JAIST Repository
https://dspace.jaist.ac.jp/

Title	他者のコメントの引用を考慮したオピニオンマイニング
Author(s)	岡山，有希
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11340
Rights	
Description	Supervisor:白井清昭，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

他者のコメントの引用を考慮した オピニオンマイニング

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

岡山 有希

2013 年 3 月

修 士 論 文

他者のコメントの引用を考慮した オピニオンマイニング

指導教官　白井清昭 准教授

審査委員主査　白井清昭 准教授
審査委員　島津明 教授
審査委員　東条敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1110015 岡山 有希

提出年月: 2013 年 2 月

Copyright © 2013 by Okayama Yuki

概 要

近年では, ブログやマイクロブログといった個人が情報を発信するメディアが一般的となり, ウェブ上から大規模テキストが容易に得られるようになった. ウェブテキストの中でも, 特にブログ記事は, さまざまなトピックについて個人の思想や感情, 評判などを表現しているため, 世間の人々がトピックに対してどのような考えを抱いているのか知ることができる貴重な情報源として利用できる. しかし, ブログ記事ではしばしば他者のコメントが引用される. 引用された他者のコメントは, ブログ著者がトピックに対してどのような立場をとっているのかを自動的に判断する際に誤判定を引き起こす要因となりうる. そこで, 本研究では, あるトピックに対して記述されたブログ記事をウェブから取得し, ブログ記事中から他者の記事やコメントを引用した箇所を検出・除去した後に, ブログの内容がトピックに対して賛成的立場・反対的立場・中立的立場のどれに当てはまるのかを分類し, 賛成・反対の立場をとるブログ件数を集計するとともに, 賛成意見文, 反対意見文をユーザーに提示することで, 世間の意見を俯瞰的に見ることができるシステムを構築することを目指す. オピニオンマイニングに関する過去の研究では, 他者の記事やコメントの引用に対して特別な処理は行われていない. これに対し, 本研究では, トピックに対する賛成・反対意見を集約するオピニオンマイニングシステムにおいて, 他者のコメントの引用を適切に処理する手法を確立することを目的とする. 本研究で提案するシステムは「コメント収集タスク」「引用箇所判定タスク」「極性判定タスク」の 3 つのタスクから構成されている. コメント収集タスクでは, 調査対象とするトピックのブログテキストをウェブから取得する. 引用箇所判定タスクでは, 他のウェブサイトから取得し転載されたテキストを検索し除去する. 極性判定タスクでは, ブログの著者がトピックに対して賛成 (肯定的) か反対 (否定的) か中立のいずれの立場を取るか判定する. 提案システムの評価実験の結果, 他者の記事やコメントの引用箇所を検出, 除去することにより, ブログ記事の極性判定の精度が 3%向上することが確認できた. 個人の意見の集約を図るオピニオンマイニングシステムでは, いかにして著者の意見文を取得するのかが重要である. ブログ記事のように著者の記事と他人の記事やコメントが混在することの多いテキストにおいて, 引用箇所を推定し除去することがオピニオンマイニングの際に有用であるといえる.

目 次

第 1 章　序論 **1**
1.1　研究の背景 1
1.2　研究の目的 1
1.3　論文の構成 2

第 2 章　関連研究 **3**
2.1　評価表現 3
2.1.1　評価表現辞書 3
2.2　オピニオンマイニングシステム 4
2.2.1　OpinionReader 4
2.2.2　調停要約 8
2.2.3　言論マップ 10

第 3 章　提案システム **11**
3.1　概要 11
3.2　システムアーキテクチャ 14

第 4 章　提案手法 **19**
4.1　コメント収集タスク 19
4.1.1　ブログ検索 19
4.1.2　文分割 20
4.2　引用箇所検出タスク 23
4.2.1　ブロック単位の引用箇所の検出/除去 23
4.2.2　文単位の引用箇所の検索/除去 25
4.3　極性判定タスク 27
4.3.1　極性判定対象文の選択 27
4.3.2　文の極性判定 28
4.3.3　ブログの極性判定 29
4.3.4　集計 29

第 5 章　評価 **31**
5.1　評価クエリ 31

5.2 引用箇所検出の評価 31
5.2.1 引用箇所検出の評価方法 31
5.2.2 引用箇所判定の評価結果 32
5.3 文の極性判定の評価 33
5.3.1 文の極性判定の評価方法 33
5.3.2 文の極性判定の評価結果 34
5.4 ブログ極性判定の評価 35
5.4.1 ブログ極性判定の評価方法 35
5.4.2 ブログ極性の評価結果 36

第 6 章 結論 **45**
6.1 結論 45
6.2 今後の課題 45

付 録 A 引用を示唆するキーワード一覧 **49**

第1章　序論

1.1　研究の背景

近年では, ブログやマイクロブログといった個人が情報を発信するメディアが一般的となっている. それに伴い, ウェブ上から大規模テキストが容易に得られるようになった. テキストマイニングは, これら大規模テキストから有用な知見を得るための技術の一つである. また, テキストマイニングを利用しウェブ上の記事から有用な情報を取得する研究が行われている. 例えば,“blogWatcher”[1] はブログを収集, 監視し, 収集したブログをマイニングすることで,「キーワードの月間出現数推移」「ホットキーワード」「評判表現」「おすすめ blog の提案」などといった有用な情報を取得しユーザーに提示するシステムである. また,“OpinionReader”[2] は, 与えられたトピックに対する主観情報を集約し, トピックの論点を示すキーワードを 2 次元グラフ上に表示することで, 賛否両論が対立する構図をわかりやすく可視化するシステムである. さらに, ポータルサイトなどの営利目的を主とするウェブサイトでも, テキストマイニングにより得られた「話題のキーワード」や「人気評判ランキング」などをユーザーに提供するサービスが見られるようになった. このように, ウェブ上の記事をテキストマイニングし有用な情報を得ることは, もはや一般的な技術の一つとなっている. また, ウェブ上の情報の中でも, 特にブログ記事は, さまざまなトピックについて個人の思想や感情, 評判などを表現している. そのため, ブログ記事は, 世間の人々がトピックに対してどのような考えを抱いているのか知ることができる貴重な情報源として利用できる. オピニオンマイニングは, このような意見情報をテキストマイニングする技術であり, これら個人が発信する情報を全体として捉えて, 世間の意見を俯瞰的に見ることができるため, 有用な技術として注目されている.

1.2　研究の目的

本研究では, あるトピックに対して記述されたブログ記事を, ウェブ上から取得し, ブログ記事中から他者の記事やコメントを引用した箇所を検出除去した後に, ブログの内容がトピックに対して賛成的立場・反対的立場・中立的立場のどれに当てはまるのかを分類する. そして, 賛成・反対の立場をとるブログ件数とともに, 賛成意見文, 反対意見文をユーザーに提示することで, 世間の意見を俯瞰的に見ることができるシステムを構築することを目指す. ブログ記事では, 記事を引用して情報を補足したり, 転載した情報に対する意見を表明するケースがたびたびみられる. そのため, ブログ記事は引用された他者のコメント

と著者のコメントが混在しており, ブログ著者がトピックに対してどのような立場をとっているのかを自動的に判断する際に問題となる. 例えば, 原子力発電の再稼働について賛成的立場の引用記事を転載し, その記事に対しブログ著者が反対を表明するコメントをしているブログ記事では, 著者はトピックに対して反対的立場でブログ記事を構成しているのにも関わらず, 賛成的立場の記事の内容からブログ全体が賛成の立場にあると誤認識してしまう可能性がある. また, ブログ記事では著者の意見が記述されておらず, 報道ニュースなどの他者の記事のみを掲載しているものも多い. その際, ブログ著者は間接的に自らの意見を表明しているとも考えられるが, ブログ著者がトピックに対して明示的に意見を表明しているブログと同等に扱っていいか疑問が残る. よって, ブログ記事から他者の記事やコメント除去して分析を行えば, 記事の中でブログ著者自身の意見に重みを持たせた解析が可能となる. 後述するように, 実験の結果からおよそ 60 %のブログ記事に他者の記事の引用・転載があることがわかった. コピー＆ペーストなどによって引用, 転載が容易に行えるため, ウェブ上の記事は情報の補足などに他者の記事やコメントを利用する傾向が強いと考えられる. そのため, 他者の意見の引用部分を正しく取り扱うことは, ウェブを対象としたオピニオンマイニングでは重要な課題である. 本論文の主な目的は, トピックに対する賛成・反対意見を集約するオピニオンマイニングシステムにおいて, 他者のコメントの引用を適切に処理する手法を確立することにある.

1.3 論文の構成

本論文の構成は以下の通りである.
2 章では, 関連研究としてウェブ上の記事に対するテキストマイニングにより有用な情報をユーザーに提示したシステムを述べる. 3 章では, 提案するシステムの概要を述べる. 4 章では, 本研究の提案手法の詳細について述べる. 5 章では, 本研究で実装したオピニオンマイニングシステムの評価について述べる. 6 章では, 結論として本研究により得られた知見と今後の課題について述べる.

第2章　関連研究

本章では, 関連研究について述べる. 本研究は, ブログの賛成的立場, 反対的立場, 中立的立場といった極性を決定する際に, 評価表現を利用する. まず, 評価表現の自動獲得, 半自動獲得に関する研究について述べている. 次に, ウェブ上のテキストから有用な情報を取得し, ユーザーに効果的に提示するシステムを提案している研究をいくつか挙げる.

2.1　評価表現

乾・奥村 [3] は, 個人の評価に関する情報を**評価情報**, 評価情報の良し悪しに関する軸を**評価極性**, 評価情報がテキスト内で記述された表現を**評価表現**, 評価表現とその表現が持つ評価極性の組[1]を集めたものを評価表現辞書と呼んでいる. 評価分析システムの多くは, 文書集合から文書, 文あるいは語句などの単位について, 肯定/否定の評価極性を判定する為に, 評価表現辞書を利用している. 本研究においても, ブログのテキスト文中に含まれる評価表現を判定する際に, 評価表現辞書を利用し, ブログ著者の立場を推測する上での手がかりとしている.

2.1.1　評価表現辞書

小林らは, ウェブ上のテキストから評価表現を半自動的に収集する方法を提案している [4]. 小林らは, テキストから < 対象 >< 属性 >< 評価値 > のような典型的な文型 (共起パタン) を利用し, 共起パタンを満たす属性と評価値を表わす表現を取得することで評価表現を網羅的に収集している. 実験では, 異なるドメインを持った 2 つのクエリに対して人手及び半自動で属性表現と評価値を取得する. その結果, 半自動収集の方が人手よりも効率的な収集を行うことができ, さらに網羅性も優れていることを示している.

東山らは, 述語の選択選好性に着目し, 教師あり学習によって名詞の評価極性の獲得を行う手法について提案している [5]. 述語の選択選好性とは, 名詞と述語の言語的制限, 例えば「防ぐ」のヲ格はネガティブな名詞となりやすいことである. 東山らは, 名詞と述語の特定パタンを持つ関係性に着目し, 述語と共起する名詞が肯定的利用または反対的利用をされやすいのかを推定し, 名詞の評価極性を取得している. 具体的には, 名詞に対して 3 種

[1] 例:良い-肯定

類の述語を補う事で名詞を事態化し, その事態に対して望ましいか否かを判別することで, 名詞の評価極性を判定する. つまり, 以下のとき名詞の評価極性を判定する.

1. (〜する) を付加して動詞化
2. (〜だ) を付加して形容詞化
3. (〜が増える) や (〜がある) を付加して動詞化

例えば,「信頼」のように < がある > というプラスの演算子を付加してポジティブな意味を持つときにはポジティブな名詞とし,「ミス」のように < が減る > のようなマイナスの演算子を付加してポジティブな意味を持つときネガティブな名詞としている.

鍛治らは,HTML 文書から自動構築した評価文コーパスを用いて評価表現辞書を自動構築する方法を提案している [6]. 鍛治らは, 評価文に使われる定型的な表現に着目して評価表現を抽出している. まず,「定型文」,「箇条書き形式のブロック」,「表」の中に, 特定パタンが出現するとき, それらを評価文として抽出しコーパスを作成する. そして, 構築したコーパスから評価表現を抽出する. 評価表現の抽出は, 全てのコーパスに出現する形容詞と形容詞句を対象とし, それらが好評表現あるいは不評表現として出現した頻度を PMI(pointwiseMutualInformation) を用いて評価し, 閾値を超える表現を評価表現として取得している.

本研究では, 2 種類の評価表現辞書を利用する. 一つは, 小林の日本語評価極性辞書 (用言編)[4] を一部改編し，人手で評価極性情報を付与した評価表現約 5 千件の辞書である. もう一つは, 日本語評価表現辞書 (名詞編)[5] である. これは, 東山らの定義した名詞の評価極性を利用し, 評価極性を持つ（複合）名詞，約 8 千 5 百表現に対して評価極性情報を半自動的に付与した辞書である.

2.2 オピニオンマイニングシステム

2.2.1 OpinionReader

藤井は, あるトピックに対して記述された主観的意見の集合を入力として与えたとき, それらを集約して, 賛成・反対に分類したうえで, グラフとして表示することにより, トピックに対する主観的情報を可視化するシステムである OpinionReader を提案している [2]. 藤井は, ウェブなどの意見文の情報から時事問題や商品の評判などに対する意志決定を行うとき, その手順は以下に分解できるとしている.

