

文エンコーダによるクエリ指向要約モデルの強化

木村 輔^{1,a)} 田上 諒^{1,†1,b)} 宮森 恒^{1,c)}

受付日 2019年3月9日, 採録日 2019年7月6日

概要: 膨大な量の文書が溢れる現代において, 与えられたクエリに着目した要約を生成するクエリ指向文書要約の重要性が高くなると予想される. 深層モデルに基づく生成型の自動要約手法では, 長期的な情報の記憶が可能となる Long-Short Term Memory (LSTM) は欠かせない. しかし翻訳タスクにおいて, 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 (2017) 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することは, 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. そこで我々は, 原文の単語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. 単語単位の注意機構と文単位の注意機構を組み合わせることで, 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の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. 最新のクエリ指向要約モデルと, そのモデルを提案手法で拡張したモデルの評価実験から, 原文のトークン数が大きいデータセットにおいても, 提案手法によって ROUGE が改善されることを確認した.

キーワード: クエリ指向文書要約, 生成型要約, 文エンコーダ, 注意機構, 自然言語処理

Query-focused Summarization Enhanced with Sentence Attention Mechanism

TASUKU KIMURA^{1,a)} RYO TAGAMI^{1,†1,b)} HISASHI MIYAMORI^{1,c)}

Received: March 9, 2019, Accepted: July 6, 2019

Abstract: In today's society where the enormous amount of documents are overflowing, it is expected that the query-focused text summarization, which generates summaries focusing on the given specific queries, will be more important. In the automatic summarization methods based on the deep neural model, Long-Short Term Memory (LSTM) which enables long-term information storage is essential. However, Koehn et al. report that the translation quality is degraded in the encoding of texts longer than 60 tokens entered into the LSTM. Even in the summarization task, it is considered to be a problem that the encoding of a long sentence fails, as a factor that greatly deteriorates the quality of the summary result. In this paper, we propose a method to generate a summary by introducing a sentence unit vector in addition to the word unit vector of the original document. We aim to generate summaries considering the importance degree of a sentence unit and the relations between sentences, by learning both the attention mechanism of word unit and the attention mechanism of sentence unit. From the experiment by ROUGE metrics of the latest query-focused summarization model and the model expanded by the proposed method, it was confirmed that ROUGE score improved by the proposed method even in the data set with the original document consisting of many tokens.

Keywords: query-focused summarization, abstractive summarization, sentence encoder, attention mechanism, natural language processing

¹ 京都産業大学大学院先端情報学研究科
Division of Frontier Informatics, Kyoto Sangyo University,
Kyoto, 603-8555 Japan

^{†1} 現在, 株式会社 JR 西日本 IT ソリューションズ
Presently with JR WEST IT Solutions Company

^{a)} i1658047@cc.kyoto-su.ac.jp

^{b)} tagami@j-wits.co.jp

^{c)} miya@cc.kyoto-su.ac.jp

1. はじめに

インターネットの継続的な発展にともない, 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*1。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、すべ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、与えられた文書を自動で要約する研究はさかんに取り組まれている。

非クエリ指向文書要約 (Generic summarization) は、特定の観点を想定しない自動要約である。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、クエリ指向文書要約 (Query-focused summarization) は、ユーザから与えられたクエリが示す、ある特定の観点に沿った要約の出力を目的としている。非クエリ指向文書要約は、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。たとえば、原文中の文やフレーズを組み合わせる要約を出力する抽出型では、原文の文頭や接続詞によるい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、クエリ指向文書要約は、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 [2] が提案した深層モデルに基づく生成型の非クエリ指向要約が一定の成功を収めたことで、生成型の自動文書要約はより活発に研究されている。このモデルは、Sequence to Sequence [3] という機械翻訳において提案されたモデルに注意機構 [4] 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、デコードごとに更新し要約を出力できる。また、クエリ指向文書要約では、Nema ら [5] 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、入力の原文の各単語ベクトルに加えて、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。

深層モデルに基づく生成型の自動要約手法では、長期的な情報の記憶が可能となる Long-Short Term Memory (LSTM) 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 [1] 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したりすることは、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。そこで我々は、原文の単語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、単語単位の注意機構と文単位の注意機構を組み合わせることで、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、最新のクエリ指向型要約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される要約に与える影響を明らかにする。

本稿の構成は以下のとおりである。2 章で関連研究について述べ、3 章で本モデルのタスクを定式化する。4 章でベースとなる最新のクエリ指向要約手法について述べ、5 章で提案手法の目的とシステム構成について詳述する。6 章で実験と考察について述べる。最後に、7 章で結論と課題を述べる。

2. 関連研究

1950 年代後半に始まった自動文書要約の研究は、原文中から抽出した重要文を要約として出力する Luhn ら [6] が提案した抽出型の手法が主流となっていた。それ以降、重要文抽出は様々な研究されており、原文中の出現頻度が高い重要語 [6]、文の位置情報 [7]、接続詞のような手掛かり語 [7] 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とともない、抽出した重要文を短くする文圧縮 [8] もさか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)*2 という初めての自動文書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)*3 や、国立情報学研究所 (NII)*4 が主催する NTCIR のタスクとして Text Summarization Challenge (TSC)*5 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。それにともない、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 [9], [10] では、これまでの研究と同様に単語の出

