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† 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．特に，現実の利用を考えた場合，単に，文書全体を要約する非クエリ指向文書要約ではなく，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）と，Recurrent Neural Network（RNN）によってエンコードした，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，Attention メカニズムが欠かせない．しかし，翻訳タスクにおいて，Koehn ら [10] は，実験により，LSTM へ入力された，60 トークンよりも長い文書の符号化において翻訳の品質が低下する，と報告している．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や，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，我々は，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．本モデルにより，文長が長い原文が入力された場合でも，モデルが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，文ベクトルの有無により，生成される要約に，どのような影響があるのか明らかにする．

キーワード　クエリ指向文書要約，文エンコーダ，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，テキスト，音声，画像，動画といった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，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，スニペットやリード文といった，原文の要約を提供されることが多い．以上の背景により，与えられた文書を自動で要約する研究は，盛んに取り組まれている．

自動要約は，要約を，どのような観点でまとめるかで，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，これまでの研究では，原文における重要文を，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などの，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，ユーザから与えられたクエリによって，要約に含むべき文の種類が変化する．そのため，これまで提案された手法では，文書検索と同様に，クエリと原文の各文の関連性を，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 [18] 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 [19] という機械翻訳において提案されたモデルに，Attention メカニズム [2] という注意機構を備えることで，入力された原文において，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．また，クエリ指向文書要約では，Nema ら [16] が，深層モデルに基づく生成型の要約を行う Encoder-Decoder モデルを提案している．このモデルでは，入力の原文の各単語ベクトルに加えて，入力のクエリにも Attention 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，Attention によって生成されるベクトルが，各デコードステップにおいて，ハードまたはソフトに，直交するように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）

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

と，RNN によって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，Attention メカニズムが欠かせない．しかし，翻訳タスクにおいて，Koehn ら [10] は，実験により，LSTM へ入力された，60 トークンよりも長い文書の符号化において翻訳の品質が低下する，と報告している．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，我々は，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．本モデルにより，文長が長い原文が入力された場合でも，モデルが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，文ベクトルの有無により，生成される要約に，どのような影響があるのか明らかにする．

本稿の構成は以下の通りである．2 章で，関連研究について述べる．3 章で本モデルのタスクを定式化し，4 章で，提案手法の目的とシステム構成について詳述する．5 章で実験とその結果，6 章で考察を述べる．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，Luhn ら [13] が提案した，原文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，原文中の，出現頻度が高い重要語 [13]，文の位置情報 [6]，接続詞のような手掛かり語 [6] 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，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，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盛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」と「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 [15] の出現で，深層学習による単語の分散表現に続き，文の分散表現が盛んに研究されている．Paragraph Vector [11] 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 [9] 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，および，$m-1$ 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表現の学習を目指した，InferSent [5] 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．その一方で，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，品質の高い文表現を学習する DisSent [17] 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，コーパス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されている．

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，本提案手法では，文エンコーダとして，Skip-Thought を選択した．また，本研究の実験データセットである Debatepedia 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，本研究の文エンコーダとして，より適切であると考えた．

クエリ指向要約における抽出型要約の研究として，Yousefi-azar ら [21] 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 モデルを提案している．また，Cao ら [3] 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]

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/

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，デコード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 [16] 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 固有の問題を解決するために，Attention によって生成される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 $\mathbf{d}^{token} = t_1^d, t_2^d, ..., t_{l_d}^d$，および，トークン t のシーケンスから構成されるクエリ $\mathbf{q} = t_1^q, t_2^q, ..., t_{l_q}^q$ 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，Subword 単位のどれかを用いることが多く，本論文では，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，3 つめの入力として，次の 1 つを定義する．センテンス s のシーケンスから構成される原文 $\mathbf{d}^{sentence} = s_1^d, s_2^d, ..., s_{l_s}^d$ とする．ここで，s_m は，1 文のみから構成される，トークンのシーケンス $s_m = t_1^{s_m}, t_2^{s_m}, ..., t_{l_{s_m}}^{s_m}$ である．

