

# 議題への明示的な賛否表現を含まない議員発言を対象とした 議会議事録のスタンス分類

猫本隆哲  
豊橋技術科学大学  
nekomoto.takanori.pf@tut.jp

秋葉友良  
豊橋技術科学大学  
akiba@cs.tut.jp

増山 繁  
東京理科大学  
masuyama@rs.tus.ac.jp

## 1 はじめに

NTCIR-15のQA Lab-PoliInfo-2タスク<sup>1)</sup>では、地方議会議事録から議員の意見やその理由・条件を要約して表示するといった複雑な実世界の質問応答技術を目指している[1]。近年、フェイクニュースが多大な悪影響を及ぼしていることから、ファクトチェックの重要性を再認識されている。情報獲得のために幅広く利用されている一般的なWeb検索エンジンでは、Eli Pariserが提唱した“filter bubble”[2]と呼ばれる概念によって、ユーザーが自分の視点に合わない情報から遠ざかることから、ファクトチェックが困難となっている。ファクトチェックのためには、議事録などの一次情報源をしっかりと把握しておく必要がある。

QA Lab-PoliInfo-2 Stance Classificationタスクでは、東京都議会における政治家の発言から、政治家の立場(スタンス)を推定することを目的としている<sup>2)</sup>。より具体的には、議会議事録から、議会中に議論される各議題について、各会派の賛否を分類する問題に取り組んだ。日本の議会議事録の記述は、議員ごとの発言の書き起こしであり、ほとんどの議員は討論の冒頭で複数の議案に対して簡単な賛否を述べることから、ここから一定数の議案に対してのスタンスが判明する。よって、Stance Classificationの評価結果では、この冒頭部分のみを用いることで90%以上の正解率で賛否分類できることが示されている[3]。しかし、一部には冒頭部分で賛否表明を行わない、または冒頭のみでは賛否表明されない議案が存在するといった場合があり、そのような場合に対応するには、冒頭以外の討論部分から賛否分類を行う必要がある。

本研究では、ある議案に対して明示的な賛否の表現のない討論の発言から賛否分類を行う問題に焦点

を当てた。賛成や反対といったターゲット単語のみに着目する単純なルールベース分類ではなく、これを除いた議員の発言全体から学習データを作成し、より多くの分野やタスクへの応用を可能とするためにBERT[4]ベースのスタンス分類器を構築した。

## 2 スタンス分類

本節では、QA Lab-PoliInfo-2 Stance Classificationタスクにおけるデータ構造、分類対象とする議事録の構造、本研究の問題設定について詳細を述べる。

### 2.1 データ構造

分類器には、会議録データと表1から構成される出題ファイルを入力として与え、空欄となる会派の賛否を埋めた出題ファイルを出力とする。

表1 出題ファイル例

Field name	Description	Example
ID	識別番号	Poliinfo2-StanceClassification- JA-Dry-Training-02543
Prefecture	都道府県	東京都
Meeting	会議名	平成31年第1回定例会, 第1回臨時会
MeetingStartDate	会議開始日	2019/2/20
MeetingEndDate	会議終了日	2019/3/28
Proponent	提案者	知事提出議案
BillClass	大カテゴリ	予算
BillSubClass	小カテゴリ	31年度予算
Bill	議案名	一般会計
BillNumber	議案番号	第一号議案
SpeakerList	議員と会派	{"増子ひろき": "都ファースト", "吉原修": "自民党"}
ProsConsPartyListBinary	会派と賛否(2値) 賛成, 反対	{"都ファースト": "賛成", "自民党": "反対"}
ProsConsPartyListTernary	会派と賛否(3値)※辞書型 賛成, 反対, 言及なし	{"都ファースト": "賛成", "自民党": "反対"}

議員らの発言は、出題ファイルのMeetingDateを参照し、期日が一致する会議録から抽出する。また、この他に収録されているフィールドは分類器への付加情報および、議員の発言をより詳細に検索するために用いる。

1) <https://poliinfo2.net/>

2) <https://poliinfo2.net/stance-classification-task/>

## 2.2 会議録データ

入力として利用可能な会議録データには、定例会や臨時会の他に委員会等の速記録があるが、後者は賛否が伴う議決を行うものではないため、スタンス分類に直結するのは前者での議員の発言である。定例会における議員の実際の発言を図 1 に示す。

"Speaker": "百四番 (村松みえ子君)",  
"Utterance": "私は、日本共産党都議団を代表して、第二百二十八号議案、東京都養育院条例を廃止する条例外六議案に反対の立場から討論を行います。  
東京都養育院は、福祉と医療、研究が連携した総合的なサービス提供を最大の特徴としており、世界的にも高い評価を受けております。ところが今回の提案は、これから迎える本格的な高齢社会に向け、養育院の、この福祉と医療と研究の一体的な運営を一層強化する必要があるときに、逆にそれを後退させるものであり、反対です。..."

