小樽商科大学 学生論文賞 第2次審査用論文

Budget Argument Miningにおける予算と会議録の連結手法の研究

2018268

永渕 景祐

商学部 社会情報学科

2021年12月20日提出

目 次

1章　はじめに

国の予算は，内閣が毎会計年度の予算を作成し，国会に提出して，その審議を受け議決を経ることで成立する[1]．また地方自治体の予算は，都道府県知事や市町村長が毎会計年度の予算を作成し，議会に提出して，その審議を受け議決を経ることで成立する[2]．成立した予算と国会・議会の会議録は，一般に公開されている．しかし，会議録のテキスト量は膨大であり，知りたい予算についての情報を探すことは困難である．従って，どのような審議を経て予算が成立したのかを把握しづらいという問題点がある．

自然言語で表現された議論の推論構造を自動的に識別・抽出するタスクとして，Argument Mining がある [3]．Argument Mining の目的として，Support(支持) や Attack(反論) などの談話単位間の論述関係や，Claim(主張)，Premise(根拠) などの談話単位の担う機能を予測する論述構造解析がある [11]．これらの研究では，解析対象を小論文などの独話的な論述文，つまり単一文書としており，複数文書にまたがる解析はなされていない．

そこで Argument Mining の考えを基に，国や地方自治体が公開する予算文書と会議録の議論を結びつけることを目的とした **Budget Argument Mining** というタスクが設計された [14]．Budget Argument Mining では，会議録中に表れる**金額表現に着目**して，以下の 2 つのタスクを行う．

1. 会議録中の金額表現の機能を示す**議論ラベルの付与**
2. 会議録中の金額表現と**関連する予算項目の連結**

議論ラベルの付与は，ラベルが多値の**文書分類**タスク，関連する予算項目の連結は，予算項目の文と会議録の文の類似度を計算する**類似文章検索**タスクと見なすことができる．

これらのタスクを解く際，文書を数値表現に変換したものである**文書ベクトル**を用意する必要がある．この文書ベクトルには，文脈に依存するものと文脈に依存しないものが存在する．

従って本研究では，文脈に依存しない文書ベクトルと，文脈に依存する文書ベクトルを用いて，議論ラベルの付与タスクと関連する予算項目の連結タスクでそれぞれ比較実験を行い，Budget Argument Mining おいて文脈を考慮すべきかどうかを明らかにすることを目的とする．

以下，2 章では関連研究を紹介し，3 章では，Budget Argument Mining について述べる．4 章，5 章では，本研究で行った実験について述べ，6 章で考察を述べる．最後に，7 章にて結論を述べる．

[1] https://elaws.e-gov.go.jp/document?lawid=321CONSTITUTION
[2] https://elaws.e-gov.go.jp/document?lawid=322AC0000000067

2章　関連研究

2.1　文書のベクトル化

機械学習の手法を用いて文書分類や類似文書検索を行うには，文書をベクトルに変換する必要がある．ここでは，文書ベクトルを得るために必要な前処理である単語分割と，文書ベクトルを得る代表的な手法を紹介する．

2.1.1　単語分割

単語や文書をベクトルに変換するためには，**単語分割**が必要となる [9]．日本語では，MeCab[1]や Sudachi[8] などの形態素解析ツールを用いて単語分割を行うことが一般的である．形態素解析は，辞書と呼ばれる単語に品詞情報が付与されたコーパスを利用して行われる．辞書には，短単位の UniDic[2]や固有表現に強い NEologd[10] など，様々な単語長のものが存在する．どの長さの単位で単語分割を行うかが，その後のタスクの結果に大きな影響を及ぼすため，どの辞書を利用するかを検討する必要がある．本研究では，単語分割に Sudachi[8] を用いている．Sudachi は，UniDic と NEologd をベースに調整した辞書を用いた形態素解析ツールであり，短単位，中単位，固有表現相当の長単位の 3 つの分割モードを利用することができる．

2.1.2　文書ベクトル

ここでは，単語分割がなされた文書から，文書ベクトルを得る手法を紹介する．

単純な文書ベクトルへの変換手法として，単語の出現頻度を用いた Bag of Words，そしてその重みづけの手法として TF-IDF が挙げられる．**Bag of Words** は，文書の集合全体から，語彙を作成し．各文書での各単語の出現回数を含んだ特徴量ベクトルへと変換する [7]．例えば，次のような単語分割された文が与えられたとする．

1. 私 は 猫 が 好き です
2. あなた は 犬 が 好き です
3. 私 は 犬 も 好き です が 猫 の 方 が 好き です

[1] https://taku910.github.io/mecab/
[2] https://ccd.ninjal.ac.jp/unidic/

これらの文に含まれる単語から，語彙を学習する．この語彙を利用して，各文を各単語の出現回数を特徴量としたベクトルに変換できる．図 2.1 に，Bag of Words で文書ベクトルを作成している様子を示す．

	あなた	が	です	の	は	も	好き	方	犬	猫	私
文1	0	1	1	0	1	0	1	0	0	1	1
文2	1	1	1	0	1	0	1	0	1	0	0
文3	0	2	2	1	1	1	2	1	1	1	1

図 2.1: Bag of Words の例

TF-IDF は，単語の出現頻度である TF(Term Frequency) と逆文書頻度である IDF(Inverse Document Frequency) の積である [7].

