

2019年度 博士論文

**クエリ指向テキスト要約のための
ニューラル言語モデルに関する研究**

2019年12月25日

京都産業大学先端情報学研究科

学生証番号: 658047

木村 輔

Research on Neural Language Model
for Query-focused Automatic Text Summarization

by

Tasuku Kimura

Doctoral Dissertation

Division of Frontier Informatics

Kyoto Sangyo University

December 2019

目次

第1章 序論	1
1.1 本論文の背景	1
1.2 本論文のアプローチ	3
1.3 本論文の構成	5
第2章 関連研究	6
2.1 緒言	6
2.2 テキスト要約	6
2.3 文の分散表現の獲得	7
2.4 文エンコーダと注意機構を組み合わせたテキスト要約	7
2.5 ゲート機構による注意機構の強化	8
2.6 深層学習によるクエリ指向テキスト要約	8
第3章 地方議会会議録を対象にした クエリ指向テキスト要約	9
3.1 緒言	9
3.2 地方議会会議録データセットの概要	11
3.3 提案手法	16
3.3.1 モデル構築の方針	16
3.3.2 提案モデル	16
3.4 実験	21
3.4.1 目的	21
3.4.2 学習設定	21
3.4.3 評価方法	21
3.4.4 結果	24
3.5 考察	25
3.5.1 トークナイザの効果	25
3.5.2 多様性機構の効果	25
3.5.3 出力長制御の効果	25
3.6 結言	27
第4章 文エンコーダを追加した クエリ指向テキスト要約	28
4.1 緒言	28
4.2 問題の定式化	30
4.3 Diversity driven Attention Model	31
4.3.1 Word Representation	32
4.3.2 Document Encoder	32
4.3.3 Query Encoder	32

4.3.4	Query Attention Mechanism	32
4.3.5	Document Attention Mechanism	33
4.3.6	Diversity Cell	33
4.3.7	Summary Decoder	34
4.4	提案手法	35
4.4.1	Sentence Representation	35
4.4.2	Sentence Encoder	35
4.4.3	Sentence Attention Mechanism	36
4.4.4	提案手法による既存手法の拡張	36
4.5	実験	38
4.5.1	データセット	38
4.5.2	評価指標	38
4.5.3	モデルの設定と学習の詳細	39
4.5.4	結果	39
4.6	考察	40
4.7	結言	42

第5章 原文を俯瞰する機構を備えた

クエリ指向テキスト要約 43

5.1	緒言	43
5.2	問題の定式化	45
5.3	Triple-Hierarchical Attention Model	46
5.3.1	Word Representation	46
5.3.2	Sentence Representation	47
5.3.3	Document Encoder	47
5.3.4	Sentence Encoder	47
5.3.5	Query Encoder	48
5.3.6	Query Attention Mechanism	48
5.3.7	Sentence Attention Mechanism	48
5.3.8	Document Attention Mechanism	49
5.3.9	Diversity Cell	49
5.3.10	Summary Decoder	50
5.4	提案手法	51
5.4.1	Adaptive Attention Mechanism	51
5.4.2	提案手法による既存手法の拡張	51
5.5	実験	53
5.5.1	実験1: Debatepedia データセットにおける実験	54
5.5.2	実験2: Wikipedia データセットにおける実験	57
5.6	結言	66

第6章 結論 67

第1章 序論

1.1 本論文の背景

現代において、インターネット、主に、World Wide Web は、新聞やニュース番組に並ぶ情報源となっている。日ごと爆発的に増加する Web 上のデジタルデータは、有益な情報からフェイクニュースのような悪意のある情報まで玉石混淆の様相を呈している。限られた時間の中で、膨大な情報の中からユーザの情報要求を満たす情報のみを効率よく取得するためには、元となる膨大な情報に含まれる、ユーザの情報要求に関するトピックについて、簡単に内容を確認できる短いテキストなどが重要な役割を果たす。

クエリ指向要約は、入力文書に含まれる特定のトピックのみについてまとめられた、より短い文書を自動生成する処理である。様々なトピックを含む膨大なテキストから、ユーザが知りたい特定のトピックを含む要約を生成することにより、ユーザは、短い時間で要約元の内容を把握できるため、クエリ指向要約は、情報の獲得や取捨選択の効率化に大きく貢献する。

情報獲得の例として、真偽が不明な情報を検証するための一次情報源の獲得が挙げられる。昨今、社会的な影響を及ぼすようになったフェイクニュースの登場により、真偽不明の情報の正確性や妥当性を検証する事実検証は、情報取得の際に重要視されつつある。Google[1] は、フェイクニュースの拡散防止の一環として、検索エンジンの検索結果に含まれるニュース記事に対して、事実検証の結果（事実か虚偽か、またはどちらともいえない）を表示する機能の導入を発表している。第三者機関が行ったクエリ指向テキスト要約は、この事実検証や検証結果の確認の効率化に貢献できる。例えば、政治家の発言について事実検証する場合、公式の報告書である会議録のような一次情報源から、検証したい情報のみを取得する必要がある。しかし、このような資料は膨大な文書長を有するものも多く、短時間に該当箇所を確認することは困難である。そこで、クエリ指向テキスト要約により、一次情報源を対象に、政治家の発言の中心的なトピックに関連する要約を生成することで、短時間に一次情報源の内容を参照でき、効率的な事実検証および検証結果の確認に繋がる。

また、情報の取捨選択の例として、検索結果内容の把握の効率化が挙げられる。Web から真に必要な情報を含む Web ページを取得するために検索エンジンを用いることは、いまや人々の生活に浸透している。しかし、検索エンジンの検索結果として提示される Web ページ数も膨大な量になる場合が多く、提示された全ての Web ページを確認することは現実的ではない。そこでユーザは、検索結果と共に提供される、Web ページを自動要約したテキストを読むことで、検索結果から必要な Web ページのみを取捨選択することができる。このテキストはスニペットと呼ばれ、ユーザの情報要求であるクエリに応じて、Web ページの要約内容を適切に変化させた内容となっている。このように、クエリ指向テキスト要約は、効率的な情報の取得や取捨選択のためには、無くてはならない存在であるといえる。

一方で、本論文で対象とする、クエリ指向テキスト要約のための生成型の手法は、データセット

の整備状況と、生成アプローチによるテキスト要約手法が抱える問題から、これまでのところほとんど報告されていない。

テキスト要約は、要約作成時に特定の観点を考慮するかどうかにより、汎用テキスト要約とクエリ指向テキスト要約に分類される。

汎用テキスト要約は、原文の内容を、より短くまとめたテキストの出力を目的としており、これまで主流として研究されてきた。そのため、様々なワークショップのタスクとして扱われてきたことで、数多くのデータセット [2, 3, 4] が整備されている。

一方、クエリ指向テキスト要約のデータセットは、あまり多いとはいえない状況である [5, 6]。また、既に整備されているデータセットの多くは、原文のトークン数が、数十トークン程度と少ない場合が多かった。しかし、本来、クエリ指向テキスト要約の原文は、トークン数が多いことが想定される。クエリ指向テキスト要約では、同一の原文に対して異なるクエリが与えられると、それぞれに適切な異なる要約が生成されることが期待される。原文は、様々なクエリに対応する複数のトピックを含むことが想定され、その結果、原文のトークン数が必然的に多くなることが考えられるためである。よって、原文のトークン数が少ないデータセットは、本来、原文のトークン数が多いことが想定されるクエリ指向テキスト要約に対しては、必ずしも適切であるとはいえない。以上より、クエリ指向テキスト要約の特徴に適したデータセットは、未だ多くは整備されていない状態であるといえる。

また、テキスト要約は、原文を抽出して要約を生成するか否かにより、抽出型テキスト要約と生成型テキスト要約に分類される。

抽出型テキスト要約では、入力である原文から重要な句や文を抽出し、それらを組み合わせることで、最終的な要約を出力する。要約に用いる句や文を原文から直接取得するため、出力される内容は、原文の内容と高い一貫性を保つ傾向にある。しかし、断片的に得られた語句を組み合わせる際に、可読性の低いテキストを出力しやすい欠点も抱えている。また、要約に含まれる語句は、抽出によって取得されるため、より短い語句への言い換えのような原文にない表現を出力することが難しい。

一方、生成型テキスト要約は、入力である原文を元に要約中の語句を確率的に予測することで、最終的な要約を出力する手法である。語句を予測する手法には言語モデルが用いられることが多く、特にニューラル言語モデルによって生成されるテキストは、可読性が高い傾向にある。また、言語モデルの学習過程で獲得した言い換え表現など、原文に存在しない語句を出力することが可能である。しかし、原文を中間表現へ変換する手法や、言語モデルの精度によって、入力文書の意味する内容から逸脱した内容の要約を出力する問題があることを指摘されている。以上より、生成型テキスト要約は、抽出型テキスト要約よりも可読性が高く、言い換え表現など入力テキストに含まれない語句を生成できる一方で、生成した要約の品質を確保することが困難であることが指摘されている。

1.2 本論文のアプローチ

本論文では、クエリ指向テキスト要約のための生成型言語モデルの研究を発展させるべく、クエリ指向テキスト要約に、より適する新たなデータセットの構築と、クエリ指向要約が抱える課題の解決を目指した新規のニューラルネットワークを用いた言語モデル（以下、ニューラル言語モデルと呼ぶ）を提案する。

具体的には、まず、クエリ指向テキスト要約のための新たなデータセットとして、議員等の膨大な発言を含む地方議会会議録、会議録中の特定トピックを指すクエリ、および、トピックに関する要約の3点から構成されるデータセットを構築する。このデータセットを用いてクエリ指向要約のためのニューラル言語モデルの精度を検証し、当該モデルが抱える課題を整理する。

次に、生成アプローチにおけるクエリ指向要約の課題の一つである、単語数の多い要約元が入力された際に、生成要約の品質が低下する課題の解決を目指し、文単位の入力も扱えるよう拡張した新規のニューラル言語モデルを提案する。また、会議録の要約データセットと比較して、更に多くの単語を含む原文を提案手法の実験対象とするために、新たなデータセットとして、English Wikipedia[7] および Simple English Wikipedia[8] からデータセットを構築した。

最後に、生成アプローチによるテキスト要約では、入力文書の意味する内容から逸脱した内容の要約を出力する課題が指摘されている。この課題の解決を目指し、要約元の全体の俯瞰と局所的な注視を制御する機構を導入した、新規のニューラル言語モデルを提案し、その有効性を検証する。

本論文では、以上の事項を次のようなアプローチで実現する。

1. クエリ指向テキスト要約のための新たなデータセットを構築し、当該データセットにおける、生成アプローチを用いたクエリ指向テキスト要約の精度を検証し、クエリ指向におけるニューラル言語モデルが抱える課題を整理する。まず、「地方議会会議録」と「当該会議のニュースレター」が、「議員等の膨大な発言を含む原文」と「特定トピックに関する会議録の要約」の対応関係にある点に着目することで、新しいクエリ指向要約のためのデータセットを構築する。当該データセットは、悪意のあるフェイクニュースの登場により重要性が増している、既出情報の事実検証において活用されることが期待される。これは、膨大な発言を含む地方議会会議録から、事実検証したい特定トピックに関する要約を生成することで、要約を生成せず直接会議録の特定トピックに該当する箇所を確認する場合と比較して、確認に要する作業時間を短縮でき、事実検証の効率化に繋がるためである。本研究では、地方議会会議録の議員の発言を原文、地方議会の議題をクエリとみなし、地方議会に基づくニュースレター上の議員発言の要約を生成する問題と捉え、クエリ指向テキスト要約の枠組みで取り組む。
2. クエリ指向テキスト要約のための新たなデータセットを構築し、入力文書を文単位で扱う機構を導入した、新規のニューラル言語モデルを提案する。クエリ指向テキスト要約では、同一の入力文書に対して複数のクエリとその要約のペアが存在し得る。各クエリは、入力文書の根幹となるトピックが同じでも、少しずつ異なるトピックを併せ持っている。たとえば「京都」についてまとめた記事に対して、「京都の地理」というクエリや「京都の文化」というクエリが想定される。そのため多様なクエリに対応可能な入力文書となればなるほど、一般に、入力文書中のトピック数も増え、結果として、入力文書の単語数も増加すると考えられ

る。しかし、生成アプローチの言語モデルにおいて多く利用される、入力文書を単語単位で扱うエンコード手法は、単語数の多い入力のエンコードに失敗することが指摘されている。また、これまで提案された手法では、そもそもデータセットの入力文書中の単語数が少ない場合が多く、この問題について議論できる状態ではなかった。そこで本研究では、Wikipediaから自動生成した、単語数の多い入力文書を含むデータセットを構築し、既存のモデルに入力文書を文単位で扱う機構を導入したモデルを提案することで、この課題の解決と検証を目指す。

3. 要約元の全体の俯瞰と、局所的な注視を制御する注意機構を導入した、新規のニューラル言語モデルを提案する。要約生成時に、入力文書中の特定の単語を注視できる注意機構を用いた場合、局所的な特定の単語の影響が増大し、相対的に入力文書全体を俯瞰することが困難になると推察される。これは、入力文書の意味する内容から逸脱した内容の要約を出力する、生成アプローチを用いたクエリ指向要約が抱える課題の一要因であると考えられる。生成する要約内容の一貫性を保つためには、常に入力文書の特定の単語を注視するだけでなく、時には入力文書全体を俯瞰することも重要である。テキスト要約の既存研究で提案された注意機構により出力を制御する手法は、入力文書中の各単語をより強調するか、または削除するかを制御する手法であった。一方、本研究では、入力の全体を元に要約を生成するか、それとも注目された単語を元に要約生成するか、その比重をゲート機構により適応的に制御する機構を提案する。この手法により、生成する要約内容の一貫性をより保つことが可能なニューラル言語モデルの構築を目指す。

1.3 本論文の構成

本論文では、クエリ指向テキスト要約のためのニューラル言語モデルに関する研究について論じる。また本論文は本章を含む全6章で構成される。

第2章では、本論文の研究に関わりの深いニューラル言語モデルによるテキスト要約の先行研究と、提案手法に導入した機構に関連する手法について紹介し、本研究の位置づけを明確に示す。

第3章では、クエリ指向テキスト要約のための新たなデータセットの構築し、当該データセットにおける、生成アプローチを用いたクエリ指向テキスト要約の精度を検証する。そして、実験の結果を通して、クエリ指向要約におけるニューラル言語モデルが抱える課題を整理する。本章では、地方議会会議録を対象にしたクエリ指向テキスト要約を扱うことで、この目的の達成を目指す。地方議会会議録の議員の発言を原文、地方議会の議題をクエリとみなし、地方議会に基づくニュースレター上の議員発言の要約を生成する問題と捉え、クエリ指向テキスト要約の枠組みで取り組む。

第4章では、クエリ指向テキスト要約のための新たなデータセットの構築し、入力文書を文単位で扱う機構を導入した、新規のニューラル言語モデルを提案する。クエリ指向テキスト要約は、一般に、入力文書が単語を多く含んでいることが想定される。これは、様々なトピックを含む可能性が高い入力文書に対して、複数のクエリとその要約のペアが存在し得るため、入力文書の単語数が自然と多くなると考えられるからである。しかし、生成アプローチの言語モデルにおいて頻繁に利用される、入力文書を単語単位で扱うエンコード手法は、単語数の多い入力のエンコードに失敗することが指摘されている。また、これまで提案された手法では、そもそもデータセットの入力文書の単語数が数十単語程度と少ない場合が多く、この問題について議論できる状態ではなかった。そこで本研究では、Wikipediaから自動生成した単語数の多い入力文書を含むデータセットを構築する。また、入力文書の単語数の増加に頑健な、入力文書を文単位で扱う機構を導入したモデルを提案する。

第5章では、入力文書全体の俯瞰と入力文書の局所的な注視を制御する機構を備えた、生成アプローチによるクエリ指向テキスト要約について提案する。生成アプローチによるテキスト要約では、入力文書の意味する内容から逸脱した内容の要約を出力する課題があると指摘されている。生成する要約と入力原文の内容の一貫性を保つためには、常に入力文書の特定の単語を注視するだけでなく、時には入力文書全体を俯瞰することも重要である。そこで本研究では、入力の全体を元に要約を生成するか、それとも注視した単語を元に要約を生成するか、その比重をゲート機構により適応的に制御する機構を提案する。この手法により、生成する要約と入力原文の内容の一貫性をより保つことが可能なニューラル言語モデルの構築を目指す。実験では、既存データセットと、第4章で構築したデータセットのそれぞれで、既存手法と提案手法の精度を比較する。

第6章では本論文の各研究成果についてまとめる。

第2章 関連研究

2.1 緒言

本論文は、クエリ指向テキスト要約のためのニューラル言語モデルに関する研究をまとめている。そこで本章では、まずテキスト要約の起こりから、機械学習によるテキスト要約の登場について簡潔に説明する。その後、本論文の研究と特に関わりの深い以下の4点の関連研究について述べることで、本論文の研究の位置づけを明確に示す。

- 文の分散表現の獲得
- 文エンコーダと注意機構を組み合わせたテキスト要約
- ゲート機構による注意機構の強化
- 深層学習によるクエリ指向テキスト要約

2.2 テキスト要約

1950年代後半に始まったテキスト要約の研究は、本文中から抽出した重要文を要約として出力する Luhn ら [9] が提案した抽出型の手法が主流となっていた。それ以降、重要文抽出は様々研究されており、本文中の出現頻度が高い重要語 [9]、文の位置情報 [10]、接続詞のような手掛かり語 [10] などを用いる抽出手法が提案されている。また同時期に、構文解析器の精度が向上したことに伴い、抽出した重要文を短くする文圧縮 [11] も盛んに研究されていた。主に文圧縮は、原文から構文解析木を生成し、重要文抽出における重要語や構文構造における各文節の重要度を元に、不要な文節を枝刈りすることで実現されている。

2000年代初期には、Text Summarization Evaluation (SUMMAC) [12] という初めてのテキスト要約の評価型ワークショップが開催された。続いて、Document Understanding Conference (DUC) [13] や、国立情報学研究所 (NII) [14] が主催する NTCIR-2[15] のタスクとして Text Summarization Challenge (TSC) [16] が開催されたことで、多くの原文とその要約のデータセットが整備された [5, 6]。それに伴い、機械学習によって重要文を抽出する研究が増加した。機械学習を用いた手法 [17, 18] では、これまでの研究と同様に単語の出現頻度に加え、N-gramなどを各文の特徴量とし、support vector machine (SVM) [19] や Hidden Markov Model (HMM) といったモデルを用いて各文の重要度を推定している。

現在では、画像処理や機械翻訳の分野で成功を取めた、深層学習によるテキスト要約が盛んに研究されている。本論文の研究は、全て深層学習によるニューラル言語モデルを用いたテキスト要約手法である。

2.3 文の分散表現の獲得

近年, Word2Vec[20] の出現で, 深層学習による単語の分散表現に続いて文の分散表現が盛んに研究されている.

Paragraph Vector[21] は, 深層学習による文の分散表現の最初期の手法の一つである. この手法では, Word2Vec の CBOW と Skip-gram のそれぞれを拡張した 2 種類の教師なし学習が提案された. 同様に Kiros らは, Skip-gram を着想とした教師なし学習の手法である Skip-Thought[22] を提案している. この手法はエンコードした m 番目の文から, $m+1$ 番目の文と $m-1$ 番目の文を正しく予測させることで文エンコーダを学習する.

教師あり学習の手法では, 文間の関係から文の分散表現の学習を目指した InferSent[23] が提案されている. 含意関係認識のデータセットを用いて文間の関係の 3 値分類タスクを訓練することで, Skip-thought と同等の精度でありながら, 学習データと学習時間を削減することに成功した. 一方で, InferSent の問題点として, データセットの構築のコストが高い点が挙げられる. Nie らは, 談話マーカ予測により自動的に構築されたデータセットを用いることで, 品質の高い文の分散表現を学習する DisSent[24] を提案した. 談話マーカや文の区切りに基づきデータセットが構築されるため, データセット全体を学習する教師なし学習と比較して, ターゲットを絞った速い学習が可能になると報告している.

第 4 章および第 5 章では, 原文を特徴ベクトルへ変換する手法として, トークン単位の分散表現に加えて, 文単位の分散表現として Skip-Thought を用いている. これは Skip-Thought が, InferSent や DisSent と比較してデータセットの構築が容易であるからだ. また各研究のデータセットである Wikipedia[7] および Debatepedia[25] には, 含意関係や談話マーカに依らない様々な文の関係が存在することを踏まえ, 特定の文間の関係のみを含むデータセットが学習対象である InferSent や DisSent より, すべての種類の文を学習対象とする Skip-Thought が文の分散表現としてより適切であると考えた.