*1 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>

*2 TIPSTER Text Summarization Evaluation Conference (SUMMAC) Overview, http://www-nlpir.nist.gov/related_projects/tipster_summac/

*3 Document Understanding Conferences, <http://duc.nist.gov>

*4 National Institute of Informatics, <http://www.nii.ac.jp/>

*5 Text Summarization Challenge Home Page, <http://lr-www.pi.titech.ac.jp/tsc/>

現頻度に加え、N-gramなどを各文の特徴量とし、support vector machine (SVM) 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収めた、深層学習による自動要約がさかんに研究されている。ここでは、本研究と関連する「文の分散表現」、「文エンコーダと注意機構を組み合わせた文書要約」、「ゲート機構と組み合わせた注意機構」、および、「深層学習によるクエリ指向文書要約」について述べる。

近年、Word2Vec [11] 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さかんに研究されている。Paragraph Vector [12] は、深層学習による文の分散表現の最初期の手法の1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 [13] 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 [14] 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あげられる。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 [15] 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。

InferSentやDisSentと比較して、データセットの構築が容易であるため、本提案手法はSkip-Thoughtを文エンコーダとして選択した。また本研究のデータセットであるWikipedia、および、Debatepediaには、含意関係や談話マーカによらない様々な文の関係が存在することをふまえ、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、本研究の文エンコーダとしてより適切であると考えた。

文エンコーダと注意機構を組み合わせた自動文書要約も活発に研究されている。Tanら [16] 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 [17] 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、各文ベクトルから単語をデコードする

encoder-decoderである。またLiら [18] 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 [19] は、BiGRU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、単語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LSTMやGated Recurrent Units (GRU) を文エンコーダとして用いている。一方、提案手法では事前学習した文エンコーダを用いるため、文の特性をよりとらえ、文の分散表現を活かした要約生成が期待される。

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 [20] は、画像の説明文生成の研究において、アテンション機構で生成したベクトルをゲート機構によってさらに強調する手法を提案している。Dhingraら [21] 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とり範囲は0.0~1.0ではないため、厳密にはゲート機構を用いたとはいえない。

これらの手法に対し、本提案手法の1つであるAdaptive Attention Mechanismは、入力の全体を表現するベクトルを使用するか、それともアテンション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。

クエリ指向要約における抽出型の研究として、Yousefiazarら [22] 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などで与えられる入力に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モデルを提案している。またCaoら [23] 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。

クエリ指向要約における生成型の研究として、Kiddon

ら [24] 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 [5] は、深層モデルに基づく生成型のクエリ指向文書要約の 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 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。

3. 問題の定式化

ここではクエリ指向要約が対象とするタスクを定義する。

まず、クエリ指向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。本稿では、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 $\mathbf{s}_m = t_1^m, t_2^m, \dots, t_{l_{s_m}}^m$ 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。

以上をふまえ、クエリ指向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 $\mathbf{q}$ について要約した文書 $\mathbf{o}^{system}$ を生成すること。

4. Diversity driven Attention Model

ここでは、Nema らがいくつか提案した Diversity driven Attention Model [5] のうち、特に精度が良かった **SD₂** について説明する。

Diversity driven Attention Model を図 1 に示す。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

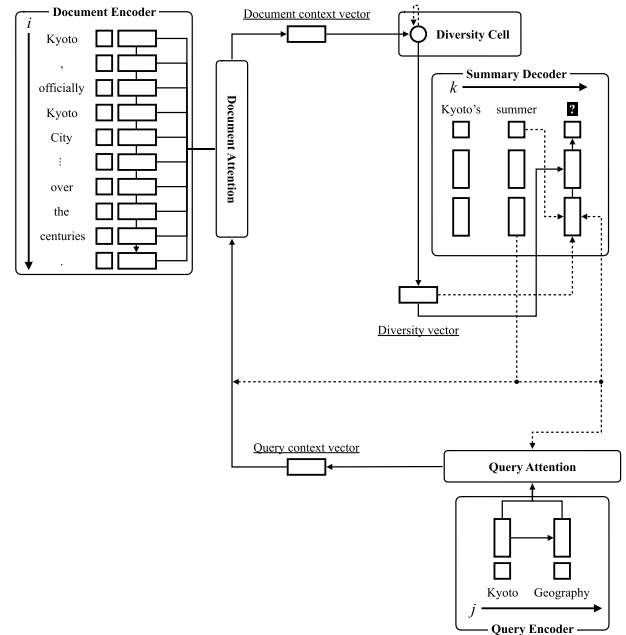


図 1 Diversity driven Attention Model. 2つのエンコーダ、および、それぞれの注意機構と、1つのデコーダ、および、1つの Diversity Cell から構成される。破線はデコードステップ $k-1$ を表現し、実線はデコードステップ k を表現する

Fig. 1 Diversity driven Attention Model is composed of the query encoder, the document encoder, the attention mechanism of each encoder, the summary decoder, and the diversity cell. The broken line represents the processing at the decoding step $k-1$, and the solid line represents the processing at the decoding step k .