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t,d)$$

$tf(t,d)$ は文書 d における単語 t の出現頻度である．IDF は，次のように求められる．

$$idf(t,d) = \log \frac{n_d}{1 + df(d,t)}$$

n_d は文書の総数， $df(d,t)$ は単語 t を含んでいる文書 d の個数を表す．TF-IDF を用いることで，出現頻度が高い，つまり重要ではない単語の重みを減らすことができる．図 2.2 に，TF-IDF で Bag of Words と同じ語彙から文書ベクトルを作成している様子を示す．文 2 の「犬」や文 1 の「猫」を見ると，特徴量が重みづけされ，他と比べて大きな値となっていることがわかる．

	あなた	が	です	の	は	も	好き	方	犬	猫	私
文1	0	0.37	0.37	0	0.37	0	0.37	0	0	0.48	0.48
文2	0.58	0.34	0.34	0	0.34	0	0.34	0	0.44	0	0
文3	0	0.39	0.39	0.33	0.19	0.33	0.39	0.33	0.25	0.25	0.25

図 2.2: TF-IDF の例

これらの手法のメリットは，簡単に文書ベクトルを得ることができることである．しかし，単語の種類に応じて特徴量が多くなり，次元数は大きなものとなる．また，語順などを考慮しないことから，文脈から得られる情報は失われてしまう．

単語の出現頻度を用いる方法の他には，文書中に出現する単語を，言語モデルを用いて単語ベクトルに変換し，その平均を取ることで文書ベクトルを得るという手法がある．言

語モデルとは，文章の出現しやすさを確率によってモデル化したものである [9]．言語モデルには，Word2Vec[4] や fastText[1] のような文脈依存しないものと，ELMo[5] や BERT[2] のような文脈依存するものがある．

2.2 文書分類

本研究における議論ラベルの付与は，金額表現を含む文章を 7 つのカテゴリに分類するタスクだと見なすことができる．文書分類とは，文書を与えられたカテゴリに分類するタスクである [9]．具体的には，文書を数値表現，つまりベクトルに変換し，その特徴量を基に分類問題を解くということである．分類問題を解くためのアルゴリズムとしては，ロジスティック回帰や SVM，決定木やランダムフォレストなどが挙げられる．また近年では，BERT[2] のような事前学習された強力な言語モデルを，文書分類タスクに合わせてファインチューニングするという手法が取られている．BERT による文書分類は．4.2.2 にて説明を行う．

2.3 類似文書検索

類似文書検索とは，ある文書が与えられたときに，その文書に内容が近い文書をデータベースの中から選択するというタスクである [9]．文書をベクトルに変換することで，2 文書間のコサイン類似度を求めることができる．コサイン類似度は，正規化されたベクトル同士の内積であり，この値はベクトル間のなす角度が小さくなるほど大きくなる [9]．コサイン類似度は，次のように求められる．

$$sim(x, y) = \frac{x \cdot y}{|x||y|}$$

Bag of Words などの文脈を考慮しない文書ベクトルは，多義語に対処できないなどの問題が存在する．そこで，BERT などの文脈を考慮する文書ベクトルを作成できる言語モデルの利用が考えられる．BERT はファインチューニングを行うことで，様々な自然言語処理タスクに適用できる．しかし，類似文書検索のようなタスクでは，精度は高いものの，BERT の構造の理由で非常に処理に時間がかかる．そこで，精度を落とさずに処理時間を高速化できるように，文書ベクトルの計算に BERT を特化させたモデルが，Sentence-BERT である [6]．2 つの文書をペアにして Sentence-BERT に入力することで，コサイン類似度などをラベルとしてファインチューニングが可能である．

3章　Budget Argument Mining

3.1　タスク概要

Budget Argument Mining は，NTCIR-16 QA Lab-PoliInfo-3[1]のサブタスクである．Budget Argument Mining では，国や地方自治体が公開する予算文書と会議録の議論を結びつけることを目的としており，会議録中に表れる**金額表現に着目**して，以下の 2 つのタスクを行う [14].

1. 会議録中の金額表現の機能を示す**議論ラベルの付与**
2. 会議録中の金額表現と**関連する予算項目の連結**

図 3.1 にタスクの大まかな流れを示す．タスクの入出力は，予算データと会議録データの JSON を入力し，金額表現を含むに対して議論ラベルの付与，関連する予算項目の連結の処理を行い，その結果の会議録データの JSON を付与するというものである．

図 3.1: タスクの大まかな流れ

本タスクで分析対象となっている予算・会議録は，国会，小樽市，茨城県，福岡市の 4 つである．Budget Argument Mining のデータセットの詳細については，3.4 で述べる．

3.2　議論ラベルの付与

議論ラベルとは，会議録に含まれる議員の発言に対して，議論の流れを理解するために必要となるラベルである [14]．議論ラベルは，以下の 7 種類である．

- Premise : 過去・決定事項
- Premise : 未来（現在以降）・見積

[1]https://poliinfo3.net/

- Premise : その他（例示・訂正事項など）
- Claim : 意見・提案・質問
- Claim : その他
- 金額表現ではない
- その他

議論ラベルのアノテーションは,「読点」による区切りを目安として,「節」以上「文」以下の範囲で行われている [14]．図 3.2 に，議論ラベルの付与の具体例を示す．ここでは，国会での発言を例に挙げている．この例では,「計上したところであります」という，今年度予算に関する金額表現を含む根拠が記述されていることから，議論ラベルを「Premise: 未来（現在以降）・見積」としている．