2.4 文エンコーダと注意機構を組み合わせたテキスト要約

文エンコーダと注意機構を組み合わせたテキスト要約も活発に研究されている.

Tan ら [26] は, 原文の重要文を識別するために, グラフに基づく注意機構を導入した Hierarchical encoder-decoder を提案している. Hierarchical encoder-decoder[27] は, 原文を単語単位と文単位の 2 段階でエンコードし, デコード時には要約の文単位でデコードした後に, 予想要約の各文ベクトルから単語をデコードする Encoder-Decoder[28] である. また Li ら [29] は, 2 種類の情報選択戦略を組み込んだ Encoder-Decoder を提案している. 情報選択は, 文単位でエンコードされた各ベクトルから不要な情報を削除するゲート機構と, ゲーティングされた文ベクトルから特定のベクトルを強調する注意機構によって実現される. Zhou ら [30] は, 双方向 Gated Recurrent Units (BiGRU) [31] でエンコードした単語ベクトルを文ベクトルに基づいてゲーティングする Encoder-Decoder を提案している.

入力文書を文ベクトルのみでエンコードする Tan らや Li らの手法は, トークン数の多い原文に強いモデルの構築が期待される. 一方, 提案手法や Zhou らの手法と異なり, トークン単位の原文をデコードに用いないため, 原文の重要語を損なう可能性が高くなることが考えられる. また

Zhou らの手法を含むこれらの既存手法では、事前学習されていない Long-Short Term Memory (LSTM) [32, 33] や GRU を文エンコーダとして用いている。一方、第 4 章および第 5 章の提案手法では、事前学習した文エンコーダを用いるため、文の特性をより捉え、文の分散表現を活かした要約生成が期待される。

2.5 ゲート機構による注意機構の強化

注意機構にゲート機構を組み合わせた手法はこれまでも研究されている。Xu ら [34] は、画像の説明文生成の研究において、注意機構で生成したベクトルをゲート機構によって更に強調する手法を提案している。Dhingra ら [35] は、文書とクエリの 2 入力を与えられるタスクにおいて、クエリベクトルから算出される値によって文書ベクトルのアテンションを生成する Gated-Attention Module という機構を提案している。ただし、この手法において、クエリから算出される値の取る範囲は 0.0 ~ 1.0 ではないため、厳密にはゲート機構を用いた手法とは言い切れない。

これらの手法に対し、第 5 章の提案手法である適応的な注意機構は、入力の全体を表現するベクトルを使用するか、それとも注意されたベクトルを使用するかを、ゲート機構により各デコードステップにおいて適応的に制御する機構である。この手法により、モデルはデコードの状況に応じて入力の全体を俯瞰するか、入力の重要なトークンに焦点を合わせるかを適応的に切り替えることができると期待される。

2.6 深層学習によるクエリ指向テキスト要約

クエリ指向テキスト要約における抽出型の研究として、Yousefiazar ら [36] は、Ensemble Noisy Auto-Encoder (ENAE) という単一文書要約手法を提案している。TF-IDF など与えられる入力が疎になることを低減するために、各文書の局所的な語彙の利用と入力にランダムなノイズを含める Auto-Encoder [37] を提案している。また Cao ら [38] は、文書や文書クラスタのための分散表現学習と注意機構を利用した手法を提案している。彼らは重要文のランク付けにおいて、クエリ関連性によるランク付けとセンテンス顕著性によるランク付けが独立していることを指摘し、それらを同時に考慮したランク付けモデルを提案している。

クエリ指向テキスト要約における生成型の研究として、Kiddon ら [39] は、与えられた料理名と材料リストから料理レシピを自動生成する Neural Checklist Models を提案している。このモデルでは、出力するレシピの記述内容に一貫性を持たせるために、ベクトルに変換された料理名をデコード時の入力に加える手法を提案している。またモデルの内部の Checklist によって、与えられた材料リストの使用状況を管理している。Nema ら [40] は、深層モデルに基づく生成型のクエリ指向テキスト要約のための Encoder-Decoder モデルを提案している。このモデルでは、入力の原文の各単語ベクトルに加えてクエリにも注意機構を用いることで、各デコードステップにおける重要なクエリを出力に反映させている。また、同じフレーズを繰り返し出力する Recurrent Neural Network (RNN) の問題を解決するために、注意機構によって生成される原文のコンテキストベクトルが、各デコードステップにおいて直交するように変換する手法を提案している。

第3章 地方議会会議録を対象にした クエリ指向テキスト要約

3.1 緒言

近年、フェイクニュースの影響により、既出の情報について真偽を検証する、事実検証の重要性が再確認されている。例えば、フェイクニュースのターゲットとなりやすい政治家の発言について事実検証するとき、地方議会会議録のような一次情報源を確認する必要がある。しかし、このような会議録は、議事の進行や提出された資料を含む膨大な文量を有しているため、議員の意見を一目で確認することは難しい。

この問題を受けて、情報の真偽判断に主眼を置いたワークショップが次々立ち上がっている。Fake News Challenge (FNC) [41] は、フェイクニュースに関する問題の解決を目指したプロジェクトである。2016年に結成されたこのプロジェクトは、世界中の学会および企業から100人超のボランティアと71のチームが協力し発足した。彼らはその最初のタスクとして、ニュースの見出しと記事本文が同一の立場であるかを判断する Fake News Challenge Stage 1 (FNC-1) [41] を開催した。また、CLEF-2018 Fact Checking Lab[42] というワークショップでは、単にテキストの真偽判断を目指すだけでなく、政治的な議論の各発話のうち、事実検証すべき発言を推測するタスクを提案した。このように情報の真偽を検証する研究は大きく注目されている。

国立情報学研究所 (NII) [14] が主催するワークショップの NTCIR-14[43] では、近年のフェイクニュースの問題に対するタスクとして、3種類のサブタスクで構成される QA Lab-PoliInfo Task[44] を実施した。そのうちのひとつである Summarization Task は、ある発話者の1発言から、その意図が損なわれない要約の生成を想定したサブタスクである。このようにして生成された要約は、既出情報の事実検証を始め、ある政策に対する議員の立場判断など、幅広い活用が見込まれる。

本章では、NTCIR-14の QA Lab-PoliInfo Task における Summarization Task に対するチーム KSU の提案モデルと結果について述べる。Summarization Task では、原文となる地方議会会議録における議員の1発言の他に、議題となるトピックや制限字数などが入力として与えられる。これらの入力に対して、要約モデルは、発言の意図を歪めない要約を出力する必要がある。地方議会において議員は、1度の登壇で複数のトピックを含む発話をする。そのため、原文全体を俯瞰できる要約生成を目的とする汎用テキスト要約の枠組みを用いて、議員の1発言を要約すると、議題となるトピック以外のトピックも含む概要が生成される可能性が高い。

そこで筆者は、議題となるトピックをクエリとみなし、このサブタスクをクエリ指向テキスト要約の枠組みで捉えることで、入力されたトピックのみを含む要約の生成を目指した。クエリ指向のテキスト要約は、複数のトピックから構成された原文と、ユーザがシステムに与えたクエリを入力とし、クエリに関連するトピックを中心に原文を要約する処理である。また、Summarization Task では、生成する要約を制限字数以内に収める必要があるため、菊池ら [45] が提案した、出力

長を制御する機構を提案モデルに導入した。

以降、3.2節では、研究対象となる地方議会会議録と構築したデータセットについて、3.3節では、データセットの特性を踏まえた提案モデルの方針と構築について、3.4節では、実験の目的とその結果について、3.5節では、実験の考察について、3.6節では、結言について述べる。

3.2 地方議会会議録データセットの概要

QA Lab-PoliInfo の Summarization Task は、地方議会会議中のあるメインピックに基づく 1 発言から、その発言中のサブピックを中心とした要約を生成するタスクである。QA Lab-PoliInfo のオーガナイザは、**要約データセット**を作成するために、地方議会会議録の本文から人手で 1 発言を抽出し、対応するニュースレターから要約やサブピックを抽出した。ここでニュースレターとは、会議録全体の活動報告や議員等の発言内容の要約から作成された市民を対象とした広報誌である。

筆者は学習に用いる要約データセットを作成するために、オーガナイザと同様に議員発言の一次情報源として地方議会会議録（以下、**会議録全文データセット**）とその要約としてニュースレター（以下、**質疑応答データセット**）を用いた。また Web 上から収集した対象となった会議を以下に示す。

- 東京都議会 [46, 47]: 平成 13 年から平成 29 年に実施された 63 件の会議
- 板橋区議会 [48, 49]: 平成 21 年から平成 29 年に実施された 37 件の会議

要約データセット

このデータセットは、地方議会会議録内の 1 発言を、あるサブピックについて要約したテキストのペアを 1 オブジェクトとするデータセットである。Summarization Task のフォーマルランで用いた要約データセットの 1 オブジェクトの項目を表 3.1 に示す。

表 3.1: 要約データセットの 1 オブジェクトの項目（ただし説明の簡略化のため、一部項目は省略している）

項目	説明
ID	各オブジェクトに一意に付与された識別子
Date	会議の実施日
Meeting	会議名
Speaker	発言した議員等の氏名と所属
MainTopic	発言全体の主軸となるトピック
SubTopic	発言中の質問または回答に当たるトピック
Summary	本タスクのターゲットとなる要約
Length	生成する要約の字数
Source	要約元となる 1 発言の全文

また実際のデータ例について表 3.2 に示す。例では、発言のメインピックである八ッ場ダムを中心とした原文から、サブピックである地下水についてまとめられた要約を生成する必要がある。本研究では、クエリ指向テキスト要約の枠組みを用いて、サブピックに基づく地方議会会議録中の発言の要約生成を目指した。

表 3.2: 要約データセットの例

項目	内容
ID	Summarization-2018-JA-FormalTest-00001
Date	23-9-29
Meeting	平成 23 年 第 3 回定例会
Speaker	松下玲子 (民主党)
MainTopic	自然の脅威に対しダムは有効か ハッ場ダムは工事中止の判断へ
SubTopic	地下水
Summary	災害時を含め積極的な活用推進を。
Length	50 字以内
Source	地産地消の取り組みは電力だけにとどまることなく、今こそエネルギーや水の自立的な都市東京を目指すべきです。他県の犠牲、他県の人々の生活を犠牲にしなければ、東京が利水上も治水上も本当に成り立たないのかどうか検証が必要と考えます。今回の福島原発事故後の三月末には、放射性物質の飛散により、水道水からも乳児の飲用基準を超える放射性元素が検出されました。その後、非検出が続いており、検査体制も強化されているようではありますが、今後、台風や豪雨等で堆積していた放射性物質が川に流され、水道水から新たに放射性物質が検出されるおそれも想定されます。水道水の検査体制や放射性物質の除去等、水道水の安全対策にどのように取り組んでいるのか伺います。また、地表に堆積している放射性物質、いわゆるホットスポットともいわれている放射性物質汚染の激しい地域を調査し、除染を行うことが今後の重要な課題でもあり、国の決定を待たずとも、区市町村と連携して取り組んでいただきたいと要望いたします。

会議録全文データセット

このデータセットは、地方議会会議録における発言内の1文を1オブジェクトとするデータセットである。データセットの1オブジェクトの項目を表3.3に示す。

表 3.3: 会議録全文データセットの1オブジェクトの項目（ただし説明の簡略化のため、一部項目は省略している）

項目	説明
id	各オブジェクトに一意に付与された識別子
date	対応するニュースレターの公開日
speakerPosition	発言した議員等の所属等
speakerName	発言した議員等の氏名
speech	発言内の1段落

また表3.2の発言の冒頭に対応する1オブジェクトは、表3.4のように格納される。

表 3.4: 会議録全文データセットの例

項目	内容
id	TokyoPref-DA-2011-3-13-6-19
date	2011.09.29
speakerPosition	七十八番
speakerName	松下玲子
speech	地産地消の取り組みは電力だけにとどまることなく、今こそエネルギーや水の自立的な都市東京を目指すべきです。他県の犠牲、他県の人々の生活を犠牲にしなければ、東京が利水上も治水上も本当に成り立たないのかどうか検証が必要と考えます。
id	TokyoPref-DA-2011-3-13-6-20
date	2011.09.29
speakerPosition	七十八番
speakerName	松下玲子
speech	今回の福島原発事故後の三月末には、放射性物質の飛散により、水道水からも乳児の飲用基準を超える放射性物質が検出されました。その後、非検出が続いており、検査体制も強化されているようではありますが、今後、台風や豪雨等で堆積していた放射性物質が川に流され、水道水から新たに放射性物質が検出されるおそれも想定されます。水道水の検査体制や放射性物質の除去等、水道水の安全対策にどのように取り組んでいるのか伺います。

質疑応答データセット

このデータセットは、地方議会会議録における質疑応答の各要約のペアを1オブジェクトとするデータセットである。データセットの1オブジェクトの項目を表3.5に示す。

表 3.5: 質疑応答データセットの1オブジェクトの項目（ただし説明の簡略化のため、一部項目は省略している）

項目	説明
id	各オブジェクトに一意に付与された識別子
date	ニュースレターの公開日
title	会議名
mainTopics	質疑応答全体の主軸となるトピック
subTopic	各質疑応答が持つ個別のトピック
questionSpeakerPosition	質問の発言した議員等の所属等
questionSpeakerName	質問の発言した議員等の氏名
questionSpeech	質問の発言の全文
answerSpeakerPosition	回答の発言した議員等の所属等
answerSpeakerName	回答の発言した議員等の氏名
answerSpeech	回答の発言の全文

また表3.2の発言に関連する1オブジェクトおよび回答の発言は、表3.6のように格納される。

表 3.6: 質疑応答データセットの例

項目	内容
id	TokyoPref-DA-2011-3-13-Spk000
date	2011.09.29
title	平成 23 年第 3 回定例会
mainTopics	自然の脅威に対しダムは有効か ハッ場ダムは工事中止の判断へ
subTopic	地下水
questionSpeakerPosition	民主党
questionSpeakerName	松下玲子
questionSpeech	災害時を含め積極的な活用推進を。
answerSpeakerPosition	水道局長
answerSpeakerName	増子敦
answerSpeech	今後とも、地盤沈下や水質の動向に配慮し、可能な限り地下水の活用を図る。

要約データセットの自動構築手順

ここでは、収集した会議録全文データセットおよび質疑応答データセットから、要約データセットを構築する手順について述べる。

1つのメインピックのみを持つ発言を会議録から抽出するために、会議録全文データセットを対象にテキストを分割した。まず会議録全文データセットの連続する同一人物の発言を全て連結することで、ある人物が一度の登壇で発言したテキストを生成する。この処理により、この時点の質疑応答に関する1つの発言は、複数のメインピックを含むことが想定される。

次に各発言をメインピック毎に分割するために、Utiyamaらが提案したテキスト分割手法 [50] を用いた。この手法は、単語分布に基づくテキスト分割手法であり、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に分割位置を決定する。この処理により、この時点の質疑応答に関する1つの発言は、単一のメインピックを含むことが想定される。また各発言は次の処理のために、全文検索システムである Solr [51] を用いてインデキシングされる。

1つのメインピックのみを持つ発言に、1つのサブピックのみを持つ要約を紐づけるために検索システムを用いた。まず質疑応答データセットの紐付け対象となる要約が持つ会議名、公開日および発言者名によって検索対象をフィルタリングした。その後、対象となる要約のみをクエリにして最上位1件の発言を取得した。ただし要約の内容が「取組は。」の場合のみ、クエリとしてサブピックを用いた。

以上の処理によって得られた発言と検索に用いた要約をペアとする 19,689 件から構成される要約データセットを構築した。

3.3 提案手法

ここでは、まず学習データセットの特性を踏まえた上でモデル構築の方針について述べ、その後、方針を元にした提案モデルについて詳述する。

3.3.1 モデル構築の方針

3.2節でも述べたように、まず要約モデルの学習のために下記の会議を対象に、Web上から収集した以下の会議録全文データと各会議録に対応する要約対象データからコーパスを作成した。

- 東京都議会: 平成13年から平成29年に実施された63件の会議
- 板橋区議会: 平成21年から平成29年に実施された37件の会議

なお東京都議会については、QA Lab-PoliInfo Taskのフォーマルランで使用された年度の会議2件はコーパスに含めない。

しかし、(1)構築した19,689件のデータセットは、深層学習で用いるには十分な量とは言い難かった。また、(2)特定の会議録から構築したため、開発コーパス中の42,343件の語彙だけを用いると、フォーマルランのみで出現する未知語への対応が難しくなることが考えられた。

そこで筆者は、Byte Pair Encoding (BPE)の一種であるSentencePiece [52]により構築された語彙によって入力文をトークナイズし、コーパスの拡張と未知語への対応をすることで、2つの問題の解決を試みた。SentencePieceで用いられるサブワードは、学習データ中の高頻度の文字列を1語として扱い、一方で低頻度の文字列を部分文字列もしくは文字単位まで分割する手法である。このように処理することで特定の言語においては、事実上、未知語が存在しなくなるため、問題(1)に対処できる。またSentencePieceは、ユニグラムに基づくトークナイザを構築できるため、同じ入力に対して複数の分割候補を確信度付きで出力できる。これにより、データセットから学習データを動的にサンプリングすることで学習データの拡張ができるため、問題(2)に対処できる。

Summarization Taskでは、生成する要約は入力されたトピックに沿った内容である必要がある。そこで提案手法ではトピックをクエリとみなしエンコーダで中間表現へ変換し、注意機構 [53]を拡張することで、各デコードステップにおいてトピックに基づく要約の生成を目指した。またSummarization Taskでは、単に短い要約ではなく、指定された字数以内の要約を生成する必要がある。そこで筆者は出力長を制御するために、菊池らが提案したLenEmb [45]を提案モデルに導入した。最後に、同じ単語を繰り返し生成するRecurrent Neural Network (RNN)の固有の問題に対応するために、筆者はNemaらが提案したDiversity Cell Module [40]を提案モデルに導入した。Diversity Cellは、直前に生成した予測語に対して、現在の予想ベクトルが直交するように演算することで、出力するトークンの多様性の改善を目指した機構である。

3.3.2 提案モデル

筆者は、ベーシックなEncoder-Decoderモデル [28]に、トークナイザ、多様性制御および出力長制御の3つの機構を組み合わせ、6種類の要約モデルを構築した。各モデルの組み合わせについて、表3.7に示す。また提案モデルの例としてKSU-05モデルの構成を図3.1に示す。

表 3.7: 各モデルにおけるモジュールの組み合わせ

モデル名	トークナイザ	多様性制御	出力長制御
KSU-01	MeCab	✓	—
KSU-02	MeCab	—	—
KSU-03	SentencePiece	✓	—
KSU-04	SentencePiece	—	—
KSU-05	MeCab	✓	✓
KSU-06	SentencePiece	✓	✓

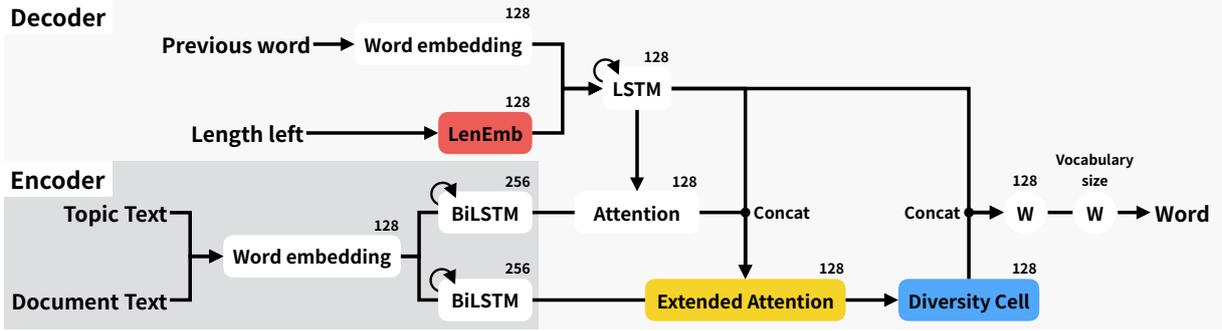


図 3.1: KSU-05 のモデル構成

ここでは、まずベースとなる Encoder-Decoder について説明する。その後、出力への多様性制御機構、出力長制御機構およびクエリとなるトピックベクトルを考慮した注意機構についてそれぞれ詳述する。