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文書ベクトルに、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。

4.1 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、モデルの内部のすべてのエンコーダとデコーダで共有される。

4.2 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) [25] を用いて次のように表す。

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

ここで、 h_i^d は δ_3 次元の特徴ベクトルである。

4.3 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

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。

$$h_j^q = \text{GRU}_q(h_{j-1}^q, e(t_j^q)), \quad (2)$$

ここで、 h_j^q は δ_4 次元の特徴ベクトルである。

4.4 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 4.7 節で説明する 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。

$$a_{k,j}^q = v_q^\top \tanh(W_q h_{k-1}^o + U_q h_j^q), \quad (3)$$

$$\alpha_{k,j}^q = \frac{\exp(a_{k,j}^q)}{\sum_{j'=1}^{l_q} \exp(a_{k,j'}^q)}, \quad (4)$$

$$q_k = \sum_{j=1}^{l_q} \alpha_{k,j}^q h_j^q, \quad (5)$$

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。

4.5 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。

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_q q_k), \quad (6)$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \quad (7)$$

$$d_k = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d, \quad (8)$$

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。

4.6 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 **SD₂** を次の式で定義した。

$$\begin{pmatrix} i_k \\ f_k \\ o_k \\ c'_k \\ g_k \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} d_k \\ h_{k-1}^{dc} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \quad (9)$$

$$c_k = \sigma(i_k) \odot \tanh(c'_k) + \sigma(f_k) \odot c_{k-1}, \quad (10)$$

$$c_k^{diverse} = c_k - \sigma(g_k) \frac{c_k^\top c_{k-1}}{c_{k-1}^\top c_{k-1}} c_{k-1}, \quad (11)$$

$$h_k^{dc} = \sigma(o_k) \odot \tanh(c_k^{diverse}), \quad (12)$$

$$d'_k = h_k^{dc}, \quad (13)$$

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。

4.7 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。

$$h_k^o = \text{GRU}_o(h_{k-1}^o, [e(t_{k-1}^o); d'_{k-1}]), \quad (14)$$

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。

$$t_k^o = \text{softmax}(Wf(W_{dec} h_k^o + V_{dec} d'_k)), \quad (15)$$

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。

5. 提案手法

我々は、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、長文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のモデルの問題ととらえ、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。また、原文やクエリのコンテキストベクトルをデコード時につねに用いることは、必ずしも適切であるとはいえない。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。

ここでは 4 章で説明したモデルに、文エンコーダとその

注意機構, および, Adaptive Attention Mechanism 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. Adaptive Attention Mechanism の構成を説明し, 最後に 4 章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.

5.1 Sentence Representation

本提案手法では, Sentence Representation として Skip-Thought [13] 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.

$$\vec{eh}_{m,n}^s = \text{GRU}_{\text{skip}}^{\text{forward}}(\vec{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (16)$$

$$\overleftarrow{eh}_{m,n}^s = \text{GRU}_{\text{skip}}^{\text{backward}}(\overleftarrow{eh}_{m,n-1}^s, e(t_{m,n}^s)), \quad (17)$$

$$eh_{m,n}^s = [\vec{eh}_{m,n}^s; \overleftarrow{eh}_{m,n}^s], \quad (18)$$

$$eh_m^s = eh_{m,l_{sm}}^s, \quad (19)$$

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_{sm} の特徴ベクトルである.

5.2 Sentence Encoder

単語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単語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 のみではとら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征ベクトル $eh^s = eh_1^s, \dots, eh_m^s, \dots, eh_{l_s}^s$ について, 我々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.

$$\vec{h}_m^s = \text{GRU}_d(\vec{h}_{m-1}^s, eh_m^s), \quad (20)$$

$$\overleftarrow{h}_m^s = \text{GRU}_d(\overleftarrow{h}_{m-1}^s, eh_m^s), \quad (21)$$

$$h_m^s = [\vec{h}_m^s; \overleftarrow{h}_m^s], \quad (22)$$

ここで, h_m^s は δ_7 次元の特徴ベクトルである.

5.3 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.

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s), \quad (23)$$

$$\alpha_{k,m}^s = \frac{\exp(a_{k,m}^s)}{\sum_{m'=1}^{l_s} \exp(a_{k,m'}^s)}, \quad (24)$$

$$s_k = \sum_{m=1}^{l_s} \alpha_{k,m}^s h_m^s, \quad (25)$$

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.

5.4 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, ある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.

$$\text{gate}_k = \sigma(W_{\text{gate}} e(t_{k-1}^o) + U_{\text{gate}} h_{k-1}^o + Z_{\text{gate}} v_k), \quad (26)$$

$$\hat{v}_k = \text{gate}_k \odot v_k + (1 - \text{gate}_k) \odot h_{l_v}^v, \quad (27)$$

ここで, $W_{\text{gate}} \in \mathbb{R}^{\delta_8 \times \delta_2}$, $U_{\text{gate}} \in \mathbb{R}^{\delta_8 \times \delta_5}$, $Z_{\text{gate}} \in \mathbb{R}^{\delta_8 \times \delta_8}$ 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.