図 1 討論例

議員は必ず議長から発言権が与えられた上で、始めにいくつか議案への簡単な賛否表明を行った上で、それぞれの議案についての詳細な討論を行う。ここでは、前者の発言を賛否表明文、後者を討論文と定義する。賛否表明文を図 2 に示す。

"Utterance": "私は、日本共産党都議団を代表して、第二百二十八号議案、東京都養育院条例を廃止する条例外六議案に反対の立場から討論を行います。"

図 2 賛否表明文例

賛否表明文は、議長から発言権が与えられた議員が始めに述べる形式的な賛否表明である。多くの場合、ここから特定議案(議案番号)の賛否表明が得られ、Stance Classification タスクの評価結果では、賛否表明文のみを参照することで 90%以上の正解率が実現されることが分かっている [3]。

次に討論文を図 3 に示す。

東京都養育院は、福祉と医療、研究が連携した総合的なサービス提供を最大の特徴としており、世界的にも高い評価を受けております。ところが今回の提案は、これから迎える本格的な高齢社会に向け、養育院の、この福祉と医療と研究の一体的な運営を一層強化する必要があるときに、逆にそれを後退させるものであり、反対です。..."

図 3 討論文例

討論文では、賛否表明を行なった議案への詳細な言及がなされる。賛否表明文と比較すると、同一議案への言及でありながらも、それについての根拠が明確に述べられている。しかし、討論文では賛成や反対といった単語は必ずしも出現しない。また、実際の発言は非常に長いものとなっている。

## 2.3 問題設定

賛否表明文を用いることで高い精度で賛否分類を行うことが可能であるが、陽に言及されない議題には対応できない。そこで本研究では、図 3 に示した討論文から、議案に対する賛否を分類する問題に焦点を当てる。そのために、議事録の各議員の各議案に対する討論文と、その議題への賛否を対応付けたペアデータを人手で構築した。このペアデータを学習データとテストデータに分割し、二値分類器の構築、評価を行なった。

## 3 手法

本節では、スタンス分類を行うための分類器の構築方法、学習データへの前処理、および分類器の精度を検証するための実験方法について述べる。

### 3.1 会派の特定

東京都議会におけるスタンス分類を行うための準備として、討論を行う議員の所属会派を事前に特定する必要がある。表 1 に示したように、会派と発言者の対応は SpeakerList から分かるが、代表質問者以外の議員は自身で所属政党を名乗らなければ会派との対応が取れない問題点がある。そこで、都議会 HP の検索エンジン<sup>3)</sup>と BeautifulSoup4 [5] を用いて定例会等に出席したすべての議員の所属政党を収録した辞書を別途で作成した。また、議員によっては政党移動または政党名変更等があるため、それを考慮した都議会議員 322 名のネスト型辞書を構築した。一例を図 4 に示す。

### 3.2 BERT を用いた分類

2.3 節で定義した討論文を対象に議員のスタンスを 2 値(賛成 or 反対)に判定する分類器を構築した。この分類器は、BERT の MRPC タスクに基づき作成したものであり、入力を単一の討論文としている。モデルには、公開されている日本語事前学習済みモデルの BASE 通常版を用いた [6]。このモデルは、

3) <https://search.metro.tokyo.lg.jp/>

```

"和泉なおみ": {
  "日本共産党": [
    "2014/3/5",
    "2015/6/17",
    "2016/6/7",
    "2017/3/15",
    "2017/3/28",
    "2018/3/14",
    "2019/3/12",
    "2020/2/27"
  ]
},
"中村ひろし": {
  "立憲・民主": [
    "2019/2/26",
    "2020/2/27"
  ],
  "民進・立憲": [
    "2018/3/1"
  ],
  "民進党": [
    "2016/6/8",
    "2017/9/26"
  ],
}

```

図4 議員/所属党派辞書

JUMAN [7] による単語分割に基づいて学習されているため、同じ形態素解析器を使用している。学習データの作成には、Poliinfo2 から提供されているデータを参照し、発言権が与えられた議員の討論文から議案名もしくは議案番号に完全に一致する発言のみを抽出し、使用した。