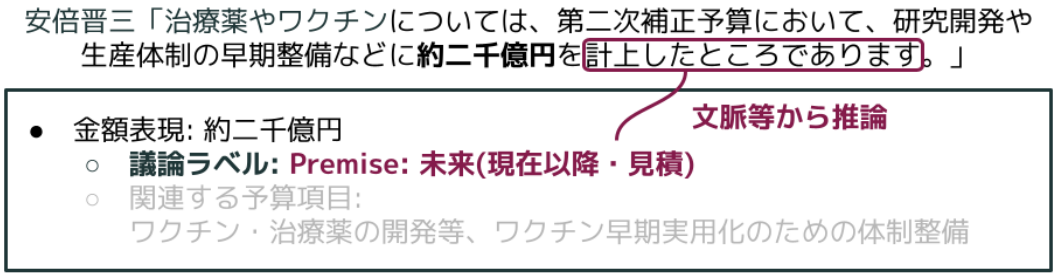

図 3.2: 議論ラベルの付与の具体例

3.3 関連する予算項目の連結

会議録中の予算に関する発言と，その内容に関連した予算項目を対応づける．本タスクでは，予算に関する発言の候補として，**金額表現を含む発言**を対象としている．関連する予算項目の連結のアノテーションは，金額表現を含む発言中の金額や文脈を，予算項目の予算額や説明と照応することで行われている．図 3.3 に，関連する予算項目を連結する具体例を示す．図 3.2 と同じく，国会での発言・予算項目を例に挙げている．この例では，発言中の「治療薬やワクチン」「研究開発」「生産の早期整備」などの予算項目名と関連がある表現や，予算項目の「600 億円」「1,455 億円」の合計額が発言中の「約二億円」と一致するといった情報から,「ワクチン・治療薬の開発など」「ワクチン早期実用化のための体制整備」の 2 つの予算項目を，発言に対応づけている．

3.4 データセット

Budget Argument Mining の参加者には，予算のデータ (budget.json)，会議録の訓練用データ (minutes-training.json)，テスト用データ (minutes-test.json) が配布される．予算のデータには，分析対象の予算のデータが，図 3.1 の形式で保存されている．会議録データには，分

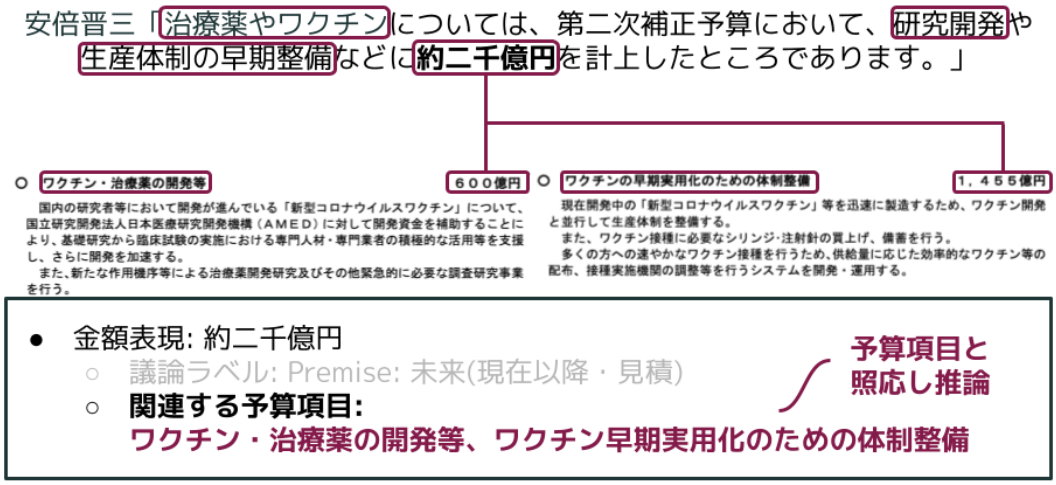

図 3.3: 関連する予算項目の連結の具体例

析対象の会議録のデータが，国会は図 3.3，地方議会は図 3.2 の形式で保存されている．日本語の自然言語処理フレームワークである GiNZA[12] の version4 で固有表現抽出を行うことで，議員の発言である speech・utterance 内の金額表現を抽出している．会議録の訓練用データには，正解の議論ラベルの付与と関連する予算項目の連結がされている．データに含まれる金額表現と予算項目のまた，会議録のテスト用データには，議論ラベルと関連する予算項目がマスクされている．本研究では，3 つのデータのうち，予算のデータと会議録の訓練用データの 2 つを実験に利用する．予算のデータ内の予算項目数と，会議録の訓練用データ内の金額表現数を，表 3.4 に示す．

表 3.1: 予算のデータ形式

フィールド名	説明
budgetId	予算 ID
budgetTitle	予算タイトル
typesOfAccount	会計種別
department	管轄省庁・部局名
url	元データの URL
budgetItem	予算項目名
categories	上位階層
budget	今年度予算額
budgetLastYear	前年度予算額
description	説明
budgetDifference	比較増減額

