

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

木村 輔[†] 田上 諒[†] 宮森 恒[†]

[†] 京都産業大学 大学院 先端情報学研究科 〒603-8555 京都府京都市北区上賀茂本山

E-mail: [†]{i1658047,i1788124,miya}@cc.kyoto-su.ac.jp

あらまし 膨大な量の文書が溢れる現代において、自動文書要約のニーズは高まっているが、現実の利用を考えた場合、単に、文書全体を要約するのではなく、与えられた観点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。非クエリ指向の文書要約では、一般的に、原文書全体の内容を損なわない要約の生成が要求されるため、原文書の文頭や接続詞による言い換え表現などの、原文書の談話構造に関連する手掛かりを活用しやすい。一方、クエリ指向の文書要約では、与えられた観点によって要約に含めるべき文の種類が変化するため、原文書が持つ談話構造を要約にそのまま反映するのは、必ずしも適切とはいえない。そこで我々は、与えられたクエリを目標ベクトルに変換し、文単位のベクトルで表された文書とともに常に注意機構に埋め込むことによって、与えられた観点に沿った一貫性を保持した要約を生成する手法を提案する。実験では、目標ベクトルの有無により、生成される要約にどのような影響があるのか検証する。

キーワード クエリ指向文書要約, 要約観点, 一貫性, 深層学習, 自然言語処理

1. はじめに

インターネットの継続的な発展に伴い、電子メール、ウェブニュース、マイクロブログや動画共有サービスをはじめ、多種多様なデジタルデータが絶えず生産、流通し、蓄積されている。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^(注1)。

このよう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ため、検索エンジンやニュース配信サービスでは、スニペットや、ニュース記事を要約したリード文といった、原文書の要約が提供されることが多い。このような背景により、与えられた文書を自動要約する研究は、盛んに取り組まれている。

自動要約は、どのような観点でまとめた要約を生成するかで2種類に分類できる。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、原文書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、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、原文書を構成する内容を出来る限り損なわずに要約する必要があるため、これまでの研究では、原文書における重要文を、いかに重複なく抽出、圧縮、生成できるかに焦点が当てられてきた。例えば、原文書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、原文書の文頭や接続詞による言い換え表現などの、原文書の談話構造に関連する手掛かりを活用している。また、原文書の文を一度ベクトルへ変換し、そのベクトルから要約を出力す

る生成型では、注意機構によって、入力された原文書において、どの単語が重要であるかを、入力単語や直前の出力単語から逐次注視できる手法が提案されている。一方、クエリ指向文書要約は、記述すべき内容によって要約に含めるべき文の種類が変化するため、これまで提案された手法では、文書検索と同様に、ユーザから与えられたクエリを用いて、原文書の各文の関連性としてTF-IDFなどによって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[14]が提案した、深層モデルに基づく生成型の自動要約が一定の成功を取ったことで、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 [15] という機械翻訳において提案されたモデルに、Attention [1] という注意機構を備えることで、入力された原文書において、どの単語が重要であるかを逐次注視しながら要約文を出力できる。Nemaら[13]は、深層モデルに基づく生成型のクエリ指向文書要約を行う Encoder-Decoder モデルを提案している。このモデルでは、入力の原文書の各単語ベクトルに加えて、クエリにも Attention を用いることで、各時間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問題を解決するために、Attention によって生成される文脈ベクトルが、各時間ステップにおいて直交するよう変換する手法を提案している。しかし、このモデルは、入力された文書を、1つの単語シーケンスと捉え、GRU によって中間表現へ変換しているため、文書長が非常に長くなった場合、先頭部分の情報が十分に保持されるとは限らない。

本稿では、生成型の自動文書要約において、入力に、原文書を表す文単位のベクトルの集合、および、要約内容が目指すべき観点を目標ベクトルとして与えることで、要約観点の一貫性を保持したクエリ指向の要約を出力する Encoder-Decoder モデルを提案する。提案モデルでは、ユーザのクエリから生成した目標ベクトルを入力に含むことより、要約に含めるべき内容

(注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>

をユーザの要求に応じて柔軟に変化できる自動要約を目指した。出力単語ベクトルの生成に際して、常に目標ベクトルを埋め込むことによって、要約観点の一貫性を保持した要約の生成が期待できる。また、文書を文単位のベクトルのシーケンスとして表現することで、文書単位のベクトルで表す場合と比べ、入力された文書の文書長が非常に長い場合にも、入力の先頭部分の情報を保持できることが期待される。