1. 対象の話題 (商品や時事問題) に関する文書をウェブから収集する
2. 収集した文書から主観的な記述を抽出する
3. 抽出した主観的記述を「肯定/否定」や「賛成/反対」などの極性に応じて分類する

4. 主観的記述を集約し, さらに可視化する

5. 可視化された内容を吟味して,「肯定/否定」から一方を選択する. 対象の話題が商品の場合は, 肯定を選んだ場合に, その商品を購入する

OpinionReader は, 手順 (4.) に焦点を当てたシステムであり, トピックに対する主観的な情報を効率よく取得することができ, 内容を吟味することで合理的な立場を選択する事ができるため, ユーザーの意志決定を支援する効果がある. また, 手順 1〜3 は人手か既存の手法によって完了している事を前提としている. 図 2.1 にシステムの概要を示す.

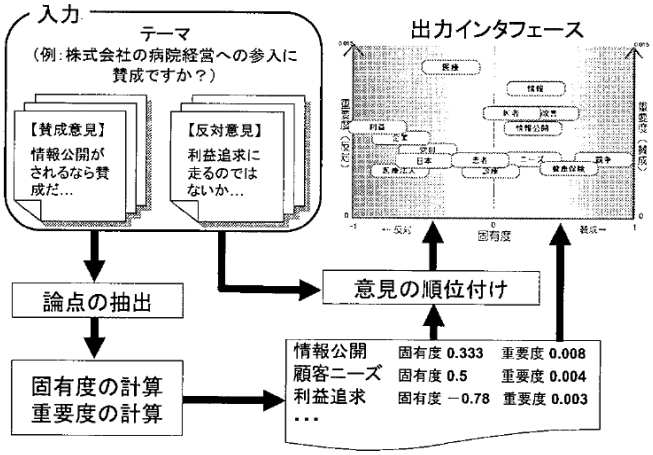

図 2.1: OpinionReader の概要

まず, 時事問題や商品の話題など, 賛成意見と反対意見に分かれたテキストを入力として与える. 次に, テキストから, 意見の「論点」を表すキーワードを抽出する. さらに, 論点について, 賛成意見と反対意見のどちらで多く論じられているのかを表す「固有度」を求める. そして, 論点の頻出度を表す「重要度」を計算し, 固有度と重要度に応じて 2 次元空間上に配置することにより, トピックに対する主観情報の可視化をはかっている. 以下, 各モジュールの内容を紹介する.

論点の抽出

論点の抽出は, まずキーワードとなるような名詞句を論点として抽出する. また, キーワードの意味がより通るようにするために動詞句も抽出する. 具体的には表 2.1 のパターンに合致する名詞句を抽出する. (a)〜(g) に合致するパターンが重複して現れる場合は, 網羅性を重視して, 各パターンを個別に抽出する. 例えば,「高い医療費を削減」という文字列からは,(a),(e),(g) によって「医療費」,「高い医療費」,「医療費を削減する」を抽出する. さらに, 名詞句の他に単独の名詞も論点として抽出する. ただし, 一般的な語を防ぐために,

表 2.1: 名詞句として抽出するパターン

	パターン	例
(a)	名詞の連続	
(b)	名詞/の/名詞	情報/公開, 独占/禁止/法
(c)	名詞/(と | や | ・)/名詞	医療/の/質
(d)	形容動詞語幹/な/名詞	水/と/油,　先輩/・/同僚
(e)	形容詞/名詞	軽い/怪我
(f)	動詞/名詞	調べる/方法
(g)	名詞/(が | を | は | の | も)/サ変動詞語幹	情報/の/公開, 情報/を/公開/

２文字以上の名詞のみを対象とし, 賛成あるいは反対の立場で１回しか出現しない名詞は論点として扱っていない．また, 論点となりにくい語を集めてストップワードとし, 単独では論点として扱っていない．動詞句の抽出は, 意見テキストの係り受け解析を行い, 名詞と助詞で構成される文節が動詞に係っている表現を抽出する．例えば,「病院が利益追求に走る」という文からは,「病院が走る」と「利益追求に走る」を論点として抽出する．しかし, 主語や目的語となる名詞が「代名詞」か「非自立語」である場合は抽出しない．また, 助動詞は削除する. 否定形や様相の抽出には, 助動詞が重要であるが複雑な表現も存在するため, 今後の課題としている.

固有度の計算

固有度は,「ある論点 A がどちらの立場で多く論じられているか」を表わす尺度である．論点 A の固有度を,「論点 A について論じている意見を一つ選んだときに, それが賛成派 (pro) の意見である」という条件付き確率と,「論点 A について論じている意見を一つ選んだときに, それが反対派 (con) の意見である」という条件付き確率の差で計算する．具体的には, 論点 A の固有度を式 (2.1) で計算する.

$$P(pro|A) - P(con|A) \tag{2.1}$$

式 (2.1) 中の条件付き確率は, 式 (2.2) を用いて計算する.

$$\frac{\text{論点 } A \text{ について言及している立場 } X \text{ の意見数}}{\text{論点 } A \text{ について言及している意見の総数}} \tag{2.2}$$

式 (2.1) は-1 以上 1 以下の値を取る. 賛成派だけが言及している論点の固有度は 1, 反対派だけが言及している論点の固有度は-1 となる．また, 両方の立場で言及されている論点の固有度は 0 に近い値となる.

重要度の計算

一つの論点に関する重要度は, 賛成派と反対派で異なる値を持つ. 立場 X における論点 A の重要度は,「論点 A が立場 X においてどれだけ多くの意見で論じられているのか」を表わす. 具体的には式 (2.3) を用いて, 立場 X における論点から無作為に一つを選んだときに, それが A である条件付き確率を計算する.

$$\frac{\text{論点 } A \text{ の立場 } X \text{ における出現頻度}}{\sum_i \text{論点 } i \text{ の立場 } X \text{ における出現頻度}} \tag{2.3}$$

上式の論点 i は, 論点 A と同じ立場に分類された論点である. ただし, 固有度が 0 の論点は, 賛成と反対の両方に対して重要度を計算し, その平均を重要度とする.

意見の順位付け

意見の順位付けでは, ユーザーが指定した論点について言及している複数の意見に対して, その論点への関連度を計算して順位づけている. まず, 対象となっている論点を A とし, 論点 A について立場 X で言及している意見の集合を $S_{A,X}$ とする. $S_{A,X}$ における高頻度語は, 論点 A に関する議論に重要である. そこで,$S_{A,X}$ における高頻度語を含む意見ほど, 論点 A への関連度が高いといえる. 具体的には,$S_{A,X}$ に対する順位付けは以下の手順で行っている.

1. 賛成と反対の立場ごとに,$S_{A,X}$ 中の意見テキストすべてを形態素解析し, 内容語の出現頻度を調べる. 名詞, 動詞, 形容詞を内容語とする.

2. 意見の関連度を式 (2.4) で計算する.

$$\frac{\sum_{w \in s} \text{内容語 } w \text{ の } S_{A,X} \text{ における出現頻度}}{s \text{ に含まれる内容語数}} \tag{2.4}$$

ここで,w は s に含まれる内容語である.

論点の可視化

論点を固有度と重要度によって 2 次元平面に可視化する. 固有度を横軸にとり, 重要度を縦軸にとっている. これにより, 論点が賛成派, 反対派どちらの立場で使われているか, どの程度頻度が高いかユーザーに一目でわかるように提示している. 図 2.2 は実際の出力結果である.

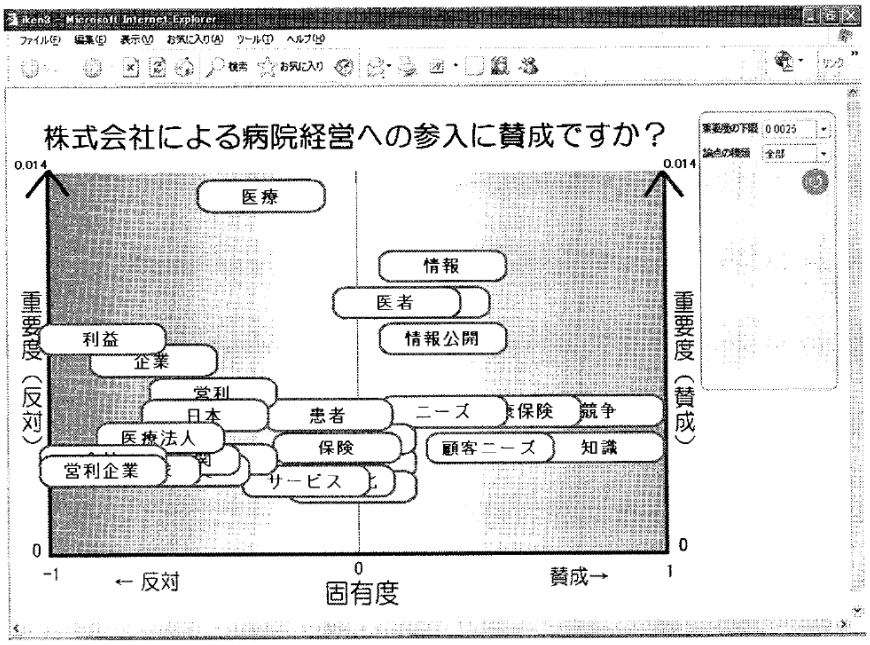

図 2.2: OpinionReader で論点の分布を可視化した例

2.2.2 調停要約

Shibuki et al は, トピックに対する調停要約を自動で生成する手法を提案している [10]. 調停要約とは, トピックに対する肯定的意見と否定的意見を対比させ, さらに対立関係が矛盾することなく説明できる記述を提示する要約である. これにより, 一見対立しているようにみえるが実際は両立可能な関係にある二言明に対して, 両立可能となる状況を提示することで利用者の信憑性判断を支援することを目的としている. 図 2.3 は調停要約の例である. 例では,「ディーゼルエンジンは環境に有害か」というクエリに対する肯定的意見と否定的意見, さらに両意見が矛盾すること無く両立する意見を提示している. 次に,Shibuki et al が提案するシステムを説明する. 図 2.4 は提案システムの概要である.

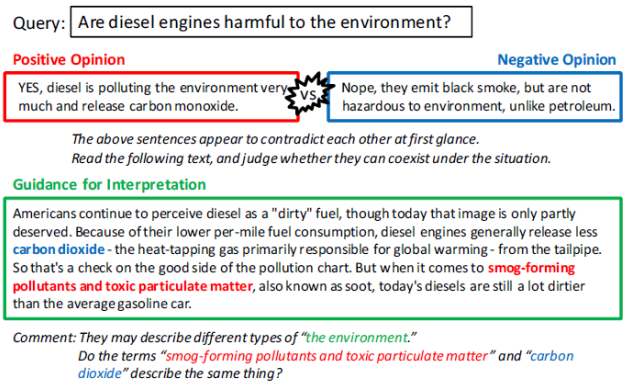

図 2.3: 調停要約の例

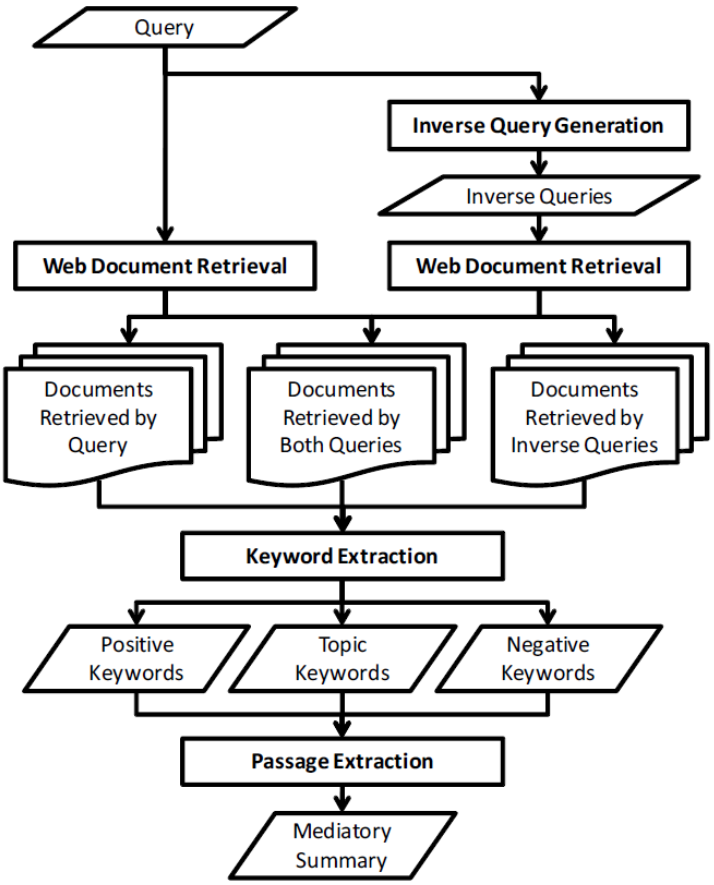

図 2.4: 提案システムの概要

まず, トピックに関する文をクエリとし, クエリとその否定文をそれぞれ用いてウェブ文書の検索を行う. 次に, クエリにより検索された文書と否定文により検索された文書, 両者にて検索された文書から特徴語を抽出する. そして, クエリにより検索された文書に多く見られる特徴語を肯定的特徴語, 否定文により検索された文書に多く見られる特徴語を否定的特徴語, クエリ中の単語において肯定的特徴語と否定的特徴語に含まれない単語をトピック特徴語とし, これら特徴語を用いてウェブ文書を検索し, パッセージを抽出する. そして, 三種類の特徴語を言及しているか否かによってパッセージのスコア付けを行い, 調停要約として適切なパッセージを出力することにより調停要約を自動生成している.