Encoder-Decoder

Encoder-Decoder は、系列データを入出力とする深層モデルである。系列データ間の変換が可能のため、主に翻訳、要約および対話などのニューラル言語モデルとして広く利用されている。

エンコーダの役割は、入力された系列データを特徴ベクトルへ変換することである。主に入力された系列データの要素を分散表現へ変換する埋め込み層と、分散表現へ変換された系列データを入力とし要素の前後関係の特徴ベクトルへ反映する RNN 層から構成される。

本研究では、原文およびクエリであるサブトピックのベクトル変換にエンコーダを用いた。また埋め込み層として Word2Vec[20]、RNN 層として双方向 LSTM (BiLSTM) [54] をそれぞれ用いた。入力される原文を $\mathbf{d} = t_1^d, t_2^d, \dots, t_{l_d}^d$ と定義したとき、エンコードステップ i の特徴ベクトル $h_i^d \in \mathbb{R}^{\delta_3}$ を生成するエンコーダは以下の式で表される。

$$\vec{h}_i^d = \text{LSTM}_d^{\text{forward}}(\vec{h}_{i-1}^d, e(t_i^d)), \quad (3.1)$$

$$\overleftarrow{h}_i^d = \text{LSTM}_d^{\text{backward}}(\overleftarrow{h}_{i-1}^d, e(t_i^d)), \quad (3.2)$$

$$h_i^d = [\vec{h}_i^d; \overleftarrow{h}_i^d], \quad (3.3)$$

ここで、 \vec{h}_i^d は入力シーケンスを順方向に入力していることを、 \overleftarrow{h}_i^d は入力シーケンスを逆方向に入力していることを示し、 $[\ast; \ast]$ は2つのベクトルの concatenate 演算を示す。また、 $e(\ast)$ は、1つのトークン $t \in \mathbb{R}^{\delta_1}$ を δ_2 次元の特徴ベクトルへ変換する埋め込み層である。

同様にクエリであるトピック $\mathbf{q} = t_1^q, t_2^q, \dots, t_{l_q}^q$ のエンコーダは以下の式で表される。

$$\vec{h}_j^q = \text{LSTM}_q^{\text{forward}}(\vec{h}_{j-1}^q, e(t_j^q)), \quad (3.4)$$

$$\overleftarrow{h}_j^q = \text{LSTM}_q^{\text{backward}}(\overleftarrow{h}_{j-1}^q, e(t_j^q)), \quad (3.5)$$

$$h_j^q = [\vec{h}_j^q; \overleftarrow{h}_j^q], \quad (3.6)$$

ここで、 $h_j^q \in \mathbb{R}^{\delta_4}$ は、エンコードステップ j のトピックの特徴ベクトルである。

デコーダは、特徴ベクトルへ変換された系列データを元に、新たな系列データを生成することができる。最初期には、エンコーダの出力である特徴ベクトル h_i^d の最終エンコードステップのみをエンコーダからの入力とすることが多かった。しかし近年では、デコーダへ入力される特徴ベクトルは、各デコードステップ k 毎に注意機構によって再計算され、そのたびに特徴ベクトルが更新される。本研究においても注意機構を導入した。そこでまず注意機構について詳述し、その後、デコーダについて説明する。

原文に対応する注意機構の役割は、エンコードされた特徴ベクトル h_i^d の各エンコードステップ i およびデコーダが出力した特徴ベクトル $h_{k-1}^o \in \mathbb{R}^{\delta_5}$ から、デコードステップ k において重要なエンコードステップについて加重平均をとったコンテキストベクトル $d_k \in \mathbb{R}^{\delta_3}$ を生成することである。デコードステップ k の原文のコンテキストベクトル d_k は次の式で算出される。

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d), \quad (3.7)$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \quad (3.8)$$

$$d_k = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d. \quad (3.9)$$

同様にデコードステップ k のトピックのコンテキストベクトル $q_k \in \mathbb{R}^{\delta_4}$ は次の式で算出される。

$$a_{k,j}^q = v_q^\top \tanh(W_q h_{k-1}^o + U_q h_j^q), \quad (3.10)$$

$$\alpha_{k,j}^q = \frac{\exp(a_{k,j}^q)}{\sum_{j'=1}^{l_q} \exp(a_{k,j'}^q)}, \quad (3.11)$$

$$q_k = \sum_{j=1}^{l_q} \alpha_{k,j}^q h_j^q. \quad (3.12)$$

デコーダの役割は、デコードステップ $k-1$ の特徴ベクトル $h_{k-1}^o \in \mathbb{R}^{\delta_5}$ および要約のトークン $t_{k-1}^o \in \mathbb{R}^{\delta_1}$ を入力とし、デコードステップ k における特徴ベクトル $h_k^o \in \mathbb{R}^{\delta_5}$ を出力することである。

$$h_k^o = \mathbf{LSTM}_o(h_{k-1}^o, e(t_{k-1}^o)). \quad (3.13)$$

その後、特徴ベクトル h_k^o とコンテキストベクトル d_k から、デコードステップ k における要約のトークン $t_k^o \in \mathbb{R}^{\delta_1}$ を予測する。

$$t_k^o = \text{softmax}(Wf(W_{dec}h_k^o + V_{dec}d_k)), \quad (3.14)$$

ここで活性化関数 f は恒等関数である。

Diversity Cell

この機構の役割は、各デコードステップ k において、同じトークンを繰り返し生成する RNN の問題点を解決することである。そこで Nema らは繰り返し同じトークンをデコードしないよう Long-Short Term Memory (LSTM) [32, 33] の実装を拡張し、入力ベクトル d_k を各デコードステップ k において互いに直交するベクトル $d'_k \in \mathbb{R}^{\delta_3}$ へ変換する機構 **SD₂** を次の式で定義した。

$$\begin{pmatrix} i_k \\ f_k \\ o_k \\ \hat{c}_k \\ g_k \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} d_k \\ h_{k-1} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \quad (3.15)$$

$$c_k = \sigma(i_k) \odot \tanh(\hat{c}_k) + \sigma(f_k) \odot c_{k-1}, \quad (3.16)$$

$$c_k^{diverse} = c_k - \sigma(g_k) \frac{c_k^\top c_{k-1}}{c_{k-1}^\top c_{k-1}} c_{k-1}, \quad (3.17)$$

$$d'_k = \sigma(o_k) \odot \tanh(c_k^{diverse}). \quad (3.18)$$

LenEmb

菊地らは、Encoder-Decoder の出力長を制御する機構として 4 種類の手法を提案している。そのうち LenEmb は、デコーダの LSTM の入力に長さ埋め込みベクトルを追加することで、出力長を制御する手法である。長さの埋め込み層 $W_{le} \in \mathbb{R}^{\delta_7 \times \delta_6}$ を用いて、各デコードステップ k における残りの出力長 $l_k^o \in \mathbb{R}^{\delta_6}$ を δ_7 次元の長さ埋め込みベクトルへ変換する。そして LSTM に重み行列 $W_l \in \mathbb{R}^{\delta_5 \times \delta_7}$ を追加することで、残りの長さ情報に応じた文生成が可能となる。残りの出力長 l_k^o はエンコード処理の後に初期化され、以下の条件に従って値が更新される。

$$l_1^o = \text{length}, \quad (3.19)$$

$$l_{k+1}^o = \begin{cases} 0 & (l_k^o - \text{len}(t_k^o) \leq 0) \\ l_k^o - \text{len}(t_k^o) & (\text{otherwise}) \end{cases}, \quad (3.20)$$

ここで $\text{len}(\ast)$ はデコードステップ k における出力トークン t_k^o の長さを返す関数であり, $length$ はモデルに与える最終出力長である.

クエリを考慮した注意機構

Summarization Task では, 生成する要約は入力されたトピックに沿った内容である必要がある. そこで提案手法ではトピックをクエリとみなしエンコードで中間表現へ変換し, 注意機構を拡張することで, 各デコードステップにおいてトピックに基づく要約の生成を目指した.

この機構の役割は, エンコードされた原文の各特徴ベクトル h_i^d から, デコードステップ k において重要なエンコードステップについて加重平均をとった, コンテキストベクトル d_k を生成することである. ここで, トピックのコンテキストベクトル q_k によって, 原文のコンテキストベクトル d_k の生成をコントロールするべきと考えた. そこで, 原文のコンテキストベクトル d_k の式は, トピックのコンテキストベクトル q_k を受け取るパラメータ $Z_q \in \mathbb{R}^{\delta_3 \times \delta_4}$ を持つ.

デコードステップ k における, 原文のコンテキストベクトル d_k は, 次の式で算出される.

$$a_{k,i}^d = v_d^\top \tanh(W_d h_k^o + U_d h_i^d + Z_q q_k), \quad (3.21)$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \quad (3.22)$$

$$d_k = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d. \quad (3.23)$$

提案手法の各モデルでは, デコードステップ k における原文の注意機構を, 式 (3.7) ~ 式 (3.9) からそれぞれ式 (3.21) ~ 式 (3.23) へ変更し導入することとした.

3.4 実験

3.4.1 目的

本実験の目的は、各機構単体の有効性およびその組み合わせの有効性について実験により明らかにすることである。そこでベースラインとなる Encoder-Decoder に各機構を導入したモデルを構築し、実験により精度を比較する。

3.4.2 学習設定

学習データセットとして 3.2 節で作成したデータセット 19,689 件を用いた。また評価データセットとして QA Lab-PoliInfo の Summarization Task で用いたフォーマルランのテストデータセット 146 件を用いた。

各モデルの構築に Chainer[55, 56] および ChainerMN[57, 58]、BPE に基づくトークナイザに SentencePiece、辞書に基づくトークナイザに MeCab[59] を用いた。SentencePiece の語彙数を 8,000、MeCab の語彙数を 42,343 とした。モデルの各層のパラメータは図 3.1 に示す通りである。また各モデルは最適化手法として Adam[60] ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $eps = 10^{-8}$) を用い、コア数 4、ミニバッチサイズ 24、エポック数 30 で学習した。

3.4.3 評価方法

本実験では、自動評価指標である ROUGE[61] および人手による評価指標である Quality question scores[62] の 2 種類の評価指標によってモデルの精度を比較する。

ROUGE は、Recall-Oriented Understudy for Gisting Evaluation の略であり、今日において最も代表的なテキスト要約における自動評価指標の一つである。この評価指標は、正解にあたる参照要約とモデルが出力したシステム要約間のエレメントの一致率に基づいて算出される。ROUGE は、このエレメントの定義に応じて様々な評価指標が提案されている。本研究では人手による主観評価と相関性が高いと報告されている ROUGE-N ($N = 1, 2$)、ROUGE-SU4 および ROUGE-L を用いた。

Quality question scores は、QA Lab-PoliInfo のオーザナイザによって設計された評価指標である。このスコアは生成された要約の品質を評価するために設計され、生成要約の内容の評価、生成要約の文法の評価および内容と文法を踏まえた総合評価の三種類の観点で構成されている。

ここでは、まず各 ROUGE の算出方法について詳述したのち、Quality question scores について紹介する。

ROUGE-N

ROUGE-N は、エレメントの単位を N-gram に定義した評価指標である。このとき ROUGE-1 は unigram に基づく評価指標であり、ROUGE-2 は bigram に基づく評価指標である。システム要

約を $S = \{e_1^S, e_2^S, \dots, e_m^S\}$ と定義し、参照要約を $R = \{e_1^R, e_2^R, \dots, e_n^R\}$ と定義した場合、ROUGE-N は以下に示す式によって算出される。

$$\text{ROUGE-N}(S, R) = \frac{\sum_{e^R \in \text{n-gram}(R)} \sum_{e^S \in \text{n-gram}(S)} \text{Count}_{\text{match}}(e^R, e^S)}{\sum_{e^R \in \text{n-gram}(R)} \text{Count}(e^R)}, \quad (3.24)$$

ここで $\text{n-gram}(S)$ はシステム要約中の N-gram の集合を表し、 $\text{n-gram}(R)$ は参照要約中の N-gram の集合を表す。また $\text{Count}(e^R)$ は参照要約の N-gram の数を返す関数であり、 $\text{Count}_{\text{match}}(e^R, e^S)$ は参照要約とシステム要約間で一致した N-gram の数を返す関数である。

ROUGE-SU

ROUGE-SU について詳述するために、まず ROUGE-S について定義する。ROUGE-U は、エレメントの単位を skip-bigram に定義した評価指標である。skip-bigram とはテキスト中の順序に基づく任意の 2 要素の組み合わせである。このとき各要素は隣接する必要がないため、スキップを許した bigram といえる。skip-bigram では連続した一致を必要としないが要素の順序を捉えることができるため、隣接しない要素間の係り受けの類似などを捉えることができる。ROUGE-S は以下に示す式によって算出される。

$$\text{Recall}_{\text{skip}}(S, R) = \frac{\sum_{e^R \in \text{skip}(R)} \sum_{e^S \in \text{skip}(S)} \text{Count}_{\text{match}}(e^R, e^S)}{C(n, 2)}, \quad (3.25)$$

$$\text{Precision}_{\text{skip}}(S, R) = \frac{\sum_{e^R \in \text{skip}(R)} \sum_{e^S \in \text{skip}(S)} \text{Count}_{\text{match}}(e^R, e^S)}{C(m, 2)}, \quad (3.26)$$

$$\text{ROUGE-S}(S, R) = \frac{(1 + \beta) \times \text{Recall}_{\text{skip}}(S, R) \times \text{Precision}_{\text{skip}}(S, R)}{\text{Recall}_{\text{skip}}(S, R) + \beta^2 \times \text{Precision}_{\text{skip}}(S, R)}, \quad (3.27)$$

ここで $\text{skip}(S)$ はシステム要約中の skip-bigram の集合を表し、 $\text{skip}(R)$ は参照要約中の skip-bigram の集合を表す。また β は $\text{Recall}_{\text{skip}}(S, R)$ と $\text{Precision}_{\text{skip}}(S, R)$ の相対的な重要度を制御する変数であり、 C は組み合わせ関数である。一方、問題点として文頭の要素と文末の要素も 1 つの skip-bigram として扱われることが挙げられる。このような skip-bigram は参照要約とシステム要約の一致度を評価するには不適切である。そこで連続したペアを生成する最大範囲を制限することが提案されている。ROUGE-S4 は、最大範囲を 4 で定義した ROUGE-S である。この場合、式 (3.25) の $C(n, 2)$ および式 (3.26) の $C(m, 2)$ は、それぞれスキップ数の最大範囲における skip-bigram の数に置き換えて計算する。

ROUGE-SU は、エレメントの単位を unigram および skip-bigram に定義した評価指標である。ROUGE-S の潜在的な問題点として、参照要約とシステム要約の skip-bigram が一つも一致しない場合、その評価値が 0 になる点がある。この問題点を解決するために、ROUGE-S の算出に unigram の素性を追加した評価指標が ROUGE-SU である。ROUGE-SU は以下に示す式によって算出される。

$$\text{Recall}_{\text{su}}(S, R) = \frac{\sum_{e^R \in \text{su}(R)} \sum_{e^S \in \text{su}(S)} \text{Count}_{\text{match}}(e^R, e^S)}{n + C(n, 2)}, \quad (3.28)$$

$$\text{Precision}_{\text{su}}(S, R) = \frac{\sum_{e^R \in \text{su}(R)} \sum_{e^S \in \text{su}(S)} \text{Count}_{\text{match}}(e^R, e^S)}{m + C(m, 2)}, \quad (3.29)$$

$$\text{ROUGE-SU}(S, R) = \frac{(1 + \beta) \times \text{Recall}_{\text{su}}(S, R) \times \text{Precision}_{\text{su}}(S, R)}{\text{Recall}_{\text{su}}(S, R) + \beta^2 \times \text{Precision}_{\text{su}}(S, R)}, \quad (3.30)$$

ここで $\text{su}(S)$ はシステム要約中の unigram および skip-bigram の集合を表し、 $\text{su}(R)$ は参照要約中の unigram および skip-bigram の集合を表す。また ROUGE-S と同様に、ROUGE-SU4 は、最大範囲を 4 で定義した ROUGE-SU である。

ROUGE-L

ROUGE-L は、エレメントの単位を最長共通部分列 (Longest Common Subsequence, LCS) に定義した評価指標である。LCS とは 2 つの系列において一致する部分列のうち、最長の部分列のことである。このとき部分列の各要素は隣り合う必要がなく、一方で順序は元の系列と同じである必要がある。例えば系列 A-B-C-D と系列 A-C-X-Y-D の共通部分列は、A, C, D, A-C, A-D および A-C-D の 6 種類となり、LCS は A-C-D となる。ROUGE-L において LCS は、 u 個の文から構成される参照要約 R の 1 文 r_i と v 個の文から構成されるシステム要約 S の各文 s_1, s_2, \dots, s_v 毎に算出され、下記の式によってスコア化される。

$$\text{LCS}_{\text{score}}(S, r_i) = \frac{\text{Count}_{\text{lcs}}(S, r_i)}{\text{Count}(r_i)}, \quad (3.31)$$

ここで $\text{Count}_{\text{lcs}}(S, r_i)$ は参照要約 R の 1 文 r_i とシステム要約 S の各文 s_1, s_2, \dots, s_v の LSC に含まれる要素数、 $\text{Count}(r_i)$ は r_i の要素数を表す。以上を踏まえ、ROUGE-L は以下に示す式によって算出される。

$$\text{Recall}_{\text{lcs}}(S, R) = \frac{\sum_{r_i \in R} \text{LCS}_{\text{score}}(S, r_i)}{u}, \quad (3.32)$$

$$\text{Precision}_{\text{lcs}}(S, R) = \frac{\sum_{r_i \in R} \text{LCS}_{\text{score}}(S, r_i)}{v}, \quad (3.33)$$

$$\text{ROUGE-L}(S, R) = \frac{(1 + \beta) \times \text{Recall}_{\text{lcs}}(S, R) \times \text{Precision}_{\text{lcs}}(S, R)}{\text{Recall}_{\text{lcs}}(S, R) + \beta^2 \times \text{Precision}_{\text{lcs}}(S, R)}. \quad (3.34)$$

Quality question scores

QA Lab-PoliInfo のオーザナイザは、人手による要約の評価指標として Quality question scores を設計した。このスコアは生成された要約の品質を評価するために設計され、生成要約の内容の評価 (content)、生成要約の文法の評価 (well-formed) および内容と文法を踏まえた総合評価 (total) の三種類の観点で構成されている。各観点は人手によってグレード A からグレード C まで 3 段階評価される。また正解要約とは異なる正解を許容するためにグレード X を含むこととして

いる。ある要約モデルが出力したシステム要約の集合 M における、各観点 $view$ に対する Quality question scores である $QQ(M, view)$ は下記の式によって算出される。

$$QQ(M, view) = \frac{\sum_{S \in M} g(S, view)}{|M|}, \quad (3.35)$$

$$g(S, view) = \begin{cases} 2 & (\text{グレード } A) \\ 1 & (\text{グレード } B) \\ 0 & (\text{グレード } C) \\ \alpha & (\text{グレード } X) \end{cases}, \quad (3.36)$$

ここで α は、正解要約以外の正しい要約であるとオーガナイザに評価されたとき 2 の値となり、それ以外の場合は 0 の値となる。

3.4.4 結果

QA Lab-PoliInfo の Summarization Task におけるフォーマルランの実験結果を表 3.8 および表 3.9 に示す。なお表 3.8 の content の項目について、例外の正解を認めない場合 ($\alpha=0$) および例外の正解を認める場合 ($\alpha=2$) のそれぞれについて結果を示す。

表 3.8: 各モデルにおける Quality question scores の結果。各スコアの最大値は 2 である

モデル名	content		well-formed	total
	$\alpha = 0$	$\alpha = 2$		
KSU-01	0.043	0.043	1.955	0.048
KSU-02	0.076	0.121	1.745	0.071
KSU-03	0.091	0.157	1.715	0.104
KSU-04	0.111	0.167	1.419	0.093
KSU-05	0.048	0.078	1.692	0.048
KSU-06	0.078	0.169	1.535	0.091

表 3.9: 各モデルにおける ROUGE-N ($N=1,2$) , ROUGE-L および ROUGE-SU4 の結果。

モデル名	Recall		F-measure	
	N1	N2	L	SU4
KSU-01	0.161	0.028	0.197	0.061
KSU-02	0.187	0.044	0.212	0.082
KSU-03	0.175	0.036	0.196	0.065
KSU-04	0.174	0.045	0.197	0.073
KSU-05	0.230	0.029	0.201	0.067
KSU-06	0.226	0.040	0.195	0.069

3.5 考察

3.5.1 トークナイザの効果

まず KSU-01 と KSU-03, KSU-02 と KSU-04 および KSU-05 と KSU-06 をそれぞれ比較することで, 各トークナイザが生成要約へ与えた影響について考察する.

