地方議会における事実確認のための会議録分割

田上 諒† 木村 輔† 杉本 翔† 勝山 光‡ 宮森 恒†
京都産業大学大学院 先端情報学研究科† 京都産業大学 コンピュータ理工学部‡
{i1788124, i1658047, i1888097, g1544377, miya}@cc.kyoto-su.ac.jp

1 はじめに

新聞記事や書籍，ブログなどでは，ある人物の発言の一部分のみを抜粋して，引用するという機会がよくある．ここで問題としてあげられるのは，発言の一部が欠落したことにより，発言者の意図とは異なる印象を，読者に与えてしまう可能性があることである．よって，発言者の本来の意図を確認するために，当該発言の一次情報を確認したいという要求が出てくる．そのような確認をしたい場合，一般的には，文書検索による手法が考えられるが，結果として返された一文書に対して，さらに人手で，発言箇所を探さなければならないのは，非効率である．したがって，発言者の意図を効率よく確認するには，さらに該当範囲を切り出して，一次情報として提示する必要性がある．

本稿では，地方議会の会議録をデータセットとして用い，図 1 に示すような，与えられたトピックに対する議員の発言範囲を切り出す手法について述べる．会議録中の，特定トピックに関して発言されている個所を切り出すことによって，発言者の意図を効率的に確認することができるようになるだけでなく，発言内容の要約や，ある議題に対する議員の立場の判断などへの活用が期待される．

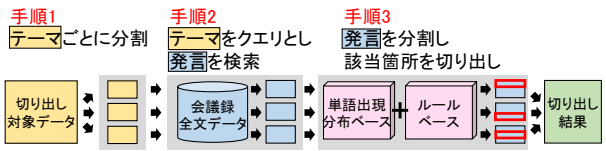

図 1: 提案する切り出し手法の概要

2 関連研究

文の切り出しというタスクについては，議論マイニング [2] と呼ばれる分野でも活用されている．議論マイニングでは，ディベートや新聞記事，ソーシャルネットワーキングサービスなどで行われる様々な議論に対して，ユーザごとの主張やその根拠を切り出したうえで，そのユーザの意見を要約したり，立場などを分類して，各主張の関係性を解析したりする．ここで重要なのは，ユーザの主張だけでなく，その根拠などについても紛れなく，かつ，余分な情報を含めずに，元のテキストから切り出す必要がある点である．QA Lab PoliInfo タスクは，これら議論マイニングとも関連がある．当該タスクは，いくつかのサブタスクから構成されているが，その 1 つとして，膨大な会議録の議員の発言データの中から，対象の箇所のみを切り出すサブタスクが存在する．本稿では，当該タスクで提供されたデータセットを用い，提案手法の精度を評価する．

3 データセットの概要

本稿では，NTCIR-14[1] QA Lab PoliInfo タスク[2]で提供された，地方議会会議録をデータセットとして使用する．データセットは，**会議録全文データ**（3.1 節）と，**切り出し対象データ**（3.2 節）の 2 種類から構成される．

地方議会では，代表質問および一般質問と呼ばれる時間が設けられており，議員は，自治体の職員などに対して，様々な質問ができる．地方議会の特徴は，議員が複数の質問を持っている場合，質問ごとに職員が都度回答するのではなく，様々なテーマの質問を 1 回の登壇ですべて発言した後に，職員も同様に，各質問に対して，順番に回答する点である．なお，本稿では，議員などが一度の登壇で話した全発話内容を 1 発言と定義する．

3.1 会議録全文データ

会議録全文データには，平成 23 年～27 年の東京都議会本会議 73 会議分の発言が，すべて 1 つのリストに収まっており，発言内の 1 文につき，1 オブジェクトとして格納されている．1 オブジェクト内の項目は，会議をまたがって一意に振られた**行番号**，会議の**実施日**，**発言者**，発言内の 1 文を示す**本文**である[3]．表 1 は，オブジェクトのデータ例である．

表 1: 会議録全文データの例

項目	内容
行番号	22231
実施日	平成 24 年 2 月 28 日
発言者	宮崎章
本文	風邪を引いておりまして，聞きにくいところはお許しをいただきたいと思っております．
行番号	22232
実施日	平成 24 年 2 月 28 日
発言者	宮崎章
本文	平成二十四年第一回東京都議会定例会に当たり，東京都議会自由民主党を代表して質問をいたします．

