都議会議事録における自動要約のための数値情報自動修正手法の提案

松井我颯[1]　石川晴基[2] 中島陽子[3]　本間宏利[3] 秋葉友良[2]
[1] 釧路工業高等専門学校 電子情報システム工学専攻 [2] 豊橋技術科学大学
[3] 釧路工業高等専門学校　創造工学科
s230710@kushiro.kosen-ac.jp，ishikawa.haruki.yu@tut.jp，
{yoko，honma}@kushiro-ct.ac.jp，akiba@cs.tut.ac.jp

概要

自動要約技術は地方議会の議事録やネットニュースなどの情報の処理に有用だが，生成型要約において数値情報の誤りが問題となっている．本研究は，自動要約における数値情報の正確性を向上させるため，原文との比較に基づく修正手法を提案する．提案手法は，数値情報の誤りを特定し，原文に即した正確な数値への修正を可能にすることで，より信頼性の高い要約生成を目指す．

1　はじめに

近年，政治関連のフェイクニュースが世界的な社会問題になっている．誤情報の拡散は公共の認識を歪め，混乱を招く恐れがある．この問題に対処するために，多くの研究者が自然言語処理技術，特に自動要約の応用に注目している．現在主流の生成型自動要約技術は，大量のテキストデータから重要な情報を抽出し，簡潔に表現することで，情報の迅速な理解と判断を支援することができる [1]．しかし，小川らも指摘しているように，生成型自動要約技術には正確な数値情報の取得に失敗し，不正確な数値で要約を行ってしまうケースが見られる [2]．数値データは重要な情報であるため，このような誤りは，政治関連のニュースにおいて特に重要な問題となり得る [3] [4]．

そこで本研究では，自動要約に含まれる数値情報の正確性を評価し，原文との比較に基づいて正確な数値に修正する手法を二つ提案する．都議会議事録における答弁とその自動要約文を対象として比較実験を行い，二つの手法の性能を分析し，それぞれの優位性と適用範囲を検証する．

表 1: 都議会議事録の答弁と要約文の例

答弁	まず、インクルーシブな公園整備についてのお尋ねでございます。誰もが自分らしく輝くことのできるダイバーシティーの実現に向けまして、都立公園において、障害の有無にかかわらず全ての子供たちが安全に楽しむことができる遊び場、これを整備することは重要でございます。都といたしまして、今年度、障害児の保護者、そして障害者団体、障害児保育の現場、ユニバーサルデザインの有識者など、さまざまな方々にヒアリングを行ってまいりました。その中で、体を支える力が弱い子供さんたちが揺れる感覚を楽しめるそんな遊具や直射日光を避けることのできる休憩場所の設置など、さまざまなご意見をいただいたところでございます。こうしたご意見を踏まえまして、現在、砧公園と府中の森公園を対象に、具体的な設計を行っておりまして、平成三十一年度末の完成を目指し整備を進めてまいります。今後とも、都立公園でこうした取り組みを進めていくことで、障害の有無にかかわらず、全ての子供たちがともに遊び、また、学ぶ機会を積極的に提供してまいります。
要約文	障害者団体等にヒアリング。砧公園と府中の森公園を対象に 31 年度末完成を目指す。

2　テストデータ

テストデータには NTCIR-17 QA Lab-PoliInfo-4 の Answer Verification タスクで配布されたトレーニングデータに前処理を施したテキストを使用する [5]．表 1 に，実験に使用する都議会議事録の答弁 (原文) と，その要約文の例を示す．

2.1　データセットのフォーマットについて

表 2 に Answer Verification で使用されたデータセットのフォーマットを掲載する．このうち，会議が行わ

れた日付 (Date)，答弁の要約 (AnswerSummary)，答弁の全文 (AnswerOriginal)，要約の正誤 (PredictedClass) を使用する．

表 2: Answer Verification で使用されたデータセットのフォーマット

property	details
ID	識別番号
Meeting	会議名
Date	会議が行われた日付
Headlines	質問の発言全体の趣旨
SubTopic	サブトピック
QuestionSpeaker	質問者の名前
QuestionSummary	質問の要約
AnswerSpeaker	答弁者の名前
AnswerSummary	答弁の要約
AnswerOriginal	答弁の全文
PredictedClass	要約の正誤