表 3.8 より, トークナイザとして SentencePiece を用いることで, content および total のスコアは増加するが, well-formed のスコアが減少することがわかる. これは SentencePiece によって未知語へ対応できた一方, 不自然な文法を出力する可能性が増加したためと考えられる. また例外の正解を含めない content のスコア ($\alpha = 0$) と比較して, 例外の正解を含めた content のスコア ($\alpha = 2$) は, SentencePiece を導入することでスコアが最大 2 倍以上改善していることがわかる. content についての詳細な内訳を調査したところ, 正解要約とは異なる内容の正解数が増加したことを確認できた. これにより総合的な要約の評価である total についてもスコアが改善したと考えられる.

表 3.9 より, SentencePiece を用いることで ROUGE-2 (N2) が一貫して改善することがわかる. 人手による評価において content のスコアが改善したことから, 他の ROUGE についても精度改善が期待されたがその通りにはならなかった. これは正解要約が 1 種類しかなかったためと考えられる.

3.5.2 多様性機構の効果

次に KSU-01 と KSU-02 および KSU-03 と KSU-04 をそれぞれ比較することで, Diversity Cell が生成要約へ与えた影響について考察する.

表 3.8 より, Diversity Cell を用いることで, well-formed のスコアが増加したことがわかる. 実際に生成された要約について調査したところ, RNN 特有の同じ語句の繰り返し生成問題が改善されたことを確認できた. 一方で, content のスコアが減少したことから, 必ずしも各予測単語ベクトルが直交の関係にあるべきではないといえる. また total のスコアについては一貫的な増減を確認できなかった.

表 3.9 より, Diversity Cell を用いることで, ROUGE-1 (N1) 以外の全ての ROUGE が一貫して低下することがわかる. この結果は, content のスコアが減少した表 3.8 の結果と符合する.

3.5.3 出力長制御の効果

最後に KSU-01 と KSU-05 および KSU-03 と KSU-06 をそれぞれ比較することで, LenEmb が生成要約へ与えた影響について考察する.

表 3.8 より, LenEmb を用いることで, well-formed のスコアが減少することがわかる. また content および total のスコアについては一貫した増減を確認できなかった. 実際に生成された要約を確認したところ, 出力内容が残りの制限字数に大きく影響を受けていることがわかった. つまり残りの制限字数によって生成する要約が決定される傾向があり, 生成要約の内容がトピックに従っていないと考えられる. 一方で LenEmb を用いないモデルと比較して, 出力した要約の出

力長は指定された字数に近い値を取ることが多いことがわかった。この結果より、LenEmb は期待された能力を発揮できたことがわかる。

表 3.9 より、LenEmb を用いることで、ROUGE-1 (N1) , ROUGE-2 (N2) および ROUGE-SU4 (SU4) が一貫して改善することがわかる。LenEmb により出力要約の出力長が増加することから、Recall に基づく ROUGE-1 および ROUGE-2 が改善することは自然な結果である。一方で ROUGE-SU4 についても改善したことから、出力要約の出力長が増加しつつも不必要な語句を多く出力しなかったことがわかる。

3.6 結言

本章では、地方議会会議録中の発言を対象とした要約生成を、クエリ指向テキスト要約の枠組みで取り組んだ。具体的には、クエリを入力できるよう拡張した Encoder-Decoder に、多様性制御や出力長制御を目的とした機構を組み合わせることで、複数の要約モデルを実装した。

各モデルが生成した要約に対して、自動評価指標と人手による評価指標のそれぞれで精度を比較した。その結果、BPE に基づくトークナイザをモデルに導入することによって、人手による評価指標における総合的な要約の品質スコアが改善されることを確認した。一方、これまで人手による評価と相関性が高いとされていた、いくつかの自動評価指標においては、辞書に基づくトークナイザを導入したモデルが最大精度となることがわかった。人手による評価と自動評価の結果が乖離した原因として、本実験において正解要約が1種類しか存在せず、複数の正解要約から算出できる ROUGE の強みを発揮できなかったためと考えられる。

また、生成要約の出力長や生成要約の内容の多様性を制御する機構が、生成要約の品質の低下に繋がることが確認された。これは、生成アプローチを用いたクエリ指向要約が抱える課題として指摘されている、入力文書の意味する内容から逸脱した内容の要約を出力する現象が生じたためと考えられる。そのため、本研究では、Summarization Task の主目的である、発言の意図を歪めない要約手法については、課題を残す結果となった。そこで、今後の研究として、多様性機構や注意機構の改善などを進めることで、原文の内容と生成要約の内容の一貫性保持を目指した要約手法について考案する予定である。

第4章 文エンコーダを追加した クエリ指向テキスト要約

4.1 緒言

インターネットの継続的な発展に伴い、テキスト、音声、画像、動画のような非構造データは増加し、いまやビッグデータという名で広く認知されている。国際的なデジタルデータの総量は、2020年には約40ゼタバイトへ拡大すると報告されている[63]。このような巨大なデータ群から必要な情報のみを抽出、収集する際、全てのデータに目を通すのは現実的ではない。そのため、検索エンジンやニュース配信サービスでは、原文の要約であるスニペットやリード文を提供することが多い。以上の背景から、与えられたテキストを自動で要約する研究は盛んに取り組まれている。

汎用テキスト要約は、特定の観点を想定しないテキスト要約である。これは単に、原文の概要を表現する要約の出力を目的としている。一方、クエリ指向テキスト要約は、ユーザから与えられたクエリが示す特定の観点に沿った要約の出力を目的としている。汎用テキスト要約は、原文を構成する内容を損なわずに要約する必要がある。そのため先行研究では、いかに原文の重要文を重複なく抽出、圧縮、生成できるかに焦点が当てられてきた。例えば、原文中の文やフレーズを組み合わせて要約を出力する抽出型では、原文の文頭や接続詞による言い換え表現のような、原文の談話構造に関連する手掛かりを活用している。一方、クエリ指向テキスト要約は、ユーザが与えたクエリに応じて要約に含めるべき文の種類は変化する。そのため、これまで提案された手法では、クエリと原文の各文の関連性を TF-IDF などにより重み付けすることで、要約の候補となる重要文を抽出している。

近年、Rushら[3]が提案したニューラル言語モデルを用いた汎用テキスト要約が一定の成功を収めたことで、ニューラル言語モデルを用いたテキスト要約はより活発に研究されている。提案されたモデルは、Sequence to Sequence[28]という機械翻訳において提案されたモデルに注意機構[53]を備えている。注意機構の導入により、入力された原文中の各トークンの重要度を、デコード毎に更新し要約を出力できる。また、クエリ指向テキスト要約では、Nemaら[40]が深層モデルに基づく生成型の Encoder-Decoder モデルを提案している。このモデルでは、入力の原文の各トークンベクトルに加えて、入力のクエリにも注意機構を用いることで、各ステップにおける重要なクエリを出力に反映させている。また、同じフレーズを繰り返し出力する Recurrent Neural Network (RNN) 固有の問題を解決するために、注意機構によって生成されるベクトルが、各ステップにおいてハードまたはソフトに直交するよう変換する手法を提案している。

ニューラル言語モデルを用いたテキスト要約手法では、長期的な情報の記憶が可能となる Long-Short Term Memory (LSTM) [32, 33] と、RNN でエンコードした各ステップのベクトルから特定のステップの重要視を可能とする注意機構が欠かせない。しかし翻訳タスクの実験において、60 トークンよりも長い文書を LSTM でエンコードした場合に翻訳の品質が低下すると Koehnら[64]

は報告している。要約タスクにおいても、長文のエンコードが失敗したり、文の関係性の消失を引き起こすことは、要約結果の品質を大きく低下させる要因として問題であると考えられる。

そこで筆者は、原文のトークン単位のベクトルに加え、原文の文単位のベクトルを導入した要約生成手法を提案する。また、トークン単位の注意機構と文単位の注意機構を組み合わせて学習させることで、文単位の重要度と文間の関係性を考慮した要約生成を目指す。これによりトークン数が多い原文が入力された場合でも、提案手法は頑健に働くことが期待される。実験では、最新のクエリ指向テキスト要約のためのニューラル言語モデルに、文単位ベクトルの機構を追加した要約モデルについて検証し、文ベクトルの有無が生成要約に与える影響を明らかにする。

以降、4.2節では、研究対象となるクエリ指向テキスト要約の問題の定式化について、4.3節では、ベースラインとなる最新のクエリ指向テキスト要約手法について、4.4節では、提案手法の目的とベースラインを拡張したモデルの構成について、4.5節では、実験の結果とその考察について、4.7節では、結言について述べる。

4.2 問題の定式化

ここではクエリ指向テキスト要約が対象とするタスクを定義する。

まず、クエリ指向の要約タスクの入力は、一般に次の2つから構成される。トークン t のシーケンスから構成される原文 $\mathbf{d}^{token} = t_1^d, t_2^d, \dots, t_{l_d}^d$ 、および、トークン t のシーケンスから構成されるクエリ $\mathbf{q} = t_1^q, t_2^q, \dots, t_{l_q}^q$ である。ここで、原文 \mathbf{d} は1文以上を含む文の集合体である。またトークン t には、単語単位、文字単位、サブワード単位のいずれかを用いることが多い。本章では、トークン t として単語単位を選択した。そのためトークン t は、 $t \in \mathbb{R}^{\delta_1}$ と表せる。ここで、 δ_1 は語彙数である。

次に本提案モデルでは、これらに加えて3つめの入力として次の1つを定義する。文 \mathbf{s} のシーケンスから構成される原文 $\mathbf{d}^{sentence} = \mathbf{s}_1^d, \mathbf{s}_2^d, \dots, \mathbf{s}_{l_s}^d$ である。ここで、 \mathbf{s}_m は1文のみから構成される、トークン t のシーケンス $\mathbf{s}_m = t_1^{s_m}, t_2^{s_m}, \dots, t_{l_{s_m}}^{s_m}$ である。

最後に、出力は次の1つから構成される。トークン t のシーケンスから構成される要約 $\mathbf{o} = t_1^o, t_2^o, \dots, t_{l_o}^o$ である。またモデルの出力要約を \mathbf{o}^{system} 、正解要約を $\mathbf{o}^{reference}$ とする。

以上を踏まえ、クエリ指向の生成要約を以下のように定義する。トークン数 l_d 、文数 l_s の原文 \mathbf{d} と、トークン数 l_q のクエリ \mathbf{q} が与えられたとき、原文 \mathbf{d} より短いトークン数 $l_{o^{system}} (< l_d)$ で構成され、クエリ \mathbf{q} について要約した出力 \mathbf{o}^{system} を生成すること。

4.3 Diversity driven Attention Model

ここでは、Nema らがいくつか提案した Diversity driven Attention Model[40] のうち、特に精度が良かった SD_2 について説明する。本章ではこのモデルをベースラインモデルとして扱う。

Diversity driven Attention Model を図 4.1 に示す。

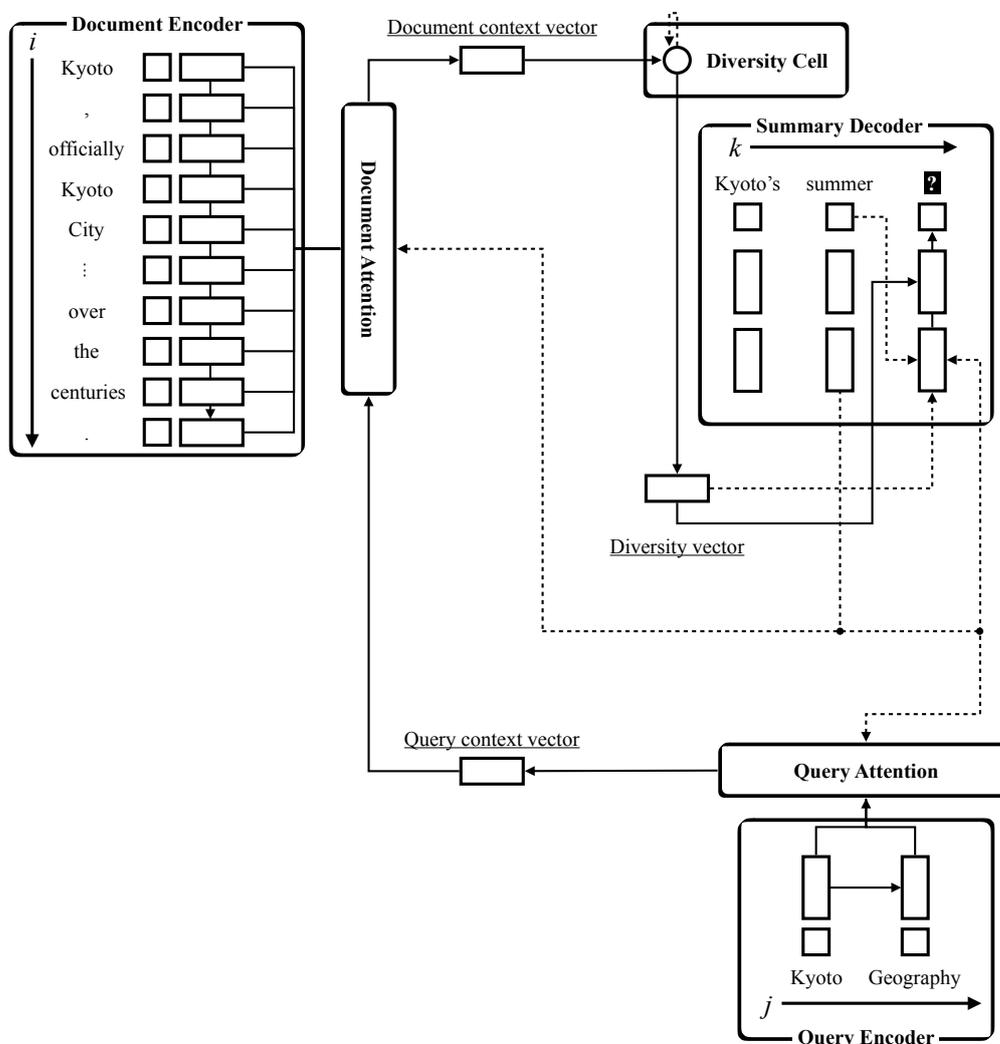


図 4.1: Diversity driven Attention Model. 2つのエンコーダ、および、それぞれの注意機構と、1つのデコーダ、および、1つの Diversity Cell から構成される。破線はデコードステップ $k-1$ を表現し、実線はデコードステップ k を表現する。

このモデルは、原文とクエリを特徴ベクトルへ変換する2つのエンコーダ、それぞれに対応する2つの注意機構、アテンションされた原文の特徴ベクトルに多様性を与える1つの Diversity Cell、そして、要約を生成する1つのデコーダから構成される。以降、モデルの各層についてデータの処理順に従って説明する。

4.3.1 Word Representation

1つのトークン t を特徴ベクトルへ変換する δ_2 次元の word embed を $e(t)$ で表す。この word embed は、モデルの内部の全てのエンコーダとデコーダで共有される。

4.3.2 Document Encoder

この機構の役割は、入力である1つのトークンシーケンスで構成される原文 $\mathbf{d}^{token} = t_1^d, t_2^d, \dots, t_{l_d}^d$ を、各エンコードステップ i における特徴ベクトル h_i^d へ変換することである。このエンコーダは、RNNとして Gated Recurrent Units (GRU) [31] を用いて次のように表す。

$$h_i^d = \mathbf{GRU}_d(h_{i-1}^d, e(t_i^d)), \quad (4.1)$$

ここで、 h_i^d は δ_3 次元の特徴ベクトルである。

4.3.3 Query Encoder

この機構の役割は、入力である1つのトークンシーケンスで構成されるクエリ $\mathbf{q} = t_1^q, t_2^q, \dots, t_{l_q}^q$ を、各エンコードステップ j における特徴ベクトル h_j^q へ変換することである。このエンコーダは、RNNとして GRU を用いて次のように表す。

$$h_j^q = \mathbf{GRU}_q(h_{j-1}^q, e(t_j^q)), \quad (4.2)$$

ここで、 h_j^q は δ_4 次元の特徴ベクトルである。

4.3.4 Query Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_j^q の各エンコードステップ j と 4.3.7 項で説明する Summary Decoder が出力する特徴ベクトル h_{k-1}^o から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル q_k を生成することである。

デコードステップ k のクエリのコンテキストベクトル q_k は次の式で算出される。

$$a_{k,j}^q = v_q^\top \tanh(W_q h_{k-1}^o + U_q h_j^q), \quad (4.3)$$

$$\alpha_{k,j}^q = \frac{\exp(a_{k,j}^q)}{\sum_{j'=1}^{l_q} \exp(a_{k,j'}^q)}, \quad (4.4)$$

$$q_k = \sum_{j=1}^{l_q} \alpha_{k,j}^q h_j^q, \quad (4.5)$$

ここで、 $W_q \in \mathbb{R}^{\delta_4 \times \delta_5}$, $U_q \in \mathbb{R}^{\delta_4 \times \delta_4}$, $v_q \in \mathbb{R}^{\delta_4}$, q_k は δ_4 次元の特徴ベクトル, h_{k-1}^o は δ_5 次元の特徴ベクトルである。

4.3.5 Document Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_i^d の各エンコードステップ i , Summary Decoder が出力する特徴ベクトル h_{k-1}^o , および、クエリのコンテキストベクトル q_k から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル d_k を生成することである。

d_k の式は、 q_k を受け取るパラメータ $Z_q \in \mathbb{R}^{\delta_3 \times \delta_4}$ を持つ。デコードステップ k のコンテキストベクトル d_k は次の式で算出される。

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_q q_k), \quad (4.6)$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \quad (4.7)$$

$$d_k = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d, \quad (4.8)$$

ここで、 $W_d \in \mathbb{R}^{\delta_3 \times \delta_5}$, $U_d \in \mathbb{R}^{\delta_3 \times \delta_3}$, $v_d \in \mathbb{R}^{\delta_3}$, d_k は δ_3 次元の特徴ベクトルである。

4.3.6 Diversity Cell

この機構の役割は、各デコードステップ k において、同じトークンを繰り返し生成する RNN の問題点を解決することである。そこで Nema らは、LSTM の実装を拡張し、各デコードステップ k のコンテキストベクトル d_k を互いに直交するベクトル d'_k へ変換する機構 **SD**₂ を次の式で定義した。

$$\begin{pmatrix} i_k \\ f_k \\ o_k \\ c'_k \\ g_k \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} d_k \\ h_{k-1}^{dc} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \quad (4.9)$$

$$c_k = \sigma(i_k) \odot \tanh(c'_k) + \sigma(f_k) \odot c_{k-1}, \quad (4.10)$$

$$c_k^{diverse} = c_k - \sigma(g_k) \frac{c_k^\top c_{k-1}}{c_{k-1}^\top c_{k-1}} c_{k-1}, \quad (4.11)$$

$$h_k^{dc} = \sigma(o_k) \odot \tanh(c_k^{diverse}), \quad (4.12)$$

$$d'_k = h_k^{dc}, \quad (4.13)$$

ここで、 $W_i, W_f, W_o, W_g, W_c \in \mathbb{R}^{\delta_3 \times \delta_3}$, $U_i, U_f, U_o, U_g, U_c \in \mathbb{R}^{\delta_3 \times \delta_3}$, d'_k は、 δ_3 次元のベクトルである。

4.3.7 Summary Decoder

この機構の役割は、デコードステップ $k-1$ のコンテキストベクトル d'_{k-1} と、デコードステップ $k-1$ のトークン t_{k-1}^o を入力とし、デコードステップ k における特徴ベクトル h_k^o を出力することである。

$$h_k^o = \mathbf{GRU}_o(h_{k-1}^o, [e(t_{k-1}^o); d'_{k-1}]), \quad (4.14)$$

ここで、 $[e(t_{k-1}^o); d'_{k-1}]$ は2つのベクトルの concatenate 演算を示す。

その後、特徴ベクトル h_k^o とコンテキストベクトル d'_k から、デコードステップ k におけるトークン t_k^o を予測する。

$$t_k^o = \text{softmax}(Wf(W_{dec}h_k^o + V_{dec}d'_k)), \quad (4.15)$$

ここで、 $W \in \mathbb{R}^{\delta_1 \times \delta_2}, W_{dec} \in \mathbb{R}^{\delta_2 \times \delta_5}, V_{dec} \in \mathbb{R}^{\delta_2 \times \delta_3}$ である。また、活性化関数 f は恒等関数である。