表 3.2: 地方議会会議録のデータ形式

フィールド名	説明
date	日付
localGovernmentCode	自治体コード（6 桁）
localGovernmentName	自治体名
proceedingTitle	議会名
url	URL
proceeding	発言者と発言内容
└ speakerPosition	発言者の役職
└ speaker	発言者
└ utterance	発言内容
└ moneyExpressions	発言に含まれる金額表現
└ moneyExpression	金額表現
└ relatedID	関連する予算 ID リスト
└ argumentClass	議論ラベル

表 3.3: 国会会議録のデータ形式

フィールド名	説明
issueID	会議録 ID
imageKind	イメージ種別
searchObject	検索対象箇所
session	国会回次
nameOfHouse	院名
nameOfMeeting	会議名
issue	号数
date	開催日付
closing	閉会中フラグ
speechRecord	
└ speechID	発言 ID
└ speechOrder	発言番号
└ speaker	発言者
└ speakerYomi	発言者読み
└ speakerGroup	発言者所属会派
└ speakerPosition	発言者肩書き
└ speakerRole	発言者役割
└ speech	発言
└ startPage	発言が掲載されている開始ページ
└ createTime	レコード登録日時
└ updateTime	レコード更新日時
└ speechURL	発言 URL
└ moneyExpressions	発言に含まれる金額表現
└ moneyExpression	金額表現
└ relatedID	関連する予算項目 ID リスト
└ argumentClass	議論ラベル
meetingURL	会議録テキスト表示画面の URL
pdfURL	発言 URL

会議録データ内の金額表現と
表 3.4: 予算データ内の予算項目の個数

対象	令和元年	令和 2 年
国会	なし	金額表現: 165 個 予算項目: 36 個
小樽市	金額表現: 144 個 予算項目: 113 個	金額表現: 85 個 予算項目: 116 個
茨城県	金額表現: 147 個 予算項目: 76 個	金額表現: 129 個 予算項目: 103 個
福岡市	金額表現: 294 個 予算項目: 156 個	金額表現: 284 個 予算項目: 168 個

4章　実験1: 議論ラベルの付与

4.1　実験の目的

Budget Argument Mining における議論ラベルの付与は，金額表現を含む発言を文書と見なすと，文書を議論ラベルに合わせて分類する，ラベルが 7 値の文書分類タスクと見なすことができる．文書分類の手法としては，文書ベクトルを作成し，その特徴量を基に分類器を訓練するという手法がよく用いられている．しかし，文書ベクトルを作成する手法には，Bag of Words や Word2vec[4] のように文脈に依存しないものと，BERT[2] のように文脈に依存するものがある．そのため本実験では，文脈に依存しない文書ベクトルと，文脈に依存する文書ベクトルを用いて，文書分類器を作成しスコアを算出することで，議論ラベルの付与では文脈を考慮すべきかどうかを明らかにすることを目的とする．

4.2　実験方法

以下に，実験の手順を示す．

1. Budget Argument Mining の会議録の訓練用データから，金額表現を含む発言 1 文と議論ラベルのペアを抽出する．
2. 発言文が重複している文を削除する．
3. 訓練用データを 60%，検証用データ，テスト用データを各 20%ずつの割合で，議論ラベルを基に層化分割する．
4. それぞれの手法に合わせて訓練，検証，評価を行う．

3.2 で述べたとおり，議論ラベルのアノテーションは「読点」による区切りを目安として，「節」以上「文」以下の範囲で行われている [14]．このことを考慮して本実験では，文書ベクトルに変換する対象文を，金額表現を含む発言 1 文としている．

また，金額表現を含む発言 1 文中に，複数の金額表現が列挙されることがあるが，金額表現に付与された議論ラベルがそれぞれ異なっている場合がある．この場合，特徴量が同じなのに正解のラベルが違うということになる．これは，分類器の訓練の際に影響を及ぼすこととなる．このことを考慮して本実験では，発言文が重複している文を削除している．

本実験では，文脈を考慮しない手法である Bag of Words，TF-IDF，GiNZA を用いて文書ベクトルを取得し，ロジスティック回帰，LiniearSVM，SVM のアルゴリズムを用いて分類器を構築する手法と，文脈を考慮する手法である BERT[2] を用いて分類器を構築する手法で比較を行なった．評価指標には，Accuracy(正解率) を用いている．

$$Accuracy = \frac{\text{出力した議論ラベルの正解数}}{\text{議論ラベルの総数}}$$

以下に，それぞれの手法での詳細な手順を示す．

4.2.1 アルゴリズムを用いて分類器を構築する手法

まず，Bag of Words，TF-IDF を用いた手法について述べる．Bag of Words，TF-IDF における文書ベクトルを得るまでの流れを図 4.1 に示す．まず訓練・検証・テストデータの発言文を，Sudachi[8] で形態素解析し，単語分割する．分割モードは中単位に統一している．次に単語分割した発言文を，scikit-learn[1]の CountVectorizer または TfidfVectorizer を用いて，それぞれ Bag of Words モデルまたは TF-IDF モデルを作成し，文書ベクトルへと変換する．語彙は，訓練データからモデルを作成する際に，同時に学習している．