本稿の構成は以下の通りである。2章では、関連研究と本研究の位置付けについて述べ、3章では、提案手法の目的とシステム構成について詳述する。4章で実験とその結果、5章で考察を述べる。最後に、6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、Luhnら[11]が提案した、原文書中の重要文を抽出し、要約として出力する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、原文書中の、出現頻度が高い重要語[11]、文の位置情報[4]、接続詞のような手掛かり語[4]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、文圧縮も盛んに研究されていた。主に、文圧縮は、原文書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)^(注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、続いて、Document Understanding Conference (DUC)^(注3)や、日本では国立情報学研究所(NII^(注4))が主催するNTCIRのタスクとして、Text Summarization Challenge (TSC)^(注5)が開催されたことで、多くの原文書とその要約のデータセットが整備された。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、SVMや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盛んに研究されている。ここでは、本研究と関連するクエリ指向の自動文書要約について述べる。

抽出型の研究として、Yousefiazarら[18][17]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、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。

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

(注3) : Document Understanding Conferences : <http://duc.nist.gov>

(注4) : 国立情報学研究所/National Institute of Informatics : <http://www.nii.ac.jp/>

(注5) : Text Summarization Challenge Home Page : <http://1r-www.pi.titech.ac.jp/tsc/>

また、Caoら[2]は、文書や文書クラスタのための分散表現学習と、Attention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[7]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、出力されるレシピの記述内容に一貫性を持たせるために、料理名をゴールベクトルへ変換し、Decode時の入力にする手法を提案している。また、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。また、Nemaら[13]は、深層モデルに基づく生成型のクエリ指向文書要約のEncoderDecoderモデルを提案している。このモデルでは、入力の原文書の各単語ベクトルに加えて、クエリにもAttentionを用いることで、各時間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て出力する問題を解決するために、Attentionによって生成される文脈ベクトルが、各時間ステップにおいて直交するよう変換する手法を提案している。

本稿では、目標ベクトルを用いた、クエリ指向自動要約のための生成モデルを提案する。本モデルは、KiddonらやNemaらと同様に、クエリにあたる入力をベクトル表現に変換し、要約生成に用いることで、要約内容の一貫性の保持を目指している。一方、入力する原文書については、先行研究のように単語単位のベクトルへ変換せず、文単位のベクトルへ変換することで、より意味を考慮した要約が生成できるモデルの構築を目指している。

3. 提案手法

本節では、目標ベクトルを組み込んだ、クエリ指向自動要約生成モデルについて説明する。図1に、我々の提案するモデルのネットワークを示す。

なお、簡単のため、各式中の変換行列 W は、バイアス項 b を含むものとする。

3.1 入力変数の定義

要約対象である i 番目の原文書 $D^{(i)}$ は、文 $X_j^{(i)}$ のシーケンスと定義する。よって、 $D^{(i)} = \{X_1^{(i)}, \dots, X_j^{(i)}, \dots, X_J^{(i)}\}$ と表現できる。また、 $X_j^{(i)}$ 、および、 $D^{(i)}$ に対応する目標 $G^{(i)}$ は、それぞれの語彙、 $doc_V = \{doc_w_1, \dots, doc_w_m, \dots, doc_w_{|doc_V|}\}$ 、および、 $goal_V = \{goal_w_1, \dots, goal_w_n, \dots, goal_w_{|goal_V|}\}$ のシーケンスで定義される。ここで、 $|doc_V|$ 、 $|goal_V|$ は、 doc_V および $goal_V$ のサイズである。

3.2 Encoder

提案モデルのEncoderの目的は、原文書 $D^{(i)}$ の各文 $X_j^{(i)}$ 、および、要約の目標 $G^{(i)}$ を、それぞれ、文ベクトル $h_X_j^{(i)}$ の集合 $h_D^{(i)}$ 、および、目標ベクトル $h_G^{(i)}$ の中間表現へ変換することである。

本モデルでは、次式によって、入力された各文を中間ベクトルへ変換する。ここで、 $LSTM$ は、Hochreiterらが提案したLong-Short Term Memory [6] を示す。また、 $h_X_j^{(i)}$ によって