4.4 提案手法

筆者は、複数の文から構成された原文 \mathbf{d} を、単一のトークンシーケンスとして扱うことによって、トークン数の多い入力のエンコードの失敗や文の関係性の消失を引き起こしている点を既存モデルの問題と捉え、文単位のシーケンス $\mathbf{d}^{sentence}$ を追加したモデルを提案する。

ここでは4.3節で説明したモデルに、文エンコーダとその注意機構を追加した提案手法について説明する。まずエンコードステップにおける Sentence Encoder の構成について述べ、次にデコードステップにおける文単位の注意機構の構成について詳述する。最後に4.3節で説明したモデルに本提案手法を導入するにあたって拡張が必要な関数について明記する。

4.4.1 Sentence Representation

本提案手法では、Sentence Representation として Skip-Thought[22] を用いた。このモデルは、エンコードした m 番目の文から $m+1$ 番目の文と $m-1$ 番目の文をデコードするよう学習するモデルである。この学習により、Skip-Thought のエンコーダが生成する特徴ベクトルは、自身の前後の文との関係や情報を保持することが期待される。ここでは、 m 番目の文 \mathbf{s}_m における、エンコードステップ n のエンコーダの式についてのみ下記に示す。

$$\vec{eh}_{m,n}^s = \text{GRU}_{skip}^{forward}(\vec{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (4.16)$$

$$\overleftarrow{eh}_{m,n}^s = \text{GRU}_{skip}^{backward}(\overleftarrow{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (4.17)$$

$$eh_{m,n}^s = [\vec{eh}_{m,n}^s; \overleftarrow{eh}_{m,n}^s], \quad (4.18)$$

$$eh_m^s = eh_{m,l_{s_m}}^s, \quad (4.19)$$

ここで、 $\vec{eh}_{m,n}^s$ は入力シーケンスを順方向に入力していることを、 $\overleftarrow{eh}_{m,n}^s$ は入力シーケンスを逆方向に入力していることを示す。また、 $eh_{m,n}^s$ は、 δ_6 次元の最終ステップ l_{s_m} の特徴ベクトルである。

4.4.2 Sentence Encoder

トークン単位のエンコーダである Document Encoder では、まず Word Representation の手法によって入力された各トークンを分散表現へ変換し、その後、RNN を通して特徴ベクトル h_i^d へ変換する。この処理により、Word Representation のみでは捉え切ることが困難であるトークンの順序が持つ意味を、特徴ベクトル h_i^d として表現することが可能となる。同様に、Sentence Representation の手法によってエンコードされた文単位の特徴ベクトル $eh^s = eh_1^s, \dots, eh_m^s, \dots, eh_{l_s}^s$ について、筆者は文の分散表現を直接用いるのではなく、RNN を通して特徴ベクトル h_m^s へ変換するべきと考えた。この処理により、文の順序が持つ意味を特徴ベクトル h_m^s として表現できることが期待される。そこで本提案手法では、文単位のエンコーダである Sentence Encoder を導入す

る。このエンコーダは、RNNとして双方向 Gated Recurrent Units (BiGRU) を選択し、次のように表す。

$$\vec{h}_m^s = \mathbf{GRU}_d(\vec{h}_{m-1}^s, eh_m^s), \quad (4.20)$$

$$\overleftarrow{h}_m^s = \mathbf{GRU}_d(\overleftarrow{h}_{m-1}^s, eh_m^s), \quad (4.21)$$

$$h_m^s = [\vec{h}_m^s; \overleftarrow{h}_m^s], \quad (4.22)$$

ここで、 h_m^s は δ_7 次元の特徴ベクトルである。

4.4.3 Sentence Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_m^s の各エンコードステップ m から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル s_k を生成することである。デコードステップ k のコンテキストベクトル s_k は、次の式で算出される。

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s), \quad (4.23)$$

$$\alpha_{k,m}^s = \frac{\exp(a_{k,m}^s)}{\sum_{m'=1}^{l_s} \exp(a_{k,m'}^s)}, \quad (4.24)$$

$$s_k = \sum_{m=1}^{l_s} \alpha_{k,m}^s h_m^s, \quad (4.25)$$

ここで、 $W_s \in \mathbb{R}^{\delta_7 \times \delta_5}$, $U_s \in \mathbb{R}^{\delta_7 \times \delta_7}$, $v_s \in \mathbb{R}^{\delta_7}$, s_k は δ_7 次元の特徴ベクトルである。

4.4.4 提案手法による既存手法の拡張

提案手法によって拡張された Diversity driven Attention Model を図 4.2 に示す。

クエリ指向テキスト要約では、クエリに沿った要約を出力する必要があるため、原文の各トークンや文の重要度はクエリに応じて変化する。そのため Nema らの手法では、クエリのコンテキストベクトル q_k に応じて、トークン単位のコンテキストベクトル d_k が決定される。文単位の原文を扱える提案手法では、クエリから原文の各トークンの重要度を決定するプロセスを細分化し、 q_k に応じて文単位のコンテキストベクトル s_k を生成し、 s_k に応じて d_k を生成することで、原文から選択した文を元に要約を生成する人間の要約生成プロセスにより近いモデルの構築を目指した。そこでデコードステップ k における文単位の注意機構を、式 (4.23) から式 (4.26) へ、以下の式に変更し、導入することとした。また同様に、トークン単位の注意機構は、式 (4.6) から式 (4.27) へ変更することとした。

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s + Z_q q_k), \quad (4.26)$$

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_q s_k). \quad (4.27)$$

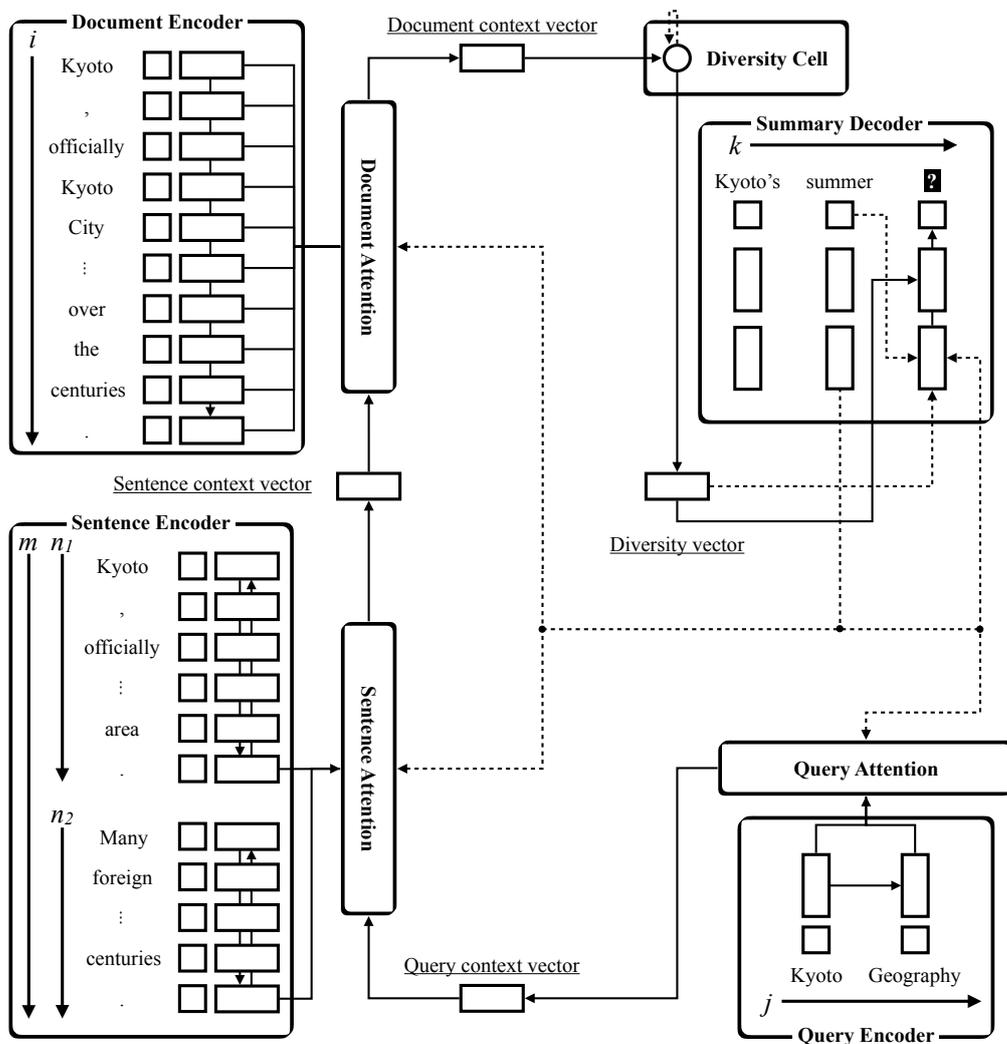


図 4.2: 提案手法によって拡張された Diversity driven Attention Model. 3つのエンコーダ, および, それぞれの注意機構, 1つのデコーダ, および, 1つの Diversity Cell から構成される. 破線はデコードステップ $k-1$ を表現し, 実線はデコードステップ k を表現する.

4.5 実験

本実験の目的は、提案手法である文ベクトルを考慮した要約生成モデルと、ベースラインとなる文ベクトルを用いない要約生成モデルのそれぞれが出力した要約を評価、比較することで、文ベクトルの導入が出力した要約へ与えた影響について明らかにすることである。提案手法と比較するモデルとして、4.3節で説明した Diversity driven Attention Model を用いた。

4.5.1 データセット

要約の目標を含む要約タスクのデータセットは、あまり多いとは言えない [5, 6]。そのため、筆者は、Wikipedia[7] と Simple English Wikipedia[8] を用いて、データセットを作成した。Simple English Wikipedia は、Wikipedia の英語版であり、基本的にベーシック英語とスペシャル・イングリッシュで記述された、語彙が簡略化されたウェブ百科事典である。作成したデータセット中のデータは、 (D, Q, S) のタプル形式で構成される。ここで、 D をあるタイトルを持つ Wikipedia 一つの記事、 Q を D に対応する Simple English Wikipedia のあるセクションのタイトル、 S を Q に対応する Simple English Wikipedia 内の文書とする。セクションのタイトルには、対象のセクション自身が持つタイトルの他に、Wikipedia 記事の階層構造に沿った、そのセクションの親となるタイトル全てを含む。例えば、ある Simple English Wikipedia 記事の階層構造が、最上位の階層から「記事タイトル: Person Name/セクション: Abstract/セクション: Personal life/サブセクション: Education」となっている場合、対象のセクションが「Personal life」であった場合、 S には「記事タイトル: Person Name/セクション: Personal life」が含まれる。

Wikipedia の全ての記事数が 5,570,022 件であるのに対して、Simple English Wikipedia の全ての記事数は 131,459 件と 2.4%程度しか存在しないため、Wikipedia と同じ記事タイトルを持つ記事、107,168 件のみを取得した。本実験では、上記の方法で取得したペアデータの 75,000 件からデータセットを作成した。データセット中の訓練データ、開発データおよび評価データの内訳について表 4.1 に示す。なおデータセットには、 D 、 Q および S の各トークン数のヒストグラムを参考

表 4.1: English Wikipedia と Simple English Wikipedia から作成したデータの件数。

データセット	件数
訓練データ	29,389
開発データ	6,300
評価データ	6,300

に、各トークン数がそれぞれ、4,500、6 および 300 以下の記事のみを含めることとした。

4.5.2 評価指標

本実験の目的は、文ベクトルを考慮した要約生成モデルと、文ベクトルを用いない要約生成モデルを比較することで、文ベクトルが、出力した要約へ与えた影響について明らかにすることで

ある。各モデルについて、要約モデルが出力したシステム要約と正解要約間の ROUGE-N ($N = 1, 2$) および ROUGE-L[61] により評価した。

4.5.3 モデルの設定と学習の詳細

本実験で用いた提案手法のモデルとベースライン手法のモデルの各パラメータについて述べる。まず D , Q および S の語彙の次元を 131,718, 単語埋め込みベクトルの次元を 128, エンコーダの隠れ層の次元を 128, 注意機構の隠れ層の次元を 128, デコーダの隠れ層の次元を 128 とした。また提案手法で用いた文エンコーダの次元数と 256 とした。語彙については出現頻度が 20 以上のトークンを用いた。よって最適化対象のパラメータ数は、提案手法のモデルでは、35,219,721, ベースライン手法のモデルでは、34,478,856 となった。各モデルは、最適化手法として Adam[60] ($\alpha=0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\text{eps} = 10^{-8}$) を用い、コア数 3, ミニバッチサイズ 4, エポック数 10 で学習した。また、word embed および Skip-Thought は、それぞれ事前学習させたパラメータを用いた。モデルの実装には、Chainer[55, 56], および、ChainerMN[57, 58] を用いた。また形態素解析器として、The Stanford CoreNLP[65] を用いた。

4.5.4 結果

実験の結果を表 4.2 に示す。

表 4.2: Wikipedia データセットに対する各モデルの ROUGE による比較。

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
提案モデル	33.55	21.01	32.05
比較モデル	33.48	20.97	31.75

表 4.2 より、すべての ROUGE において、既存手法の精度より提案手法の精度がわずかに高いことがわかる。

4.6 考察

各手法の出力例を表 4.3 および表 4.4 に示す.

表 4.3: 出力された要約文の一例 (記事タイトル: Thanjavur district, クエリ: Thanjavur district)

クエリ: Thanjavur district
正解要約: thanjavur district is one of the three two district in tamil nadu , india . its headquarters is thanjavur city . this district is known as the “ granary of south india ” and “ rice bowl of tamil nadu ” . this is one of the important district for the tamil nadu agriculture and economy . <eos>
提案モデル: thanjavur district is one of the three two districts of the state of tamil nadu , in southeastern india . its headquarters is thanjavur . the district is in the delta of the cauvery river and is mostly agrarian . as of two zero one one taluk of the taluk is the main rice producing region in the state and hence known as the rice bowl of tamil nadu . <eos>
比較モデル: kanyakumari district is one of the three two districts of the state of tamil nadu , in southeastern india . its headquarters is the headquarters of the district of <unk> . the district is in the delta of the cauvery river and is mostly educated at the river <unk> , in the delta of the cauvery river and is mostly educated at the river <unk> , in the indian state of tamil nadu , in southeastern india . the district is in the delta of the cauvery river and is mostly educated . <eos>

表 4.3 では、インドに存在する「Thanjavur district」についての English Wikipedia の記事の正解要約と各モデルが生成したシステム要約である。この例では、クエリとして「Thanjavur district」のみが与えられたため、提案手法では、概要についての要約文を出力していることが確認できる。一方、既存手法では、インドに存在する「Kanyakumari」についての概要を生成している。この現象は、既存手法に限った問題ではないが、この例においては、既存手法がより過学習を起していることがわかる。

表 4.4 では、表 4.3 と同じく、インドに存在する「Thanjavur district」についての English Wikipedia の記事の正解要約と各モデルが生成したシステム要約である。ただし、この例では、「Thanjavur district」の他に、クエリとして「agriculture and tourism」がモデルに与えられる。既存手法では、識字率や人口について要約しているため適しているとは言い難い。一方、提案手法では、有名な米産地であるタミルナドゥの井について要約できている。ただし、人口についても触れているため、完全な要約とは言えない。

表 4.4 から、既存手法において、モデルが同一のフレーズの繰り返しを生成することが確認できる。これは、既存手法に限らず、提案手法についても同様の問題が発生している。Diversity cell により、同じトークンを繰り返し生成する問題は改善されたが、少なくとも SD_2 を用いた場合、長い文を生成したとき、繰り返しの周期が、トークン単位から文脈単位へスケールアップしたと考えられる。

4.7 結言

本章では、原文の単語単位のベクトルに加え、文単位のベクトルを導入し、要約を生成する手法を提案した。実験により、わずかではあるが、提案手法によって ROUGE が改善したことを確認できた。一方、各モデルが生成した要約において、文またはフレーズの単位で同じ内容が繰り返して生成されることが確認できた。この問題を解決するため、単語単位の多様性を与える Diversity Cell を改良した、文単位の多様性を与える機構を追加する予定である。また本章では、独自に構築したデータセットのみで提案手法の精度を検証した。今後は、Nema らが構築したデータセットにおいても比較実験をすることで、提案手法の一貫した精度向上を確認できるか調査を進める予定である。

第5章 原文を俯瞰する機構を備えた クエリ指向テキスト要約

5.1 緒言

検索エンジンとクエリによってインターネット空間から得られる検索結果のランキングは、刹那的にパーソナライズされたインターネット空間へ繋がるホームページといえる。近年では、検索エンジンの検索結果に、スニペットと呼ばれる Web ページの概要が付随していることが当たり前となってきた。ユーザはスニペットを読むことにより、最小限の時間で検索結果から必要な Web ページのみを効率よく選定できる。このようにテキスト要約は、我々の生活になくてはならない研究分野といえる。

Google[1]では、ユーザが与えたクエリと関連性が高い部分をページコンテンツから抜粋することで、スニペットを自動的に生成している [66]。このような、ユーザから与えられたクエリが示す、特定の観点に沿った要約の出力を目的とするテキスト要約をクエリ指向テキスト要約という。

これまで筆者は、ニューラル言語モデルを用いたクエリ指向テキスト要約について研究を進めてきた。主に、生成型のテキスト要約では、要約の対象である原文を元に、要約中の語句を確率的に予測することで最終的な要約を生成する。一方、抽出型のテキスト要約は、原文の語句を抽出し組み合わせることで要約を出力する。抽出型と比べ生成型は、言い換え表現の獲得のような、原文に存在しない語句や言い回しを出力することが可能である。また、注意機構 [53] の登場により、原文の各トークンの重要度を反映した要約生成が可能となった。しかし、出力された要約内容が原文の内容から逸脱する課題があると指摘されている。

筆者は、生成する要約内容の一貫性を保つためには、原文全体で表現される特徴を捉える必要があると考えた。現在、ニューラル言語モデルとして主流となっている注意機構を備えた Encoder-Decoder[28]において、もし原文を俯瞰するベクトルを生成したいのであれば、注意機構で算出される原文中の全トークンの重要度を同一にする必要がある。このような重要度の算出結果は、特定のトークンへの注意を目的とする注意機構にとって期待されていない動作である。

そこで本研究は、入力の全体を表現するエンコーダの最終出力のベクトルを使用するか、それとも、注意機構により特定のエンコードステップが強調されたベクトルを使用するかを、ゲート機構によって適応的に制御する機構を提案する。この手法により、デコードの状況に応じて、入力の全体を俯瞰するか、入力の重要なトークンに焦点を合わせるかを、適切に切り替えるモデルの構築を目指した。実験では、適応的な注意機構の有無が生成要約に与える影響を明らかにするために、最新のクエリ指向テキスト要約のニューラル言語モデルおよび第4章で提案したニューラル言語モデルのそれぞれに、提案手法を導入し、実験によって各モデルの精度を比較した。

以降、5.2節では、研究対象となるクエリ指向テキスト要約の問題の定式化について、5.3節では、ベースラインとなる第4章で提案したクエリ指向テキスト要約手法について、5.4節では、提

案手法の目的とベースラインを拡張したモデルの構成について、5.5節では、実験の結果とその考察について、5.6節では、結言について述べる。

5.2 問題の定式化

ここではクエリ指向テキスト要約が対象とするタスクを定義する。

まず、クエリ指向の要約タスクの入力は、一般に次の2つから構成される。トークン t のシーケンスから構成される原文 $\mathbf{d}^{token} = t_1^d, t_2^d, \dots, t_{l_d}^d$ 、および、トークン t のシーケンスから構成されるクエリ $\mathbf{q} = t_1^q, t_2^q, \dots, t_{l_q}^q$ である。ここで、原文 \mathbf{d} は1文以上を含む文の集合体である。またトークン t には、単語単位、文字単位、サブワード単位のいずれかを用いることが多い。本章では、トークン t として単語単位を選択した。そのためトークン t は、 $t \in \mathbb{R}^{\delta_1}$ と表せる。ここで、 δ_1 は語彙数である。

次に本提案モデルでは、これらに加えて3つめの入力として次の1つを定義する。文 \mathbf{s} のシーケンスから構成される原文 $\mathbf{d}^{sentence} = \mathbf{s}_1^d, \mathbf{s}_2^d, \dots, \mathbf{s}_{l_s}^d$ である。ここで、 \mathbf{s}_m は1文のみから構成される、トークン t のシーケンス $\mathbf{s}_m = t_1^{s_m}, t_2^{s_m}, \dots, t_{l_{s_m}}^{s_m}$ である。