2.2.3 言論マップ

水野らは, 言論マップを自動的に生成する手法を提案している [8]. 言論マップとは, トピックに関連した文を文間関係認識に基づいて, 賛成意見あるいは反対意見か否か分類し, それぞれに対する根拠を含むか否かを判定しそれらを俯瞰的に示す手法である. まず, 自然文をクエリとしてパッセージ検索を行い, 検索対象文を獲得する. そして, クエリと対象文の 2 文間の意味的関係を分類し, 俯瞰的に表示することで言論マップを生成する. 水野らの研究では, 文間関係認識技術を最も重要な問題としている. 文間関係認識は 2 文間の意味的関係を分類する技術である. 彼らは, 分類の対象とする関係を「同意」「同意根拠」「対立」「対立根拠」「その他」の 5 つに分類している. また, 分類は 2 段階で行っており, まず < その他 > と < 関係あり > に分類する. この分類は, 検索対象文中にクエリが内容的に対応する部分があるかどうかによって判断される. そして,< 関係あり > に分類された文対を < 同意 > と < 対立 > に分類する. この分類は, クエリ中の単語と, 対応づけられた検索対象文側の単語とが反義語や否定の関係にあるかによって判断される. さらに,< 同意 > と < 対立 > に分類された文対の検索対象文中でクエリの内容に相当する部分に対して, その根拠となる情報が修飾関係にある文対をそれぞれ < 同意根拠 > と < 対立根拠 > に細分類する. 最後に, 分類により得られた関係をそれぞれ俯瞰的に表示してユーザーに提示している.

第3章　提案システム

3.1　概要

本論文では, 調査対象とするトピックに関するクエリを入力として与えたとき, ウェブ上のクエリを含むブログ記事を検索し, 著者がトピックに対してどのような立場を表明しているのかによって「賛成」「反対」「中立」の３値に分類し, それぞれに分類されたブログ記事の数を集計するシステムを提案する. ここでのクエリは,「原発　再稼働」「消費税　増税」のようなキーワードの組として与えられるものとする. さらに, 本論文では, 他者のコメントの引用に着目する. ブログ記事では, 新聞の記事や他者のコメントを引用しつつ, 自分の意見を述べる場合が多い. 他者のコメントの引用は, ブログ記事がトピックに対して賛成, 反対, 中立なのかを判定する際に障害になり得る. その主な２つの理由を以下に述べている.

1. 引用記事と著者の立場が異なるブログ記事
   引用した記事の意見とブログ著者の意見が異なる場合, ブログ記事の極性判定に注意を要する. 図 3.1 は, 引用記事と著者の立場が異なるブログ記事の実際の例である. 引用記事では, 使用済み核燃料の直接処分に関する肯定的な意見を述べているが, それに対してブログ著者は否定的な意見を述べており, 全体としては使用済み核燃料の処分に対する否定的な記事となっている. ところが, 引用記事中の肯定的な評価表現から, ブログ全体で肯定的な意見を表明していると誤って判定してしまう可能性がある.

2. 引用記事だけで著者の意見が述べられてないブログ記事
   図 3.2 では, 引用記事をコピー＆ペーストしてそのままブログに掲載しているが, 著者のコメントは全く述べられていない. このような場合, 著者は記事の引用によって間接的に自分の意見を表明しているとも考えられる, しかし, このような記事を, 著者が明確に意見を表明しているブログ記事と同等に扱ってよいのかには疑問が残る. 全く同一の記事が掲載されている場合, 引用した記事と引用された記事を１つの賛成意見もしくは反対意見としてカウントすべきという考え方もあろう. よって, 本研究では, このような著者の意見が明確でない記事を除外し, 著者が明確に意見を表明している記事の数を求めることを試みる.

図 3.1: 引用記事と著者の立場が異なるブログ記事の例

図 3.2: 引用記事だけで著者の意見が述べられていないブログ記事の例

上記の状況を踏まえ, 本研究では,「賛成」「反対」のブログ記事数を 2 通りの方法で集計する．1 つは, ブログ記事の本文に出現する全ての語を手がかりに「賛成」「反対」の判定 (極性判定) を行い集計する方法である．もう一つは, ブログ記事の中から他者のコメントを引用している箇所を検出し, それを除いた残りのテキストを手がかりとし, 極性判定を行う方法である．前者の方法で集計された記事数を「総記事数」, 後者の方法で集計された記事数を「ユニーク記事数」としてともに提示する．さらに, ブログ記事の中から賛成もしくは反対を表明している文を抽出し, 賛成意見文, 反対意見文としてユーザーに提示する.

3.2 システムアーキテクチャ

提案システムの処理の流れを図 3.3 に示す. また, 図 3.4 は, 本研究で想定しているシステムの出力例である.

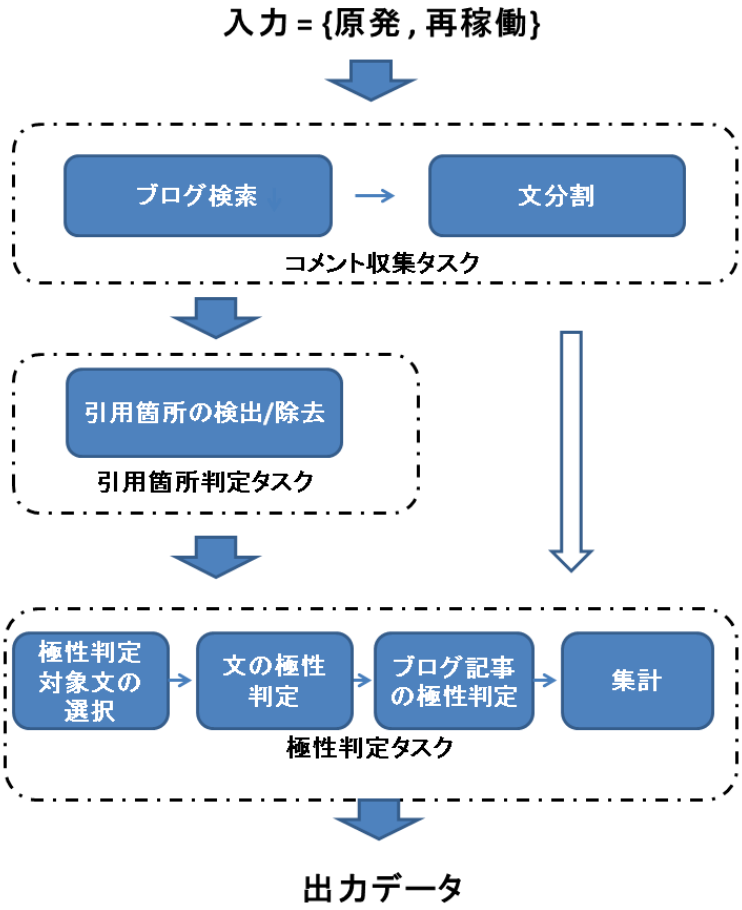
入力 = {原発 , 再稼働}
ブログ検索
文分割
コメント収集タスク
引用箇所の検出/除去
引用箇所判定タスク
極性判定
対象文の
選択
文の極性
判定
ブログ記事
の極性判定
集計
極性判定タスク
出力データ

図 3.3: システムの概要

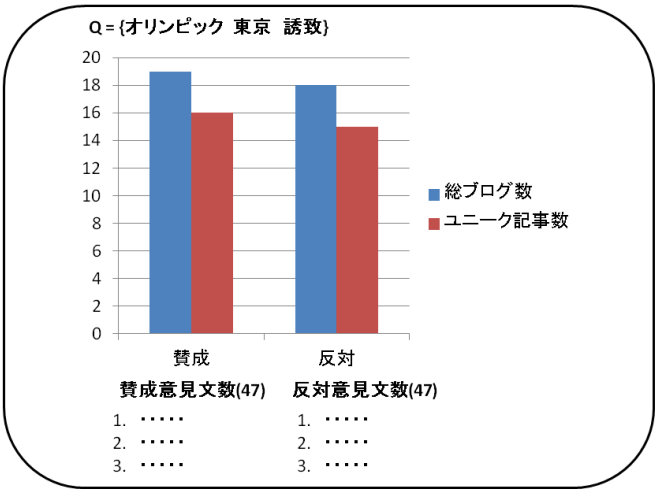

図 3.4: 本研究で想定するシステムの出力

システムは「コメント収集タスク」「引用箇所判定タスク」「極性判定タスク」の 3 つのタスクから構成されている. 各タスクの内容を以下に述べる.

1. コメント収集タスク
   コメント収集タスクでは, ブログ検索エンジンを用いて調査対象としたいトピックのブログページを検索し, ページ全体から本文となる部分を検出して, ブログ記事のテキストを取得する.

   このタスクは 2 つのモジュールで実現される. 「ブログ検索」モジュールでは, ブログ検索エンジンを利用して調査対象とするトピックを表わすクエリを含むブログページを検索する. 本論文では, ブログ記事が掲載されたウェブページを「ブログページ」, ブログ著者が記述したテキストを「ブログ記事」と呼ぶ. 次に,「文分割」モジュールでは, 検索されたブログページからいくつかのヒューリステックを用いてブログ記事を検出した後,HTML タグや句読点を境界として文単位に分割する.

2. 引用箇所判定タスク
   引用箇所検出タスクでは「引用箇所の検出/除去」モジュールを使い, ブログ記事から, ほかのウェブサイトから取得し転載された記事などの「引用されたテキスト」を検索し, 除去する.

   このタスクは 1 つのモジュールで実現される. 「引用箇所の検出/除去」モジュールでは, 引用箇所を示唆するキーワードや HTML タグを手がかりとし, ブログ記事の中から外部のウェブページから引用されたと思われる文を検索し除去する.

3. 極性判定タスク
   極性判定タスクでは, ブログの極性, すなわちブログの著者が調査対象トピックに対して賛成 (肯定的), 反対 (否定的), 中立のいずれの立場を取るかを判定する. 引用箇所を除いたブログ記事の文集合から, トピックに言及している文を選別する. これらを「極性判定対象文」と呼ぶ. 次に, これらの文について極性判定を行う. 最後に, 文単位の極性判定の結果を基にブログ記事全体の極性を判定する.

   このタスクは 4 つのモジュールで実現される. 「極性判定対象文の選択」では, トピックに対する意見が述べられていると思われる文を選択する. 基本的には, 引用箇所として除去された文以外の文の中で, クエリのキーワードの近傍に出現する文を極性判定対象文とする. 「文の極性判定」モジュールでは, 極性判定対象文の極性 (賛成もしくは反対) を判定する. 文の中に含まれる名詞および用言の評価表現を手がかりとし, クエリのトピックについて賛成もしくは反対を表明しているかを判定する. 本研究では, 賛成あるいは反対を表明した文を「意見文」と呼ぶ. 「ブログ記事の極性判定」モジュールでは, ブログ記事全体の極性を判定する. ここでは, 文の極性判定結果のスコアの重み付け和によって, ブログ著者がトピックに対して「賛成」「反対」「中立」のうちどの立場にあるのかを判定する. 「集計」モジュールでは, トピックに対し「賛成」もしくは「反対」と判断されたブログ記事数をカウントし, ブ

ログ記事内に存在する賛成, 反対を表わすコメントをユーザーに効果的に提示する. また,3,1 節で述べたように, 引用箇所を除去せずにブログ記事全体を用いて極性判定した結果を集計した「総記事数」(図 3.3 の白抜き矢印の処理) と引用箇所を除いたテキストを用いて極性判定した結果を集計した「ユニーク記事数」の 2 通りの結果を示す.

第4章　提案手法

本章では, 提案手法の詳細について述べている．前章で述べたように, システムは 3 つのタスク「コメント収集タスク」「引用箇所検出タスク」「極性判定タスク」と 7 つのモジュール「ブログ検索」「文分割」「引用箇所の検出/除去」「極性判定対象の選択」「文の極性判定」「ブログ記事の極性判定」「集計」によって構成される．以下, それぞれのタスク, それぞれのモジュールの詳細について述べる．

4.1　コメント収集タスク

コメント収集タスクでは「ブログ検索」「文分割」モジュールを利用し, ブログ検索エンジンを用いて調査対象のトピックに関連するブログページを検索し, ページ全体から本文となる部分を検出して, ブログ記事のテキストを取得する．

4.1.1　ブログ検索

「ブログ検索」モジュールでは, ブログ検索エンジンにクエリを入力し, クエリを含むブログページの URL を出力上位から順に取得する．５章で述べる評価実験では, 上位５０件の URL を取得しているが, この設定はある程度自由に変更できる．

入力クエリ

システムに入力するクエリは,「原発　再稼働」「消費税　増税」など調査対象とするトピックを表わす語の組み合わせとする．

ブログ検索エンジン

ブログ検索エンジンは Yahoo!が公開している「Yahoo!ブログ検索」[1]を利用する．このエンジンでは, 検索対象を一般のウェブページ以外のブログページに限定して検索を行う．また,Yahoo!が運営しているブログだけでなく, 他社によって運営されているブログも検索の対象となる．

[1]http://search.yahoo.co.jp/blog

4.1.2 文分割

「文分割」モジュールでは, まず取得したブログページの URL から HTML 文書をダウンロードする. 次に, ヒューリステックを用いてブログ記事 (本文) を取得する. 最後に, 取得したテキストを文単位に分割する. 具体的な手順は, 以下の通りである.