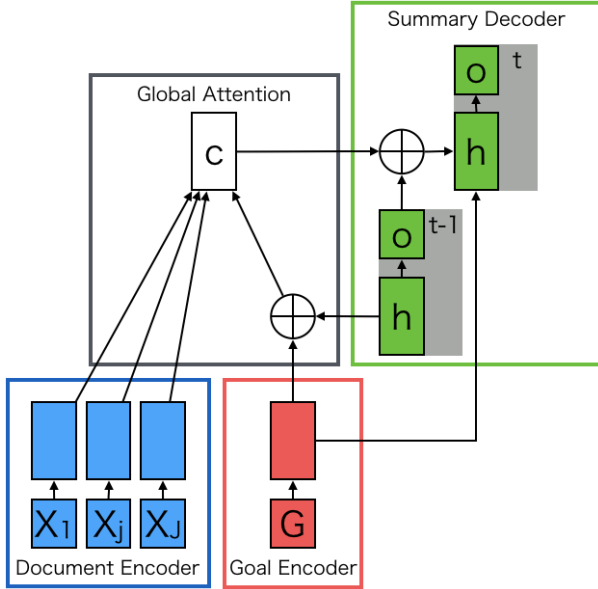


図1 提案モデルのEncoderでは、原文書 $D^{(i)}$ に含まれる各文 $X_j^{(i)}$ を文ベクトルへ変換する。また、ユーザのクエリ $G^{(i)}$ を目標ベクトルへ変換する。Decoderでは、目標ベクトルと時刻 $t-1$ のDecoderの隠れ層 $h_{t-1}^{(i)}$ を用いて、各時刻 t における各文ベクトルへのAttentionを算出し、文脈ベクトルを得る。また、目標ベクトルを考慮したGRU (GRUGV) から出力された $h_t^{(i)}$ と文脈ベクトルから、要約文の単語 $o_t^{(i)}$ を推定する。

表現された原文書を $h_D^{(i)} = \{h_X_1^{(i)}, \dots, h_X_j^{(i)}, \dots, h_X_J^{(i)}\}$ と定義する。

$$h_X_{j_m}^{(i)} = LSTM(W^{doc} doc_w_m, h_X_{j_m-1}^{(i)}) \quad (1)$$

$$h_G_n^{(i)} = LSTM(W^{goal} goal_w_n, g_X_{n-1}^{(i)}) \quad (2)$$

ここで、学習対象の行列は、 W^{doc} , W^{goal} である。また、 $LSTM$ 内の各重み行列も学習対象となる。

3.3 Decoder

提案モデルのDecoderの目的は、時刻 t における文脈ベクトル $context_t^{(i)}$ と、中間表現 $h_t^{(i)}$ を入力とし、時刻 t における要約文の単語ベクトル o_t を出力することである。

$$o_t^{(i)} = \tanh(W_h h_t^{(i)}) \quad (3)$$

また、要約文の単語 w_t は、次の式を用いて o_t から求められる。

$$w_t = \text{softmax}(W_o o_t^{(i)}) \quad (4)$$

ここで、学習対象の行列は、 W_h , W_o である。

3.3.1 Gated Recurrent Unit with Goal Vector

中間表現 $h_t^{(i)}$ を得るために、Decoderでは、独自の改良を加えたGated Recurrent Unit (GRU) [3], Gated Recurrent Unit with Goal Vector (GRUGV) を用いて、目標ベクトル $h_G^{(i)}$ と、時刻 $t-1$ における出力ベクトル o_{t-1} と文脈ベクトル $context_t^{(i)}$ を連結したベクトルを入力とし、ある時刻 t の中間表現 $h_t^{(i)}$ を出力する。ここでは、一般的なGRUについて説明したのち、 $h_G^{(i)}$ を入力できるように改良したGRUについて詳述する。

GRUは、各時間 t における入力 x_t と、一つ前の隠れ層 h_{t-1} が与えられたとき、次の隠れ層 h_t を出力するRNNの一つである。

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (5)$$

つまり、 h_t は、更新ゲート z_t によって、一つ前の隠れ層 h_{t-1} と新しく生成された隠れ層 \tilde{h}_t の含有率が調整されることで生成される。ここで、 \tilde{h}_t , z_t は、それぞれ次の式で計算される。

$$\tilde{h}_t = \tanh(W_{\tilde{h}} x_t + U_{\tilde{h}} r_t \odot h_{t-1}) \quad (6)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (7)$$