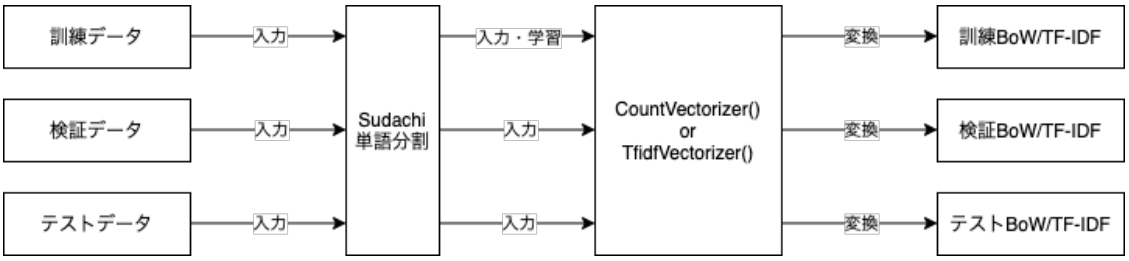

図 4.1: Bag of Words，TF-IDF のモデル作成の流れ

次に，GiNZA[12] の version5 を用いた手法について述べる．GiNZA では，単語分散表現に ChiVe[13] を利用している．Chive は，Word2Vec[4] の一種であり，大規模コーパスと Sudachi による複数粒度分割に基づく，単語ベクトルを得られる言語モデルである[2]．GiNZA では，ChiVe で得られる単語ベクトルを平均して，文書ベクトルを計算する機能を備えている．本実験ではこれを利用して，訓練・検証・テストデータの発言文を，文書ベクトルへと変換している．

最後に，分類器の作成について述べる．分類器は，scikit-learn の LogisticRegression，LinearSVC，SVC を用いて作成している．パラメータを変化させて，訓練データで分類器を学習，検証データでスコアの確認を繰り返し，最も高いスコアを出したパラメータで，学習を行なっている．学習させた分類器で，テストデータに対して議論ラベルの推論を行い，スコアを算出している．

[1]https://scikit-learn.org/stable/index.html
[2]https://github.com/WorksApplications/chiVe

4.2.2 BERT のファインチューニングによる分類器を作成する手法

BERT[2] は，東北大学が公開している事前学習済みモデル (cl-tohoku/bert-base-japanese-whole-word-masking)[3]を，本実験のタスクに特化するように学習を行うファインチューニングを行うことで，分類器を作成している．まず，訓練・検証・テストデータの発言文・ラベルから，データローダを作成する．揃える系列長は，東北大の事前学習済みモデルの BertTokenizer で訓練データの発言文を分割した際，最大単語数が 324 だったことから，336 に設定している．+2 しているのは，BERT の特殊トークンである [CLS]・[SEP] を考慮しているためである．バッチサイズは，訓練時は 32，検証・テスト時は 256 に設定している．次に，作成した訓練・検証データローダを用いて，文書分類に向けた BERT のファインチューニングを行う．ファインチューニングは，Transformers[4]の BertSequenceClassification を用いて行なっている．図 4.2 に，BertSequenceClassification の入出力関係を示す [9]．入力は，BertTokenizer でトークン化された文である．出力は，各ラベルに対してスコアが付いているテンソルとなっている．ここで最もスコアの高いものを推論するラベルとして選ぶこととなる．