最後に，出力は，次の 1 つから構成される．トークンのシーケンスから構成される要約 $\mathbf{o} = t_1^o, t_2^o, ..., t_{l_o}^o$ である．また，モデルが出力する要約を $\mathbf{o}^{system}$，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 提 案 手 法

ここでは，提案手法である，文単位のエンコーダを導入したクエリ指向要約モデルについて説明する．まず，提案手法である，各エンコードステップにおける，Document Encoder，および，Sentence Encoder の構成について述べ，次に，各デコードステップにおける，単語単位の注意機構，および，文単位の注意機構の構成について詳述する．

また，単語単位のコンテキストベクトルと比べ，文単位のコンテキストベクトルを，デコード時に常に用いることは，必ずしも適切であるとは言い難い．そこで，文単位の注意機構の出力を，LSTM などで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，Sentence Adaptive Attention Mechanism を提案する．

最後に，実験において使用する，既存のクエリ指向型要約モデルについて説明し，本提案手法を導入するにあたってモデルを拡張する部分を明記する．

4.1 Word Representation

モデル内で共有する，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．提案手法では，Word Representation として，Skip-gram を選択した．

4.2 Document Encoder

Document Encoder 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 $\mathbf{d}^{token} = t_1^d, t_2^d, ..., t_{l_d}^d$ 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) [4] を用いて，次のように表す．

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

ここで，h_i^d は δ_3 次元の特徴ベクトルである．

4.3 Sentence Representation

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．

本提案手法では，Sentence Representation として，Skip-Thought [9] を用いた．このモデルは，入力された m 番目の文をエンコードし，エンコードした特徴ベクトルから，$m+1$ 番目の文，および，$m-1$ 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．

$$\overrightarrow{eh}_{m,n}^s = \mathbf{GRU}_{skip}^{forward}(\overrightarrow{eh}_{m,n-1}^s, e(t_{m,n}^s)), \tag{2}$$

$$\overleftarrow{eh}_{m,n}^s = \mathbf{GRU}_{skip}^{backward}(\overleftarrow{eh}_{m,n-1}^s, e(t_{m,n}^s)), \tag{3}$$

$$eh_{m,n}^s = [\overrightarrow{eh}_{m,n}^s; \overleftarrow{eh}_{m,n}^s], \tag{4}$$

$$eh_m^s = eh_{m,l_{s_m}}^s, \tag{5}$$

ここで，$\overrightarrow{eh}_{m,n}^s$ 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し，$[\overrightarrow{eh}_{m,n}^s; \overleftarrow{eh}_{m,n}^s]$ は，2 つのベクトルの concatinate 演算を示す．また，$eh_{m,n}^s$ は，δ_4 次元の最終ステップ l_{s_m} の特徴ベクトルである．

4.4 Sentence Encoder

単語単位のエンコーダである Document Encoder では，まず，入力された各単語を，Word Representation の手法により，単語の分散表現へ変換し，その後，RNN を通して，最終的な特徴ベクトル h_i^d へ変換する．同様に，Sentence Representation によってエンコードされた文単位の特徴ベクトル $eh^s = eh_1^s, ..., eh_m^s, ..., eh_{l_s}^s$ 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，最終的な特徴ベクトル h_m^s へ変換するべきと考えた．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．

$$\overrightarrow{h}^s_m = \mathbf{GRU}_s(\overrightarrow{h}^s_{m-1}, eh_m^s), \tag{6}$$

$$\overleftarrow{h}^s_m = \mathbf{GRU}_s(\overleftarrow{h}^s_{m-1}, eh_m^s), \tag{7}$$

$$h_m^s = [\overrightarrow{h}^s_m; \overleftarrow{h}^s_m], \tag{8}$$

ここで，h_m^s 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 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 h_k^d を生成することである．