2.2 前処理

AnswerSummary と AnswerOriginal のテキストについて，以下の a c に示す 3 つの前処理を行う．前処理を施した AnswerOriginal と AnswerSummary をそれぞれ原文と要約文として研究を行う．

a. 和暦を西暦に変換

要約元文章では，年についての言及時に「平成 29 年」「令和 2 年」といった和暦が使用されているが，これにより年号を考慮する必要が生じ，数値の正誤判定が困難になる．そこで，判定を容易にするために，すべて西暦表記に統一する．また，要約元文章で「今年」などと記載されている場合，要約では「令和 2 年」などの数値で表記されている場合がある．このため，Date を参照し,「今年」「昨年」「元年度」などの表現も西暦に変換する．

b. 漢数字をアラビア数字に変換

数値による判定を容易にするために，漢数字をアラビア数字に変換する．この変換では，まず日本語自然言語処理ライブラリである GiNZA[1] を使用して分かち書きを行い，数値を抽出する．その後，抽出された数値を kanjize[2] を用いて変換する．

c. 数値を使用した要約の抽出

本実験は数値の誤りに着目するため，要約文に数値が含まれているデータセットのみ抽出してテストデータとする．数値が含まれているかについては，正規表現により判別する．

3 提案手法

要約文における数値の誤要約を検出し，修正するための二つの手法を提案する．

第一の手法は，原文から要約に使用されている数値と単位が含まれるセグメントを特定し，その情報を基に数値を比較する．これにより，複数の文章内に数値が含まれている場合でも，要約に使用されている数値のみ取得することが可能である．

第二の手法は，各数値の係り受け元を特定し，その情報を基に数値を比較する．この比較により，文脈上の誤りや，数値の不適切な使用を識別し，修正することが可能である．

3.1 セグメントの特定による要約の修正

原文から要約に使用されている数値と単位が含まれるセグメントを特定する．特定された原文のセグメントと要約文から，使用されている数値と単位を抽出する．その後，抽出された数値と単位を原文と要約文間で比較し，要約に使用されている数値が原文と一致しているかを評価する．

3.1.1 セグメントの特定

要約内容が含まれる箇所を特定するため，まず要約元の文章を句点（。）を基準に区切り，それによって得られる各文を個別のセグメントとして扱う．要約に使用された特定のセグメントを識別するために，テキストマイニングの分野で広く用いられる手法の一つである Term Frequency-Inverse Document Frequency（Tf-Idf）を採用する．Tf-Idf は，文書内の各単語の重要度を測定する手法であり，文書全体の中で特定の単語がどの程度重要であるかを定量的に評価することが可能である．算出した Tf-Idf 値に基づいて，要約文と各セグメント間の類似度の評価を行う．この評価は，各セグメントにおける単語の Tf-Idf 値から構成されるベクトルを基に，コサイン類似度を計算することで行う．コサイン類似度は，二つのベクトル間の角度のコサインを計算することで，これらのベクトルがどの程度似ているかを測定する手法である．

[1] https://megagonlabs.github.io/ginza/
[2] https://pypi.org/project/kanjize/

具体的には，各セグメントにおける単語の Tf-Idf 値をベクトルとして表現し，要約文と各セグメントのベクトル間でコサイン類似度を計算する．要約文と各セグメント間のコサイン類似度を比較し，最も高い値を持つセグメントをキーセグメントとして抽出する．この方法により，要約文に最も類似する内容を持つセグメントを効率的に識別できると考えられる．

3.1.2 数値と単位の抽出

GiNZA を用いて，キーセグメントと要約文を単語レベルで分割する．分割された各テキストデータにおいて，文頭から探索を開始し，数値を抽出する．抽出された数値の直後に位置する単語を該当数値の単位とする．この手法により，数値とそれに対応する単位を識別し，リストを生成する．リストの例を表 3 に示す．