1. ブログページの URL にアクセスし,HTML 文書を取得する.
2. HTML 文書の DOM(Document Object Model) を参照し, ブログページのなかでブログ記事に相当するブロック要素となるノードをヒューリステックにより選択する. 以下, ブログ記事に該当する DOM 上のノードを「本文ノード」と呼ぶ.
3. 選択した本文ノード以下のテキストをすべて取得する.
4. 取得したテキストを
 タグ及び句読点「.」で分割し, それぞれを一文とする.

以下では, ブログ記事の特徴について述べた後, ステップ 2 の本文ノードの検出方法の詳細を説明する.

ブログ記事の特徴

ブログ記事が掲載されたウェブページでは次のようなテキスト情報がよく見られる.

- ブログ著者の記事 (本文)
- ブログ読者のコメント
- アフェリエイト等の広告記事
- 関連記事へのリンク
- ブログ著者の過去記事

ブログ記事の掲載されたウェブページには, ブログ著者の記事以外にも, 見出しや広告, 読者のコメントなど著者の記述以外のテキスト情報が多く見られる. これらは極性判定の際にノイズとなりうるため, 除去する. 図 4.1 はよく見られるブログページの例である. 1 番は, ブログページの本文となるブログ記事である. 2 番は関連記事へのリンクである. 3 番はブログ記事に対する読者のコメントである. 2 番や, 3 番のようなテキスト情報を除去し, 1 番のようなブログページの本文のみを取得する.

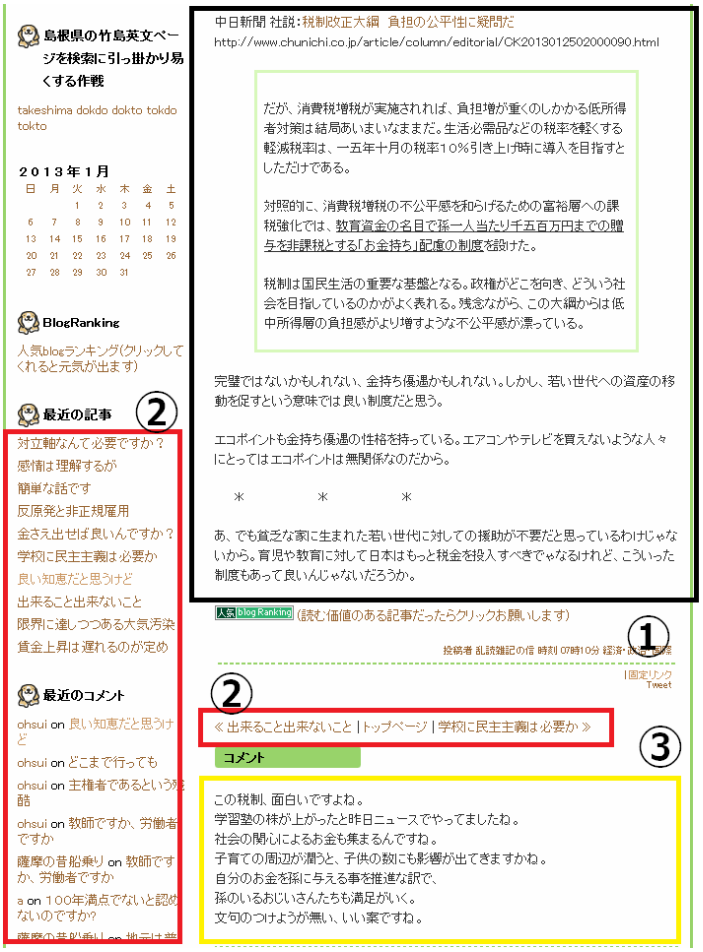

図 4.1: ブログ記事の例

ブログ記事の取得

HTML 文書から, ブログページの本文テキストであるブログ記事を取得する．本研究では, この処理を DOM 上において本文に対応するノードを特定する問題に帰着する．予備実験により,「消費税　増税」「原発　再稼働」というクエリに対して得られたウェブページ 100 件の本文ノードを調査した．結果, や など,‘body’ のような本文を示唆するキーワードを class 属性に持つ div ノードが本文ノードに対応するページが 87 件あった．さらに, レンタルサーバーを利用していない個人のページにおいても ‘body’ を class 名に含む div ノードが本文ノードであるケースが多く見られた．よって, ブログ記事が掲載されたウェブページでは, 本文ノードとして ‘body’ を class 名に含む div ノードが利用されている可能性が高いことがわかった．

しかし,HTML 文書に ‘body’ を class 名に含む div ノードが複数存在するケースや,‘body’ を class 名に含む div ノードが本文ノード以外の利用をされているケースなども少なからず見られた．そのため,‘body’ を class 名に含む div ノードが HTML 文書の中に複数存在する

とき, どれが本文ノードとなるブロック要素であるかを推定する必要がある．また,‘body’のような class 属性を持たない div ノードが本文ノードになっている可能性も考えなければならない．そこで, ブログ記事は調査対象とするトピックに関連したテキストが掲載されていることを前提とした上で, クエリとなるキーワードはブログ記事内に最も多く出現する可能性が高いことに着目した．すなわち, クエリを含むテキストの親ノードをたどり, 直近の ‘body’ を class 名に含む div ノードを本文ノードとして選択する．‘body’ を class 名に含む div ノードが複数ある時, クエリのキーワードを最も多く含むものを本文ノードとして選択する．また, もし ‘body’ を class 名に含む div ノードが取得できなかったとき, クエリを含むテキストブロックが最も多い class 属性を持つ div ノードを本文ノードとして選択した．具体的な手順を以下に示す．また, これら手順により本文ノードが選択できなかった時, 本文ノードを取得できないブログとみなす. このような, ブログに対し以降の処理は行わないものとする.

1. HTML ページからクエリを含むテキストブロックを発見する.
2. クエリを含むテキストブロックが存在したとき, ブロック要素となるノードから親ノードをたどる.
3. 自ノードもしくは親ノードに class 属性を持つ div ノードが初めて出現した時, 本文ノードの候補とする. ただし, 付近にリンクを示す <a> タグが存在する時は, クエリを含むテキストブロックは「見出し」である可能性が高いため, 本文ノードから除外する.
4. さらに親ノードをたどり class 属性を持つ div ノードが 5 回以上出現する前に,‘body’を class 名に含む div ノードが出現した際,‘body’ を class 名に含む div ノードを本文ノード候補として選択する.
5. HTML ページ中の全てのクエリを含む全てのテキストブロックに対して 1〜4 の処理を行う.
6. 本文ノード候補として ‘body’ を class 名に含む div ノードが存在する場合, クエリのキーワードが最もよく出現する div ノードを本文ノードとして選択する.
7. 本文ノード候補として ‘body’ を class 名に含む div ノードが存在しないとき,(‘body を含まない’)div ノードの候補のうち, クエリのキーワードが最もよく出現するものを本文ノードとして選択する.

ブログ記事の文分割

極性判定及び引用文判定を容易にするため, 本文ノード以下のテキストを文単位に分割する.

具体的な手順を以下に示す.

1. 本文ノード以下のテキストを
 タグで区切る.

2.
 タグで区切られたテキストを, さらに句読点 (.) で区切り, 1 文とする.

3. 1 文に区切ったテキスト全てに, ブログページの ID と行番号を付加する.

図 4.2 は文分割後のテキストの例である. blog_number はブログページの ID,line_number は行番号を表わす. また,keyword はクエリのキーワード (この例では「原発」「再稼働」) を含む文であることを表わす.

```
<blog_number:9,line_number:14><keyword:w>そうでなければ,「2030 年代までに原発ゼロを目指す」とか「経済のために再稼働が必要」などといったのんきなことは, とてもじゃないが言っていられない
<blog_number:9,line_number:15>事故処理や被害の補償で莫大なお金がかかるのに, その現実を抜きにして経済を語るのはあまりに馬鹿げている
<blog_number:9,line_number:16>日本は, 世界に対しはかり知れない重い責任を負っているのだ
<blog_number:9,line_number:17>これほどインターネットが発達した情報化社会でありながら, こうした現状がほとんど報じられないことがただただ恐ろしい
<blog_number:9,line_number:18><keyword:b>　本当のことを理解しているならば, 再稼働など論外
```

図 4.2: 文分割したテキストの例

4.2 引用箇所検出タスク

引用箇所検出タスクでは「引用箇所の検出/除去」モジュールにより, ブログ記事から, ほかのウェブサイトから取得し転載された記事などの「引用されたテキスト」を検出し, 除去する. 本研究では, ブロック単位と文単位の 2 種類の方法で引用箇所を検出する.

4.2.1 ブロック単位の引用箇所の検出/除去

ここでは, 引用箇所に該当する DOM 上のノードを特定し, 除去する. まず, 引用を示唆するキーワードを発見する. 次に, そのキーワードを含むノード及びその周辺のノードから引用箇所のブロックに該当するノードを特定する. 最後に, そのノード以下のテキストを除去する.

引用キーワードの検出

本論文では, 引用を示唆するキーワードとして 308 のキーワードを使用した. キーワードは人手で用意し, 主要な新聞の名称と, 予備実験で用いたクエリに対して取得されたブログ計 100 件に見られた引用キーワードを用いた. これらのキーワードが存在するとき, その前後のブロックには引用箇所が存在する可能性が高い. ただし, 引用を示唆するキーワードが文中に出現するときは, 他者のコメントの引用していることを意味しないことが多い. 例えば,「転載」は引用を示唆する典型的なキーワードであるが,「デジタルデータは, 容易にコピーが可能であるというその特徴から転載が行われやすい.」のような文では, ブログ記事内で他者のコメントを引用していることを示唆しない. よって, これらキーワードが次の条件のいずれかを満たすとき, そのキーワードは引用箇所の開始位置もしくは終了位置に対応するものとみなし, 引用箇所の検出に利用する.

- 前後が空白あるいは, 文頭または文末にキーワードが単独で出現する.
- キーワードの前後が「日」「１」など日付を表す文字である.
- キーワードの前後が記号である.

以下, 上記の条件を満たす引用キーワードが出現したとき, それを含む DOM 上のノードを「引用境界ノード」と呼ぶ. 図 4.3 には引用を示唆するキーワードの一部を, 図 4.4 には引用境界ノードに対応するテキストの例を示す.

RT，転載，転載開始，転載終わり，引用，掲載，NAVER まとめ，社説，毎日新聞，日本経済新聞，読売新聞，朝日新聞，New York Times

図 4.3: 引用を示唆するキーワード (抜粋)

・毎日新聞　2008 年 7 月 8 日 ・(朝日新聞社)：ロンドン暴動, 各地に拡大　五輪競技場近くでも略奪 - 国際 ・RT【今週金曜！緊急大拡散！】#大飯原発再稼働決定をただちに撤回せよ！6/22(金)18〜20 時, 首相官邸前にて原発再稼働反対の超大規模抗議行動を行います.

図 4.4: 引用を示唆するキーワードを含むテキストの例

引用ブロックの検出

引用境界ノードがブログ記事に存在するとき, ブログ記事から引用箇所となるブロック群を検出する. その検出方法は以下の 3 通りである.

- **近隣のブロック**
  引用境界ノードのテキスト長が 50 以上のとき, そのノードを引用ブロックとして検出する. また, 引用境界ノードに隣接する兄弟ノードが存在するとき, 兄弟ノード以下のブロックに含まれるテキスト長が 50 以上ならば, それを引用ブロックとして検出する. ただし, 引用境界ノードの親ノードがブログ記事本文のほとんどをカバーする時, 正確には本文ノード以下のテキスト長の 9 割以上のテキスト長を持つとき, これらのノードのブロックを引用箇所として検出しない.

- **罫線に挟まれた範囲に存在するブロック群**
  ブログ記事では, 引用範囲を罫線で挟み区別するケースが見られる. このことを踏まえ, 引用境界ノードに対応するテキストが罫線に挟まれているとき, 罫線に挟まれた部分を引用範囲として検出する. ここでは, 罫線は 4 回以上連続する文字列もしくは <hr> タグとする. 同じ文字が 4 回以上連続して出現し, 罫線と見なせる例を以下に挙げる.

——— , ********* , ○×○×○×○×○×○×

図 4.5: 罫線の例

  また, 図 4.6 は実際に引用箇所が罫線に挟まれたブログ記事の例である.