3.2 切り出し対象データ

切り出し対象データは，**会議録全文データ**の中から，切り出してほしい対象について記述されたデータであり，1 対象につき，1 オブジェクトに対応させる．1 オブジェクト内の項目は，会議の**実施日**，発言の**メイントピック**，発言の**サブトピック**，本トピックについて

[1]NTCIR-14: http://research.nii.ac.jp/ntcir/ntcir-14/
[2]QA Lab PoliInfo: https://poliinfo.github.io
[3]説明の簡略化のため，実際のデータセットに含まれる一部項目は省略している．

質問した議員（**質問者**），および質問発言の概要（**質問概要**），本トピックについて回答した職員（**回答者**），および回答発言の概要（**回答概要**）である[3]．1 オブジェクトに格納されているデータの例は，表 2 に示すとおりである．**会議録全文データ**とは異なり，質問者や回答者の名前は，正式な氏名ではなく，質問者名では，カッコ書きで所属名も含まれていたり，回答者名では，役職名のみとなっている．また，質問の概要および回答の概要には，複数のテーマが記述されている場合がある．

表 2: 切り出し対象データの例

項目	内容
実施日	平成 24 年 2 月 28 日
メイントピック	日本の未来のため東京が先頭に 帰宅困難者対策をどう具体化か
サブトピック	スポーツ
質問者	宮崎章（自民党）
質問概要	〔1〕シニアスポーツ振興に力入れよ．〔2〕スポーツ振興基本計画改定し新推進指針策定を．〔3〕スポーツ祭東京 2013 成功に向け所見は．（後略）
回答者	知事
回答概要	〔3〕区市町村と一丸となり成功させ東京オリンピック・パラリンピックへと繋げたい．

実際に，発言の切り出し処理を行う際は，1 オブジェクトごとに，その概要などに合致する発言の一部分を，**会議録全文データ**より切り出すが，1 文（**会議録全文データ**の 1 オブジェクト）のみを切り出すのではなく，合致する範囲の文をすべて切り出すことが想定されている．また，先に述べた地方議会の構成上，1 つのトピックには，それについて質問をする発言と，回答する発言が存在するため，1 オブジェクトにつき，それぞれの発言を抽出することとなる．

4 提案手法

4.1 情報源の前処理

切り出し処理を行う前に，**会議録全文データ**を，1 オブジェクトを 1 文書として検索エンジンへ登録する．その際，以下に述べるいくつかの前処理を施してから登録する．

まず，**会議録全文データ**内の各オブジェクトを，1 発言ごとにまとめる処理を行う．3.1 節でも述べたが，データセットの 1 オブジェクトは，1 発言内の 1 文単位であり，発言単位の明示的な区切りは存在しない．後の処理において，1 発言を取り扱う処理があるため，この時点で区切りを明確化する．方法としては，データセット内を順に走査し，同じ発言のオブジェクトである限り，同じセクション番号を当該オブジェクトに割り振る．走査中,「発言者が変わった」か「本文に罫線文字列が存在した」場合は，その都度，新しいセクション番号を振ることで，発言の区切りを明確化する．

次に，明確化された各発言ごとに，**発言タイプ**を推定する．我々は，全国の地方議会を参考としたうえで，地方議会における 7 つの発言タイプを定めた．この中には，議員が代表質問や一般質問などを行う際の発言（タイプ「質問」）や，職員が議員の質問に対して答弁する際の発言（タイプ「回答」）が含まれる．

発言タイプの分類には，Joulin ら [1] が提案した，テキスト分類手法を用いる．この手法を用いるには，あらかじめ，各タイプの発言例を一定量学習させる必要がある．我々は，事前に Web 上から収集した，国内 4 自治体の全 110 会議について，人手によるラベリングを行い，発言タイプ推定用のモデルを構築した．素性として，1 発言の冒頭 2 文および末端 2 文を用いた．ラベリングデータの 9 割を学習用，1 割を評価用として学習させたところ，評価用データに対する推定精度は約 99.3%であった．