5.5 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 2 に示す.

クエリ指向要約では, 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, 単語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

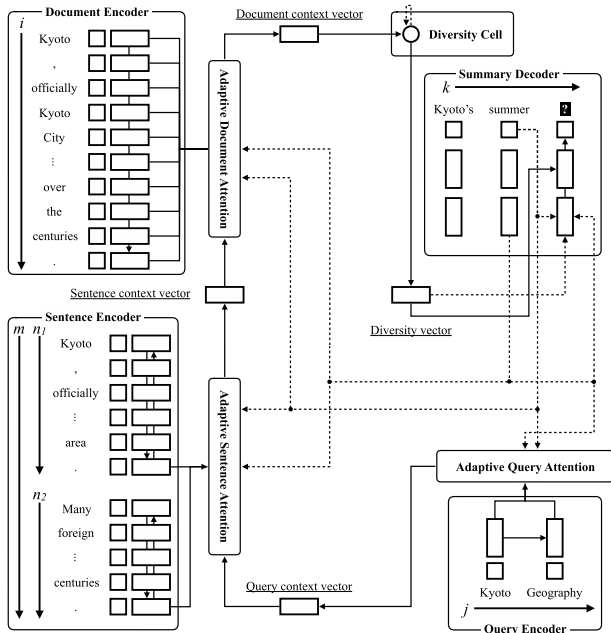


図 2 提案手法によって拡張された Diversity driven Attention Model. 3つのエンコーダ, および, それぞれの Adaptive Attention Mechanism, 1つのデコーダ, および, 1つの Diversity Cell から構成される. 破線はデコードステップ $k-1$ を表現し, 実線はデコードステップ k を表現する

Fig. 2 Proposed model is composed of the query encoder, the sentence encoder, the document encoder, the adaptive attention mechanism of each encoder, the summary decoder, and, the diversity cell. The broken line represents the processing at the decoding step $k-1$, and the solid line represents the processing at the decoding step k .

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 (23) から式 (28) へ, 以下の式に変更し, 導入することとした. また同様に, 単語単位の注意機構は, 式 (6) から式 (29) へ変更することとした.

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s + Z_q q_k), \quad (28)$$

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_s s_k). \quad (29)$$

また 5.4 節で提案した Adaptive Attention Mechanism を用いる場合は, 文単位の注意機構を, 式 (28) から式 (30) へ, 同様に, 単語単位の注意機構を, 式 (29) から式 (31) へ, 以下の式に変更することとした.

$$a_{k,m}^s = v_s^\top \tanh(W_s h_{k-1}^o + U_s h_m^s + Z_q \hat{q}_k), \quad (30)$$

$$a_{k,i}^d = v_d^\top \tanh(W_d h_{k-1}^o + U_d h_i^d + Z_s \hat{s}_k). \quad (31)$$

6. 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.

同様に, Adaptive Attention Mechanism の効果についても明らかにするために, Adaptive Attention Mechanism のみを比較モデルに導入した場合についても実験する.

まず実験 1 で, 比較モデルで用いられた Debatepedia データセットに対する各モデルの性能を比較する. 次に実験 2 で, 原文の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する.

6.1 実験 1: Debatepedia データセットにおける実験

6.1.1 設定

比較モデルとして 4 章で説明したモデルを用いた. 提案手法のモデルとして, 比較モデルに文エンコーダのみを導入したモデル 1, および, 比較モデルに Adaptive Attention Mechanism のみを導入したモデル 2 を用いた. また, 比較モデルに文エンコーダと Adaptive Attention Mechanism の両方を導入したモデル 3 も提案手法として用いた.

実験データとして, Nema ら [5] が Debatepedia^{*6} から構築した 13,573 件のデータセット^{*7} を用い, ミニバッチサイズ 16, 32, および, 64 の 3 種類について, 10 交差検証により性能を比較した. ただし, Nema らの実験では, モデルへのデータの受け渡しを行う際, 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, 本稿の実験では, より公平な実験とするためにこの増強は行っていない. 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, 本稿では, より広く用いられている DUC2004 Task2 の ROUGE パラメータを用いることとした. 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 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, テストデータにおいて 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, 本稿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.

6.1.2 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 カテゴリーを含む, 663 件の討論から構成されている. データセット中の

^{*6} Welcome to Debatepedia! - Debatepedia, the Wikipedia of Debates, http://www.debatepedia.org/en/index.php/Welcome_to_Debatepedia%21

^{*7} DiversityBasedAttentionMechanism/data at master PrekshaNema25/DiversityBasedAttentionMechanism, <https://github.com/PrekshaNema25/DiversityBasedAttentionMechanism/tree/master/data>

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。

6.1.3 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 1 (文エンコーダのみ導入)、モデル 2 (Adaptive Attention Mechanism のみ導入)、モデル 3 (両提案手法を導入)、および、比較モデルの各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^{*8} 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 以上の単語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。また提案手法で用いた、Skip-Thought の次元数、および、Sentence Encoder の次元数を 256 とした。