最後に、出力は次の1つから構成される。トークン t のシーケンスから構成される要約 $\mathbf{o} = t_1^o, t_2^o, \dots, t_{l_o}^o$ である。またモデルの出力要約を \mathbf{o}^{system} 、正解要約を $\mathbf{o}^{reference}$ とする。

以上を踏まえ、クエリ指向の生成要約を以下のように定義する。トークン数 l_d 、文数 l_s の原文 \mathbf{d} と、トークン数 l_q のクエリ \mathbf{q} が与えられたとき、原文 \mathbf{d} より短いトークン数 $l_{o^{system}} (< l_d)$ で構成され、クエリ \mathbf{q} について要約した出力 \mathbf{o}^{system} を生成すること。

5.3 Triple-Hierarchical Attention Model

ここでは、第4章で筆者が提案した Triple-Hierarchical Attention Model モデルについて説明する。

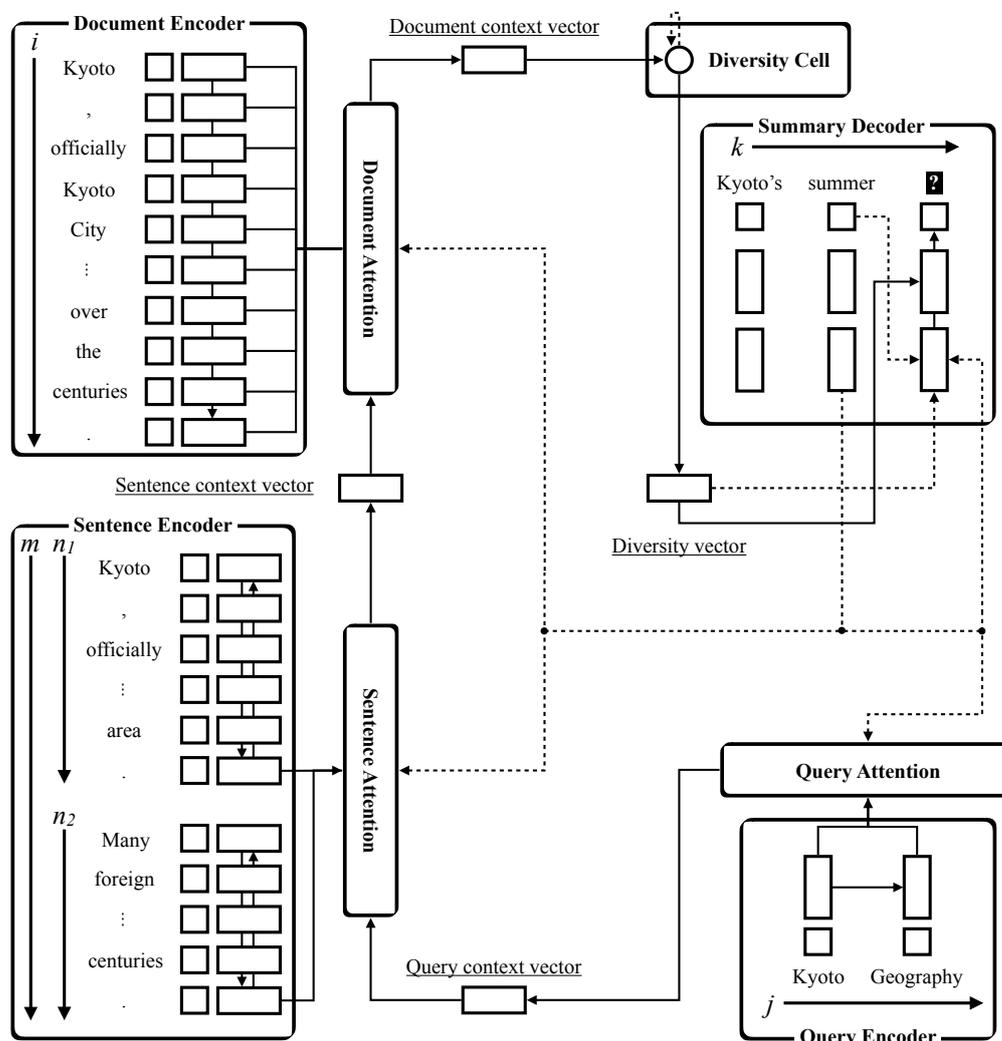


図 5.1: Triple-Hierarchical Attention Model. 3つのエンコーダ、および、それぞれの注意機構と、1つのデコーダ、および、1つの Diversity Cell から構成される。破線はデコードステップ $k-1$ を表現し、実線はデコードステップ k を表現する。各注意機構は階層的に構成されている。

このモデルは、トークン単位の原文、文単位の原文およびクエリを特徴ベクトルへ変換する3つのエンコーダ、それぞれに対応する3つの注意機構、アテンションされたトークン単位の原文の特徴ベクトルに多様性を与える1つの Diversity Cell、そして、要約を生成する1つのデコーダから構成される。以降、モデルの各層についてデータの処理順に従って説明する。

5.3.1 Word Representation

1つのトークン t を特徴ベクトルへ変換する δ_2 次元の word embed を $e(t)$ で表す。この word embed は、モデルの内部の全てのエンコーダとデコーダで共有される。

5.3.2 Sentence Representation

この機構の役割は、1つの文 s を特徴ベクトルへ変換することである。

Sentence Representation として Skip-Thought[22] を用いた。このモデルは、エンコードした m 番目の文から $m+1$ 番目の文と $m-1$ 番目の文をデコードするよう学習するモデルである。この学習により、Skip-Thought のエンコーダが生成する特徴ベクトルは、自身の前後の文との関係や情報を保持することが期待される。ここでは、 m 番目の文 s_m における、エンコードステップ n のエンコーダの式についてのみ下記に示す。

$$\vec{eh}_{m,n}^s = \mathbf{GRU}_{skip}^{forward}(\vec{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (5.1)$$

$$\overleftarrow{eh}_{m,n}^s = \mathbf{GRU}_{skip}^{backward}(\overleftarrow{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (5.2)$$

$$eh_{m,n}^s = [\vec{eh}_{m,n}^s; \overleftarrow{eh}_{m,n}^s], \quad (5.3)$$

$$eh_m^s = eh_{m,l_{s_m}}^s, \quad (5.4)$$

ここで、 $\vec{eh}_{m,n}^s$ は入力シーケンスを順方向に入力していることを、 $\overleftarrow{eh}_{m,n}^s$ は入力シーケンスを逆方向に入力していることを示す。また、 $eh_{m,n}^s$ は、 δ_3 次元の最終ステップ l_{s_m} の特徴ベクトルである。

5.3.3 Document Encoder

この機構の役割は、入力である1つのトークンシーケンスで構成される原文 $\mathbf{d}^{token} = t_1^d, t_2^d, \dots, t_{l_d}^d$ を、各エンコードステップ i における特徴ベクトル h_i^d へ変換することである。このエンコーダは、Recurrent Neural Network (RNN) として Gated Recurrent Units (GRU) [31] を用いて次のように表す。

$$h_i^d = \mathbf{GRU}_d(h_{i-1}^d, e(t_i^d)), \quad (5.5)$$

ここで、 h_i^d は δ_4 次元の特徴ベクトルである。

5.3.4 Sentence Encoder

この機構の役割は、Sentence Representation の手法によってエンコードされた文単位の原文の特徴ベクトル $eh^s = eh_1^s, \dots, eh_m^s, \dots, eh_{l_s}^s$ を、各エンコードステップ m における特徴ベクトル h_m^s へ変換することである。このエンコーダは、RNN として双方向 Gated Recurrent Units (BiGRU) を選択し、次のように表す。

$$\vec{h}_m^s = \mathbf{GRU}_d(\vec{h}_{m-1}^s, eh_m^s), \quad (5.6)$$

$$\overleftarrow{h}_m^s = \mathbf{GRU}_d(\overleftarrow{h}_{m-1}^s, eh_m^s), \quad (5.7)$$

$$h_m^s = [\vec{h}_m^s; \overleftarrow{h}_m^s], \quad (5.8)$$

ここで、 h_m^s は δ_5 次元の特徴ベクトルである。

5.3.5 Query Encoder

この機構の役割は、入力である1つのトークンシーケンスで構成されるクエリ $\mathbf{q} = t_1^q, t_2^q, \dots, t_{l_q}^q$ を、各エンコードステップ j における特徴ベクトル h_j^q へ変換することである。このエンコーダは、RNN として GRU を用いて次のように表す。

$$h_j^q = \mathbf{GRU}_q(h_{j-1}^q, e(t_j^q)), \quad (5.9)$$

ここで、 h_j^q は δ_6 次元の特徴ベクトルである。

5.3.6 Query Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_j^q の各エンコードステップ j と 5.3.10 項で説明する Summary Decoder が出力する特徴ベクトル h_{k-1}^o から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル q_k を生成することである。デコードステップ k のクエリのコンテキストベクトル q_k は次の式で算出される。

$$a_{k,j}^q = v_q^\top \tanh(W_q h_{k-1}^o + U_q h_j^q), \quad (5.10)$$

$$\alpha_{k,j}^q = \frac{\exp(a_{k,j}^q)}{\sum_{j'=1}^{l_q} \exp(a_{k,j'}^q)}, \quad (5.11)$$

$$q_k = \sum_{j=1}^{l_q} \alpha_{k,j}^q h_j^q, \quad (5.12)$$

ここで、 $W_q \in \mathbb{R}^{\delta_6 \times \delta_7}$, $U_q \in \mathbb{R}^{\delta_6 \times \delta_6}$, $v_q \in \mathbb{R}^{\delta_6}$, q_k は δ_6 次元の特徴ベクトル, h_{k-1}^o は δ_7 次元の特徴ベクトルである。

5.3.7 Sentence Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_m^s の各エンコードステップ m , Summary Decoder が出力する特徴ベクトル h_{k-1}^o およびクエリのコンテキストベクトル q_k から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル s_k を生成することである。筆者は、文単位の原文のコンテキストベクトル s_k の生成を、クエリのコンテキストベクトル q_k によってコントロールするべきと考えた。そこで本注意機構は、 q_k を受け取るパラメータ $Z_q \in \mathbb{R}^{\delta_5 \times \delta_6}$ を持つ。デコードステップ k のコンテキストベクトル s_k は、次の式で算出される。

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s + Z_q q_k), \quad (5.13)$$

$$\alpha_{k,m}^s = \frac{\exp(a_{k,m}^s)}{\sum_{m'=1}^{l_s} \exp(a_{k,m'}^s)}, \quad (5.14)$$

$$s_k = \sum_{m=1}^{l_s} \alpha_{k,m}^s h_m^s, \quad (5.15)$$

ここで、 $W_s \in \mathbb{R}^{\delta_5 \times \delta_7}$, $U_s \in \mathbb{R}^{\delta_5 \times \delta_5}$, $v_s \in \mathbb{R}^{\delta_5}$, s_k は δ_5 次元の特徴ベクトルである。

5.3.8 Document Attention Mechanism

この機構の役割は、エンコードされた特徴ベクトル h_i^d の各エンコードステップ i , Summary Decoder が出力する特徴ベクトル h_{k-1}^o , および、クエリのコンテキストベクトル q_k から、デコードステップ k において重要なエンコードステップについて加重平均をとった、コンテキストベクトル d_k を生成することである。筆者は、トークン単位の原文のコンテキストベクトル d_k の生成を、文単位の原文のコンテキストベクトル s_k によってコントロールするべきと考えた。そこで d_k の式は、 s_k を受け取るパラメータ $Z_s \in \mathbb{R}^{\delta_4 \times \delta_5}$ を持つ、デコードステップ k のコンテキストベクトル d_k は次の式で算出される。

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_q s_k), \quad (5.16)$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \quad (5.17)$$

$$d_k = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d, \quad (5.18)$$

ここで、 $W_d \in \mathbb{R}^{\delta_4 \times \delta_7}$, $U_d \in \mathbb{R}^{\delta_4 \times \delta_4}$, $v_d \in \mathbb{R}^{\delta_4}$, d_k は δ_4 次元の特徴ベクトルである。

5.3.9 Diversity Cell

Nema ら [40] が提案したこの機構の役割は各デコードステップ k において、同じトークンを繰り返し生成する RNN の問題点を解決することである。Long-Short Term Memory (LSTM) [32, 33] の実装を拡張し、各デコードステップ k のコンテキストベクトル d_k を互いに直交するベクトル d'_k へ変換する機構 **SD₂** は次の式で定義される。

$$\begin{pmatrix} i_k \\ f_k \\ o_k \\ c'_k \\ g_k \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} d_k \\ h_{k-1}^{dc} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \quad (5.19)$$

$$c_k = \sigma(i_k) \odot \tanh(c'_k) + \sigma(f_k) \odot c_{k-1}, \quad (5.20)$$

$$c_k^{diverse} = c_k - \sigma(g_k) \frac{c_k^\top c_{k-1}}{c_{k-1}^\top c_{k-1}} c_{k-1}, \quad (5.21)$$

$$h_k^{dc} = \sigma(o_k) \odot \tanh(c_k^{diverse}), \quad (5.22)$$

$$d'_k = h_k^{dc}, \quad (5.23)$$

ここで、 $W_i, W_f, W_o, W_g, W_c \in \mathbb{R}^{\delta_4 \times \delta_4}$, $U_i, U_f, U_o, U_g, U_c \in \mathbb{R}^{\delta_4 \times \delta_4}$, d'_k は、 δ_4 次元のベクトルである。

5.3.10 Summary Decoder

この機構の役割は、デコードステップ $k-1$ のコンテキストベクトル d'_{k-1} と、デコードステップ $k-1$ のトークン t_{k-1}^o を入力とし、デコードステップ k における特徴ベクトル h_k^o を出力することである。

$$h_k^o = \mathbf{GRU}_o(h_{k-1}^o, [e(t_{k-1}^o); d'_{k-1}]). \quad (5.24)$$

その後、特徴ベクトル h_k^o とコンテキストベクトル d'_k から、デコードステップ k におけるトークン t_k^o を予測する。

$$t_k^o = \text{softmax}(Wf(W_{dec}h_k^o + V_{dec}d'_k)), \quad (5.25)$$

ここで、 $W \in \mathbb{R}^{\delta_1 \times \delta_2}$, $W_{dec} \in \mathbb{R}^{\delta_2 \times \delta_7}$, $V_{dec} \in \mathbb{R}^{\delta_2 \times \delta_4}$ である。また、活性化関数 f は恒等関数である。

5.4 提案手法

ニューラル言語モデルを用いたテキスト要約において、生成される要約が入力の原文とは異なる内容になる問題が指摘されている。特に、注意機構によって原文やクエリの特定のトークンを強調したベクトルを用いる場合、入力ベクトル全体を俯瞰することを難しくするため、原文と要約間の内容の一貫性を保つことがより困難になっていると考えられる。そこで、注意機構の出力を LSTM などを用いられるゲート機構により制御することで、適応的にコンテキストベクトルを用いる Adaptive Attention Mechanism を提案する。この機構により要約モデルは、重要なトークンを注視する働きと入力全体を俯瞰する働きを、バランスよく扱えるようになることが期待される。

ここでは本章の 5.3 節で説明したモデルに、Adaptive Attention Mechanism を追加した提案手法について説明する。まず Adaptive Attention Mechanism の構成を説明し、それから 5.3 節で説明したモデルに本提案手法を導入するにあたって拡張が必要な関数について明記する。

5.4.1 Adaptive Attention Mechanism

この機構の役割は、各デコードステップ k において、コンテキストベクトル $v_k \in \mathbb{R}^{\delta_8}$ を適応的に利用するために、ゲート機構によって制御されたベクトル \hat{v}_k を生成することである。ここで v_k は、 d_k 、 q_k 、および、 s_k のうち、いずれかのコンテキストベクトルを表す。デコードステップ k の適応的なコンテキストベクトル \hat{v}_k は、次の式で算出される。

$$gate_k = \sigma(W_{gate}e(t_{k-1}^o) + U_{gate}h_{k-1}^o + Z_{gate}v_k), \quad (5.26)$$

$$\hat{v}_k = gate_k \odot v_k + (1 - gate_k) \odot h_{l_v}^v, \quad (5.27)$$

ここで、 $W_{gate} \in \mathbb{R}^{\delta_8 \times \delta_2}$ 、 $U_{gate} \in \mathbb{R}^{\delta_8 \times \delta_7}$ 、 $Z_{gate} \in \mathbb{R}^{\delta_8 \times \delta_8}$ である。また、 $h_{l_v}^v$ は、 h^v における最終ステップ l_v の特徴ベクトルである。

5.4.2 提案手法による既存手法の拡張

提案手法によって拡張された Triple-Hierarchical Attention Model を図 5.2 に示す。

5.4 節で提案した Adaptive Attention Mechanism を適用するために文単位の注意機構を、式 (5.13) から式 (5.28) へ、同様に、トークン単位の原文の注意機構を、式 (5.16) から式 (5.29) へ、以下の式に変更することとした。

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s + Z_q \hat{q}_k), \quad (5.28)$$

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_s \hat{s}_k). \quad (5.29)$$

5.5 実験

本実験の目的は、提案手法である Adaptive Attention Mechanism を導入した要約生成モデルと、提案手法の機構を用いない要約生成モデルのそれぞれが出力した要約を評価、比較することで、適応的な注意機構の導入が出力した要約へ与えた影響について明らかにすることである。

Triple-Hierarchical Attention Model を拡張した提案モデル（以下、**モデル 3**）と比較するモデルとして、第 4 章の 4.3 節で説明した Diversity driven Attention Model（以下、**モデル 0**）および 5.3 節で説明した Triple-Hierarchical Attention Model（以下、**モデル 1**）を用いた。また Adaptive Attention Mechanism 単体の効果についても明らかにするために、Adaptive Attention Mechanism のみを Diversity driven Attention Model に導入したモデル（以下、**モデル 2**）についても実験した。

本章では、原文の平均トークン数および生成要約の内容の傾向の異なる、2 種類のデータセットでそれぞれ実験した。まず実験 1 では、先行研究で用いられた Debatepedia データセットに対する各モデルの性能を比較した。次に実験 2 では、第 4 章で筆者が構築した、原文の平均トークン数がより大きい Wikipedia データセットに対する各モデルの性能を比較した。

5.5.1 実験1：Debatepedia データセットにおける実験

設定

実験データとして、Nemaら[40]がDebatepedia[25]から構築した13,573件のデータセット[67]を用い、ミニバッチサイズ16, 32, 及び、64の3種類について、10交差検証により性能を比較した。また、Nemaらの実験で用いられたROUGEの設定の詳細が確認できなかったため、本章では、より広く用いられているDUC2004 Task2のROUGEパラメータ[68]を用いることとした。

ただし、Nemaらの実験では、モデルへのデータの受け渡す際、Word Representationを用いたトークンの入れ替えにより訓練データ数が約15倍に増強されているが、本章の実験では、より公平な実験とするために増強していない。さらに、Nemaらの実験での語彙空間は、約220万語彙を有する学習済みのGloVe[69]から構成されているが、各10交差検証の訓練データ、検証データおよび評価データにおいて1回以上利用された語彙のみを抽出することで、平均37,000語彙に縮小されている。一方、本章の実験での語彙空間は、より適切な比較とするため、あらかじめEnglish Wikipediaのみから学習した約13万語彙をそのまま用いている。

データセット

Debatepediaは、重要な議題の賛成意見と反対意見の本文と、それぞれの要約文が複数まとめられている討論のインターネット百科事典である。またこのデータセットは、政治、法、環境、健康、道徳、宗教などの53カテゴリを含む、663件の討論から構成されている。データセット中のデータは、 (D, Q, S) のタプル形式で構成され、 D は各意見の本文、 Q は D に対応した議論となる1文、 S は Q に対応する要約文となっている。

モデルの設定と学習の詳細

本実験で用いた各モデルのパラメータについて述べる。

まず、fine tuningの対象であるWord RepresentationおよびSentence Representationについて説明する。各モデルで共通して用いたSkip-gramは、English Wikipedia[7]の全記事を対象として事前学習した。事前学習では、出現頻度が20以上のトークンを語彙として採用し、語彙の次元を131,718、単語埋め込みベクトルの次元を128とした。同様に、モデル1およびモデル3で用いたSkip-ThoughtについてもEnglish Wikipediaの全記事によって事前学習した。

Document Encoderの隠れ層の次元、各注意機構の隠れ層の次元、および、Summary Decoderの隠れ層の次元を128とした。またモデル1およびモデル3で用いたSkip-Thoughtの次元数、および、Sentence Encoderの次元数を256とした。

Debatepediaデータセットを用いた先行研究であるモデル0を参考に、各モデルは最適化手法としてAdam[60] ($\alpha=0.0004$, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$)を用い、コア数1、エポック数50で学習し、早期終了によりパラメータの学習を停止した。

モデルの実装には、Chainer[55, 56]、および、ChainerMN[57, 58]を用いた。また品詞タガーとして、The Stanford CoreNLP[65]を用いた。

結果

各モデルについて、要約モデルが出力した要約と正解要約間の再現率ベースの ROUGE-N[61] ($N = 1, 2$) および ROUGE-L により評価した。評価モジュールとして、要約評価ライブラリである pythonrouge[70] を用い、オプションは DUC 2004 の Task2 の設定 [68] を適用した。実験結果を表 5.1 に示す。

表 5.1: Debatepedia データセットに対する各モデルの ROUGE による比較。太文字の値は、各バッチサイズにおける最大値を示す。

モデル名	ミニバッチ	ROUGE-1	ROUGE-2	ROUGE-L
モデル 0	16	18.75	5.17	17.50
	32	16.24	3.65	15.02
	64	16.21	3.88	15.01
モデル 1	16	18.99	5.40	17.77
	32	16.39	3.93	15.21
	64	16.37	4.05	15.21
モデル 2	16	16.75	4.33	15.58
	32	14.47	3.12	13.38
	64	12.61	2.20	11.60
モデル 3	16	16.13	4.06	15.03
	32	14.76	3.25	13.65
	64	13.25	2.67	12.30

考察

表5.1より、モデル0と比べモデル1の各ROUGEの値が上回ることを確認できる。一方、Adaptive Attention Mechanismのみを導入したモデル2は、モデル0よりも各ROUGEの値が低くなった。また、モデル2に文エンコーダの導入したモデル3では、全てのROUGEの値ではないものの一部改善が確認できる。以上の結果から、第4章の実験結果と同じく、文エンコーダの導入により各ROUGEの値が向上することを確認できた。また、各ミニバッチサイズ間の比較から、ミニバッチのサイズが小さくなるにつれて性能が向上する傾向があることを確認できた。

一方、実際に生成された要約を確認したところ、原文やクエリ中に存在しない意味がよく似た別のトークンを、要約として生成しやすいことが全てのモデルにおいて確認された。実験に用いたデータセットについて確認したところ、公開されているデータセットを構築する際にも、データを増強する手段として、原文、および、クエリの数トークンについて、Word Representationの次元上で距離が近い他のトークンへ置換されていることが判明した。これにより、一部のデータにおいて、原文、クエリ、および、要約のそれぞれの間で、整合性が損なわれていることを確認した。

そこで筆者は、各交差検証の評価データから、原文や要約間の整合性を人手で確認したデータをそれぞれ10件ずつ、計100件を収集し、gold standardとして新たな正解データとした。gold standardによる評価結果について表5.2に示す。表5.2より、整合性を確認したgold standardに

表 5.2: 人手で選択された gold standard 評価データに対する、各モデルの ROUGE による比較.