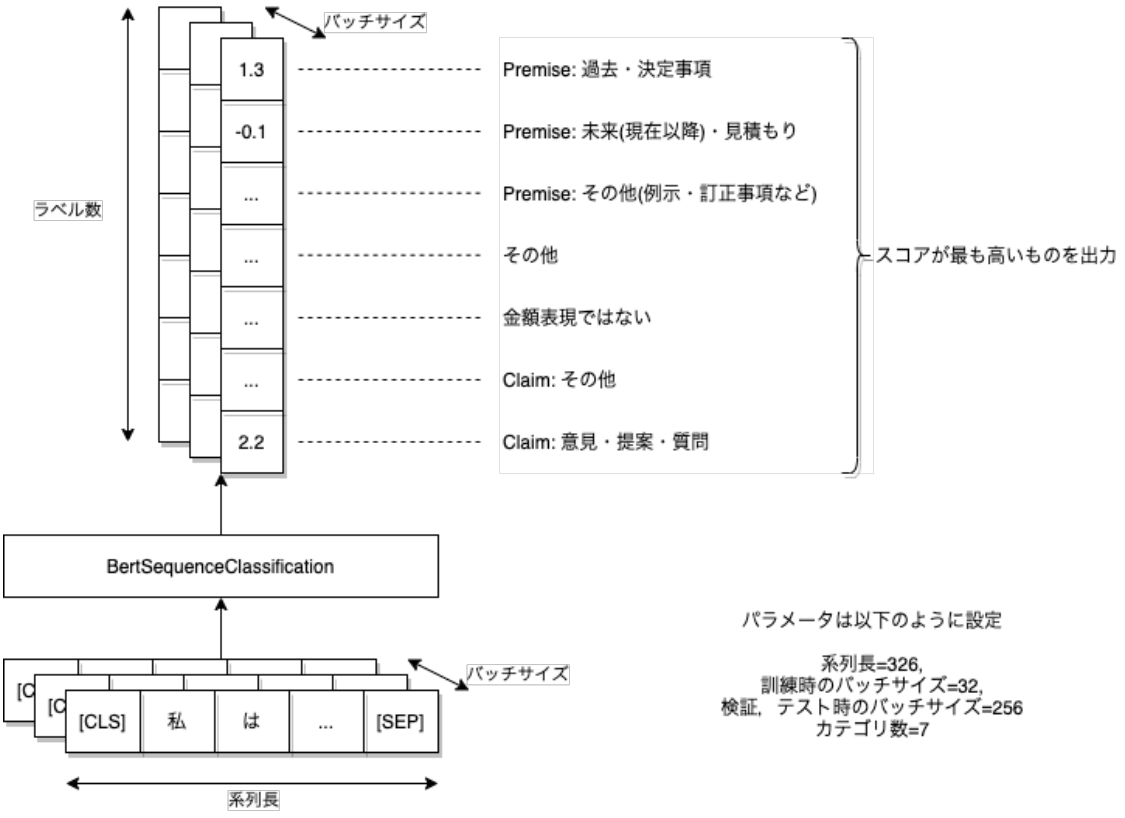

図 4.2: 本実験における BertSequenceClassification の入出力関係

[3]https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking
[4]https://huggingface.co/docs/transformers/index

ファインチューニングはBERTによる自然言語処理入門[9]の第6章を参考に，epoch数10，学習率1e-5で行なった．損失関数は，BERTの論文[2]に合わせてAdamを採用している．ファインチューニングが終わった後，10epoch中で検証時の損失が最も小さいベストモデルを用いて，テストデータに対し推論を行い，スコアを算出している．

4.3 実験結果

実験結果を表4.1に示す．結果は，BERTを用いて作成した分類器が，他の手法と比較して0.013-0.059高いスコアを示した．従って議論ラベルの付与には，文脈を考慮する方が良いことが示された．BERTに次いで高いスコアを示したのは，TF-IDFの文書ベクトルを用いたSVCの分類器であった．また，ロジスティック回帰とLinearSVCの線形分類器では，GiNZAのものがBag of WordsとTF-IDFのものと比較して，0.026-0.040高いスコアを示した．

表4.1: 議論ラベル付与のスコア

手法	Accuracy	パラメータ
Bag of Words & LogisticRegression	0.608	C=0.1
Bag of Words & LinearSVC	0.601	C=0.01
Bag of Words & SVC	0.627	C=10, gamma=0.01, kernel=rbf
TF-IDF & LogisticRegression	0.601	C=10
TF-IDF & LinearSVC	0.601	C=10
TF-IDF & SVC	0.647	C=10, gamma=0.1, kernel=rbf
GiNZA & LogisticRegression	0.641	C=1000
GiNZA & LinearSVC	0.634	C=10
GiNZA & SVC	0.641	C=10, gamma=1, kernel=rbf
BERT	**0.660**	4.2.2を参照

4.4 考察

文脈を考慮するBERTで作成した分類器の方が，Bag of Words，TF-IDF，GiNZAのような，文脈に依存しない文書ベクトルを用いた場合より，高いスコアを示した．これは3.2で述べた，議論ラベルのアノテーションの範囲に起因すると考えられる．議論ラベルの判断材料としては，副詞や文末の表現などの単語単体では意味が汲み取りづらいものが多い．そのため，文脈を考慮できるBERTが高いスコアを示したのだと考えられる．

GiNZAが，Bag of WordsとTF-IDFよりも高いスコアを示したのも，似た理由だと考えられる．GiNZAで利用されている単語分散表現モデルであるChiVe[13]は，Skip-gramを用いたWord2Vec[4]を基に構築されている．Skip-gramは，特定の単語からその前後の単語が何かを予測するタスクを解き，その精度が上がるように単語の分散表現を学習すると

いうものである．周辺の単語を考慮しているため，全く考慮しない Bag of Words，TF-IDF よりも高いスコアを示したのだと考えられる．

5章　実験2: 関連する予算項目の連結

5.1　実験の目的

Budget Argument Mining における関連する予算項目の連結は，金額表現を含む発言と予算項目を文書と見なすと，似ている文書を探し出す，類似文書検索タスクと見なすことができる．類似文書検索の手法としては，文書ベクトルを作成し，そのコサイン類似度を計算するという手法がよく用いられている．しかし，文書ベクトルを作成する手法には，Bag of Words や Word2vec[4] のように文脈に依存しないものと，BERT[2] のように文脈に依存するものがある．そのため本実験では，文脈に依存しない文書ベクトルと，文脈に依存する文書ベクトルを用いて，金額表現を含む発言と予算項目の類似度を計算しスコアを算出することで，予算項目の連結には文脈を考慮すべきかどうかを明らかにすることを目的とする．

5.2　実験方法

以下は，本実験における関連する予算項目を連結する手順である．

1. Budget Argument Mining の予算のデータの予算項目名 (budgetItem) と予算に関する説明 (description) を用いて，各予算項目の文書ベクトルを作成する．
2. 会議録の訓練データの金額表現を含む発言 1 文から文書ベクトルを作成する．
3. 発言の文書ベクトルと各予算項目の文書ベクトルのコサイン類似度を計算する．
4. コサイン類似度が閾値を超えたものを，関連する予算項目として，relatedID に付与する．
5. 関連する予算項目が付与し終わったら，評価を行う．