しかし、東京都議会における討論文には議題に上がった議案とは直接関連の無い内容が多く含まれており、BERT に入力できる単語数の最大値が 512 語であるため、何らかの前処理を適用する必要がある。今回は簡単な前処理として、討論文に特定の接続詞が含まれていた場合、それ以降の文は議題案とは無関係であると判断し、接続詞の後の文を全て削除している。特定の接続詞には、以下のパターンを想定している。

#### 特定の接続詞

(次に、.\*|次いで、.\*|続いて、.\*|最後に、.\*|さて、.\*)

なお、上記のパターンは、討論の途中に出現するものにはマッチせず、句読点があるもののみを考慮している。作成された学習データの総数は 442 件であり、データの各行には賛否ラベル (“0” : 反対 or “1” : 賛成) と討論文が収録されている。

### 3.3 編集距離を用いたデータ拡張

3.2 節で述べたように、学習データとして利用可能な議員の討論文は、議案名または議案番号との完全一致で取れるものに限定しているため、抽出が不十分であると考えられる。そこで、暫定的なデータ拡張として編集距離を用いて、議題に上がった議案名に関連性のある単語を討論文から抽出し、学習データ

を 442 件から 485 件へ増強した。

### 3.4 DP を用いたテキストセグメンテーション

学習データとしての討論文は、特定議案への言及文のみを収録し、その他の関連の無い議案への言及文は取り除くことが理想となる。3.2 節では、特定の接続詞以降の文を削除しているが、これもまたテキスト分割としては不十分であると考えられる。そこで、議員の発言から議題に関係のない発言を削除する手法として、Dynamic Programming (DP) を用いて討論文のセグメンテーションを行った。前提として、セグメントを考える際の最小単位は文であり、シーケンスデータでの 1 つのデータポイントは、今回の場合は 1 つの文となる [8]。実装したテキストセグメンテーションの手順は以下の通りである。

- Step 1.** テキストデータクラスを定義し、セグメント内のスコアを計算する  $h(t_1, t_2)$ 、テキストの長さを返す  $GetLength()$  を定義する。
- Step 2.**  $T$  をテキストの文のリスト、 $L$  をテキストの単語リスト (Mecab [9] を使用) とし、文間の類似語を表す  $L \times T$  スケールの  $matrix F$  を定義する。
- Step 3.** 行列  $F$  は疎行列であり、各文から単語を抽出し、 $L$  に含まれる単語が  $T$  にある場合、リスト  $L$  からその単語のインデックスを取り、 $F$  のインデックスの行に 1 を代入する。
- Step 4.** 文章  $i$  と  $j$  の類似度を表す  $matrix f$  とその転置行列の積を  $matrix d$  と定義する。同一文の類似度は  $0(d_{ii} = 0)$  とし、この  $matrix d$  を用いてセグメント内のスコアを計算する。
- Step 5.** セグメントに含まれるすべての文の類似度を足し合わせ、セグメントの長さで割ってスコアを得る。評価関数  $J$  は、セグメントの開始点を  $t_1$ 、終了点を  $t_2$  と定義する。

$$J = \frac{\sum_{i=t_1}^{t_2-1} \sum_{j=t_1}^{t_2-1} D_{ij}}{(t_2 - t_1)}$$

- Step 6.** DP の再帰関数に討論文と任意の数のセグメント数をオブジェクトとして与え、 $J$  が最大となるセグメントの分割点をインデックスとして返す。

なお、セグメントの数は DP へ任意に与えなければならぬため、ここでは討論文の長さに応じて任意の値でセグメンテーションを行っている。

### 3.5 分類器への付加情報

政治的なイデオロギーは都議会もまた例外ではなく、保守派やリベラル派といったものがあり、このような政党間では、議案の種類や提案政党などによって賛否に大きなバイアスが生じている。例として、保守派とリベラル派は双方の提案に反対することが多く、また知事が提案する議案には賛成することが多い。このような政党間や議案の提出元で変化する賛否のバイアスの解消を見込み、賛否ラベルと討論文の意味的な等価性を評価する分類器への付加情報として、表 1 に示したフィールド名に基づき、政党名 (party) と議案の提出元 (proponent) を与える実験を行った。なお、提出元は知事提出もしくは議員提出の 2 値である。

表 2 入力構成

データセット	入力	付加	出力
討論文のみ	document		賛成/反対
討論文 + 政党名	document	party	賛成/反対
討論文 + 提出元	document	proponent	賛成/反対