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
モデル0	18.60	5.25	17.23
モデル1	18.72	5.30	17.59
モデル2	16.35	4.14	14.85
モデル3	16.90	4.21	15.45

おいても、モデル1の各ROUGEの値が最大となることを確認できた。同様に、モデル2に文エンコーダの導入したモデル3では、各ROUGEの値が改善したことを確認できた。

5.5.2 実験2：Wikipedia データセットにおける実験

設定

実験データとして、English Wikipedia, および、Simple English Wikipedia[8] から構築した41,989件のデータセット用い、ホールドアウト検証により性能を比較した。

データセット

要約の目標を含む要約タスクのデータセットは、あまり多いとは言えない [5, 6]. そのため筆者は、Wikipedia と Simple English Wikipedia を用いて、独自のデータセットを作成した. Simple English Wikipedia は、Wikipedia の英語版であり、基本的にベーシック英語とスペシャル・イングリッシュで記述される語彙が簡略化されたウェブ百科事典である. 作成したデータセット中のデータは、 (D, Q, S) のタプル形式で構成される. ここで、 D をあるタイトルを持つ Wikipedia 一つの記事、 Q を D に対応する Simple English Wikipedia のあるセクションのタイトル、 S を Q に対応する Simple English Wikipedia 内の文書とする. セクションのタイトルには、対象のセクション自身が持つタイトルの他に、Wikipedia 記事の階層構造に沿った、そのセクションの親となるタイトル全てを含む. 例えば、ある Simple English Wikipedia 記事の階層構造が、最上位の階層から「記事タイトル: Person Name/セクション: Abstract/セクション: Personal life/サブセクション: Education」となっている場合、対象のセクションが「Personal life」であった場合、 S には「記事タイトル: Person Name/セクション: Personal life」が含まれる.

Wikipedia の全ての記事数が 5,570,022 件であるのに対して、Simple English Wikipedia の全ての記事数は 131,459 件と 2.4%程度しか存在しないため、Wikipedia と同じ記事タイトルを持つ記事、107,168 件のみを取得した. 本実験では、上記の方法で取得したペアデータの 75,000 件からデータセットを作成した. データセット中の訓練データ、開発データおよび評価データの内訳について表 5.3 に示す. なおデータセットには、 D 、 Q および S の各トークン数のヒストグラムを参考

表 5.3: English Wikipedia と Simple English Wikipedia から作成したデータの件数.

データセット	件数
訓練データ	29,389
開発データ	6,300
評価データ	6,300

に、トークン数がそれぞれ、4,500、6 および 300 以下の記事のみを含めることとした.

モデルの設定と学習の詳細

本実験で用いた各モデルのパラメータについて述べる. ただし、5.5.1 項の実験 1 の設定とは異なるハイパーパラメータについてのみ記述する.

各モデルは、最適化手法として Adam[60] ($\alpha=0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\text{eps} = 10^{-8}$) を用い、コア数 4, バッチサイズ 3, エポック数 10 で学習し、開発データに対する ROUGE に基づく早期終了によりパラメータの学習を停止した。

結果

各モデルについて、要約モデルが出力した要約と正解要約間の再現率ベースの ROUGE-N[61] ($N = 1, 2$) および ROUGE-L により評価した。

クエリ指向テキスト要約において、生成要約 S はクエリ Q に応じて内容が変化する。そのため、このクエリ中に未知語が多く含まれる評価データは、モデルの精度を比較する上で適切なデータとは言い難い。そこで本実験では、クエリ Q の未知語率が、評価データにおけるクエリ Q の未知語率平均の 15% を超えるデータは除外することとした。実験結果を表 5.4 に示す。

また原文のトークン数について 500 以下から 4,500 まで 500 刻みで増加させた 9 通りにおける各 ROUGE の値を調べた。その結果を図 5.3 ~ 図 5.5 に示す。横軸は原文 D におけるトークン数を表し、縦軸は各 ROUGE の値を表す。

表 5.4: Wikipedia データセットに対する各モデルの ROUGE による比較。太文字の値は最大値を示す。

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
モデル 0	38.51	22.33	33.14
モデル 1	40.41	23.32	34.45
モデル 2	39.43	23.09	34.44
モデル 3	37.51	21.69	32.52

考察

表 5.4 より、モデル 0 と比べ、文エンコーダのみを導入したモデル 1 の各 ROUGE の値が大きく向上し各 ROUGE が最高値を達成した。これは、表 5.1 の実験と同様の結果を示している。同様に、Adaptive Attention Mechanism のみを導入したモデル 2 も、比較手法の各 ROUGE を上回ることが確認できる。一方、モデル 2 に文エンコーダを導入したモデル 3 では、全ての ROUGE の値が低下したことを確認できる。図 5.3, 図 5.4, および、図 5.5 より、原文のトークン数の増加に対して、文エンコーダを導入したモデル 1 は、他のモデルの各 ROUGE を全て上回る結果となった。一方、どのトークン数においても、他の 3 モデルの ROUGE は大きな差がないことがわかる。以上の結果から、文エンコーダの導入により各 ROUGE の値が向上することを確認できた。一方、Adaptive Attention Mechanism の導入による性能向上は、原文のトークン数が小さい場合のみ確認された。

各手法の出力例を表 5.5 ~ 表 5.10 に示す。表 5.5 ~ 表 5.8 は、鯨の一種である「Lemon shark」についての English Wikipedia の記事の正解要約と各モデルが生成したシステム要約である。こ

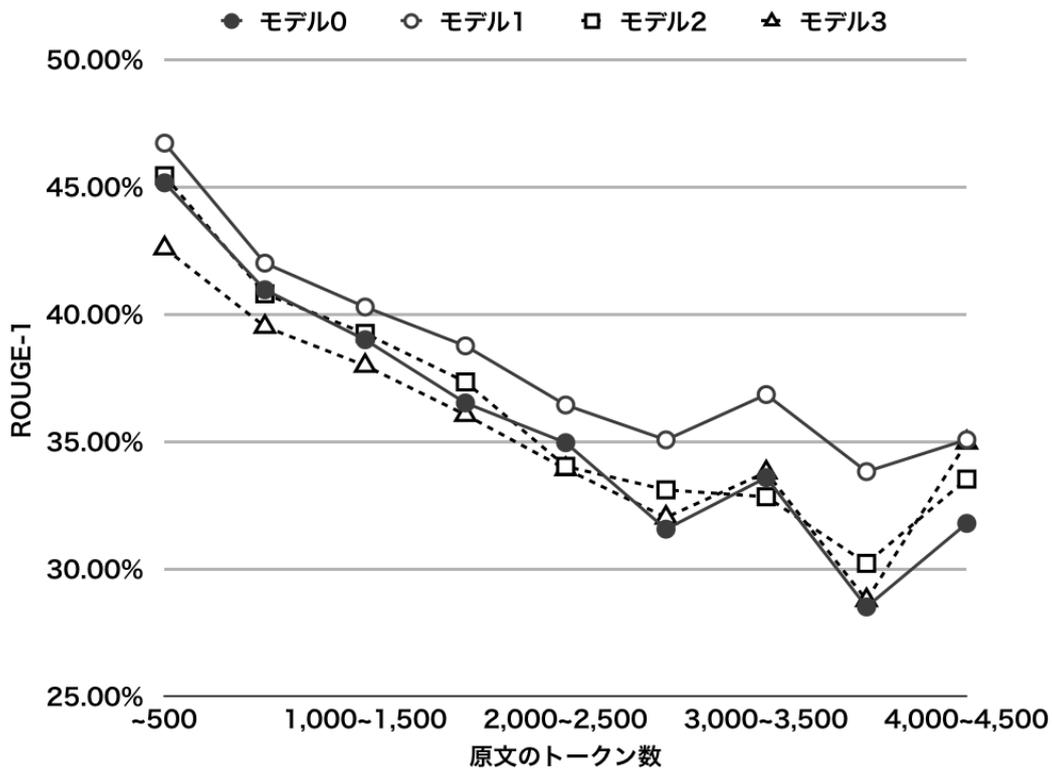


図 5.3: 原文のトークン数に対する, ROUGE-1 による各モデルの比較.

の例では, クエリとして「Lemon shark」のみが与えられたため, 各モデルは概要についての要約を出力していることが確認できる. 特にモデル2では, この鯨の特徴的な体表の色にまで言及した要約を生成している.

表5.9は, 表5.5~表5.8と同じく「Lemon shark」についての English Wikipedia の記事の正解要約と各モデルが生成したシステム要約である. ただし, この例では「Lemon shark」の他にクエリとして「habitat」がモデルに与えられる. モデル0, モデル1, および, モデル2は, 鯨の生息地を含んだ要約を生成できている. ただしモデル2以外の手法では, 鯨の食事についても触れているため, クエリに応じた完全な要約とは言えない.

表5.10は, インドに存在する「Thanjavur district」についての English Wikipedia の記事の正解要約と各モデルが生成したシステム要約である. この例では, クエリとして「Thanjavur district」のみが与えられたため, 表5.5~表5.8と同様に各モデルは概要についての要約を出力していることが確認できる. また各モデルは, 原文の冒頭とほぼ同じ内容を出力していることがわかる. 他の要約においても記事タイトルのみを含むクエリが与えられたとき, 各モデルが原文の冒頭とほぼ同じ内容を要約として出力する傾向があった.

また, 表5.5~表5.10から, 各モデルがトークン単位の繰り返しを生成しないことが確認できる. 一方, 同一のフレーズや文を繰り返し生成することも確認できる. これは, Diversity cellにより同じトークンを繰り返し生成する問題は改善されたが, 長い文を生成した場合には, 繰り返しの周期がトークン単位から文脈単位へスケールアップしたためと考えられる.

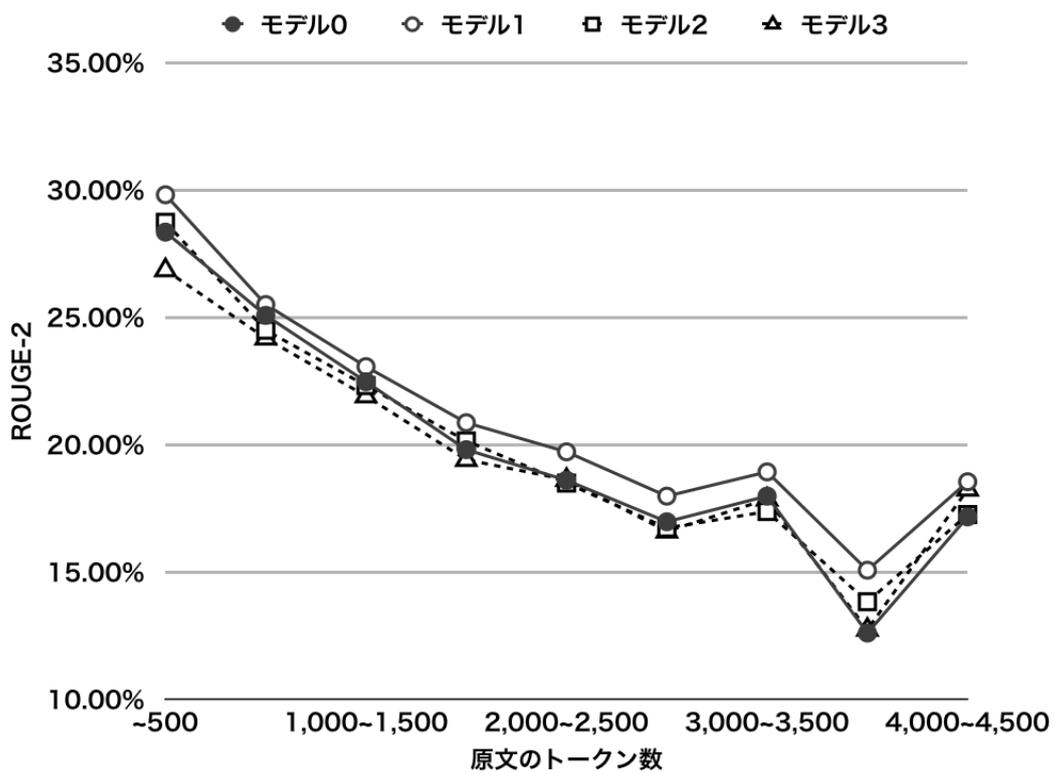


図 5.4: 原文のトークン数に対する, ROUGE-2 による各モデルの比較.

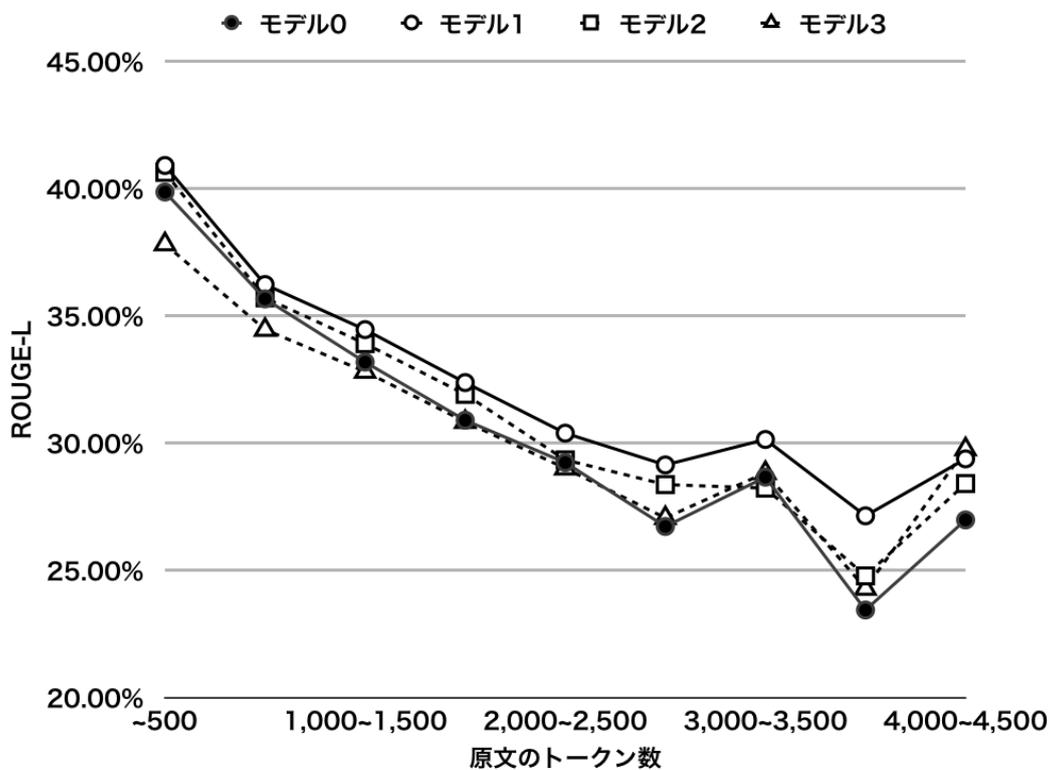


図 5.5: 原文のトークン数に対する, ROUGE-L による各モデルの比較.

表 5.6: モデル 1 が出力した要約文の一例要約文の一例 (記事タイトル:Lemon shark, クエリ:Lemon shark)

クエリ: Lemon shark
正解要約: the lemon shark is a shark that gets its name from its deep yellow back. its belly is off-white. lemon sharks are usually eight to ten feet long. they have long, sharp teeth for catching slippery fish. lemon sharks are common on the southeast coast of the united states of america.
モデル 1: the lemon shark (<unk> pantherophis, <unk> pantherophis, <unk> <unk>) is a small shark that is found in the eastern pacific ocean, caribbean oceans. the shark 's yellow appears in the western pacific ocean. they are very common in the arctic ocean. they are very common to live in the arctic ocean. they are called the “ <unk> shark ”, which is the most common species in the world.

表 5.7: モデル 2 が出力した要約文の一例要約文の一例 (記事タイトル:Lemon shark, クエリ:Lemon shark)

クエリ: Lemon shark
正解要約: the lemon shark is a shark that gets its name from its deep yellow back. its belly is off-white. lemon sharks are usually eight to ten feet long. they have long, sharp teeth for catching slippery fish. lemon sharks are common on the southeast coast of the united states of america.
モデル 2: the lemon shark (<unk> bolas) is a <unk> <unk>. it is a long shark in the arctic shark. it is a member of the family <unk>, lemon shark, and the lemon shark.