先行研究である比較モデルを参考に、各モデルは最適化手法として Adam [26] ($\alpha = 0.0004$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) を用い、コア数 1、Epoch 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^{*9} [27]、および、ChainerMN^{*10} [28] を用いた。また品詞タガーとして、The Stanford CoreNLP [29] を用いた。

6.1.4 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 [30] ($N = 1, 2$) 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^{*11} を用い、オプションは DUC 2004 の Task2 の設定^{*12} を適用した。実験結果を表 1 に示す。

6.1.5 考察

表 1 より、比較手法と比べ、文エンコーダのみを導入したモデル 1 の各 ROUGE の値が上回ることを確認できる。一方、Adaptive Attention Mechanism を導入したモデル 2

表 1 Debatepedia データセットに対する比較手法と各提案手法の 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

Table 1 Comparison between the models by ROUGE metrics for Debatepedia data set. The boldface indicates the maximum value in each mini-batch size.

モデル名	ミニバッチ	ROUGE-1	ROUGE-2	ROUGE-L
比較モデル	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

は、比較手法よりも各 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、すべての 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、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すべ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、このデータセットを構築する際、データを増強する手段として、原文、および、クエリの数単語について、単語埋め込み次元上で距離に近い他の単語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。

そこで我々は、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 2 に示す。表 2 より、整合性を確認した gold standard においても、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。

6.2 実験 2: Wikipedia データセットにおける実験

6.2.1 設定

実験データとして、English Wikipedia、および、Simple English Wikipedia から構築した 41,989 件のデータセット

^{*8} Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Main_Page

^{*9} Chainer: A flexible framework for neural networks, <https://chainer.org/>

^{*10} ChainerMN: Scalable distributed deep learning with Chainer, <https://github.com/chainer/chainermn>

^{*11} taguucci/pythonrouge: Python wrapper for evaluating summarization quality by ROUGE package, <https://github.com/taguucci/pythonrouge>

^{*12} An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems, <https://duc.nist.gov/pubs/2004slides/duc2004.intro.pdf>

表 2 人手で選択された gold standard 評価データに対する, 比較手法と各提案手法の ROUGE による比較

Table 2 Comparison between the models by ROUGE metrics for the gold standard test data which were manually selected from Debatepedia data set.

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
比較モデル	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

を用い, ホールドアウト検証により性能を比較した.

6.2.2 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, あまり多いとはいえない [31]. そのため我々は, Wikipedia と Simple English Wikipedia^{*13}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 1 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 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すべてを含む. たとえば, 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 のすべての記事数が 5,570,022 件に対して, Simple English Wikipedia のすべての記事数は 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, 評価データの内訳について表 3 に示す. なおデータセットには, D , Q , S の各トークン数のヒストグラムを参考に, トークン数がそれぞれ, 4,500, 6, 300 以下の記事のみを含めることとした.

6.2.3 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, 6.1 節の実験 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 [26] ($\alpha = 0.001$,

表 3 English Wikipedia と Simple English Wikipedia から作成したデータの件数

Table 3 Number of data created from English Wikipedia and Simple English Wikipedia.

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

表 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, 各バッチサイズにおける最大値を示す

Table 4 Comparison between the models by ROUGE metrics for Wikipedia data set. The boldface indicates the maximum value in each mini-batch size.

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
比較モデル	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

$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) を用い, コア数 4, バッチサイズ 3, Epoch 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.

6.2.4 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 [30] ($N = 1, 2$) および ROUGE-L により評価した.

クエリ指向文書要約において, 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うえで適切なデータとはいえない. 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 4 に示す.

また提案モデルは, 文エンコーダを導入することで, トークン数が大きい原文 D が入力された場合でも, 頑健に働くことが期待される. そこで我々は, 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 3, 図 4, 図 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.

6.2.5 考察

表 4 より, 比較手法と比べ, 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 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 に文エンコーダの導入したモデル 3 では, すべての ROUGE の値が低下したこ

^{*13} Wikipedia, https://simple.wikipedia.org/wiki/Main_Page

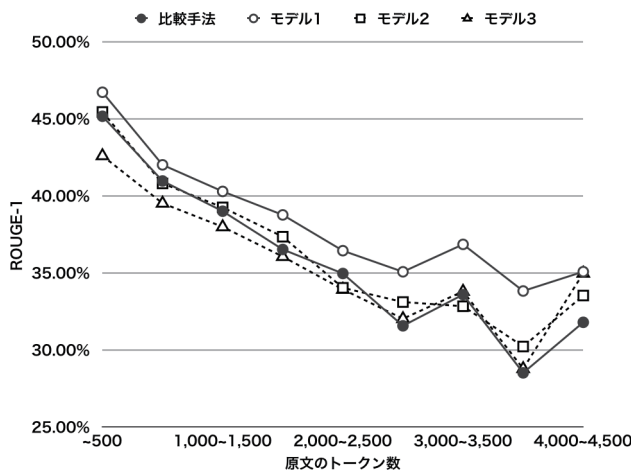


図 3 原文のトークン数に対する, ROUGE-1 による各モデルの比較
Fig. 3 Comparison between the models by ROUGE-1 metric for the number of tokens in the original texts.

