての結果は表 4)．ただし，それぞれのトピックのサブトピッククラスター T_i について (20) 式と同様に $r(T_i)$ を計算し，その降順に並べた．この値が大きいほど，否定的な意見をもつニュースサイトがクラスターにより多く含まれるようなサブトピックとなっている．

T_4 については，price, inflation, consumers などがキーワードとしてあることから，市場や物価に関連するトピックで，T_9 は，gop, republican, democrats などがみられるので，米国の政党に関するものであることが考えられる (表 2)．

4.2.4 サブトピッククラスターに関する代表的な賛否文の発見

以下ではまず，T_4 ($\in \mathcal{T}_{\text{upper}}$) を対象として実験を行なう．$T_4$ に関する否定的代表文一覧および賛成的代表文一覧を求めた．それによると主な否定的論点としては,「エネルギーや食糧価格の高騰」や「気候変動の抑制のためにエネルギー削減が必要」なことなどについてが挙げられ，賛成的論点としては,「小麦や金属類の価格高騰」や「米国での雇用増加」などがあった．

4.2.5 論争の起こっているクラスターからの論争文の発見

Algorithm 3 によって再構成したサブトピッククラスター群 $\mathcal{U}_4 = \{U_1, U_2, U_3, U_4\}$ を得た (表 5)．このうち論争が生じているとされた U_2 について，否定的または賛成的なな論争文の上位 10 件を調べた．

得られた否定的な論争文としてはほとんどが「インフレのともなうエネルギーや物価の高騰や供給不安」についてであった．T_4 の否定的代表文においても，主にエネルギーや食糧の価格高騰についてが論点となっていたため，発見された否定的論争文においては上位のトピックとの関連性がみられた．

賛成的論争文についても，同様のコストの高騰について述べているものがみられる (この情報だけでは賛否どちらの意見として発せられたかが判断できないが，感情分析の際に「物事が上昇する，高止まりする」といった表現が賛成的であると判定された可能性がある)．一方で，上位トピックの賛成的代表文には存在していなかった，ロシアが原油の輸出を再開したことで原油供給の懸念が緩和された旨の論争文も存在しており，これについては否定的論争文にもあったエネルギーの供給不安といった内容の対となる．

以上により，同じイベントに対して，賛否両面からの論争文が得られることがわかった．しかし，上記の話題はあくまで事実を報じた文であって，論争の対象としては第三者の感情をともなうような意見が期待されるため，提示される論争文とはその性質において不適当であると考えられる．

4.2.6 他トピックを扱った追加実験

T_9 ($\in \mathcal{T}_{\text{upper}}$) に関して，否定的代表文には,「Cawthorn 議員が Zelenskyy に対して過激な発言をした」ことや「そのことが他の議員から批判されている」ことが得られ，および賛成的代表文として「下院の選挙が盛況である」ことや「Cawthorn がノースカロライナ州で最も有名な人物である」ことなどが得られた．

そして T_9 のサブトピックを再構成したトピッククラスター群 $\mathcal{U}_9 = \{U_1, U_2, U_3, U_4, U_5\}$ を作成し，このうち論争が生じていると判定された U_4 について否定的または賛成的な論争文の発見を行なった．否定的な論争文としては，否定的代表文と同様のものが多く見られたが，賛成的な論争文について,「Cawthorn が 25 歳で共和党下院予備選に勝利した」ことや「Cawthorn が 1 期目にして多くの注目を集めた」などといった内容がみられた．

この結果より，否定または賛成的代表文でも Cawthorn について異なる賛否述べていた内容について扱われていたが，得られた否定的または賛成的代表文では，同氏についてより詳細な形で賛否両面からの論点が現れたと考えられる．

4.2.7 考察

上位トピック T_4 はキーワードとして price, inflation, consumers など，事柄や概念についてで，T_9 は gop, republican, democrats など，具体的な組織を表していると考えられる．T_4 での結果としては，論争文として得られたものが物価の高騰などの出来事について述べたもので，論争として望まれるような性質を持ったものが発見されなかった．これに対して，T_9 では，具体的な人物に対しての批判的あるいは賞賛的な論争文が発見され，これは論争として適当であった．具体的な組織や人物に関するトピックであれば，論争文として妥当なものが得られると考えられる．これは，特定の人物に関して述べた文は，その人物に向けた感情が表出しやすいためと考えられる．

以上より，トピックによっては論争が起こっていると判定されるものであっても，論争文の妥当性がないもの存在することが考えられる．

表 2: 下位クラスター T_i ($\in \mathcal{T}_{\text{lower}}$) のトピック例

i	T_i のキーワード	$n_{\text{neg}}(T_i)$	$n_{\text{pos}}(T_i)$	$\lvert T_i \rvert$
1	know, you, going, ...	130	14	927
2	nord, stream, pipeline, ...	281	17	865
3	stinger, javelin, armor, ...	236	16	853
4	olaf, scholz, chancellor, ...	162	24	795
5	iaea, mariano, grossi, ...	311	17	759
⋮				
6638	commitments, securing, independence, ...	0	0	10
6639	venezuelan, withdrawals, barrels, ...	0	8	8