- **<blockquote> タグがつけられた要素ブロック**
  HTML ページでは, 一般的に比較的長い文書を引用する際に, 引用範囲を <blockquote> タグで挟む. よって, ブログ本文の中で <blockquote> のノード以下の文を引用範囲として検出する. なお,<blockquote> タグによる引用箇所の検出は, 例外的に, 引用境界ノードの有無に関わらず行う.

4.2.2 文単位の引用箇所の検索/除去

前項では引用を示唆するキーワードを利用して, 引用範囲をブロック単位で判定したが, 引用キーワードを明示せず他者のコメントを引用している場合もある. そのような場合には, 引用元のリンクと引用記事が同時に記述されている場合が多い. そこで, 本文と同一の文がリンク先のウェブページに存在するとき, その文は引用された文であると判定する.

皆さんお疲れ様でした。

ヘンオジ生きた煎餅を皆さんに提供して福島県の復興を成し遂げて行きます。

皆さん感謝です。

□■□■□■□■□ コルネットより □■□■□■□■□

社説

原発再稼働問題／「周辺同意」は時代の要請だ

原発再稼働の是非を議論する場合、どこまでを「地元」として扱い、それぞれの自治体に意見を求めるべきなのか。

新たに原子力災害対策重点区域（原発から10～30キロ圏）に含まれることになった全国の市町村へのアンケートでは、約60％が「周辺自治体の同意も得る必要がある」と回答し、「立地自治体のみでいい」を大きく引き離した。

原発問題での「同意権」をどの範囲まで認めるかは、ことし7月に再稼働した関西電力大飯原発（福井県）でも議論になり、結局は県と立地自治体に限られた形になった。これまで基本的に国－県－立地市町村という枠組みに限定されてきたが、周辺市町村をないがしろにするようなことはもはやできない。

・・・(中略)・・・

安全協定を結べば相応の発言力が得られるだろう。だが、協定の有無にかかわらず、周辺自治体の意思は最終判断の要件と位置付けなくてはならない。

2012年11月21日水曜日

□■□■□■□■□

原発の再稼働を要請した人達の謝罪がないまま時間が流れています。おかしい大人の責任論、原発事故の復興は先ずは謝罪して責任を取ることが大切です。これなし原発の廃止はないでしょう。お金があれば人間の欲の殆どが解決出来ます。だから買収された住民が原発再稼働を容認するのです。容認したら責任を取る姿が見本となるように出来ればです。そして勝俣会長の謝罪はまだきいていません。

図 4.6: 罫線に挟まれた引用の例

ブログ記事中のリンクの検出, リンク先ページの取得

本文のテキスト中に http:が先頭についた文字列が含まれるとき, 及び HTML 文書の本文ノード以下に <a> タグがあるとき, その URL のページをリンク先ページとして取得する. ただし,URL が http のドメイン名から同一著者のブログと判断できるものを除外する. リンク先ページは 4.1 節と同じ処理によって本文検出及び文分割を行う.

文の類似度

ブログ本文の中で他のウェブページへのリンクが存在するとき, クエリのキーワードを含む文に対して, リンク先ウェブページの全ての文との類似度を計算する. 文間の類似度は, ブログ記事の文とリンク先の文の間の形態素単位とした編集距離と定義する. 編集距離が元の文長の 30%以下の時, 同一の文と判断し, 元の文を引用文とする. また, 形態素解析には ChaSen を利用した.[2]

[2]http://chasen.naist.jp/hiki/ChaSen

リンク先の全ての文に対して類似度を計算するのは効率的でないため, 本文中のクエリを含む文と, リンク先ページ内の文のうちのクエリを含む文との類似度のみを計算する. また, クエリ単独で出現する場合等を考慮し, 長さが 10 以上の文のみを類似度計算の対象とする.

4.3 極性判定タスク

極性判定タスクでは, ブログの極性, すなわちブログの著者が調査対象トピックに対して賛成 (肯定的), 反対 (否定的), 中立のいずれの立場を取るかを判定する. 引用箇所を除いたブログ記事の文集合から, トピックに言及している文を選別する. これらを「極性判定対象文」と呼ぶ. 次に, これらの文について極性判定を行う. 最後に, 文単位の極性判定の結果を基にブログ記事全体の極性を判定する.

4.3.1 極性判定対象文の選択

「極性判定対象文の選択」モジュールでは, ブログ記事の引用箇所を除いた文集合から, 極性判定を行う文を決定する. 調査対象とするトピックに関する意見文は, クエリが含まれている文の近傍に存在する可能性が高いという考えに基づき, クエリから一定距離にある文のみを極性判定対象文とする. ただし, 複数のクエリが互いに離れた箇所に存在するとき, それらは調査対象とするトピックと関係のない文である可能性もあるため, クエリにおける全てのキーワードが一定距離内に存在するという条件を加える. 具体的な手順は以下に示す. ただし, クエリを $Q = \{\cdots, q_i, \cdots\}$,「文分割」によって得られたブログ記事における文の集合を $S = \{\cdots, s_i, \cdots\}$ とする.

1. Q 中の全ての q_i が出現する文の範囲 [i,j] を決定する. ただし, 範囲 [i,j] において q_i が出現する文は互いに距離 2 以内に位置しなければならないとする.

2. 全ての q_i が出現する範囲 [i,j] が決定したとき, その前後 2 文 [i-2,j+2] を極性判定対象文の範囲とする.

例えば,Q={ 原発, 再稼働 }, 文集合が図 4.7 の場合を考える. 文 s_{24} に「原発」, 文 s_{25}「再稼働」が含まれている. さらに, これらの距離は 2 以内である. よって全ての q_i が出現する範囲は [24,25] となり, その前後 2 文である範囲 [22,27], すなわち s_{22} から s_{27} までの文を極性判定対象文とする. なお, 図 4.7 は, 実際のブログ記事に見られた文に対する極性判定対象文の選択例である. キーワード周辺の文はトピックに関連した意見文であるが, 極性判定対象文の範囲外には, トピックに対して無関係な文が目立つ.

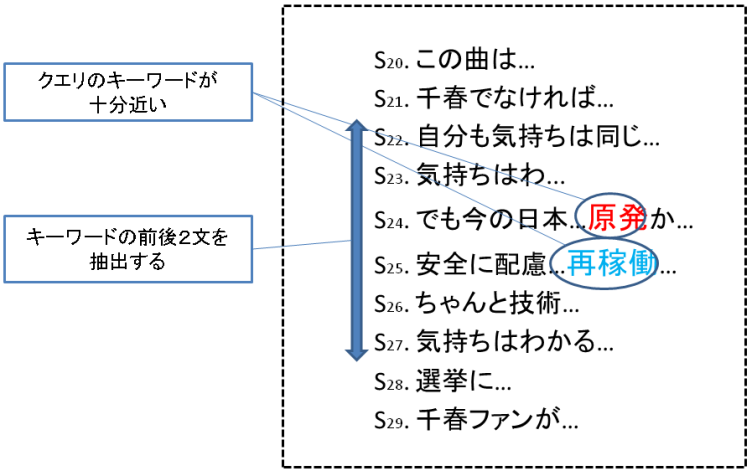

図 4.7: 極性判定対象文の選択例

4.3.2 文の極性判定

「文の極性判定」モジュールでは, 極性判定対象内の文の極性を「賛成」,「反対」,「中立」のいずれかに分類する．また, 極性判定には名詞及び用言の評価表現を手がかりとして利用する．2 章で説明したとおり, 文書集合から文書, 文あるいは語句などの単位について,「肯定」「否定」の評価極性を判定するために, 評価表現は有用である．例えば,「増税賞賛の新聞はうんざりする」という文では,「うんざり」という評価表現からネガティブな文であると推測することができる．このことを利用し, 文中の評価表現から文の極性を「賛成」「反対」「中立」に分類する．具体的には, 式 (4.1) で定義される文 s_i の極性スコア $Score(S_i)$ を求める．

$$Score(s_i) = \begin{cases} \prod_{w \in V} polar(w) \times neg(w) & \text{if } V \neq \phi \\ norm(\sum_{w \in N} polar(w)) & \text{if } N \neq \phi \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$$norm(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

式 (4.1) における V は，文 s_i 中の単語のうち，日本語評価極性辞書 (用言編) に含まれる用言の集合であり，$polar(w)$ は同辞書における極性 (+1 or −1) を表わす．また，$neg(w)$ は否定語のスコアであり，w と同一文節中に否定語が出現すれば −1，それ以外は +1 とする．すなわち, 否定語があるときは $neg(w)$ によって極性を反転し, また否定的な評価語が 2 回出現するときは二重否定とみなして肯定 (賛成) と判定している．また，文 s_i に含まれる用言の評価語を第一の手がかりとし，次に名詞の評価語を手がかりとして文の極性を判定している．一方，N は日本語評価極性辞書 (名詞編) に含まれる名詞の集合であ

り,$polar(w)$ は同辞書における極性である．名詞の場合, 式 (4.2) に示した正規化関数によってスコアを +1,−1,0 のいずれかの値に変換する．すなわち, 文内において肯定的な名詞の評価表現の数が否定的な名詞の評価表現の数より, 多いときは +1, 少ないときは −1, 同数の時は 0 となる．文の極性は,$Score(s_i)$ が +1 なら「賛成」,−1 なら「反対」,0 なら「中立」と判定する．

4.3.3 ブログの極性判定

「ブログの極性判定」モジュールでは, ブログ記事の極性を賛成, 反対, 中立に分類する．「文の極性判定」モジュールにより得られた極性判定対象文の極性に基づき, ブログ記事の極性を推測する．また, 極性判定対象文のうち, クエリを含む文や, クエリと評価表現との間に係り受け関係が見られた文などは, ブログの極性を決定する際により有効な文であると考えられる．したがって, それらの文に対する極性判定結果を重視するよう重み付けを行う．具体的には, ブログ記事 A の極性スコア $Score(A)$ を式 (4.3) のように定義する．

$$Score(A) = \sum_{s_i \in S_t} Weight(s_i) \times Score(s_i) \tag{4.3}$$

式 (4.3) における $S - t$ はブログ記事における極性判定対象文の集合である．また, 重み $Weight(s_i)$ は以下のように決定する．

1. 文 s_i において，検索クエリ q_i と評価語の間に係り受け関係[3]があるとき，重みを 5 とする．

2. 文 s_i に含まれる検索クエリの数が 3, 2, 1 個のとき，重みを 3, 2, 1.5 とする．

$Score(A)$ が 1 以上のとき，ブログ記事 A はトピックに対して「賛成」を，−1 以下のときには「反対」を表明していると判定し，それ以外は「中立」と判定する．

4.3.4 集計

「集計」モジュールでは, 賛成, 反対, 中立に分類したブログ記事の件数を集計し, ユーザーに提示する．図 4.8 は, 出力の例である (図 3.4 の再掲)．「総ブログ数」はシステムにより賛成・反対の出力が得られたブログ記事の数である. 一方,「ユニーク記事数」は, 引用箇所を除去したブログ記事を極性判定し, システムにより賛成・反対の出力が得られたブログ記事数である. また, 文の極性判定によって得られた意見文を賛成・反対に分けて提示する．

[3]係り受け解析には CaboCha を用いた．

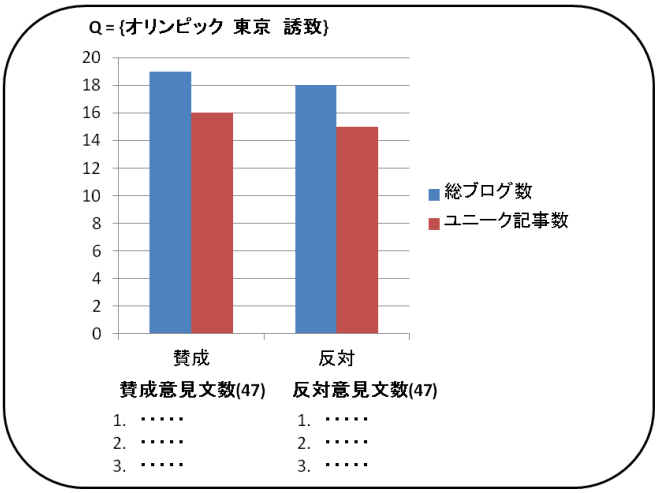
Q = {オリンピック 東京 誘致}
20
18
16
14
12
10
8
6
4
2
0
総ブログ数
ユニーク記事数
賛成
反対
賛成意見文数(47)
反対意見文数(47)
1. ・・・・・
2. ・・・・・
3. ・・・・・
1. ・・・・・
2. ・・・・・
3. ・・・・・

図 4.8: システムの出力

第5章　評価

5.1　評価クエリ

提案手法を用いたオピニオンマイニングシステムを実装し, 評価実験を行った. 実験に用いたクエリを表 5.1 に示す. クエリの Q_1 と Q_2 は提案手法の設計・開発の際に使用したクエリであり, 評価データとして利用しない.