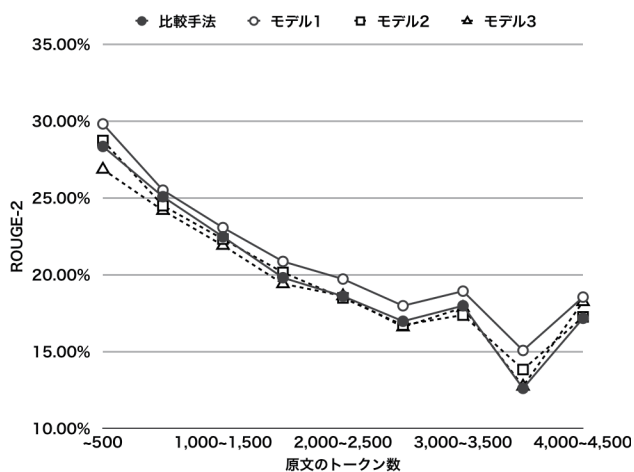


図 4 原文のトークン数に対する, ROUGE-2 による各モデルの比較
Fig. 4 Comparison between the models by ROUGE-2 metric for the number of tokens in the original texts.

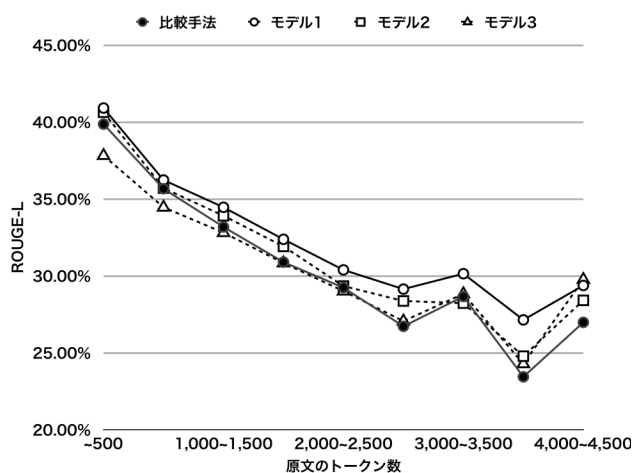


図 5 原文のトークン数に対する, ROUGE-L による各モデルの比較
Fig. 5 Comparison between the models by ROUGE-L metric for the number of tokens in the original texts.

とを確認できる. 図 3, 図 4, および, 図 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 をすべ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分かる. 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 の導入による一貫した性能向上は確認されなかった.

各手法の出力例を表 5, 表 6, 表 7 に示す. 表 5 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 6 は, 表 5 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. 比較モデル, 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いえない.

表 7 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 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分かる. 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~表 7 から, 各モデルがトークン単位の繰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返し生成することも確認できる. これは, Diversity Cell により同じ単語を繰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では, 繰返しの周期が単語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.

7. まとめ

本稿では, 原文の単語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, 要約を生成する手法を提案した. また, 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.

実験により, 入力される原文の平均トークン数の大小によらず, 文エンコーダの導入によって ROUGE の値が改善されることが分かった. また, 原文の平均トークン数が大

表 5 出力された要約文の一例（記事タイトル：Lemon shark, クエリ：Lemon shark）
Table 5 An example of summarizing the English Wikipedia article “Lemon shark”, with the given query “Lemon shark”.

[illegible]

きい Wikipedia データセットにおいては、文エンコーダのみを導入した提案手法の各 ROUGE が大きく改善されることを確認できた。一方、6.1 節の実験で用いた Debapedia 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。今後は、他の先行研究で用いられたクエリ指向性要約のデータセットに対しても性能を検証する予定である。

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

参考文献

- さい Wikipedia データセットにおいては，文エンコーダのみを導入した提案手法の各 ROUGE が大きく改善されることを確認できた．一方，6.1 節の実験で用いた 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．今後は，他の先行研究で用いられたクエリ指向性要約のデータセットに対しても性能を検証する予定である．
- 謝辞 本研究の一部は科研費 18K11557 の助成を受けたものです．ここに記して感謝の意を表します．
- 参考文献
- [1] Koehn, P. and Knowles, R.: Six Challenges for Neural Machine Translation, *Proc. 1st Workshop on Neural Machine Translation*, pp.28–39, Association for Computational Linguistics (online), DOI: 10.18653/v1/W17-3204 (2017).
- [2] Rush, A.M., Chopra, S. and Weston, J.: A Neural Attention Model for Abstractive Sentence Summarization, *Proc. 2015 Conference on Empirical Methods in Natural Language Processing*, pp.379–389, Association for Computational Linguistics (online), DOI: 10.18653/v1/D15-1044 (2015).
- [3] Sutskever, I., Vinyals, O. and Le, Q.V.: Sequence to sequence learning with neural networks, *Advances in Neural Information Processing Systems*, pp.3104–3112 (2014).
- [4] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *CoRR*, Vol.abs/1409.0473 (2014) (online), available from <http://arxiv.org/abs/1409.0473>.
- [5] Nema, P., Khapra, M.M., Laha, A. and Ravindran, B.: Diversity driven attention model for query-based abstractive summarization, *Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume*

表 6 出力された要約文の一例 (記事タイトル: Lemon shark, クエリ: Lemon shark habitat)

Table 6 An example of summarizing the English Wikipedia article “Lemon shark”, with the given query “Lemon shark habitat”.