また、忘却ゲートである r_t は、次の式で計算される。

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (8)$$

ここで、学習対象の行列は、 W_z , W_r , $W_{\tilde{h}}$, U_z , U_r , $U_{\tilde{h}}$ である。

目標ベクトル $h_G^{(i)}$ を入力するために、我々は、Kiddonら[7]の手法を参考に、GRUの \tilde{h}_t の計算式を改良し、それに伴って、目標ゲート s_t を追加した。

$$\tilde{h}_t = \tanh(W_{\tilde{h}} x_t + U_{\tilde{h}} r_t \odot h_{t-1} + U_g s_t \odot h_G^{(i)}) \quad (9)$$

$$s_t = \sigma(W_s x_t + U_s h_{t-1}) \quad (10)$$

ここで、 W_s , U_s , U_g は、新たに追加された学習対象の行列である。

最後に、文脈ベクトル $context_t$ を得るため、時刻 t における $h_D^{(i)}$ のアテンションを次式で算出する。

$$context_t^{(i)} = \sum_{j=1}^T \alpha_j h_X_j^{(i)} \quad (11)$$

$$\alpha_j = \text{softmax}(e_{jt}) \quad (12)$$

$$e_{jt} = W_a \tanh(W_e h_X_j^{(i)} + W_d [h_t^{(i)}; h_G^{(i)}]) \quad (13)$$

ここで、学習対象の行列は、 W_a , W_e , W_d である。

3.4 損失関数

目標 $G^{(i)}$ が与えられたときの、ある原文書 $D^{(i)}$ におけるモデルの損失 $loss^{(i)}$ は、次の式で計算され、提案モデルの最適化対象の損失 $loss$ は、 $loss = \sum_{i=1}^I loss^{(i)}$ となる。

$$loss^{(i)} = - \sum_{t=0}^{T_i} \log p(w_t^{(i)} | w_1^{(i)}, \dots, w_{t-1}^{(i)}, D^{(i)}, G^{(i)}; \theta) \quad (14)$$

4. 評価実験

本実験の目的は、提案手法である目標ベクトルを考慮した要約生成モデルと、目標ベクトルを用いない要約生成モデルの精度を比較することで、目標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。提案手法と比較するモデルとして、3.3.1項で説明したGRUGVを、GRUに置き換えたベースラインモデルを用いた。図2にベースライン手法のモデルのネットワークを示す。

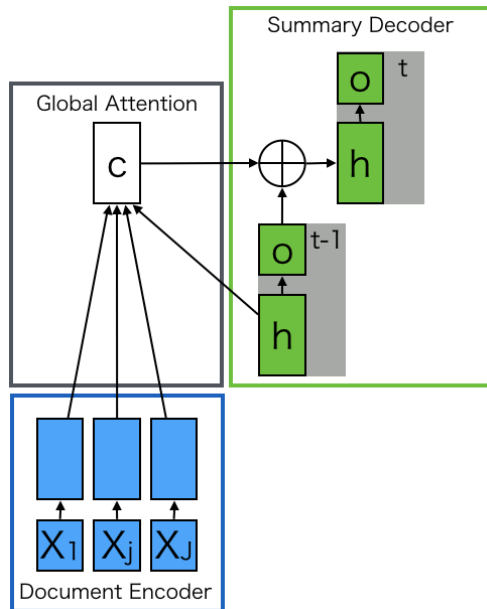


図2 比較実験に用いるベースライン手法のモデル

4.1 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。そのため、我々は、Wikipedia^(注6)と Simple English Wikipedia^(注7)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、 $(D^{(i)}, G^{(i)}, S^{(i)})$ のタプル形式で構成される。ここで、 $D^{(i)}$ を、あるタイトルを持つ Wikipedia 一つの記事、 $G^{(i)}$ を、 $D^{(i)}$ に対応する Simple English Wikipedia のあるセクションのタイトル、 $S^{(i)}$ を、 $G^{(i)}$ 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、 $S^{(i)}$ 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、5,570,022 件に対して、Simple English Wikipedia の全ての記事は、131,459 件と、2.4%程度しかないため、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 50,000 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、テストデータの内訳について表 1 に示す。なお、データセットには、 $D^{(i)}$ 、 $G^{(i)}$ 、 $S^{(i)}$ の各文書長のヒストグラムを参考に、文書長がそれぞれ、4,500、6、