クエリに含まれる「賛成」「反対」はブログ検索エンジンを利用し, 効率よく意見文を取得する為のキーワードである. 他のキーワードに加え,「賛成」「反対」を含むウェブページは意見文を含む可能性が高く, 調査対象として有用なページである可能性が高い. しかし, 評価の際に偏りが発生する可能性を考慮し, 検索上位に意見文を含むページの割合が低い Q_6-Q_8 のクエリに対してのみ利用した. ただし,4.3.1 項で述べたように, 極性判定対象文を選択する際には, クエリとして与えられた全てのキーワード群の近傍にある文を選択しているが, 「賛成」「反対」は極性判定対象文の選択の際は無視し, それ以外のキーワード群の近傍にある文を選択した.

表 5.1: クエリ一覧

Q_1: 原発 再稼働	Q_2: 消費税 増税
Q_3: 赤ちゃんポスト 批判	Q_4: オリンピック 東京 開催
Q_5: TPP メリット	
Q_6: 英語教育 小学校 (賛成 or 反対)	
Q_7: 胃瘻 (賛成 or 反対)	Q_8: 女性専用車 (賛成 or 反対)

5.2　引用箇所検出の評価

5.2.1　引用箇所検出の評価方法

本論文で提案する引用箇所検出手法を評価する. ここでは, 文を単位として, その文が引用箇所に含まれているか否かを判定したときの精度, 再現率,F 値を評価基準とした. その定義を式 (5.1)(5.2)(5.3) に示す. 各クエリについて, ブログ検索モジュールで取得した上位 50 件のブログ記事に対し, 個々の文が引用文であるかを人手で判定した. この結果を正解データとして, 精度, 再現率,F 値を算出した.

$$精度\,(P) = \frac{システムが出力した正しい引用文の総数}{システムが出力した引用文の総数} \tag{5.1}$$

$$再現率\,(R) = \frac{システムが出力した正しい引用文の総数}{引用文の総数} \tag{5.2}$$

$$\text{F 値} = \frac{2 \times P \times R}{P + R} \tag{5.3}$$

5.2.2 引用箇所判定の評価結果

引用箇所検出の評価結果を表5.2に示す. 表5.2は, クエリ毎及び6つのクエリ全て(ALL)について, システムによるブログ記事内の引用箇所の判定の精度, 再現率,F 値を示している. また, システムが出力したブログ中の文の総数を「総文数」, 人手で判定したブログ中の引用文の総数を「引用文数」として示している.

表 5.2: 引用箇所検出の評価

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
精度	0.950	1.00	0.957	0.899	0.885	0.991	0.942
再現率	0.663	0.623	0.705	0.822	0.677	0.726	0.711
F 値	0.781	0.767	0.812	0.859	0.767	0.838	0.810

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
総文数	9,913	2,919	10,626	7,628	5,436	7,629	44,151
引用文数	4,459	1,118	7,881	3,687	2,468	2,774	22,387

引用箇所検出の精度, 再現率,F 値はそれぞれ94%,71%,81%であった. 引用箇所検出の精度は比較的高いといえる. このことから, 引用を示唆するキーワードの周辺のブロックから引用箇所を正確に検出できることがわかった. しかし, 再現率はやや低い値となった. これは, 引用を示唆するキーワードが存在しなかったケース, 著者の記事と引用箇所の境界が曖昧なケースでは, 本来引用箇所として検出するべき引用文を検出できなかったためである. 引用を示唆するキーワードを増やす, あるいはより高度な境界の判定を行うことで再現率が向上する可能性がある. 本研究では, 引用を示唆するキーワードとして, 各種新聞の名称や, 予備実験で用いたクエリに対する検索結果から人手で取得した引用を示唆するキーワードを使用した. しかし, ウェブ上の媒体は, 大手マスメディア以外にも掲示板のまとめ記事など数多く存在し, また今後も増加する可能性があるため, 引用を示唆するキーワード

は自動で獲得することが望ましい．また, 本手法では, 著者自身の記述と引用箇所の境界が曖昧なケースでは, リンク先のウェブページのテキスト情報を利用し, 引用文の判定を行ったが, 効率よく引用文を判定することを優先し, クエリを含む文のみを対象としたため, 引用箇所の正確な判定を行うことができないケースが多くあった．よって, 文の類似度を測るモジュールを高速化し, 元の記事の文と引用先のウェブページ中の文の全ての組について類似度を測ることによって, 文単位の引用文の判定の正確性が向上すると考えられる．

また, 表 5.2 6 つのクエリに対して検索されたブログページにおける総文数, 引用文数はそれぞれ 44,151 文,22,387 文であった．総文数の約 50%は引用文であり, ブログ記事は他者の記事の引用が多いことがわかった．ブログ記事では, 他者による記事の引用が, 情報の補足としてだけでなく, ページのほとんどを占めるケースも多く見られ, その取り扱いに注意する必要があるといえる．本研究で行った引用箇所検出の評価は, 文単位の精度, 再現率,F 値を評価基準としたが, ブログページによって文の総数が異なるため, 理想的な評価方法とはいえない．ブログページに含まれる引用箇所をブロック単位でカウントし, 精度, 再現率をはかる評価方法も検討すべきである．

5.3 文の極性判定の評価

5.3.1 文の極性判定の評価方法

文の極性判定手法の結果を評価する．ここでは, ブログ記事中の全ての文 (引用箇所も含む) のうち, システムが「賛成」または「反対」と判定した文のみを評価の対象とする．以下, これを「極性評価対象文」とする．極性評価対象文について, 以下の 5 つのラベルを人手で付与した．

- 「賛成 (T)」 文がクエリのトピックに対して賛成を表明している
- 「反対 (T)」 文がクエリのトピックに対して反対を表明している
- 「賛成 (O)」 文がクエリのトピック以外の対象に対して賛成を表明している
- 「反対 (O)」 文がクエリのトピック以外の対象に対して反対を表明している
- 「中立」 文が賛成もしくは反対を表明していない

次に,「正解率」「正解率 (極性のみ)」を以下のように定義し, これを文の極性判定の評価基準とする．

$$\text{正解率} = \frac{\text{システムの賛成・反対の判定が完全に正しい文の数}}{\text{文極性評価対象文数}} \tag{5.4}$$

$$\text{正解率 (極性のみ)} = \frac{\text{システムの賛成・反対の判定が部分的に正しい文の数}}{\text{文極性評価対象文数}} \tag{5.5}$$

式 (5.4) における「完全に正しい」文とは, 意見文の極性が正しいだけでなく, クエリのトピックに対して賛成や反対を表明している文をさす. すなわち, 正解ラベルが「賛成 (T)」または「反対 (T)」で, システムが「賛成」または「反対」と正しく判定できた文である. 一方, 式 (5.5) における「部分的に正しい」とは, 意見文の言及対象がクエリのトピックであるかないかに関わらず, 意見文の極性が正しい場合をさす. すなわち, 正解ラベルが「賛成 (T)」「賛成 (O)」または「反対 (T)」「反対 (O)」で, システムが「賛成」または「反対」と正しく判定できた文である. ここでは, 極性評価対象文, すなわちシステムが極性ありと判定した文のみを評価の対象としている. すなわち, 式 (5.4)(5.5) の正解率は極性判定の精度に相当し, 再現率は評価していない. これは, ブログ記事中の全ての文に対して人手で極性のラベルを与えることが困難なだったためである.

5.3.2 文の極性判定の評価結果

文の極性判定の評価結果を表 5.3 に示す. 表 5.3 は, クエリ毎及び 6 つのクエリの全て (ALL) について, 正解率, 極性のみに対しての正解率, 極性評価対象文数を示したものである.

表 5.3: 文の極性判定の評価

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.163	0.114	0.160	0.167	0.203	0.182	0.180
正解率 (極性のみ)	0.224	0.371	0.240	0.396	0.480	0.462	0.426
極性評価対象文数	49	70	25	96	306	132	678

正解率は 18%, 極性のみに対しての正解率は 42%であり, ともに低い値となった. 本システムの文極性判定では, 文内の用言および名詞の評価極性のみを手がかりとした単純な手法を使用したため, 正解率が低くなったと考えられる. 文極性判定において多く見られたエラーは,「中立」の文中に名詞の評価表現が出現したため,「賛成」あるいは「反対」と誤った出力を行うケースであった. 例えば,「予備知識を書いておきました」のように, 中立的意味の文に「知識」というポジティブな名詞の評価表現が含まれるため,「賛成」の文として判定してしまうことが多かった. しかし,「予備知識を書いておきました」という文は状況によってポジティブと捉えられることもあるため, 名詞の評価表現が文極性判定に有効でないとも言い切れない. よって, 文単位ではなく周辺の文脈まで考慮した上で極性を決定しなければ精度の高い判定は難しいと考えられる. さらに,「正解率」が「正解率 (極性のみ)」よりも明らかに低いことから, トピックに関連した意見文の取得がうま

くいっていないことがわかる．本システムでは, 全てのクエリが出現する周辺の文は, クエリに対して述べられた意見文であるという前提のもとで, クエリ周辺の範囲に絞って極性判定対象文の収集を行った．しかし, 結果を見ると, 極性が正しく判定された意見文のうち58%はトピックに関係のない文であることがわかった．クエリ近傍の文の意見文が必ずしもトピックに関連するものではないことがわかる．また, ブログ記事中の総文数が 44,151 であるのに対し, 極性評価対象文数が 678 と非常に少ないことがわかった．特にクエリ Q_3 と Q_5 では, 検索されたブログの件数 (50) を下回る数となった．両クエリでは, クエリの全てのキーワードが十分近い位置 (距離 2 以内) に存在しないために, 極性判定対象文の選択に失敗しているブログが多く見られた．このような場合, トピックに関係ある意見文が見られないと仮定し, ブログ記事は「中立」と判定した．先に示す表 5.9 から, クエリ Q_3 と Q_5 の「中立」の判定の F 値は 0.838,0.850 と他のクエリと比べて高いことから, この仮定はある程度正しかったといえる．

5.4 ブログ極性判定の評価

5.4.1 ブログ極性判定の評価方法

ブログの極性判定については, 表 5.4 に示した 5 つのシステムを比較評価する．引用箇所の検出・除去については 3 通りの方法を比較する．システム 1 は, 引用箇所判定タスクを経ず, 極性判定タスクの処理を行うシステムである．すなわち, ブログ記事中に著者の記事と他者の記事が混在した可能性のあるブログをそのまま極性判定している．システム 2 は, システムによって引用箇所の検出と削除を行った後, 極性判定タスクの処理を行うシステムである．すなわち, 引用箇所と判定されたテキストは極性判定に利用しない．システム 3 は, 引用箇所を人手で判定した後に, 極性判定タスクの処理を行うシステムである．さらに,「引用箇所の検出」と「極性判定対象文の選択」の処理の順序について 2 種類の方法を比較する．システム 2-a, システム 3-a では,「引用箇所除去」の処理を行ったのちに「極性判定対象文の選択」の処理を行う．システム 2-b, システム 3-b では,「極性判定対象文の選択」処理を行ったのちに「引用箇所除去」の処理を行う．後者では, 極性判定対象文を選択する際, すなわちクエリの近傍にある文の範囲を決定する際, 引用箇所を除去していないため, 前者に比べてより多くの文が極性判定対象文として選択される可能性がある．例えば, 図 5.1 では, S_{24} にクエリの全てのキーワードが出現するため, 極性判定対象文は S_{22} から S_{26} である．また, 引用を示唆するキーワード「転載」が存在するため, 罫線に挟まれた S_{20} から S_{25} までが引用箇所である．このとき, システム 2-b, システム 3-b では極性判定対象文を選択した後に引用箇所の除去を行うため,S_{26} が極性判定対象文として残る．一方, システム 2-a,3-a では引用箇所を先に除去するため, クエリのキーワードを含む S_{24},S_{25} も除去され, 極性判定対象文は 1 つも選択されない．

各クエリ毎に検索された 50 のブログ記事について, その記事がトピックに対してどのような立場を表明しているのかを人手で判定し,「賛成」「反対」「中立」のいずれかのラベ

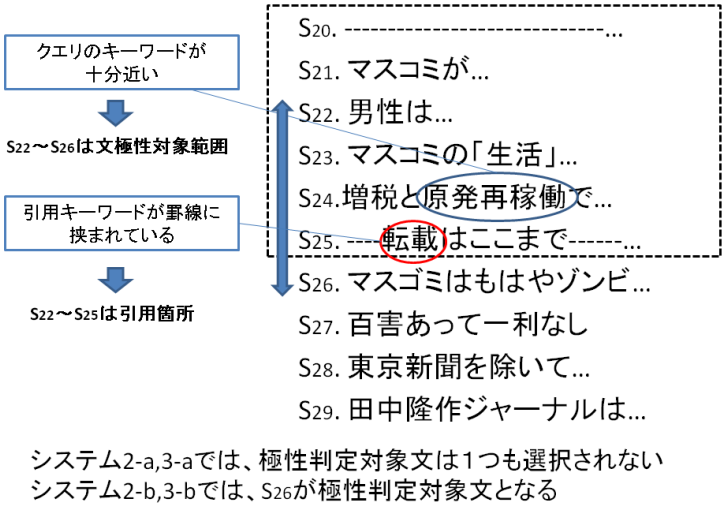

図 5.1: 極性判定対象文の選択例

表 5.4: 比較したシステム

	引用箇所検出/除去	処理の流れ
システム 1	なし	
システム 2-a	自動	引用箇所除去→極性判定対象文の選択
システム 2-b	自動	極性判定対象文の選択→引用箇所除去
システム 3-a	人手	引用箇所除去→極性判定対象文の選択
システム 3-b	人手	極性判定対象文の選択→引用箇所除去

ルを正解クラスとして付与した．さらに, 以下の評価基準によってブログ記事の極性判定の結果を評価した.