クエリ : Lemon shark habitat
正解要約 : the lemon shark lives in the tropical and subtropical waters of the coastal areas of the atlantic and pacific oceans. they stay in moderately shallow water, normally going no deeper than eight zero meters (roughly two six zero feet). they are often found in shallow waters near coasts and islands, by coral reefs, mangroves, bays, and sometimes even river mouths.
比較モデル : the lemon shark feeds on octopuses and squids. they are usually caught in the pelagic zone. the lemon shark feeds on octopuses and squids. they are usually caught in the pelagic zone. the lemon shark feeds on octopuses and squids. they are also caught in the pelagic zone. the lemon shark feeds on octopuses and squids. they are also caught in the pelagic zone. the lemon shark feeds on octopuses and squids. they are also caught in the pelagic zone. the lemon shark feeds on octopuses and squids. they are also caught in the pelagic zone.
モデル 1 : the lemon shark is found in the eastern pacific, south cuba and ecuador. they are found in the eastern pacific, south cuba and ecuador.
モデル 2 : lemon shark shark feeds on prey that live in shallow subtropical seas. they are found in shallow subtropical seas, and are known to eat crustaceans and other species.
モデル 3 : the lemon shark is a flap of the skin of the skin, and the deepest shark is the most common species. it is a hybrid of shark. it is the most common species of sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks. it is the most common area sharks are known to have a long time. it is also known as the “<unk> viper”, and the “<unk> viper”.

表 7 出力された要約文の一例 (記事タイトル: Thanjavur district, クエリ: Thanjavur district)

Table 7 An example of summarizing the English Wikipedia article “Thanjavur district”, with the given query “Thanjavur district”.

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. ...
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.
比較モデル : thanjavur district is one of the three two districts of the state of tamil nadu, in southeastern uttar india. its head-quarters 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 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
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.
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.
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>.

- 1: *Long Papers*), pp.1063–1072, Association for Computational Linguistics (online), DOI: 10.18653/v1/P17-1098 (2017).
- [6] Luhn, H.P.: The automatic creation of literature abstracts, *IBM Journal of Research and Development*, Vol.2, No.2, pp.159–165 (1958).
- [7] Edmundson, H.P.: New methods in automatic extracting, *Journal of the ACM (JACM)*, Vol.16, No.2, pp.264–285 (1969).
- [8] Filippova, K. and Strube, M.: Dependency Tree Based Sentence Compression, *Proc. 5th International Natural Language Generation Conference, INLG '08*, pp.25–32, Association for Computational Linguistics (2008) (online), available from (<http://dl.acm.org/citation.cfm?id=1708322.1708329>).
- [9] Hirao, T., Isozaki, H., Maeda, E. and Matsumoto, Y.: Extracting Important Sentences with Support Vector Machines, *Proc. 19th International Conference on Computational Linguistics - Volume 1, COLING '02*, pp.1–7, Association for Computational Linguistics (online), DOI: 10.3115/1072228.1072281 (2002).
- [10] Conroy, J.M. and O'leary, D.P.: Text Summarization via Hidden Markov Models, *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01*, pp.406–407, ACM (online), DOI: 10.1145/383952.384042 (2001).
- [11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems 26*, Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z. and Weinberger, K.Q. (Eds.), pp.3111–3119, Curran Associates, Inc. (2003) (online), available from (<http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>).
- [12] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proc. 31st International Conference on Machine Learning - Volume 32, ICML '14*, pp.II-1188–II-1196, JMLR.org (2014) (online), available from (<http://dl.acm.org/citation.cfm?id=3044805.3045025>).
- [13] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A. and Fidler, S.: Skip-Thought Vectors, *Advances in Neural Information Processing Systems 28*, Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M. and Garnett, R. (Eds.), pp.3294–3302, Curran Associates, Inc. (2015) (online), available from (<http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf>).
- [14] Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A.: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*, pp.670–680, Association for Computational Linguistics (online), DOI: 10.18653/v1/D17-1070 (2017).
- [15] Nie, A., Bennett, E.D. and Goodman, N.D.: DisSent: Sentence Representation Learning from Explicit Discourse Relations, *CoRR*, Vol.abs/1710.04334 (2017) (online), available from (<http://arxiv.org/abs/1710.04334>).
- [16] Tan, J., Wan, X. and Xiao, J.: Abstractive Document Summarization with a Graph-Based Attentional Neural Model, *Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.1171–1181, Association for Computational Linguistics (online), DOI: 10.18653/v1/P17-1108 (2017).
- [17] Li, J., Luong, T. and Jurafsky, D.: A Hierarchical Neural Autoencoder for Paragraphs and Documents, *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp.1106–1115, Association for Computational Linguistics (online), DOI: 10.3115/v1/P15-1107 (2015).
- [18] Li, W., Xiao, X., Lyu, Y. and Wang, Y.: Improving Neural Abstractive Document Summarization with Explicit Information Selection Modeling, *Proc. 2018 Conference on Empirical Methods in Natural Language Processing*, pp.1787–1796, Association for Computational Linguistics (2018) (online), available from (<https://www.aclweb.org/anthology/D18-1205>).
- [19] Zhou, Q., Yang, N., Wei, F. and Zhou, M.: Selective Encoding for Abstractive Sentence Summarization, *Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.1095–1104, Association for Computational Linguistics (online), DOI: 10.18653/v1/P17-1101 (2017).
- [20] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. and Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, *Proc. 32nd International Conference on Machine Learning*, Bach, F. and Blei, D. (Eds.), Vol.37, pp.2048–2057, PMLR (2015) (online), available from (<http://proceedings.mlr.press/v37/xuc15.html>).
- [21] Dhingra, B., Liu, H., Yang, Z., Cohen, W. and Salakhutdinov, R.: Gated-Attention Readers for Text Comprehension, *Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.1832–1846, Association for Computational Linguistics (online), DOI: 10.18653/v1/P17-1168 (2017).
- [22] Yousefiazar, M., Sirts, K., Hamey, L. and Aliod, D.M.: Query-based single document summarization using an ensemble noisy auto-encoder, *Proc. Australasian Language Technology Association Workshop 2015*, pp.2–10 (2015).
- [23] Cao, Z., Li, W., Li, S., Wei, F. and Li, Y.: AttSum: Joint Learning of Focusing and Summarization with Neural Attention, *Proc. COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp.547–556, The COLING 2016 Organizing Committee (2016) (online), available from (<http://aclweb.org/anthology/C16-1053>).
- [24] Kiddon, C., Zettlemoyer, L. and Choi, Y.: Globally Coherent Text Generation with Neural Checklist Models, *EMNLP*, pp.329–339 (2016).
- [25] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1724–1734, Association for Computational Linguistics (online), DOI: 10.3115/v1/D14-1179 (2014).
- [26] Kingma, D.P. and Ba, J.: Adam: A method for stochastic optimization, *International Conference on Learning Representations (ICLR)* (2015).
- [27] Tokui, S., Oono, K., Hido, S. and Clayton, J.: Chainer: A next-generation open source framework for deep learning, *Proc. Workshop on Machine Learning Systems (LearningSys) in the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, Vol.5 (2015).