(注6) : Wikipedia, the free encyclopedia : https://en.wikipedia.org/wiki/Main_Page

(注7) : Wikipedia : https://simple.wikipedia.org/wiki/Main_Page

300 以下の記事のみを含めることとした。

表 1 作成したデータセットの内訳

データセット	件数
訓練データ	35,179
開発データ	3,909
テストデータ	395

4.2 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 、 G 、および、 S の語彙の次元を 20,000、単語埋め込みベクトルの次元を 256、Encoder の隠れ層の次元を 256、Attention の隠れ層の次元を 256、Decoder の隠れ層の次元を 256 とした。各語彙については、最大語彙サイズを 20,000 とし、出現頻度が 2 以上の単語を語彙として用いた。また、 D 、 G では共通の語彙を用いた。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、16,957,985、ベースライン手法のモデルでは、16,629,537 となった。各モデルは、最適化手法として Adam [8] ($\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$) を用い、ミニバッチサイズ 16、Epoch 10 で学習した。それぞれ最終的な損失は、提案手法のモデルで 227.26、ベースライン手法のモデルで 240.18 となった。モデルの実装には、Chainer [16]^(注8)を用いた。また、形態素解析器として、The Stanford CoreNLP [12] を用いた。

4.3 実験結果

本実験の目的は、目標ベクトルを考慮した要約生成モデルと、目標ベクトルを用いない要約生成モデルを比較することで、目標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 [10] ($N=1,2$) により評価した。また、生成する要約文の最大長は、4.1 節の正解要約の文の長さの調査より 300 とした。実験結果を図 3 に示す。

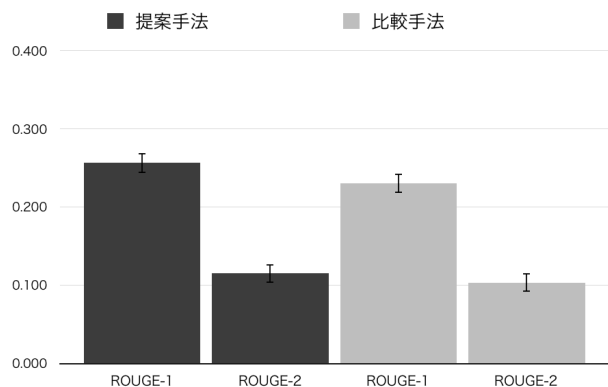


図3 提案手法とベースライン手法の ROUGE-N ($N=1,2$) による実験結果。エラーバーは標準誤差を示している。

図 3 より、要約生成の過程に目標ベクトルを用いないベースライン手法のモデルと比べ、目標ベクトルを考慮した提案手法

(注8) : Chainer: A flexible framework for neural networks : <https://chainer.org/>

のモデルの ROUGE-1, ROUGE-2 が共に高いことがわかる。

ここで、テストデータセットの、 D , G , および、 S と、各モデルが生成した要約文中のそれぞれの未知語の使用率の平均値を表 2 に示す。表 2 より、 G における未知語使用率が高い

表 2 未知語の割合の平均値

種別	未知語使用率の平均値
D	0.072
G	0.374
S	0.057
生成要約 (提案手法)	0.197
生成要約 (ベースライン手法)	0.181

ことがわかる。 G の入力に、精度に与える影響を確認することが実験の目的であるため、未知語使用率が高い目標 $G^{(i)}$ を含むデータの結果は、実験結果として適しているとは言い難い。よって、図 3 の各データのうち、 G の未知語使用率の平均値以上を除いた 219 件における実験結果を図 4 に示す。また、同様に、各手法が生成した要約文においても、未知語使用率が高いことがわかる。よって、図 3 の各生成要約のうち、各手法における生成要約の未知語使用率の平均値以上を除いた、それぞれ 253 件、232 件における実験結果を図 5 に示す。

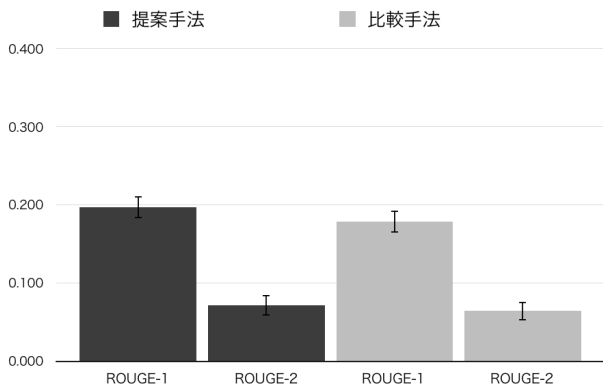


図 4 G の未知語利用率の高いテストデータを除いた、提案手法とベースライン手法の ROUGE-N ($N=1,2$) による実験結果。エラーバーは標準誤差を示している。