また，Budget Argument Mining のデータセットの relatedID の値には，アノテーション時に金額表現に対して関連する予算項目があると判断され予算 ID を付与されたものと，ないと判断され null(要素なし) とされたものがある．そのため全ての金額表現に対し，一つの予算項目を付与するということができない．従って本実験では，コサイン類似度が閾値を超えたものを付与するという手法をとっている．

コサイン類似度は，以下のように計算する．

$$sim(x, y) = \frac{x \cdot y}{|x||y|}$$

本実験では，scikit-learn[1] の cosine_similarity を用いて計算している．コサイン類似度の閾値は，それぞれの手法で，0.05 単位で変動させ，最も F 値が高かった場合の閾値としている．

評価指標については，Precision(適合率)，Recall(再現率)，F 値を用いる[2]．

$$Precision = \frac{\text{出力に含まれる正解の数}}{\text{出力の数}}$$

$$Recall = \frac{\text{出力に含まれる正解の数}}{\text{正解の数}}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

以下に，それぞれの手法での詳細な手順を示す．

5.2.1 文脈を考慮しない文書ベクトル

まず，Bag of Words，TF-IDF を用いた手法について述べる．Bag of Words，TF-IDF におけるコサイン類似度の計算までの流れを図 5.1 に示す．まず，予算データの予算項目名 (budgetItem)，予算に関する説明 (description) と，会議録データの発言文を，Sudachi[8] で形態素解析し，単語分割する．分割モードは，小単位，中単位，長単位の 3 通りすべてで行なっている．ここで，品詞が名詞である単語のみを抽出する．次に，単語分割した予算項目名，予算に関する説明と，発言文を，scikit-learn の CountVectorizer または TfidfVectorizer を用いて，それぞれ Bag of Words モデルまたは TF-IDF モデルを作成し，文書ベクトルへと変換する．語彙は，予算項目名と予算に関する説明からモデルを作成する際に，同時に学習している．

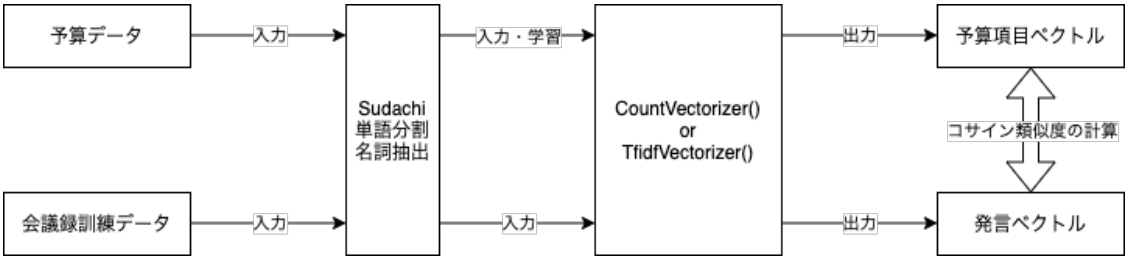

図 5.1: Bag of Words，TF-IDF のモデル作成の流れ

次に，GiNZA[12] の version5 を用いた手法について述べる．4.2 で述べた通り，GiNZA では，単語分散表現に ChiVe[13] を利用しており，得られる単語ベクトルを平均して文書ベクトルを計算できる．本実験でもこれを利用して，予算項目名，予算に関する説明と，発言文を文書ベクトルへと変換している．しかし，こちらでは名詞句のみを抽出して文書ベクトルに変換している．

[1] https://scikit-learn.org/stable

[2] https://poliinfo3.net/tasks/budget-argument-mining/

5.2.2 文脈を考慮した文書ベクトル

文脈を考慮した文書ベクトルを作成するモデルの一つに，BERT[2] が挙げられる．この BERT を文書ベクトルの作成に特化させるようにファインチューニングしたモデルが，Sentence-BERT である [6]．Sentence-BERT は，文脈を考慮した高精度な文書ベクトルを，高速に計算可能なモデルである．本実験では，この Sentence-BERT を用いて，関連する予算項目の連結を行う．

本実験で用いる Sentence-BERT は，sentence-transformers[3]を用いて，東北大が公開している事前学習済みモデル 2 つ (cl-tohoku/bert-base-japanese-whole-word-masking，cl-tohoku/bert-base-japanese-v2) を基に，京都大学の JSNLI データセットを 1epoch 分学習させることで作成した 2 つのモデルである．

この Sentence-BERT を利用して，予算項目名，予算に関する説明と，発言文を文書ベクトルへと変換している．

5.3 実験結果

実験結果を表 5.1 に示す．結果は，Sudachi の分割モード A(短単位) で単語分割し TF-IDF でベクトル化した手法が，他の手法と比較して 0．006-0.150 高い F 値を示した．従って関連する予算項目の連結には，文脈を考慮しない方が良いことが示された．また，Bag of Words，TF-IDF，GiNZA のように文脈に依存する文書ベクトルを用いた場合の方が，Sentence-BERT のように文脈に依存する文書ベクトルを用いた場合より，高い F 値を示した．適合率では，分割モード C(長単位) で単語分割し TF-IDF でベクトル化した手法が，最も高いスコアを示した．再現率では，GiNZA でベクトル化した手法が，最も高いスコアを示した．