- 正解率

$$\text{正解率} = \frac{\text{システムの賛成・反対・中立の判定が正しいブログの数}}{\text{総ブログ数}} \tag{5.6}$$

- クラス別の精度, 再現率,F 値
  「賛成」「反対」「中立」のそれぞれのクラスについて, 判定の精度, 再現率,F 値を算出した.

5.4.2 ブログ極性の評価結果

ブログ記事の極性判定の正解率を以下に示す．表 5.5 はシステム 1, 表 5.6 はシステム 2-a と 2-b, 表 5.7 はシステム 3-a と 3-b の結果である.

表 5.5: ブログ記事の極性判定の正解率 (記事全体で判定)

システム 1 (全体)

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.680	0.540	0.680	0.600	0.400	0.420	0.553
正解ブログ数	34	27	34	30	20	21	166
総ブログ数	50	50	50	50	50	50	300

表 5.6: ブログ記事の極性判定の正解率 (引用箇所を自動除去)

システム 2-a (引用箇所除去→対象文選択)

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.700	0.520	0.720	0.600	0.500	0.480	0.587
正解ブログ数	35	26	36	36	25	24	176
総ブログ数	50	50	50	50	50	50	300

システム 2-b (対象文選択→引用箇所除去)

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.680	0.520	0.720	0.600	0.480	0.480	0.580
正解ブログ数	34	26	36	30	24	24	174
総ブログ数	50	50	50	50	50	50	300

表 5.7: ブログ記事の極性判定の正解率 (引用箇所を人手で除去)

システム 3-a (引用箇所除去→対象文選択)

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.700	0.540	0.740	0.600	0.520	0.520	0.607
正解ブログ数	35	27	37	30	26	26	182
総ブログ数	50	50	50	50	50	50	300

システム 3-2b (対象文選択→引用箇所除去)

	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
正解率	0.720	0.540	0.720	0.620	0.540	0.500	0.607
正解ブログ数	36	27	36	31	27	25	182
総ブログ数	50	50	50	50	50	50	300

記事全体でのブログの極性判定を行うシステムと比べて, 引用箇所を除去してブログの極性判定を行うシステム方が正解率が高い. この結果から, 引用箇所を除去した後に極性判定を行い, ブログ著者自身の意見に重みを持たせた解析を行うという本研究のアプローチは正しかったといえる. また, 引用箇所を人手で除去したとき, システム 3-a とシステム 3-b の正解率は同じであった. 一方, 提案手法によって引用箇所を除去したとき, わずかではあるが, 引用箇所を除去してから極性判定対象文を選択するシステム 2-a の方が,2 つの処理の順序を逆にしたシステム 2-b よりも F 値が高かった.

次に, 賛成, 反対, 中立のクラス毎の精度, 再現率,F 値を示す. 表 5.8 はシステム 1, 表 5.9 はシステム 2-a と 2-b, 表 5.10 はシステム 3-a と 3-b の結果である.

表 5.8: ブログ記事の極性判定 クラス別の評価 (記事全体で判定)

システム 1(全体)

		Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	精度	0.667	0.0062	0.000	0.182	0.105	0.077	0.15
賛成	再現率	0.267	1.000	0.000	0.500	0.400	0.143	0.29
賛成	F 値	0.381	0.118	0.000	0.267	0.167	0.100	0.19
反対	精度	0.000	0.444	0.667	0.500	0.500	0.389	0.42
反対	再現率	0.000	0.444	0.154	0.385	0.500	0.636	0.42
反対	F 値	0.000	0.444	0.250	0.435	0.500	0.483	0.42
中立	精度	0.811	0.880	0.727	0.793	0.692	0.684	0.78
中立	再現率	0.857	0.550	0.941	0.697	0.333	0.406	0.64
中立	F 値	0.833	0.677	0.821	0.742	0.450	0.510	0.70

表 5.9: ブログ記事の極性判定 クラス別の評価 (引用箇所を自動除去)

システム 2-a (引用箇所除去→対象文選択)

		Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	精度	0.800	0.000	0.000	0.154	0.000	0.077	0.143
賛成	再現率	0.267	0.000	0.000	0.500	0.000	0.143	0.257
賛成	F 値	0.400	0.000	0.000	0.235	0.000	0.100	0.184
反対	精度	0.000	0.429	1.000	0.500	1.000	0.467	0.464
反対	再現率	0.000	0.333	0.154	0.308	0.154	0.636	0.406
反対	F 値	0.000	0.375	0.267	0.381	0.267	0.538	0.433
中立	精度	0.795	0.793	0.739	0.828	0.739	0.727	0.779
中立	再現率	0.886	0.575	1.000	0.727	1.000	0.500	0.701
中立	F 値	0.838	0.667	0.850	0.774	0.850	0.593	0.738

システム 2-b (対象文選択→引用箇所除去)

		Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	精度	0.667	0.062	0.000	0.182	0.000	0.077	0.154
賛成	再現率	0.267	1.000	0.000	0.500	0.000	0.143	0.286
賛成	F 値	0.400	0.118	0.000	0.267	0.000	0.100	0.200
反対	精度	0.000	0.375	1.000	0.500	1.000	0.467	0.448
反対	再現率	0.000	0.333	0.154	0.308	0.154	0.636	0.406
反対	F 値	0.000	0.353	0.267	0.381	0.267	0.538	0.426
中立	精度	0.795	0.846	0.739	0.774	0.739	0.727	0.780
中立	再現率	0.886	0.550	1.000	0.727	1.000	0.500	0.687
中立	F 値	0.838	0.667	0.850	0.750	0.850	0.593	0.730

表 5.10: ブログ記事の極性判定 クラス別の評価 (引用箇所を人手で除去)

システム 3-a (引用箇所除去→対象文選択)

		Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	精度	1.000	0.000	0.000	0.100	0.133	0.091	0.151
賛成	再現率	0.267	0.000	0.000	0.250	0.400	0.143	0.229
賛成	F 値	0.421	0.000	0.000	0.143	0.200	0.111	0.182
反対	精度	0.000	0.429	1.000	0.429	0.643	0.538	0.481
反対	再現率	0.000	0.333	0.231	0.231	0.500	0.636	0.391
反対	F 値	0.000	0.375	0.375	0.300	0.563	0.562	0.431
中立	精度	0.816	0.800	0.723	0.788	0.762	0.692	0.764
中立	再現率	0.886	0.600	1.000	0.788	0.593	0.562	0.741
中立	F 値	0.849	0.686	0.840	0.788	0.667	0.621	0.753

システム 3-b (対象文選択→引用箇所除去)

		Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	精度	1.000	0.067	0.000	0.200	0.143	0.083	0.182
賛成	再現率	0.267	1.000	0.000	0.500	0.400	0.143	0.286
賛成	F 値	0.421	0.125	0.000	0.286	0.211	0.105	0.222
反対	精度	0.000	0.375	1.000	0.500	0.600	0.538	0.472
反対	再現率	0.000	0.333	0.154	0.308	0.500	0.636	0.391
反対	F 値	0.000	0.353	0.267	0.381	0.545	0.583	0.427
中立	精度	0.821	0.852	0.708	0.781	0.714	0.692	0.760
中立	再現率	0.914	0.575	0.000	0.758	0.556	0.562	0.726
中立	F 値	0.865	0.687	0.829	0.769	0.625	0.621	0.743

「賛成」「反対」「中立」全体を通して, わずかではあるが, 引用箇所を除去してブログの極性判定を行うシステムの方がF値の値が高かった. この結果から, 引用箇所を除去した後に極性判定を行い, ブログ著者自身の意見に重みを持たせた解析を行うという本研究のアプローチは正しかったといえる. また,「賛成」「反対」「中立」の判定におけるF値を見ると,「中立の」F値が一番高く,「中立」のブログを正しく判定できたケースが多いことがわかった. 「賛成」と「反対」の極性判定にエラーが多かったのは, 文単位での極性判定の正解率が悪かった事も要因の一つとして挙げられるが, ブログ記事からの意見文の取得率が悪かったことも要因と考えられる. これは, クエリのキーワードを含むが調査対象とするトピックに対して論じたブログでない記事が多く検索されてしまったためである. クエリを自然言語文とした高度な検索を行うことや, 調査対象とするトピックのブログ記事に含まれる特徴語を手がかりとして, 検索されるブログ記事をさらに絞るなど, トピックに関連したブログ記事の検索精度を向上させる必要がある.

次に, 正解クラスとシステムが判定したクラスの詳細な内訳を示す. 表5.11はシステム1, 表5.12はシステム2-aと2-b, 表5.13はシステム3-aと3-bの結果である. また, 表5.14はそれぞれのシステムにおける極性判定結果の抜粋であり,「賛成」「反対」「中立」を正しく判定した件数と,「中立」のブログを「賛成」「反対」と誤判定した件数を示している. 提案システムでは, ブログページから本文が検出できなかったとき, または極性評価対象文が1つも選択できなかったとき, 無条件に「中立」と判定している. これらの表における「中立 (判定なし)」は上記のような場合を指し, システムが式 (4.3) の値に基づいて「中立」と判定した場合と区別している.

表 5.11: ブログ記事の極性判定結果の詳細 (記事全体で判定)

システム 1(全体)

正解クラス	出力クラス	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	賛成	4	1	0	2	2	1	10
賛成	反対	4	0	0	1	2	1	8
賛成	中立	0	0	0	0	1	1	2
賛成	中立 (判定なし)	7	0	3	1	0	4	15
反対	賛成	0	2	2	3	6	3	16
反対	反対	0	4	2	5	9	7	27
反対	中立	0	0	2	1	2	1	6
反対	中立 (判定なし)	0	3	7	4	1	0	15
中立	賛成	2	13	1	6	11	9	42
中立	反対	3	5	1	4	7	10	30
中立	中立	3	0	0	1	3	4	11
中立	中立 (判定なし)	27	22	32	22	6	9	118

表 5.12: ブログ記事の極性判定結果の詳細 (引用箇所を自動除去)

システム 2-a (引用箇所除去→対象文選択)

正解クラス	出力クラス	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	賛成	4	0	0	2	2	1	9
賛成	反対	3	0	0	1	2	1	7
賛成	中立	1	1	0	0	1	1	4
賛成	中立 (判定なし)	7	0	3	1	0	4	15
反対	賛成	0	1	2	5	6	3	17
反対	反対	0	3	2	4	10	7	26
反対	中立	0	2	2	0	1	1	6
反対	中立 (判定なし)	0	3	7	4	1	0	15
中立	賛成	1	13	0	6	8	9	37
中立	反対	3	4	0	3	6	7	23
中立	中立	4	1	2	1	7	7	23
中立	中立 (判定なし)	27	22	32	22	6	9	118

システム 2-b (対象文選択→引用箇所除去)

正解クラス	出力クラス	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	賛成	4	1	0	2	2	1	10
賛成	反対	4	0	0	1	2	1	8
賛成	中立	0	0	0	0	1	1	2
賛成	中立 (判定なし)	7	0	3	1	0	4	15
反対	賛成	0	2	2	3	6	3	16
反対	反対	0	3	2	4	10	7	26
反対	中立	0	1	2	2	1	1	7
反対	中立 (判定なし)	0	3	7	4	1	0	15
中立	賛成	2	13	0	6	9	9	39
中立	反対	3	5	0	3	6	7	24
中立	中立	3	0	2	2	6	7	20
中立	中立 (判定なし)	27	22	32	22	6	9	118

表 5.13: ブログ記事の極性判定結果の詳細 (引用箇所を人手で除去)

システム 3-a (引用箇所除去→対象文選択)

正解クラス	出力クラス	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	賛成	4	0	0	1	2	1	8
賛成	反対	4	0	0	1	2	1	8
賛成	中立	0	1	0	1	1	1	4
賛成	中立 (判定なし)	7	0	3	1	0	4	15
反対	賛成	0	1	0	5	5	1	12
反対	反対	0	3	3	3	9	7	25
反対	中立	0	2	3	1	3	3	12
反対	中立 (判定なし)	0	3	7	4	1	0	15
中立	賛成	0	12	0	4	8	9	33
中立	反対	4	4	0	3	3	5	19
中立	中立	4	2	2	4	10	9	31
中立	中立 (判定なし)	27	22	32	22	6	9	118

システム 3-b (対象文選択→引用箇所除去)

正解クラス	出力クラス	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	ALL
賛成	賛成	4	1	0	2	2	1	10
賛成	反対	4	0	0	1	2	1	8
賛成	中立	0	0	0	0	1	1	2
賛成	中立 (判定なし)	7	0	3	1	0	4	15
反対	賛成	0	2	0	3	4	1	10
反対	反対	0	3	2	4	9	7	25
反対	中立	0	1	4	2	4	3	14
反対	中立 (判定なし)	0	3	7	4	1	0	15
中立	賛成	0	12	0	5	8	10	35
中立	反対	3	5	0	3	4	5	20
中立	中立	5	1	2	3	9	8	28
中立	中立 (判定なし)	27	22	32	22	6	9	118

表 5.14: ブログ記事の極性判定結果の詳細の抜粋

正解クラス	出力クラス	システム 1	2-a	2-b	3-a	3-b
賛成	賛成	10	9	10	8	10
反対	反対	27	26	26	25	25
中立	中立	11	23	20	31	28
中立	中立 (判定なし)	118	118	118	118	118
中立	賛成	42	37	39	33	35
中立	反対	30	23	24	19	20

表 5.14 から引用箇所を除去した後に極性判断を行うシステム 2-a,2-b,3-a,3-b は, 記事全体で極性判定を行うシステム 1 と比べて,「賛成」と「反対」のブログ正解数はほぼ同じなのに対して,「中立」のブログの正解数が向上している. また,「中立」クラスをシステムが「賛成」または「反対」と誤検出した数が減少している. すなわち, 本来「中立」のブログなのにも関わらず, 引用箇所に含まれる評価表現により「賛成」または「反対」と誤判定してしまう場合が減っているとわかる. このことから, オピニオンマイニングにおいて引用箇所を除去することは, 他者のコメントに含まれる評価表現による誤判定をある程度抑制していることがわかる.