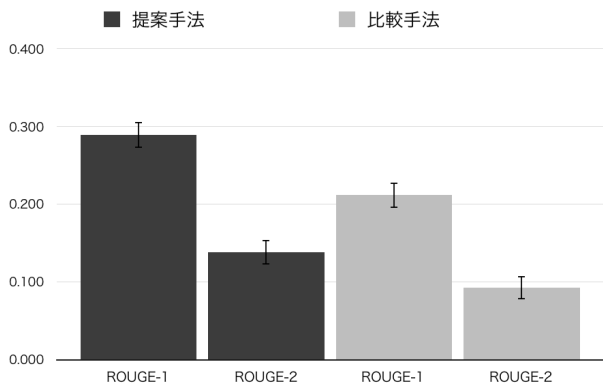


図 5 各生成要約の未知語利用率の高いテストデータを除いた、提案手法とベースライン手法の ROUGE-N ($N=1,2$) による実験結果。エラーバーは標準誤差を示している。

図 4, 5 にいずれにおいても、要約生成の過程に目標ベクトルを用いないベースライン手法のモデルと比べ、目標ベクトルを考慮した提案手法のモデルの ROUGE-1, ROUGE-2 が共に高いことがわかる。

また、評価指標として用いた ROUGE-N は、システム要約の文書長が長くなるとスコアが高くなる傾向を持つ。そこで、図 3 の各データのうち、システム要約の文書長が、0 ~ 99, 100 ~ 199, 200 ~ 299 のそれぞれに 3 種類に分け集計したグラフを、図 6 ~ 図 8 に示す。

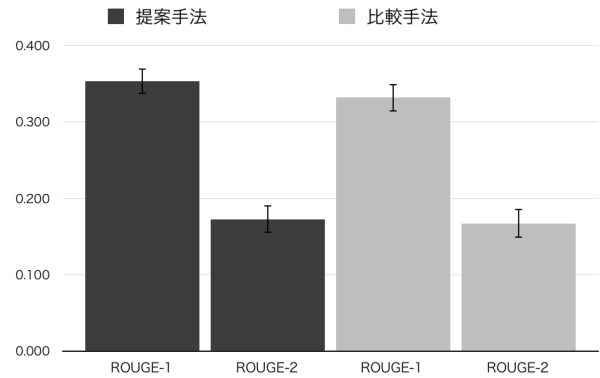


図 6 システム要約の文書長が 99 以下のテストデータにおける、提案手法とベースライン手法の ROUGE-N ($N=1,2$) による実験結果。なお、各データの件数は、提案手法が 247 件、ベースライン手法が 225 件である。また、エラーバーは標準誤差を示している。

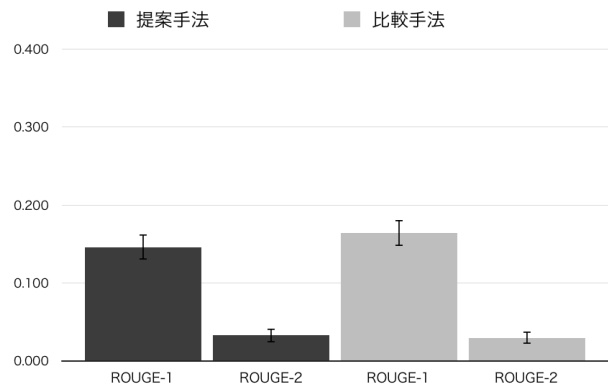


図 7 システム要約の文書長が 100 以上、199 以下のテストデータにおける、提案手法とベースライン手法の ROUGE-N ($N=1,2$) による実験結果。なお、各データの件数は、提案手法が 31 件、ベースライン手法が 29 件である。また、エラーバーは標準誤差を示している。