我々は，文単位コンテキストベクトル h_k^s によって，単語単位のコンテキストベクトル h_k^d の生成をコントロールするべきと考えた．そこで，h_k^d の式は，h_k^s を受け取るパラメータ $Z_s \in \mathbb{R}^{\delta_3 \times \delta_5}$ を持つ．よって，デコードステップ k のコンテキストベクトル h_k^d は，次の式で算出される．

$$a_{k,i}^d = v_d^\top \tanh(W_d s_k^o + U_d h_i^d + Z_s h_k^s), \tag{9}$$

$$\alpha_{k,i}^d = \frac{\exp(a_{k,i}^d)}{\sum_{i'=1}^{l_d} \exp(a_{k,i'}^d)}, \tag{10}$$

$$h_k^d = \sum_{i=1}^{l_d} \alpha_{k,i}^d h_i^d, \tag{11}$$

ここで，$W_d \in \mathbb{R}^{\delta_3 \times \delta_6}$, $U_d \in \mathbb{R}^{\delta_3 \times \delta_3}$，$v_d \in \mathbb{R}^{\delta_3}$，$h_k^s$ は δ_5 次元の特徴ベクトルである．また，s_k^o は，デコードステップ k における，Decoder の出力ベクトルを表す，δ_6 次元の特徴ベクトルである．

4.6 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 h_k^s 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 h_k^s は，次の式で算出される．

$$a_{k,m}^s = v_s^\top \tanh(W_s s_k^o + U_s h_m^s), \tag{12}$$

$$\alpha_{k,m}^s = \frac{\exp(a_{k,m}^s)}{\sum_{m'=1}^{l_s} \exp(a_{k,m'}^s)}, \tag{13}$$

$$h_k^s = \sum_{m=1}^{l_s} \alpha_{k,m}^s h_m^s, \tag{14}$$

ここで，$W_s \in \mathbb{R}^{\delta_5 \times \delta_6}$, $U_s \in \mathbb{R}^{\delta_5 \times \delta_5}$，$v_s \in \mathbb{R}^{\delta_5}$ である．

4.7 Sentence 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 h_k^s 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 $h_k'^s$ を生成することである．デコードステップ k の適応的なコンテキストベクトル $h_k'^s$ は，次の式で算出される．

$$gate_k = \sigma(W_{gate} e(t_{k-1}^o) + U_{gate} s_{k-1}^o + Z_{gate} h_k^s), \tag{15}$$

$$h_k'^s = gate_k \odot h_k^s + (1 - gate_k) \odot h_{l_s}^s, \tag{16}$$

ここで，$W_{gate} \in \mathbb{R}^{\delta_5 \times \delta_2}$，$U_{gate} \in \mathbb{R}^{\delta_5 \times \delta_6}$，$Z_{gate} \in \mathbb{R}^{\delta_5 \times \delta_5}$ である．また，$h_{l_s}^s$ は，h_m^s における最終ステップ l_s の特徴ベクトルである．

4.8 Diversity driven Attention Model

ここでは，Nema らがいくつか提案した，Diversity driven Attention Model [16] のうち，特に精度が良かった，the soft diversity である $\mathbf{SD_2}$ を用いた手法について説明する．Diversity driven Attention Model を図 1 に示す．このモデルは，文書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．

4.8.1 Query Encoder

クエリエンコーダの役割は，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 $\mathbf{q} = t_1^q, t_2^q, ..., t_{l_q}^q$ 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．

$$h_j^q = \mathbf{GRU}_q(h_{j-1}^q, e(t_j^q)), \tag{17}$$

ここで，h_j^q は δ_7 次元の特徴ベクトルである．

4.8.2 Attention Mechanism

この機構の役割は，エンコードされた各特徴ベクトル h_i^d と h_j^q の各エンコードステップから，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 h_k^d を生成することである．