表 3: 取得した数値と単位の例

	テキストデータ	数値と単位
原文	そのため 2020 年度に 都教育委員会は、都立学校 70 校の 普通教室、特別教室等に無線 LAN 環境を整備いたします。	「2020 年度」 「70 校」
要約文	2015 年度に都立学校 80 校 の普通教室、特別教室等に整備。	「2015 年度」 「80 校」

3.1.3 要約の正誤判定

最後に，生成されたキーセグメントのリストと要約文のリストを比較する．要約文のリスト内の数値と単位がキーセグメントのリストに含まれていない場合，その要約を誤りと判定する．

3.2 係り受け解析に基づく数値探索による要約の修正

第二の手法は，係り受け解析を使用する．解析には spaCy[3]を使用しており，モデルは日本語データで学習されている `ja_ginza_electra`[4] を用いる．

3.2.1 係り受け解析

原文と要約文から各文内の単語や記号 (トークン) を取得し，これらをリストとして扱う．原文と要約文それぞれのトークンリストから数値と単位を取得し，どの事柄に関する数値なのか特定する．この処理は GiNZA で解析した際に付与された係り受けラベルと品詞ラベルに基づいて行われる [6]．要約文への係り受け解析の結果を木構造のグラフで表示した例を図 1, 2 に示す．図 1 では，単位である「年」が数値である「2018」に係るように，数詞修飾語を意味する係り受けラベル nummod が付与されている．また,「2018」には数詞を意味する品詞ラベル NUM が付与されている．

係元の抽出を行うために，数詞の位置を特定する．文頭から NUM ラベルが付与されているトークン (NUM トークン) を探索し，そのトークンを探索の開始位置とする．図 1 ではこれが「2018」にあたる．次に，NUM トークンに nummod ラベルが付与されている場合，係元をその数値の単位として取得する．図 1 ではこれが「年」にあたる．図 2 のように nummod ラベルではなく，複合名詞を意味する compound ラベルが付与されている場合がある．この場合は，NUM トークンの次に格納されているトークンと，その係元のトークンを接続し，単位として取得する．図 2 では，これが「年」と「末」を結合した「年末」にあたる．次に，要約文から単位の係元の原形を取得する．ここで取得したトークンをキートークンとする．ここで取得したキートークンは数値がどの事柄に関しているかを表している．取得したキートークンが非「する」「いく」のような自立可能動詞である場合，さらにその係元を探索し，キートークンとする．図 1, 2 ではそれぞれ「発送」と「拡充」にあたる．その後，原文からキートークンを探索し，係先にある数値と単位を取得する．以上の処理により，数値，単位，キートークンが格納されているリストを生成する．表 4 に生成したリストの例を示す．

表 4: 取得した数値とキートークンの例

	テキストデータ	数値とキートークン
原文	(前略) そのため、2020 年度に 都教育委員会は、都立 学校 70 校の普通教室、 特別教室等に無線 LAN 環境を整備いたします。	「2020 年度，整備」 「70 校，教室」
要約文	2015 年度に都立学校 80 校の普通教室、 特別教室等に整備。	「2015 年度，整備」 「80 校，教室」

3.2.2 要約の正誤判定

原文と要約文のリストを比較し，要約文のキートークンが原文にも含まれている場合，そのキートークンが係っている数値を比較する．数値が一致しなければ，

[3]https://github.com/explosion/spaCy/
[4]https://huggingface.co/megagonlabs/transformers-ud-japanese-electra-base-discriminator/

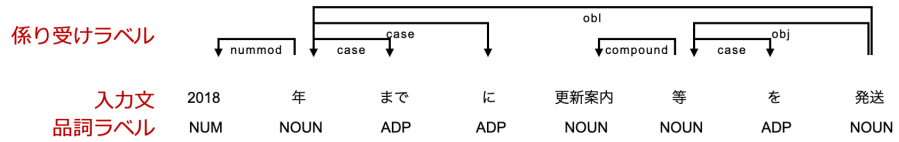

図 1: 係り受けラベルと品詞ラベルの例 1

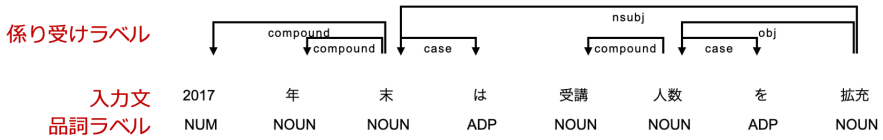

図 2: 係り受けラベルと品詞ラベルの例 2

その要約を誤りと判定する．例えば，表 4 のようにキートークン「整備」が係っている「2020 年度」と「2015 年度」の数値が異なる場合，要約文中の「2015 年度」を「2020 年度」に修正する．同様に,「普通教室、特別教室」というキートークンが係っている「70 校」と「80 校」の数値についても同じ処理を行う．