図 6 より、生成したシステム要約の文書長が 99 以下の場合、要約生成の過程に目標ベクトルを用いないベースライン手法のモデルと比べ、目標ベクトルを考慮した提案手法のモデルの ROUGE-1, ROUGE-2 が共に高いことがわかる。一方、図 7, および、図 8 より、生成したシステム要約の文書長が 100 以上の場合、ベースライン手法のモデルと比べ、提案手法のモデルの ROUGE-1, ROUGE-2 がわずかに低いことが確認できる。

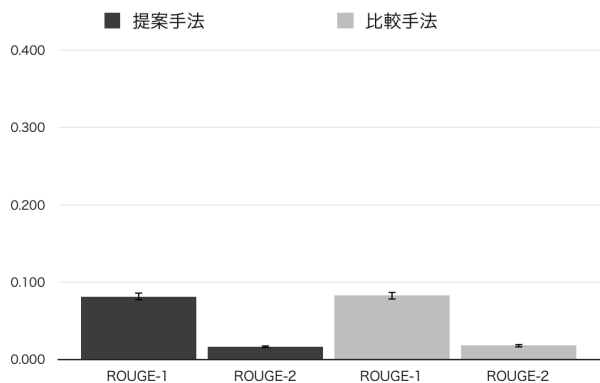


図 8 システム要約の文書長が 200 以上、299 以下のテストデータにおける、提案手法とベースライン手法の ROUGE-N (N=1,2) による実験結果。なお、各データの件数は、提案手法が 117 件、ベースライン手法が 141 件である。また、エラーバーは標準誤差を示している。

5. 考察

4 章の比較実験より、図 3～図 6 の結果から、目標ベクトルを用いないベースライン手法と比べ、目標ベクトルを考慮した提案手法の評価が、ROUGE-1、ROUGE-2 共に高いことがわかる。このことより、提案手法の Decoder で用いた GRUGV によって、目標から生成したベクトルを考慮することで、ベースライン手法と比べて、より適切な要約文の生成ができたと考えられる。

各手法の出力例を、図 9、10 に示す。図 9 は、スイスに存在する自治体「Fully」についての Wikipedia 記事から生成したテストデータの例と生成された要約文である。この例では、目標として「Fully」のみが与えられたため、提案手法、ベースライン手法ともに、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、図 10 に示す、ある人物の Wikipedia 記事から生成されたテストデータでは、その人物の名前と「personal life」が目標として入力されている。ベースライン手法では、「どこで生まれた」や、「どの大学で学んだ」のような、該当記事における「Early life and education」のセクションに当たる要約文を生成されている。それに対して、提案手法では、「誰と結婚した」や、「何人の子供を授かった」のような、クエリで指示された観点についての要約文を生成できていることが確認できる。

表 2 より、提案手法、ベースライン手法のいずれにおいても、生成した要約文中に占める未知語の割合が高いことがわかる。また、特に、 G 中の未知語の割合が高いことがわかる。これは、開発データセットに対して、本実験で設定した語彙サイズが不足していることを示している。このような未知語の問題に対しては、語彙サイズの増加や、原文書中の単語を要約に反映できる Pointer Network 機構 [5] をモデルに追加するなどして改善できると考えられる。

また、図 7、および、図 8 から、生成したシステム要約の文書長が長い場合、提案手法の精度が、ベースライン手法よりわずかに低くなることが確認された。それぞれの手法について、

ROUGE-N の値が、図 3 の実験結果の、平均値を超えるシステム要約について内容を確認したところ、RNN が抱える問題の一つである同じフレーズの繰り返しが発生していることがわかった。この問題に対しては、Nema ら [13] が提案した、各時間ステップにおける文脈ベクトルを、それぞれソフトに直交させる機構をモデルに追加するなどして改善できると考えられる。

6. まとめ

与えられたクエリを目標ベクトルに変換し、文単位のベクトルで表された文書とともに常に注意機構に埋め込むことによって、与えられた観点に沿った一貫性を保持した要約を生成する手法を提案した。比較実験により、出力単語ベクトルの生成に、常に目標ベクトルを埋め込む提案手法が、ベースライン手法より精度向上したことを確認した。

提案手法では、入力された文書は、文単位にベクトルへ変換されるため、文間の関係性が損なわれてしまう問題点がある。今後、この問題を回避するため、文の順序に着目した文ベクトル生成手法である skip-thought vector [9] をモデルに追加する予定である。

文献

- [1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [2] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li. Attsum: Joint learning of focusing and summarization with neural attention. *arXiv preprint arXiv:1604.00125*, 2016.
- [3] 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. *arXiv preprint arXiv:1406.1078*, 2014.
- [4] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.
- [5] J. Gu, Z. Lu, H. Li, and V. O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR*, abs/1603.06393, 2016.
- [6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [7] C. Kiddon, L. Zettlemoyer, and Y. Choi. Globally coherent text generation with neural checklist models. In *EMNLP*, pages 329–339, 2016.
- [8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [9] 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*, pages 3294–3302, 2015.
- [10] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [11] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.
- [12] 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*, pages 55–60, 2014.