## 4 結果・考察

分類器の評価実験は、出題データ全てを賛成と分類する majority ベースラインと、BERT を用いたものの 2 種類に分けられる。加えて、BERT については分類器を変えずにいくつかの学習データの構築方法を提案し、精度の比較を行った。ハイパーパラメータの設定は、入力の最大系列長を 128、ミニバッチサイズを 32、学習係数を  $2e-5$ 、エポック数を 5 とし、モデルの評価には k 交差検証を用いた。各手法での分類結果を 3 に示す。DA は 4.2 節のデータ拡張手法であり、DP は 4.3 節のものである。提出元での majority は、知事提出議案を賛成、議員提出議案を反対で統一とし、政党ごとの majority は統一化が困難であるため、無しとしている。

表 3 評価結果, k=5 交差検証 avg, train:442, train(DA):485

	document only	document + proponent	document + party
Majority	0.7352	0.8167	
BERT	<b>0.8507</b>	0.8509	0.8370
BERT+DA	0.8439	0.8419	<b>0.8688</b>
BERT+DP	0.8484	<b>0.8666</b>	0.8685
BERT+DA+DP	0.8212	0.8637	0.8460

結果から、討論文のみの入力ではデータ拡張等を

施さない状態が最も精度が高く、討論文+提出元では DP 手法、討論文+政党名ではデータ拡張手法が最も高い精度となった。これらの提案手法は全てにおいて、majority ベースラインを上回っているが、付与情報による大幅な精度向上は見込めなかった。

フィールド名である政党名と提出元の付加により、一定の精度向上が見られた要因としては、3.5 節で述べた政党間、提案者によって生じる賛否のバイアスの解消によるものと考えられる。しかしながら、正解ラベルに分類された討論文が、フィールド情報を与えたことで誤分類されることがあるため、精度向上へ寄与する手法としては一長一短であるといえる。また、実験データとしては非常にサンプル数が少なく、BERT での日本語構文解析においてはより大規模なコーパスと、より高い pre-training 時の epoch 数によって、さらなる精度向上が実現することが示唆されていることから [10]、さらなる大規模なデータセットの構築が必要である。

## 5 おわりに

本研究では、東京都議会における議員のスタンス分類タスクにおいて、BERT を用いた議員の発言をスタンス分類するための学習データの作成方法、および所属政党や提出元などのフィールド情報を付加し、モデルの評価を行なった。評価結果から、提案手法は単純な majority ベースラインを上回る性能が発揮され、また政治的なフィールド情報を付与することで一定の精度向上に寄与することが分かった。

今回は東京都議会に限定して学習を行ったが、データ数が非常に少ないため、今後は地方議会等に範囲を広げ、より大規模なデータセットを構築し、提案手法の有効性を検証する。

## 謝辞

本研究は JSPS 科研費 19K11980 の助成を受けた。

## 参考文献

- [1] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Teruko Mitamura, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Tatsunori Mori, Kenji Araki, Satoshi Sekine, and Noriko Kando. Overview of the ntcir-15 qa lab-poliinfo-2 task. *Proceedings of The 15th NTCIR Conference*, 12 2020.
- [2] Eli Pariser. The filter bubble: What the internet is hiding from you. 2011.
- [3] Takanori Nekomoto, Ryoto Ohsugi, Tomoyosi Akiba, Shigeru Masuyama, and Daiki Shirato. akbl at the ntcir-15 qa lab-poliinfo-2 tasks. *The 15th NTCIR Conference*, p. 155, 2020.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [5] Gábor László Hajba. *Website Scraping with Python: Using BeautifulSoup and Scrapy*. Apress, 2018.
- [6] Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Improving the accuracy of japanese parsing with bert. *The Association for Natural Language Processing 25th Annual General Meeting*, pp. 205–208, 2019.
- [7] Daisuke Kawahara. Japanese morphological analysis system juman version 3.61. <http://pine.kuee.kyoto-u.ac.jp/nl-resource/juman.html>, 1999.
- [8] V. Petridis A. Kehagias, P. Fragkou. Linear text segmentation using a dynamic programming algorithm. *In Proceedings of the EACL*, pp. 171—178, 2003.
- [9] 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告. NLP, 自然言語処理研究会報告, Vol. 161, pp. 89–96, 5 2004.
- [10] 柴田知秀, 河原大輔, 黒橋禎夫. Bert による日本語構文解析の精度向上. 言語処理学会 第 25 回年次大会, pp. 205–208, 2019.