以上の前処理により，**会議録全文データ**の各オブジェクトには,「同じ発言内のオブジェクト同士が同じ番号となるセクション番号」と「推定された発言タイプ」の 2 種類の項目が新たに付与される．本手法では，この状態のオブジェクトを，1 文書として検索エンジンに登録する．

4.2 情報源からの文の切り出し処理

図 1 に示す手順で処理を進める．

4.2.1 手順 1：複数テーマへの分割

まず初めに，**切り出し対象データ**と関連する文書を，情報源より検索する．本章で使用する**切り出し対象データ**においては，3.2 節で示した「質問概要」および「回答概要」が該当する．しかし，3.2 節でも述べたように，本データセットでは，1 つの概要内に，複数のテーマが含まれている場合がある．よって，あらかじめ概要を分割し，テーマごとに手順 2 以降を実施して，最終的に結果を統合することとする．データセットでは，概要内に複数のテーマが存在する場合，見出し番号で明示的に区別できるように整形されているため，ルールベースによって複数テーマへ分割する．

4.2.2 手順 2：文書検索

手順 1 で得られた各テーマごとに，4.1 節で前処理を行った情報源に対して，検索を行う．検索エンジンには，議員の 1 発言内の 1 文ごとに，1 文書として登録されているため，検索で得られる関連文書は，任意の発言内の 1 文である．検索に必要なクエリは，テーマの文字列と，当該**切り出し対象データ**のサブトピックの文字列を連結させたものとする．また，検索の際には，以下に 2 種類の条件によって，検索結果をフィルタリングする．

- **切り出し対象データ**内の実施日と，発言日が一致する文書のみを検索対象とする
- 発言者フィルタ（後述）で対象となった文書のみを検索対象とする

ここで，発言者フィルタについて述べる．先に述べたクエリを使用して，検索を行った場合，表層的に類似した，別の議員の発言がヒットする可能性がある．**切り出し対象データ**では，質問側については，発言者が氏名の形で記述されている．よって，質問発言を検索する際には，この情報をもとに，当該議員が発言したもののみを検索対象とするフィルタをかける．対となる回答側については，**切り出し対象データ**に氏名の記述はないが，3 章で説明した会議録の構成から，先の質問側の検索でヒットした発言の直後の回答発言内に，切り出したい部分があることは明らかであるため，ヒットした質問発言の直後の，発言タイプが「回答」である文書のみを検索対象とする．1 つの質問に対し

て，複数人が回答している場合は，それらすべてを対象とする．

ただし，**会議録全文データ**と**切り出し対象データ**との間で，氏名表記の揺らぎにより，発言者フィルタによって，1 件も文書がヒットしない場合は，発言者フィルタを使用しない．そのような理由でフィルタが使用できない場合や，最初から意図的にフィルタを使用しない場合，代替として，質問発言を検索する際には，発言タイプが「質問」である文書を，回答発言を検索する際には，タイプが「回答」である文書を，検索対象とする．

以上により，質問側および回答側で，最も関連する文書を 1 件（1 文）のみ取得する．そして，セクション番号の情報をもとに，検索結果から 1 発言を取得する．ただし，質問側および回答側それぞれの発言内で，どの文が検索時にヒットした文かどうかの情報は保持しておく．

4.2.3 手順 3：発言の分割

手順 2 で得られた 1 つの発言から，**手順 1** で得られた指定のテーマについて発言している箇所を切り出す．3 章で述べたように，議員は，一度の発言において，複数のテーマについて質問または回答を行うため，なんらかの分割処理によって，1 発言を単一のテーマごとに分割し，指定テーマと合致する範囲を，切り出し結果とする．分割処理は 2 段階構成とし，各段階で切り出し結果を出力する．

1 段階目には，発言を分割する処理として，Utiyama ら [3] が提案した，単語分布に基づくテキスト分割手法を用いる．この手法は，テキスト内の単語の出現分布のみを指標として，分割確率が最大となるような分割を選択する手法であり，汎用的なテキストに対して用いることができる．この手法により，理想的には，**手順 2** で得られた 1 発言が，任意個のセグメントに分割され，各セグメントは単一のテーマの質問または回答となる．よって，手順 2 で保持した「検索時にヒットした文」を含むセグメントを，指定テーマの切り出し結果とする．