表 5.1: 関連する予算項目の連結のスコア

手法	Precision	Recall	F 値
Bag of Words modeA	0.145	0.139	0.142
Bag of Words modeB	0.175	0.128	0.148
Bag of Words modeC	0.146	0.104	0.121
TF-IDF modeA	0.234	0.169	**0.196**
TF-IDF modeB	0.267	0.147	0.190
TF-IDF modeC	**0.294**	0.128	0.178
GiNZA	0.034	**0.180**	0.057
SBERT whole-word-masking	0.056	0.039	0.048
SBERT v2	0.038	0.065	0.046

[3]https://www.sbert.net/index.html

5.4 考察

文脈に依存する文書ベクトルを用いた場合の方が，文脈に依存する文書ベクトルを用いた場合より，高いF値を示したのは，会議録中の金額表現を含む発言文では,「(予算項目名)の予算額は，(金額表現) となっている」のような表現が多い可能性が考えられる．予算項目に関する説明が発言中でなされている場合は，文脈に依存する文書ベクトルを用いた場合の方が，スコアが高くなると考えられる．しかし，予算項目に関する説明が省かれている場合は，文脈や語順よりも字面などの表層的な情報がより重要になってくるため，このような結果になったと考えられる．

また，本実験を行う前に，金額表現を含む1文以外に，前方の1文を加えた2文，後方の1文を加えた2文，前後の1文を加えた3文をSentence-BERTに入力し，スコアを算出するという比較実験を行なっていた．この事前実験の目的は，文脈をどの程度考慮すべきかを明らかにするためであった．表5.2に，事前実験の結果を示す．事前実験の結果としては，金額表現を含む1文を入力した場合が，他と比べて高いF値を示した．この結果を受けて，本実験での対象文の長さを金額表現を含む1文としている．また，後方の1文を加えた2文を入力した場合に，再現率の向上が見られた．このことから，金額表現を含む1文の後方の文には、予算に関する表現が含まれる可能性があると考えられる．

表5.2: 各入力文の長さでのSentence-BERTのスコア

入力文の長さ	Precision	Recall	F値
金額表現を含む1文	**0.040**	0.065	**0.050**
前方の1文を加えた2文	0.025	0.048	0.033
後方の1文を加えた2文	0.035	**0.067**	0.046
前後の1文を加えた3文	0.021	0.041	0.028

6章　結論

本研究では，Budget Argument Mining における，議論ラベルの付与と関連する予算項目の連結を行う際，どのような手法が適しているかを比較実験をすることで明らかにした．

議論ラベルの付与では，Bag of Words，TF-IDF，GiNZA，BERT による文書分類で比較実験を行い，BERT を用いた手法が，他と比較して 0.013-0.059 高いスコアを示した．このことから，議論ラベルの付与では文脈を考慮した方が良いことが示された．

関連する予算項目の連結では，Bag of Words，TF-IDF，GiNZA，Sentence-BERT による類似文書検索で比較実験を行い，短単位で単語分割した TF-IDF を用いた手法が，他と比較して 0．006-0.150 最も高い F 値を示した．このことから，関連する予算項目の連結では，文脈や語順よりも字面などの表層的な情報が重要だということが示された．

これらの結果は，予算や議会会議録というデータセットの性質に起因することが考えられる．5 の考察でも述べたように，金額表現を含む文の後方の文に，予算項目に関連する文が続く傾向がある可能性ある．従って今後は，それらのデータセットに関する詳細な分析を行っていく．

参考文献

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[2] 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.

[3] John Lawrence and Chris Reed. Argument mining: A survey. Computational Linguistics, 45(4):765–818.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[5] ME Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee, and L Zettlemoyer. Deep contextualized word representations. arxiv 2018. arXiv preprint arXiv:1802.05365, 12, 2018.

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 11 2019.

[7] クイープ 福島真太朗 Sebastian Raschka, Vahid Mirjalili. 『Python 機械学習プログラミング-達人データサイエンティストによる理論と実践-第 3 版 (impress top gear)』. impress top gear. インプレス RD/インプレスビジネスメディア, 3rd edition, 2020.

[8] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a japanese tokenizer for business. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), may 2018.

[9] 金田健太郎 近江崇宏. 『BERT による自然言語処理入門 -Transformers を使った実践プログラミング-』. オーム社, 2021.

[10] 奥村 学 佐藤 敏紀, 橋本 泰一. 単語分かち書き辞書 mecab-ipadic-neologd の実装と情報検索における効果的な使用方法の検討.

[11] 栗林 樹生, 大内 啓樹, 井之上 直也, 鈴木 潤, Paul Reisert, 三好 利昇, and 乾 健太郎. 論述構造解析におけるスパン分散表現. 自然言語処理, 27(4):753–779, 2020.

[12] 松田寛. Ginza - universal dependencies による実用的日本語解析. 自然言語処理, 27(3):695–701, 2020.

[13] 海川祥毅 岡一馬 内田佳孝 浅原正幸 真鍋陽俊, 岡照晃. 複数粒度の分割結果に基づく日本語単語分散表現. In 言語処理学会第 25 回年次大会 (NLP2019), pages NLP2019–P8–5. 言語処理学会, 2019.

[14] 木村泰知, 永渕景祐, 乙武北斗, 佐々木稔, et al. 予算項目に関連する議論を対応づける budget argument mining のデータセット構築. 研究報告自然言語処理 (NL), 2021(10):1–9, 2021.