まず，デコードステップ k のクエリのコンテキストベクトル h_k^q は，次の式で算出される．

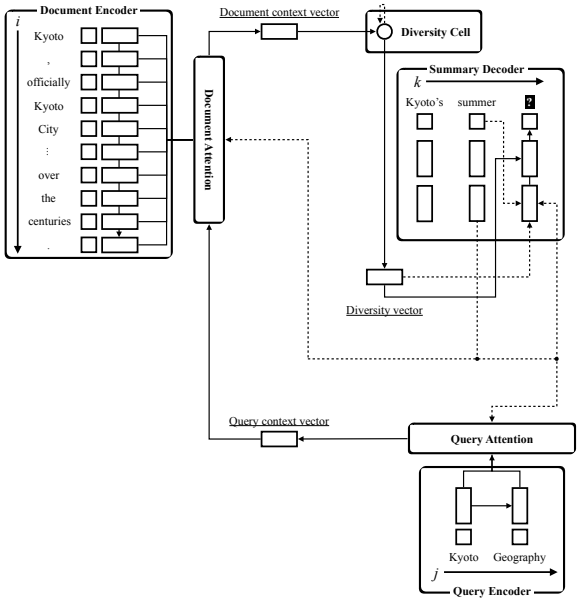

図 1　Diversity driven Attention Model. 2 つのエンコーダ，および，それぞれの注意機構と，1 つの要約デコーダ，および，1 つの多様性機構から構成される．破線は，デコードステップ $k-1$ を表現し，実線は，デコードステップ k を表現する．

$$a^q_{k,j} = v_q^\top \tanh(W_q s^o_k + U_q h^q_j), \tag{18}$$

$$\alpha^q_{k,j} = \frac{\exp(a^q_{k,j})}{\sum_{j'=1}^{l_q} \exp(a^q_{k,j'})}, \tag{19}$$

$$h^q_k = \sum_{j=1}^{l_q} \alpha^q_{k,j} h^q_j, \tag{20}$$

ここで，$W_q \in \mathbb{R}^{\delta_7 \times \delta_6}$, $U_q \in \mathbb{R}^{\delta_7 \times \delta_7}$，$v_q \in \mathbb{R}^{\delta_7}$ である．

Nema らは，クエリのコンテキストベクトル h^q_k によって，原文のコンテキストベクトル h^d_k の生成をコントロールするべきと考えた．そこで，h^d_k の式は，h^q_k を受け取るパラメータ $Z_q \in \mathbb{R}^{\delta_3 \times \delta_7}$ を持つ．デコードステップ k のコンテキストベクトル h^d_k は，次の式で算出される．

$$a^d_{k,i} = v_d^\top \tanh(W_d s^o_k + U_d h^d_i + Z_q q_k), \tag{21}$$

$$\alpha^d_{k,i} = \frac{\exp(a^d_{k,i})}{\sum_{i'=1}^{l_d} \exp(a^d_{k,i'})}, \tag{22}$$

$$h^d_k = \sum_{i=1}^{l_d} \alpha^d_{k,i} h^d_i. \tag{23}$$

4.8.3　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，コンテキストベクトル h^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 h'^d_k へ変換する機構 **SD$_2$** を，次の式で定義した．

$$\begin{pmatrix} i_k \\ f_k \\ o_k \\ \hat{c}_k \\ g_k \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} h^d_k \\ h_{k-1} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \tag{24}$$

$$c_k = \sigma(i_k) \odot \tanh(\hat{c}_k) + \sigma(f_k) \odot c_{k-1}, \tag{25}$$

$$c^{diverse}_k = c_k - \sigma(g_k) \frac{c_k^\top c_{k-1}}{c_{k-1}^\top c_{k-1}} c_{k-1}, \tag{26}$$

$$h_k = \sigma(o_k) \odot \tanh(c^{diverse}_k), \tag{27}$$

$$h'^d_k = h_k, \tag{28}$$

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

4.8.4　Summary Decoder

このデコーダの役割は，ステップ $k-1$ のコンテキストベクトル h'^d_{k-1} と，ステップ $k-1$ のトークン t^o_{k-1} を入力とし，ステップ k における特徴ベクトル s^o_k を出力することである．

$$s^o_k = \mathbf{GRU}_o(s^o_{k-1}, [e(t^o_{k-1}); h'^d_{k-1}]). \tag{29}$$

その後，特徴ベクトル s^o_k とコンテキストベクトル h'^d_k から，ステップ k におけるトークン t^o_k を予測する．

$$t^o_k = \mathrm{softmax}(W f(W_{dec} s^o_k + V_{dec} d'_k)), \tag{30}$$

ここで，$W \in \mathbb{R}^{\delta_1 \times \delta_2}$,$W_{dec} \in \mathbb{R}^{\delta_2 \times \delta_6}$,$V_{dec} \in \mathbb{R}^{\delta_2 \times \delta_3}$ である．また，活性化関数 f は，恒等関数である．