2 段階目では，1 段階目の切り出し結果に対して，さらにルールベースによる分割手法を適用する．これは，1 段階目の切り出し結果が，理想の切り出し箇所よりも，広く切り出されてしまった場合への対処である．地方議会では，1 発言内で，異なるテーマの質問や回答に話題を転換する際，「次に」や「について伺います.」などの言い回しが多用される．よって，テーマ転換の際によく用いられる言い回しが出現した際は，その箇所を区切り目として分割するルールを設ける．実際には，1 段階目の切り出し結果内の，検索時にヒットした文を起点として，前後方向にそれぞれ走査し，ルールに合致した文が出現した場合は，その文までを切り出し範囲とする．

以上の処理によって，**手順 1** で分割された 1 テーマに対する，切り出し結果が確定する．ここまでの処理をテーマごとに行い，最終的に，1 オブジェクト内で結果を統合する．3.2 節でも述べたように，1 オブジェクトにつき，質問側と回答側の 2 つの切り出し結果が出力される．

5 実験

本実験では，提案手法を用いることで，どの程度の精度で，ユーザなどが要求するデータを，一次情報から切り出すことができるかを調査する．同時に，提案手法のシステム内で使用する機構の有無が，精度にどの程度影響するかについて，比較実験する．

5.1 方法

実験時には，システム内で使用する機構の有無によって，精度がどの程度変化するか比較する．具体的には，表 3 に示すとおり，4.2.1 項で述べた発言者フィルタ（X）の有無，4.2.3 項で述べた単語の出現頻度に基づく分割（Y）の有無，および，ルールベースによる分割（Z）の有無の組み合わせによる，8 通りの条件によって比較する．

表 3: 各実験条件で有効とした提案手法のシステム内の機構

	機構		
条件番号	X	Y	Z
C1	—	—	—
C2	✓	—	—
C3	—	✓	—
C4	—	—	✓
C5	✓	✓	—
C6	—	✓	✓
C7	✓	—	✓
C8	✓	✓	✓

文書検索時の検索エンジンには，オープンソースの全文検索システムである Apache Solr[4] を使用する．単語の出現頻度に基づく分割を行う際の，形態素解析には，MeCab[5] を使用する．

切り出し対象データは，298 件の**訓練用データ**と，83 件の**評価用データ**に分けられる．本実験では，4.2.3 項で述べた，ルールベースによる分割処理で使用するルールについて，**訓練用データ**でよく用いられている表現を人手で判断し，ルールとして追加する．なお，情報源となる**会議録全文データ**は，**訓練用データ**と**評価用データ**で共通である．

精度の指標には，適合率，再現率，F 値を用いる．**切り出し対象データ**には，あらかじめ，人手による理想の切り出し範囲（正解範囲）が明示されているため，システムの切り出し結果と比較することで，各指標の値を得ることができる．適合率（$Precision$）は，システムが切り出した範囲のうち，どの程度が正解範囲と一致するかの割合である．再現率（$Recall$）は，正解範囲のうち，どの程度がシステムの切り出し範囲と一致するかの割合である．F 値（$F\text{-}measure$）は，適合率と再現率の調和平均であり，式 1 で求まる．

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

5.2 結果

各条件による，**評価用データ**に対する提案手法のシステムの切り出し精度は，表 4 のとおりとなった．全 83 件のオブジェクトの切り出し結果ごとに精度を算出することができるが，表 4 はそれらの平均値を示している．また，各評価指標について，**すべて**（質問および回答の両方），**質問**，**回答**のそれぞれについて算出している．

[4]Apache Solr: http://lucene.apache.org/solr/
[5]MeCab: http://taku910.github.io/mecab/

表 4: 各実験条件における評価用データに対する切り出し精度（太字は，各列の最大値を示す）

条件	すべて			質問			回答		
番号	適合率	再現率	F 値	適合率	再現率	F 値	適合率	再現率	F 値
C1	0.087	0.940	0.160	0.065	0.920	0.122	0.275	**0.982**	0.430
C2	0.112	**0.991**	0.202	0.088	**1.000**	0.162	0.278	0.973	0.433
C3	0.243	0.779	0.370	0.191	0.834	0.311	0.841	0.661	0.740
C4	0.294	0.906	0.444	0.221	0.879	0.353	0.797	0.964	0.873
C5	0.660	0.819	0.731	0.612	0.898	0.728	0.857	0.654	0.742
C6	0.267	0.759	0.395	0.208	0.806	0.331	0.949	0.660	0.778
C7	0.857	0.952	**0.902**	**0.881**	0.953	**0.916**	0.812	0.950	**0.875**
C8	**0.922**	0.796	0.854	0.905	0.865	0.885	**0.973**	0.651	0.780