第6章　結論

6.1　結論

本論文では, 引用文を考慮したオピニオンマイニングシステムを提案した. また, 提案手法の評価実験と結果について論じた. 実験の結果, トピックに関するブログ著者の立場を集計するオピニオンマイニングシステムでは, 他者の記事やコメントの引用箇所を検出, 除去して分析することにより, ブログ記事の極性判定の正解率が向上することが確認できた. 個人の意見の集約を図るオピニオンマイニングシステムでは, いかにして著者の意見文を取得するのかが重要である. ブログ記事のように著者の記事と他人の記事やコメントが混在することの多いテキストを対象としたオピニオンマイニングでは, 引用箇所を推定し除去することが特に有用であるといえる.

6.2　今後の課題

本研究における課題として, 文及びブログの評価極性判定の精度の低さがある. 既存の研究には, 係り受け構造を利用した極性判定など, 高度な処理によって極性判定を行っているものも多く, 改善の余地は十分ある. 今回の評価実験では, 中立的な意味の文に出現する名詞の評価表現の取り扱いが不適切であったことが判定誤りの主な原因であった. 名詞の評価表現は, 語自身がポジティブやネガティブの意味を持つが, 賛成あるいは反対の評価はある対象に対して抱く感情であるため, 名詞の評価表現のみで文の極性を判断することに問題があったといえる. だが, あるトピックに関するテキストでの名詞の評価表現の出現頻度により, 著者が抱くイメージを推測できる可能性がある. 例えば, 原発のトピックについてブログの極性判定を行った結果, 否定的な立場を表明したブログ記事が多く見られたが, 記事には「爆発」「アレルギー」「嘘」「災害」など否定の極性を持つ名詞の評価表現が多く見られた. これらは原発のトピックにおいては評価表現単独で否定的見解を表わすといえる. このようなトピックに固有の肯定的・否定的表現を自動的に推定できれば, テキスト内の名詞の評価表現により著者がトピックに対して抱くイメージを分析し, 正しいブログの極性判定につながる可能性があると考えられる. つまり, 名詞の評価表現は著者の思想や意見を間接的に捉えることができるといえる. また, 今回実装したシステムでは, 極性判定対象文の範囲を限定したため, 効果的な意見文の取得が行えなかった. 極性判定対象文を限定せず, ブログ記事の全ての文に対して極性判定を行い, またトピックに深く関連した文の極性を重視してブログ全体の極性判定を行えば, より多くの意見を捉えることが

できよう. 例えば, クエリのキーワードに近い文ほどトピックに深く関連していると仮定し, キーワードまでの距離に応じて文の極性に重みを与える手法が考えられる.

参考文献

[1] 奥村学, 南野朋之, 藤木稔明, 鈴木泰裕. blog ページの自動収集と監視に基づくテキストマイニング, 人工知能学会研究会資料,SIG-SW-ONT-A401-01, 2006.

[2] 藤井敦. Opinionreader:意志決定支援を目的とした主観情報の集約・可視化システム, 電子情報通信学会論文誌 D,Vol.J91-D,No.2,pp.459-470,2008.

[3] 乾孝司, 奥村学. テキストを対象とした評価情報の分析に関する研究動向, 自然言語処理,Vol.13,No.3, 2006.

[4] 小林のぞみ，乾健太郎，松本裕治，立石健二，福島俊一. 意見抽出のための評価表現の収集, 自然言語処理，Vol.12, No.2, pp.203-222, 2005.

[5] 東山昌彦, 乾健太郎, 松本裕治. 述語の選択選好性に着目した名詞評価極性の獲得, 自然言語処理，言語処理学会第 14 回年次大会論文集, pp.584-587, 2008.

[6] 鍜治伸裕, 喜連川 優. 自動構築した評価文コーパスからの評価表現辞書の構築, 日本データベース学会 Letters,Vol.6, No.1 pp.41-44, 2007.

[7] 森田一, 池田大介, 奥村学. 係り受け構造を利用した発言の賛否の分類, 人工知能学会全国大会, 2007.

[8] 水野淳太, 渡邉陽太郎, エリックニコルズ, 村上浩司, 乾健太郎, 松本祐治. 文間関係認識に基づく賛成・反対意見の俯瞰, 情報処理学会論文誌,Vol.52,No.12,pp.1140-1148,2011.

[9] 山下清美. ウェブログの心理学, 人工知能学会研究資料,SIG-SWO-A401-03, 2005.

[10] Hideyuki Shibuki, Takahiro Nagai, Masahiro Nakano,Rintaro Miyazaki, Madoka Ishioroshi, and Tatsunori Mori. A method for automatically generationg a mediatory summary to verify credibility of information on the Web, In proceedings of the COLING,pp.1140-1148,2010.

[11] Ana-Maria Popescu, and Oren Etzioni. extracting product features and opinions from reviews, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing,Pages 339-346,2005.

[12] Awais Athar. Sentiment Analysis of Citations using Sentence Structure-Based Features, Proceedings of the ACL-HLT 2011 Student Session,pages 81-87,2011.

付 録A　引用を示唆するキーワード一覧

4.2.1 項で述べた引用を示唆するキーワードの一覧を以下に示す．これらはブロック単位の引用箇所検出の際に用いる．キーワードの大部分は新聞の名称であり, 新聞記事のリンクサイトである「新聞ネット」[1]から網羅的に収集した.

NAVER まとめ, 社説, 転載開始, 転載終わり, 引用, 転載, 掲載, 時事通信, 毎日新聞, 朝日新聞, 読売新聞, 産経新聞, 産経ニュース, 日本経済新聞, 日刊スポーツ, スポーツニッポン, サンケイスポーツ, スポーツ報知, デイリースポーツ, 中日スポーツ, 西日本スポーツ, 東京スポーツ, 日刊ゲンダイ, 夕刊フジ, 北海道新聞, 函館新聞, 釧路新聞, 室蘭民報, 十勝毎日新聞, 苫小牧民報, 名寄新聞, 北海民友新聞, 道北日報, プレス空知, オホーツク新聞, 東奥日報, 陸奥新報, 津軽新報, デーリー東北新聞, 岩手日報, 岩手日日新聞, 東海新報, 胆江日日新聞, 河北新報, 三陸新報, 三陸河北新報, 大崎タイムス, 秋田魁新報, 北羽新報, 北鹿新聞, あきた北新聞, 山形新聞, 米沢新聞, 鶴岡タイムス, 酒田・鶴岡 コミュニティ新聞, 福島民報, 福島民友新聞, 夕刊いわき民報, 茨城新聞, 常陽新聞, よみうりタウンニュー, 下野新聞, 真岡新聞, 上毛新聞, 群馬経済新聞, 桐生タイムス, 埼玉新聞, 日刊文化新聞, 日刊新民報, 千葉日報, 稲毛新聞, 房日新聞, ニューファミリー新聞, 市川よみうり新聞, 市川ジャーナ, 東京新聞, 都政新報, 全東京新聞, 新宿区新聞, 西多摩新聞, 小笠原新聞, 多摩ニュータウンタイム, 足立よみうり新聞, 高島平新聞, 神奈川新聞, 横浜タウン新聞, 湘南よみうり新聞, 湘南新聞, 江ノ電沿線新聞, 多摩川新聞, 辻堂タイムズ, 神静民報, 相模経済新聞, あおばタイムズ, 市民かわら版, 新潟日報, 十日町新聞, 上越よみうり, 信濃毎日新聞, 長野日報, 南信州新聞, 市民タイムス, 軽井沢新聞, 更埴新聞, 山梨日日新聞, 山梨新報, 八ヶ岳ジャーナル, 北日本新聞, 富山新聞, 富山県市町村新聞, 北國新聞, 北陸中日新聞, 福井新聞, 日刊県民福井, 岐阜新聞, 高山市民時報, 静岡新聞, 沼津朝日新聞, 伊豆新聞, 富士ニュース, 中日新聞, 中部経済新聞, 東愛知新聞, 東海日日新聞, 尾張中央タイムス, 滋賀報知新聞, 滋賀新聞, 伊勢新聞, 東海経済新聞, 三重タイムズ, 夕刊三重新聞, ローカルみえ, 伊和新聞, 吉野熊野新聞, 京都新聞, 京都経済新聞, 両丹日日新聞, 両丹経済新聞, 洛南タイムス, 舞鶴市民新聞, 亀岡市民新聞, あやべ市民新聞, 大阪日日新聞, 河北新聞, 堺ジャーナル, 神戸新聞, 丹波新聞, 上郡民報, 奈良新聞, 紀伊民報, 紀州新聞, 南紀州新聞, ニュース和歌, ツー・ワン紀州, 日本海新聞, 山陰経済新聞, 山陰中央新報, 島根日日新聞, 山陽新聞, 岡山日日新聞, 備北民報, 中国新聞, 大陽新聞, 西広島タイム, 府中よみうり速報, 山口新聞, 宇部日報, 防府日報, 新周南新聞, 長門時事新聞, 長周新聞, 徳島新聞, tribune し

[1] http://www.fn69.com/local3.htm

こ, 四国新聞, 四国タイムズ, 愛媛新聞, 南海日日新聞, マイタウン今治新聞, 高知新聞, 西日本新聞, 有明新報, 佐賀新聞, 長崎新聞, 五島新報新聞, 壱岐日々新聞, 壱岐日報, 熊本日日新聞, 人吉新聞, 大分合同新聞, 大分団地新聞, 今日新聞, 地元新聞, あばかん, 宮崎日日新聞, 南日本新聞, 南海日日新聞, 南九州新聞, 沖縄タイムス, 琉球新報, 八重山毎日新聞, 宮古毎日新聞, 観光とけいざい, 産業新聞, 日経産業新聞, 日刊工業新聞, 日刊鉄鋼新聞, 半導体産業新聞, 生活産業新聞, 電気新聞, 電経新聞, 電波新聞, 繊研新聞, 日本繊維新聞, 繊維ニュース, 染織経済新聞, 日刊自動車新聞, 自動車タイヤ新聞, 日本ベアリング新聞, 週刊玩具通信, 商経機械新聞, 商工管財新聞, 化学工業日報, エアゾール＆スプレー産業新聞, ゴム報知新聞, 石鹸新報, 全国ドライ新聞, 洗剤新報, 塗料報知新聞, 科学新聞, エネルギーと環境, 石油化学新聞, 北海道石油新聞, ガスエネルギー新聞, 原子力産業新聞, 日刊建設工業新聞, 建設通信新聞, 建通新聞, 日刊建設産業新聞, 建設新聞, 建設速報, 日本工業経済新聞, 日本水道新聞, 日本下水道新聞, 環境新聞, 週刊住宅新聞, 住宅産業新聞, 住宅新報, 日本屋根経済新聞, セメント新聞, 運輸新聞, 航空新聞, 内運航海新聞, 日本海事新聞, 日刊海事プレス, 交通新聞, 東京交通新聞, 日経流通新聞, 生協流通新聞, 日刊通運情報, 物流ニッポン新聞, 物流ウィークリー, 日本流通産業新聞, 速報！株式情報, 日本証券新聞, 証券新報, 日本金融新聞, 金融経済新聞, 金融タイムス, 金融ファクシミリ新聞, 保険銀行日報, ニッキン, 投資日報, 税理士新聞, 消費経済新聞新聞, 文化通信, 映像新聞, 日刊合同通信, 月刊民放, New York Times, USA Today, Washington Post, Christian Science Monitor, Journal of Commerce, Wall Street Journal, Times／The Sunday Times, Daily Mirror, Independent, Financial Times, 人民日報, 時報週刊, 中央日報, 朝鮮日報, 東亜日報, 民族時報, 朝鮮新報, バンコク週報, 日刊まにら新聞, 世界日報, Japan Times, asahi.com, Daily Yomiuri On-Line, Mainichi Daily News, NHK ニュース, 時事ドットコム, WEB マガジン, 福島民報, 神奈川新聞社