- [28] Akiba, T., Fukuda, K. and Suzuki, S.: Chain-erMN: Scalable Distributed Deep Learning Framework, *Proc. Workshop on ML Systems in the 31st Annual Conference on Neural Information Processing Systems (NIPS)* (2017) (online), available from http://learningsys.org/nips17/assets/papers/paper_25.pdf.
- [29] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. and McClosky, D.: The Stanford CoreNLP natural language processing toolkit, *Proc. 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.55-60 (2014).
- [30] Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries, *Text Summarization Branches Out: Proc. ACL-04 Workshop*, Marie-Francine Moens, S.S. (Ed.), pp.74-81, Association for Computational Linguistics (2004).
- [31] Dernoncourt, F., Ghassemi, M. and Chang, W.: A Repository of Corpora for Summarization, *Proc. 11th International Conference on Language Resources and Evaluation (LREC 2018)* (chair), Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. and Tokunaga, T. (Eds.), European Language Resources Association (ELRA) (2018).

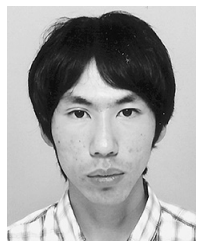


宮森 恒 (正会員)

1992年早稲田大学理工学部電子通信学科卒業, 1994年同大学大学院理工学研究科修士課程修了, 1997年同大学院理工学研究科後期博士課程修了. 1996~1997年同大学理工学部助手. 1997年郵政省通信総合研究所入所. 独立行政

法人情報通信研究機構主任研究員サブグループリーダー兼務を経て, 2008年京都産業大学コンピュータ理工学部准教授. 2013年より同大学同学部教授. 工学博士. 主に, マルチメディアデータ工学, パターン認識, 情報検索に関する研究に従事. 2006年日本データベース学会平成17年度論文賞. ACM, 電子情報通信学会, 日本データベース学会, 人工知能学会, 言語処理学会, 映像情報メディア学会各会員.

(担当編集委員 土田 正明)



木村 輔 (学生会員)

2013年京都産業大学コンピュータ理工学部インテリジェントシステム学科卒業. 2016年同大学大学院先端情報学研究科博士前期課程修了. 現在, 同大学院先端情報学研究科博士後期課程在学中. 主に, 自然言語処理, 自動テ

キスト要約, 情報抽出, パターン認識, 質問応答の研究に従事. 日本データベース学会会員.



田上 諒 (正会員)

2017年京都産業大学コンピュータ理工学部ネットワークメディア学科卒業. 2019年同大学大学院先端情報学研究科博士前期課程修了. 主に, 情報検索, 自然言語処理, 質問応答に関する研究に従事. 日本データベース学会

会員.