表 3: 上位クラスター T_i ($\in \mathcal{T}_{\text{upper}}$) のトピック例

i	T_i のキーワード	$n_{\text{neg}}(T_i)$	$n_{\text{pos}}(T_i)$	$\lvert T_i \rvert$
1	know, think, going, ...	2,690	410	14,021
2	know, you, me, ...	2,702	655	12,375
3	kharkiv, city, miles, ...	2,418	133	9,195
4	prices, inflation, consumers, ...	2,702	731	8,808
⋮				
9	gop, republican, republicans, ...	2,394	413	8,369
⋮				
99	crystal, inch, waived, ...	879	281	3,866
100	kabul, debacle, taliban, ...	951	227	3,856

表 4: サブトピッククラスター T_i ($(T_4, T_i) \in R_{\text{sub}}$) のトピック ($r(T_i)$ の降順)

i	T_i のキーワード	$n_{\text{neg}}(T_i)$	$n_{\text{pos}}(T_i)$	$\lvert T_i \rvert$	$r(T_i)$
4645	minimum, wage, paycheck, ...	14	2	48	1.7143
4831	workforce, afford, care, ...	22	4	45	1.4545
1985	spike, prices, inflation, ...	31	7	98	1.4113
1039	chains, disruptions, chain, ...	42	6	128	1.3810
1770	communities, resources, rural, ...	12	3	103	1.3750
⋮					

表 5: T_i ($(T_4, T_i) \in R_{\text{sub}}$) を再構成したサブトピッククラスター U_j ($t = 5,\ r_{\text{cont}} = 0.8$)

j	$r'(U_j)$	$n_{\text{neg}}(U_j)$	$n_{\text{pos}}(U_j)$	$\lvert U_j \rvert$	$P_{\text{cont}}(U_j)$	U_j の構成要素 ($T_i \in (\mathcal{T}_{\text{lower}})$)
1	0.5486	362	79	1,144	False	T_{23}, T_{505}, T_{974}, T_{3032}, T_{3396}, T_{4680}, T_{5348}, T_{5672}, T_{6354}, T_{6517}
2	0.8340	362	67	1,119	True	T_{1039}, T_{1240}, T_{1770}, T_{1848}, T_{1915}, T_{1995}, T_{2666}, T_{2800}, T_{3248}, T_{4645}, T_{4754}, T_{4831}, T_{5757}, T_{6182}, T_{6290}, T_{6415}
3	0.8034	335	70	1,004	True	T_{319}, T_{506}, T_{781}, T_{2211}, T_{2583}, T_{3056}, T_{3530}, T_{3573}, T_{4548}, T_{5062}, T_{6387}
4	0.5416	399	75	1,230	False	T_{602}, T_{957}, T_{1333}, T_{1460}, T_{1748}, T_{1985}, T_{2242}, T_{2998}, T_{3142}, T_{3560}, T_{4353}, T_{4368}, T_{5847}, T_{6245}, T_{6265}, T_{6377}

5 むすび

本研究では GDELT を参照し収集された Web ニュース記事群に対して，そこに存在する重複性を削減したうえで，Top2Vec によって作成されるトピッククラスター群を加工したものを利用して〈論争〉の発見および論点の整理を試みた．実験においては，個別具体的

なトピックについて論争文の発見と整理を行なったところ，本研究が望む〈論争〉の発見には扱うトピックの性質が影響することが示唆された．

今後の展望としては，Top2Vec で取得されるトピッククラスターをより記事文書の内容に即すようにそのサイズや分割を厳密に決定することや，文書への賛否情報付与の際に文の意見性と賛否情報の整合性をよりとれるような方法を考案することなどが考えられる．

参考文献

[1] Masaharu Yoshioka, Myungha Jang, James Allan, and Noriko Kando. Visualizing polarity-based stances of news websites. *NewsIR@ ECIR*, 2079:6–8, 2018.

[2] 立浪紀彦. ユーザの興味を反映したニュースサイトの多観点スタンス分析, 2020. 北海道大学大学院情報科学院修士論文.

[3] Myungha Jang and James Allan. Explaining controversy on social media via stance summarization. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1221–1224, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27, 2018.

[5] 赤峯享, 宮森恒, 加藤義清, 中川哲治, 乾健太郎, 黒橋禎夫, and 木俵豊. Web 情報の信頼性検証のための情報分析システム wisdom. **言語処理学会第 *14* 回年次大会論文集 *[*本論文集*]***, 2008.

[6] 水野淳太, 渡邉陽太郎, 村上浩司, 松吉俊, 大木環美, 乾健太郎, and 松本裕治. 言論マップ生成技術の現状と課題. **言語処理学会第 *17* 回年次大会講演論文集**, pages 49–52, 2011.

[7] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[8] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[9] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[10] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[11] Alexander Hinneburg and Hans-Henning Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. In *International symposium on intelligent data analysis*, pages 70–80. Springer, 2007.

[12] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[13] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.