4 評価実験

提案手法の有効性を検証するため，評価実験を行う．実験には 2 章で説明したテストデータ 110 件を使用する．

4.1 実験設定

原文と要約文に対してセグメントの特定による数値探索手法 (SSR:Segment Summary Revision) と係り受け解析に基づく数値探索手法 (DSR:Dependency-based Summary Revision) を適用する．評価指標は，テストデータの要約の正誤 (PredictedClass) を基準として，精度（Precision），再現率（Recall），および F1 スコアを算出し，提案手法の有効性を分析する．

4.2 実験結果

表 5 に評価結果を示す．DSR は SSR と比較して F1 スコアが高く，要約の正誤を正確に判定できている．DSR のスコアが高い要因として，要約文内に複数の数値が存在する場合に，係り受け解析をすることで文脈を考慮できる DNC が，より関連性の高い数値を適切に識別できたことが挙げられる．

判定に失敗した例として，SSR は単位を基準に比較を行うため，例えば「2020 年度規定改正を行い、2021 年度から校内管理体制を整える。」といったように原文に同じ単位が複数回出現する場合では，どちらの数値に修正すべきか正しく判断できす，修正できない．一方で，DSR は数値が関連する事柄を考慮することで，この種のミスを避けることができる．しかし，DSR ではキートークンの取得が難しい場合や，要約文のキートークンが原文には含まれていない場合，判定に失敗する場合がある．

表 5: 評価結果

手法	Recall	Precision	F-1
SSR	1.000	0.272	0.428
DSR	1.000	0.554	0.713

5 おわりに

本研究では，原文と要約文の数値と単位を含むセグメントを特定する手法と，係り受け解析を用いて原文と要約文の数値を文脈的に比較する手法を提案し，自動要約における数値情報の誤りを特定し，原文を参照して正確な数値に修正を行った．評価結果から，数値情報の誤りを修正する上で，係り受け解析による文脈の考慮有用であることが示された．

係り受け解析に基づく数値探索手法では，複数の数値が同一のキートークンを係元としている場合は対応不可能である．今後は，数値が含まれている文章だけではなく，その前後の文脈も考慮した解析をすることで改善を目指す．

参考文献

[1] 石垣達也, 高村大也, and 奥村学. ”複数文質問を対象とした抽出型および生成型要約.” 自然言語処理 26.1 (2019): 37-58.

[2] Ogawa, Yasuhiro, Yugo Kato, and Katsuhiko Toyama. ”nukl’s QA System at the NTCIR-16 QA Lab-PoliInfo-3.” Proceedings of The 16th NTCIR Conference. 2022.

[3] Igarashi, Naoki, et al. ”Forst: A Challenge to the NTCIR-16 QA Lab-PoliInfo-3 Task.” Proceedings of The 16th NTCIR Conference. 2022.

[4] Ohsugi, Ryoto, et al. ”AKBL at the NTCIR-16 QA Lab-PoliInfo-3 Task.” Proceedings of The 16th NTCIR Conference. 2022.

[5] 浅原正幸, 金山博, 宮尾祐介, 田中貴秋, 大村舞, 村脇有吾, 松本裕治. ”Universal Dependencies 日本語コーパス.” 自然言語処理 26.1 (2019): 3-36.

[6] Yasuhiro Ogawa, Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Kazuma Kadowaki, Tomoyoshi Akiba, Minoru Sasaki, and Akio Kobayashi. Overview of the NTCIR-17 QA Lab-PoliInfo-4 Task. In Proceedings of the 17th NTCIR Conference, 2023.