表 5.10: 出力された要約文の一例 (記事タイトル:Thanjavur district, クエリ:Thanjavur district)

<p>原文 (冒頭のみ) : thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern india. its headquarters is thanjavur. the district is located in the delta of the cauvery river and is mostly agrarian. as of two zero one one, thanjavur district had a population of two , four zero five , eight nine zero with a sex-ratio of one , zero three five females for every one , zero zero zero males. the district is located in central tamil nadu bounded on the northeast by nagapattinam district, on the east by tiruvarur district, on the south by the palk strait, of bay of bengal on the west by pudukkottai district, and on the north by the river, across which lie tiruchirappalli and perambalur districts. ...</p>
<p>クエリ: Thanjavur district</p>
<p>正解要約: thanjavur district is one of the three two district in tamil nadu, india. its headquarters is thanjavur city. this district is known as the “ granary of south india ” and “ rice bowl of tamil nadu ”. this is one of the important district for the tamil nadu agriculture and economy.</p>
<p>モデル 0: thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern uttar india. its headquarters is thanjavur. the district is in the delta of the cauvery river and is mostly autonomous. as of two zero one one, thanjavur district had a population of two , four zero five , eight nine zero with a sex-ratio of one , zero three five females for every one , zero zero zero males. the district is in the delta of the cauvery river and is mostly agrarian. as of two zero one one, thanjavur district had a population of two , four zero five , eight nine zero with a sex-ratio of one , zero three five females for every one , zero zero zero males. the district is in the delta of the cauvery river and is mostly agrarian. as of two zero one one, thanjavur district had a population of two , four zero five , eight nine zero with a sex-ratio of one , zero three five females for every one , zero zero zero males. the district is in the delta of the cauvery river and is mostly agrarian. as of two zero one one, thanjavur district had a population of two , four zero five , eight nine zero with a sex-ratio of one , zero three five females for every</p>
<p>モデル 1: thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern india. its headquarters is thanjavur. the district is in the delta of the nunavut of the cauvery river and is mostly historically politically.</p>
<p>モデル 2: thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern india. its headquarters is thanjavur. the district is in the delta of the cauvery river and is mostly uncultivated farmers.</p>
<p>モデル 3: thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern india. its headquarters is thanjavur district. the district is located in the delta of the cauvery river and is mostly associated with the mountains of the indus river and the river <unk>.</p>

5.6 結言

本章では、各注意機構の出力ベクトルを適応的に用いる Adaptive Attention Mechanism を導入したモデルを提案した。実験1および実験2より、第4章で提案した文エンコーダを導入した手法の一貫的な精度向上が確認できた。一方、本章の提案手法である Adaptive Attention Mechanism の精度向上は、原文の平均トークン数が大きい実験2のデータセットにおいてのみ確認できた。また実験2により、入力される原文の平均トークン数の大小に寄らず、Adaptive Attention Mechanism および文エンコーダをそれぞれ導入することによって、各 ROUGE の値が改善されることが確認できた。一方、5.5.1 項の実験で用いた Debatepedia データセットにおいて、一部のデータの整合性が損なわれていることが確認された。そのため今後は、他の先行研究で用いられたクエリ指向テキスト要約のデータセットに対しても、各モデルの性能を検証を検討している。また、本提案手法である Adaptive Attention Mechanism を、汎用テキスト要約へ適用した場合についても検証を進める予定である。

第6章 結論

本論文では、クエリ指向テキスト要約のためのニューラル言語モデルの研究を発展させるべく、クエリ指向テキスト要約に、より適する新たなデータセットの構築と、クエリ指向要約が抱える課題の解決を目指した。新規のニューラル言語モデルを提案し、当該の新規データセットを用いて提案手法の有効性を検証した。

第1章では、クエリ指向テキスト要約の研究背景を述べ、解決されていない課題を指摘した。また、指摘した課題の解決を目指し、本論文で提案した各手法について概説した。

第2章では、第3章、第4章および第5章の研究に関わりが深い関連研究について述べ、本論文の各研究の位置づけを明確に示した。

第3章では、クエリ指向テキスト要約のための新たなデータセットを地方議会会議録から構築した。当該データセットにおいて、生成アプローチを用いたクエリ指向テキスト要約の精度を検証することで、クエリ指向要約におけるニューラル言語モデルが抱える課題を整理した。人手による生成要約の品質の評価から、生成要約の出力長や生成要約の内容の多様性を制御する機構が、生成要約の品質の低下に繋がることが確認された。これは、生成アプローチを用いたクエリ指向要約が抱える課題として指摘されている、入力文書の意味する内容から逸脱した内容の要約を出力する現象が生じたためと考えられる。一方で、学習データセットが少ない場合でも、SentencePiece[52]を用いて、未知語問題へ対応し、データセットを増強することで、想定されていた正解要約以外の要約を生成できることを示した。

第4章では、既存の生成アプローチを用いるクエリ指向テキスト要約の課題であった、トークン数の多い原文の適切なエンコードを達成するために、原文を文単位で扱うエンコーダの導入と、クエリ、文単位の原文、および、トークン単位の原文を、階層的に組み合わせる注意機構を提案した。また、提案手法の精度改善を検証するために、これまで整備が進んでいなかった、トークン数が多い原文を含む新たなデータセットを構築した。実験から、既存手法に文エンコーダおよび階層的な注意機構を導入することによって、生成要約の精度が改善することを確認した。また、同一の原文に対して異なるクエリを与えた場合、生成要約の内容を適切に変更できることを確認した。

第5章では、原文の局所的な注視と、原文の全体の俯瞰の適応的な制御を目指した注意機構を提案した。実験では、先行研究のデータセットと、第4章で構築したデータセットのそれぞれで実験し、評価指標や、実際の生成要約の内容を既存手法と比較することで、提案手法の有効性について議論した。原文の俯瞰がより重要であると考えられる。原文のトークン数が多い実験データセットにおいて、提案手法による精度改善が確認された。また、原文のトークン数の増加に対する評価指標の変動について検証することで、第4章の文エンコーダ、および、適応的な注意機構のそれぞれで、原文のトークン数の増加によらず、おおよそ一貫して精度向上に貢献することを示した。本提案手法の適応的な注意機構は、汎用テキスト要約や機械翻訳の分野でも導入可能な手法となっており、今後は、汎用テキスト要約に対しても提案手法を導入し、実験結果を検証

することで、手法の改良を進める予定である。

第1章でも述べたように、現代の要約のメインターゲットといえるデジタルデータは、爆発的に増加し、また利用するユーザの嗜好も多様化が進んでいる。そのため、多様化したユーザの情報要求に対応できる柔軟な情報取得の実現は、今後、より一層重要な課題になると推測される。本論文で扱ったクエリ指向テキスト要約は、様々なトピックを含む膨大なテキストから、ユーザが知りたい特定のトピックを含む要約を生成する処理である。これにより、ユーザは、要約元の内容に含まれる関心のあるトピックのみを、より短い時間で把握できる。そのため、クエリ指向要約は、情報の獲得の効率化や、情報の円滑な取捨選択を支援する側面から、この課題に貢献する研究といえる。

本論文では、クエリ指向テキスト要約で必要とされる課題設定に沿った、トークン数の多い原文を含むデータセットを構築し、当該データセットによる実験を通して、新たに提案した、原文のトークン数の増加に頑健な要約手法、および、原文の内容と生成要約の内容の一貫性保持を目指した要約手法の有効性を示した。今後、本論文で論じた研究が、クエリ指向テキスト要約の発展に繋がることを期待したい。

謝 辞

本論文は筆者が京都産業大学大学院先端情報学研究科先端情報学専攻博士後期課程に在籍中の研究成果をまとめたものです。本論文を結ぶにあたり、本研究を遂行する上でご指導、ご鞭撻およびご援助をいただいた方々に感謝の意を表します。

同専攻教授宮森恒先生には、指導教官として本研究の実施の機会を与えて戴き、その遂行にあたって終始、ご指導を戴きました。ここに深謝の意を表します。

同専攻教授中島伸介先生、並びに、同専攻教授平石裕実先生には、副査として多くの議論と重要なご助言を戴きました。ここに深謝の意を表します。

本専攻宮森研究室の各位には、研究遂行にあたり日頃より有益なご討論ご助言を戴きました。ここに謝意を表します。特に本論文の作成にあたっては、田上諒氏、杉本翔氏および石橋陽一氏の各氏には、本論文に不可欠で有益なご討論を戴きました。ここに感謝の意を表します。

以上の方々をはじめとして、研究を進めるにあたり、多方面にわたり筆者を支えてくださった関係各位に、心から感謝申し上げます。

最後に、大学において勉学のみ集中することができましたのは、いつも温かく見守り、また、辛抱強く支援を続けて戴いた両親の力添えがあったからです。深謝の意を表して謝辞と致します。

2019年12月 木村 輔

業績リスト

学術雑誌論文

1. 木村 輔, 田上 諒, 宮森 恒. 文エンコーダによるクエリ指向要約モデルの強化. 情報処理学会論文誌データベース (TOD), Vol.12, No.4, pp.66-79, 2019
2. 田上 諒, 木村 輔, 宮森 恒. 大学入試の穴埋め型問題に対する語順を考慮した自動解答手法. 情報処理学会論文誌データベース (TOD), Vol.10, No.3, pp.45-57, 2017
3. Ryo Tagami, Tasuku Kimura, Hisashi Miyamori. Automatic Answering Method considering Word Order for Slot Filling Questions of University Entrance Examinations, 19th International Conference on Asia-Pacific Digital Libraries (ICADL 2017), LNCS 10647, pp.128-141, 2017

会議論文 (査読有)

1. Tasuku Kimura, Ryo Tagami, Hisashi Miyamori. Query-focused Summarization Enhanced with Sentence Attention Mechanism, 6th IEEE International Conference on Big Data and Smart Computing (BigComp 2019), 6B-2, pp.250-257, 2019

会議論文 (査読無)

1. Tasuku Kimura, Ryo Tagami, Hikaru Katsuyama, Sho Sugimoto and Hisashi Miyamori. KSU Systems at the NTCIR-14 QA Lab-PoliInfo Task, The 14th NTCIR Conference, 2019
2. Tasuku Kimura, Ryo Tagami, Hisashi Miyamori. KSU Team's QA System for World History Exams at the NTCIR-13 QA Lab-3 Task, The 13th NTCIR Conference, 2017

学術報告

1. 木村 輔, 田上 諒, 宮森 恒. 文エンコーダによるクエリ指向要約モデルの強化. 第11回データ工学と情報マネジメントに関するフォーラム (DEIM2019), A2-4, 2019. [学生プレゼンテーション賞]
2. 田上 諒, 木村 輔, 宮森 恒. 前提知識を考慮した根拠の妥当性判断による自動回答手法. 第11回データ工学と情報マネジメントに関するフォーラム (DEIM2019), G7-4, 2019
3. 石橋 陽一 (NAIST), 森 泰, 木村 輔, 宮森 恒. 質問文から連想した画像特徴量を用いた質問応答モデル. 言語処理学会第25回年次大会 (NLP2019), D3-4, 2019
4. 田上 諒, 木村 輔, 杉本 翔, 勝山 光, 宮森 恒. 地方議会における事実確認のための会議録分割. 言語処理学会第25回年次大会 (NLP2019), P3-28, 2019
5. 勝山 光, 杉本 翔, 田上 諒, 木村 輔, 宮森 恒. 地方議会特有の表現に基づく政策に対する発言の賛否分類. 言語処理学会第25回年次大会 (NLP2019), P7-26, 2019

6. 杉本 翔, 木村 輔, 田上 諒, 宮森 恒. 表層的個性と対話行為に基づく特定の個性を付与した文生成. 言語・音声理解と対話処理研究会, No.84, pp.17-22, 2018
7. 木村 輔, 田上 諒, 宮森 恒. 目標ベクトルによる要約観点の一貫性を保持した文書要約生成. 第10回データ工学と情報マネジメントに関するフォーラム (DEIM2018), C2-2, 2018. [学生プレゼンテーション賞][ラケシュ・アグラワル賞]
8. 田上 諒, 木村 輔, 宮森 恒. 事物間の多様な状態に基づくファクトイド型質問応答. 第10回データ工学と情報マネジメントに関するフォーラム (DEIM2018), G3-5, 2018. [学生プレゼンテーション賞]
9. 田上 諒, 木村 輔, 宮森 恒. 大学入試の穴埋め型問題に対する語順を考慮した自動解答手法. 第9回データ工学と情報マネジメントに関するフォーラム (DEIM2017), B8-2, 2017
10. 木村 輔, 宮森 恒. 要約条件と単文型知識源に基づく non-factoid 型質問応答手法. 第9回データ工学と情報マネジメントに関するフォーラム (DEIM2017), B8-3, 2017

学術講演

1. 木村 輔, 田上 諒, 宮森 恒. 文エンコーダによるクエリ指向要約モデルの強化. WebDB Forum 2019, 2019. [優秀学生ポスター発表賞]
2. 田上 諒, 木村 輔, 宮森 恒. 大学入試の穴埋め型問題に対する語順を考慮した自動解答手法. WebDB Forum 2017, 2017. [学生奨励賞]

参考文献

- [1] “Google.” <https://www.google.co.jp>.
- [2] “DUC 2004 Documents for Summarization, Tasks, and Measures.” <https://duc.nist.gov/duc2004/>.
- [3] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Association for Computational Linguistics, 2015.
- [4] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*, (Cambridge, MA, USA), pp. 1693–1701, MIT Press, 2015.
- [5] F. Deroncourt, M. Ghassemi, and W. Chang, “A repository of corpora for summarization,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, (Miyazaki, Japan), European Languages Resources Association (ELRA), May 2018.
- [6] F. Deroncourt, M. Ghassemi, and W. Chang, “A repository of corpora for summarization,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, (Miyazaki, Japan), European Languages Resources Association (ELRA), May 2018.
- [7] “Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/Main_Page.
- [8] “Wikipedia.” https://simple.wikipedia.org/wiki/Main_Page.
- [9] H. P. Luhn, “The automatic creation of literature abstracts,” *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.
- [10] H. P. Edmundson, “New methods in automatic extracting,” *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.
- [11] K. Filippova and M. Strube, “Dependency tree based sentence compression,” in *Proceedings of the Fifth International Natural Language Generation Conference, INLG ’08*, (Stroudsburg, PA, USA), pp. 25–32, Association for Computational Linguistics, 2008.
- [12] “TIPSTER Text Summarization Evaluation Conference (SUMMAC) Overview.” http://www-nlpir.nist.gov/related_projects/tipster_summac/.
- [13] “Document Understanding Conferences.” <http://duc.nist.gov>.
- [14] “National Institute of Informatics.” <http://www.nii.ac.jp/>.
- [15] “NTCIR Workshop.” <http://research.nii.ac.jp/ntcir/ntcir-ws2/work-en.html>.

- [16] “Text Summarization Challenge Home Page.” <http://lr-www.pi.titech.ac.jp/tsc/>.
- [17] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto, “Extracting important sentences with support vector machines,” in *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING ’02, (Stroudsburg, PA, USA), pp. 1–7, Association for Computational Linguistics, 2002.
- [18] J. M. Conroy and D. P. O’leary, “Text summarization via hidden markov models,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’01, (New York, NY, USA), pp. 406–407, ACM, 2001.
- [19] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, Sep 1995.
- [20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 3111–3119, Curran Associates, Inc., 2013.
- [21] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML’14, pp. II–1188–II–1196, JMLR.org, 2014.
- [22] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, “Skip-thought vectors,” in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 3294–3302, Curran Associates, Inc., 2015.
- [23] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised learning of universal sentence representations from natural language inference data,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680, Association for Computational Linguistics, 2017.
- [24] A. Nie, E. D. Bennett, and N. D. Goodman, “Dissent: Sentence representation learning from explicit discourse relations,” *CoRR*, vol. abs/1710.04334, 2017.
- [25] “Welcome to Debatepedia! - Debatepedia, the Wikipedia of Debates.” http://www.debatepedia.org/en/index.php/Welcome_to_Debatepedia%21.
- [26] J. Tan, X. Wan, and J. Xiao, “Abstractive document summarization with a graph-based attentional neural model,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1171–1181, Association for Computational Linguistics, July 2017.

- [27] J. Li, T. Luong, and D. Jurafsky, “A hierarchical neural autoencoder for paragraphs and documents,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Beijing, China), pp. 1106–1115, Association for Computational Linguistics, July 2015.
- [28] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, pp. 3104–3112, 2014.
- [29] W. Li, X. Xiao, Y. Lyu, and Y. Wang, “Improving neural abstractive document summarization with explicit information selection modeling,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 1787–1796, Association for Computational Linguistics, Oct.-Nov. 2018.
- [30] Q. Zhou, N. Yang, F. Wei, and M. Zhou, “Selective encoding for abstractive sentence summarization,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1095–1104, Association for Computational Linguistics, July 2017.
- [31] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Association for Computational Linguistics, 2014.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.
- [33] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural Comput.*, vol. 12, pp. 2451–2471, Oct. 2000.
- [34] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 2048–2057, PMLR, 07–09 Jul 2015.
- [35] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov, “Gated-attention readers for text comprehension,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1832–1846, Association for Computational Linguistics, July 2017.
- [36] M. Yousefiazar, K. Sirts, L. Hamey, and D. M. Aliod, “Query-based single document summarization using an ensemble noisy auto-encoder,” in *Proceedings of the Australasian Language Technology Association Workshop 2015*, pp. 2–10, 2015.

- [37] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, pp. 504–507, July 2006.
- [38] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, “Attsum: Joint learning of focusing and summarization with neural attention,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 547–556, The COLING 2016 Organizing Committee, 2016.
- [39] C. Kiddon, L. Zettlemoyer, and Y. Choi, “Globally coherent text generation with neural checklist models,” in *EMNLP*, pp. 329–339, 2016.
- [40] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, “Diversity driven attention model for query-based abstractive summarization,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1063–1072, Association for Computational Linguistics, 2017.
- [41] “Fake News Challenge.” <http://www.fakenewschallenge.org/#page-top>.
- [42] “Overview >CLEF-2018 Fact Checking Lab.” <http://alt.qcri.org/clef2018-factcheck/index.php?id=overview>.
- [43] “NTCIR-14 カンファレンス | NTCIR-14 | NTCIR.” <http://research.nii.ac.jp/ntcir/ntcir-14/conference-ja.html>.
- [44] “What ’ s New — NTCIR-14 QA Lab PoliInfo.” <https://poliinfo.github.io>.
- [45] Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura, “Controlling Output Length in Neural Encoder-Decoders,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, 2016.
- [46] “東京都議会.” <https://www.gikai.metro.tokyo.jp>.
- [47] “都議会だより | 東京都議会.” <https://www.gikai.metro.tokyo.jp/newsletter/>.
- [48] “板橋区議会 — 板橋区.” <http://www.city.itabashi.tokyo.jp/sub10/>.
- [49] “いたばし区議会だより — 板橋区.” http://www.city.itabashi.tokyo.jp/c_categories/index10010.html.
- [50] M. Utiyama and H. Isahara, “A statistical model for domain-independent text segmentation,” in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, (Toulouse, France), pp. 499–506, Association for Computational Linguistics, July 2001.
- [51] “Apache Solr - .” <https://lucene.apache.org/solr/>.

- [52] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.
- [53] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *CoRR*, vol. abs/1409.0473, 2014.
- [54] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” *Trans. Sig. Proc.*, vol. 45, pp. 2673–2681, Nov. 1997.
- [55] S. Tokui, K. Oono, S. Hido, and J. Clayton, “Chainer: a next-generation open source framework for deep learning,” in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, vol. 5, 2015.
- [56] “Chainer: A flexible framework for neural networks.” <https://chainer.org/>.
- [57] T. Akiba, K. Fukuda, and S. Suzuki, “ChainerMN: Scalable Distributed Deep Learning Framework,” in *Proceedings of Workshop on ML Systems in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.
- [58] “ChainerMN: Scalable distributed deep learning with Chainer.” <https://github.com/chainer/chainermn>.
- [59] “MeCab: Yet Another Part-of-Speech and Morphological Analyzer.” <https://taku910.github.io/mecab/>.
- [60] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [61] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop* (S. S. Marie-Francine Moens, ed.), (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.
- [62] Y. Kimura, H. Shibuki, H. Ototake, Y. Uchida, K. Takamaru, K. Sakamoto, M. Ishioroshi, T. Mitamura, N. Kando, T. Mori, H. Yuasa, S. Sekine, , and K. Inui, “Overview of the NTCIR-14 QA Lab-PoliInfo Task.,” in *Proceedings of the 14th NTCIR Conference*, 2019.
- [63] “総務省 | 平成 26 年版情報通信白書 | ICT の進化が促すビッグデータの生成・流通・蓄積.” <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/html/nc131110.html>.
- [64] P. Koehn and R. Knowles, “Six challenges for neural machine translation,” in *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, Association for Computational Linguistics, 2017.

- [65] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit,” in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.
- [66] “検索結果に効果的に表示されるタイトルとスニペットを作成する - Search Console ヘルプ.” <https://support.google.com/webmasters/answer/35624?hl=ja>.
- [67] “DiverstiyBasedAttentionMechanism/data at master · PrekshaNema25/DiverstiyBasedAttentionMechanism.” <https://github.com/PrekshaNema25/DiverstiyBasedAttentionMechanism/tree/master/data>.
- [68] “An Introduction to DUC 2004 Intripnsic Evaluation of Generic New Text Summarization Systems.” <https://duc.nist.gov/pubs/2004slides/duc2004.intro.pdf>.
- [69] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [70] “tagucci/pythonrouge: Python wrapper for evaluating summarization quality by ROUGE package.” <https://github.com/tagucci/pythonrouge>.