6 考察

6.1 発言者フィルタ処理の効果

C1 や C2 は，分割処理を一切行っていないため，切り出し結果は，**切り出し対象データ**が要求する範囲を含む，議員の 1 発言となる．1 発言には，要求する範囲が必ず含まれるべきであり，C1，C2 の再現率は限りなく 1.0 に近くなることが理想である．C1 の**すべて**の再現率は，0.940 であり，これは，一部の**切り出し対象データ**において，要求する範囲とまったく異なる発言を，検索時に取得してしまっているといえる．C1 に発言者フィルタを取り入れた C2 について見ると，**すべて**の再現率は 0.991 と向上していることが分かる．よって，このフィルタによって，より適切に発言を取得できていると言える．これは，類似した表層文字列の文を含む，別の議員の発言が取得されることを抑制できたためと考える．

6.2 2 種類の分割処理の効果

分割処理により，1 発言から不必要な文排除し，要求範囲を切り出すため，適合率の向上が見込まれるが，必要な箇所まで排除してしまった場合，再現率の低下につながるため，適切な箇所で分割しなければならない．C1 に対して，単語の出現頻度に基づく分割を取り入れた C3，ルールベースによる分割を取り入れた C4 は，どちらとも F 値が向上している．特に C4 は，C1 に対して，再現率をほとんど維持したまま，適合率を向上させている．また，発言者フィルタを有効としている C2，C5，C7 も，同様の傾向となっている．以上より，2 つの分類処理は共に精度向上に貢献し，さらに，ルールベースによる分割のほうが，F 値をより大きく改善させることが分かる．

ルールベースによる分割のほうが，より適切に分割された例として，次のような例が存在した．1 発言中の理想の切り出し部分と，その後に続く部分が，テーマは違うものの，どちらも同じような語彙が使われている発言が存在した．単語の出現頻度に基づく分割では，それらを同一のテーマとして扱ってしまい，適切な分割ができなくなっていた．しかし，ルールベースによる分割では，テーマの区切り目でよく用いられる表現が出現した箇所で分割を行うため，この例では適切に分割されていた．反対に，テーマの区切り目でよく用いられる表現が出現しなかったため，ルールベースによる分割では，適切な分割ができなかった例も存在した．その例では，単語の出現頻度に基づく分割を用いた場合，異なるテーマ間で使用される語彙の違いをきちんと判断し，適切に分割されていた．以上より，2 つの分割処理には，それぞれに良い点と悪い点が存在することが分かった．本章で使用している，地方議会会議録のデータセットでは，ルールベースによる分割のほうがより大きな改善が見られたが，人手で完璧なルールを作成することは困難であり，どのようなデータセットにおいても，適切なルールが作成できるとは言えないため，単語の出現頻度に基づく分割のほうが，より汎用的であると考えられる．

7 まとめ

本稿では，地方議会会議録から，要求されたトピックに対する議員の発言を切り出す手法について，2 種類の分割手法を適用した手法を提案した．実験では，提案手法による切り出しの精度について調べた結果，最高で F 値が 0.9 を上回る結果となった．また，単語出現分布による文書分割手法，および，ルールベースによる文書分割手法のそれぞれにおいて，良い点と悪い点が存在することが分かった．情報源から，要求に対する特定箇所を切り出すというタスクは，様々な分野で応用できると考えられる．本稿の実験では，ルールベースによる文書分割手法が，精度の向上に大きく貢献する結果となったが，今後は，さらに汎用的な手法を検討する必要がある．

謝辞

本研究の一部は科研費 18K11557 の助成を受けたものです．ここに記して感謝の意を表します．

参考文献

[1] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431. Association for Computational Linguistics, 2017.

[2] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, Vol. 16, No. 2, p. 10, 2016.

[3] Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 491–498, 2001.