原文書	<p>fully is a municipality in the district of <unk> in the canton of <unk> in switzerland . <eos> fully is first mentioned in the 11th century as <unk> . <eos> fully has an area , , of . <eos> of this area , <unk> % is used for agricultural purposes , while 27.7 % is forested . <eos> of the rest of the land , 5.7 % is settled (buildings or roads) and <unk> % is unproductive land . <eos> it is on the right bank of the river rhône and is well known for its wines . <eos> fully is just the administrative name of a group of several villages of <unk> (sometimes called fully) , branson , <unk> and <unk> along with a number of hamlets .</p> <p style="text-align: center;">⋮</p> <p>in fully about <unk> or (<unk> %) of the population have completed non-mandatory upper secondary education , and <unk> or (8.8 %) have completed additional higher education (either university or a fachhochschule) . <eos> of the <unk> who completed tertiary schooling , <unk> % were swiss men , 31.5 % were swiss women , 5.9 % were non-swiss men and 4.3 % were non-swiss women . <eos> , there were 8 students in fully who came from another municipality , while <unk> residents attended schools outside the municipality . <eos></p>
目標	fully
正解要約	fully is a municipality in the district of <unk> in the canton of valais in switzerland . <eos>
提案手法	the <unk> is a municipality in the district of <unk> in the canton of valais in switzerland . <eos>
比較手法	<unk> is a municipality of the district of <unk> in the canton of valais in switzerland . <eos>

図 9 出力された要約文の一例 (記事タイトル: Fully, 目標: Fully)

原文書	<p>james stuart gilmore iii (born october 6 , 1949) is an american politician and former attorney who was the <unk> governor of virginia from 1998 to 2002 and chairman of the republican national committee in 2001 . <eos> a native <unk> , gilmore graduated as a bachelor of arts and a <unk> doctor from the university of virginia , and then served in the u.s. army as a <unk> agent . <eos> he was later elected to public office as a county prosecutor , as the attorney general of virginia , and as governor of virginia . <eos></p> <p style="text-align: center;">⋮</p> <p>gilmore is the president & ceo of the free congress foundation , a conservative think tank which was founded by paul <unk> . <eos> recently , gilmore has <unk> the free congress foundation . <eos> it is now the american opportunity foundation and launched a new website in 2017 - <unk> <eos> gilmore has also served as chairman of the national council on <unk> & <unk> , a homeland security program focused on community involvement and <unk> partnerships . <eos> he is also president of usa secure , a non-profit homeland security think tank based in washington , d.c. <eos> from 2005 to 2017 he served on the board of directors of the national rifle association . <eos> official sites <eos> topic pages and databases <eos> <unk> records <eos></p>
目標	jim gilmore personal life
正解要約	gilmore has been married to <unk> <unk> since 1977 . together , they have two sons . <eos>
提案手法	<unk> was married to <unk> <unk> from 1964 until they divorced in 1989 . they had two children . <eos>
比較手法	<unk> was born in <unk> , illinois . he studied at the university of michigan . he graduated from the university of illinois . he graduated from the university of illinois . he graduated from the university of illinois . he graduated from the university of illinois . he graduated from the university of illinois in 1948 . he graduated from the university of illinois in 1948 . he graduated from the university of illinois in 1948 . <eos>

図 10 出力された要約文の一例 (記事タイトル: Jim Gilmore, 目標: Jim Gilmore Personal life)

- [13] P. Nema, M. Khapra, A. Laha, and B. Ravindran. Diversity driven attention model for query-based abstractive summarization. *arXiv preprint arXiv:1704.08300*, 2017.
- [14] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.
- [15] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [16] 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)*, volume 5, 2015.
- [17] M. Yousefiazar et al. Query-oriented single-document summarization using unsupervised deep learning. 2015.
- [18] 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*, pages 2–10, 2015.