4.9　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．文エンコーダを導入するにあたり，我々は，各デコードステップ k におけるコンテキストベクトル h^d_k 生成時に，アテンションされたクエリ h^q_k によって，直接，単語単位のアテンションが決定されるのではなく，まず，クエリによって文単位のアテンションが生成され，その後，文単位のアテンションを通して，単語単位のアテンションが決定されるべきと考えた．そこで，デコードステップ k における，文単位のアテンションを，(12) から (31) へ，以下の式に変更し，導入することとした．また，同様に，単語単位のアテンションは，(21) の式から (9) の式へ変更することとした．

$$a^s_{k,m} = v_s^\top \tanh(W_s s^o_k + U_s h^s_m + Z_q h^q_k). \tag{31}$$

また，4.7 節で提案した，適応的な文単位のコンテキストベクトルを用いる場合は，(9) から (32) へ，以下の式に変更することとした．

$$a^d_{k,i} = v_d^\top \tanh(W_d s^o_k + U_d h^d_i + Z_s h'^s_k). \tag{32}$$

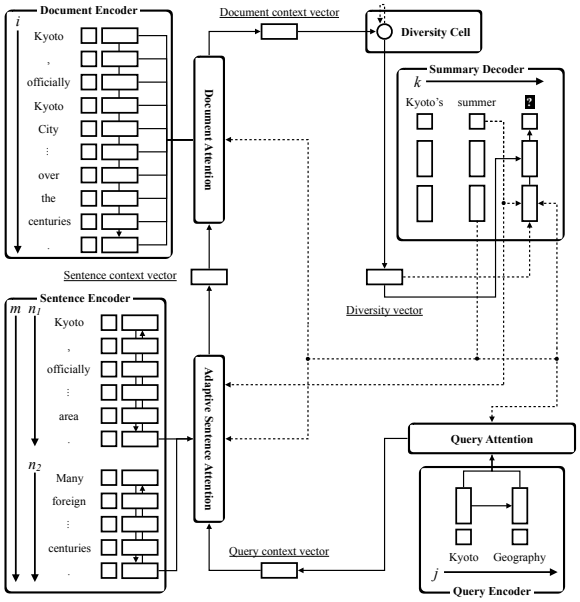

図 2　提案手法によって拡張された Diversity driven Attention Model. 3 つのエンコーダ，および，それぞれのアテンション機構と，1 つの要約デコーダ，および，1 つの多様性機構から構成される．破線は，デコードステップ $k-1$ を表現し，実線は，デコードステップ k を表現する．

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，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．

5.1 実 験 設 定

比較モデルとして，実験データセットにおいて SOTA である，4.5 節で説明したモデルを用いた．また，提案手法のモデルとして，4.9 節で説明した，提案手法により比較モデルを拡張したモデル 1，及び，さらにモデル 1 を拡張し，4.7 節で説明した，適応的な文単位のコンテキストベクトルを扱うモデル 2 を用いた．実験データとして，Nema ら [16] が，Debatepedia から構築した 13,573 件のデータセットを用い，ミニバッチサイズ 16，32，及び，64 の 3 種類について，10 交差検証により精度を比較した．

5.2 データセット

Debatepedia は，重要な議題について，賛成意見の本文，および，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．

表 1　比較手法と各提案手法の ROUGE-N(N = 1,2,L) による比較．太文字の値は，各ミニ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
比較モデル	16	27.56	1.06	26.78
	32	25.76	0.65	24.91
	64	24.26	0.44	23.38
モデル 1	16	**28.60**	1.16	**27.56**
	32	25.59	0.69	24.89
	64	24.48	0.53	23.69
モデル 2	16	28.31	**1.23**	27.42
	32	**26.97**	**0.89**	**26.14**
	64	**25.21**	**0.67**	**24.51**

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 1，モデル 2，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，各モデルで共通して用いた Skip-gram は，English Wikipedia 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，fine tuning の対象である，モデル 1 及びモデル 2 で用いた Skip-Thought についても，English Wikipedia の全記事を対象として事前学習した．

Document Encoder の隠れ層の次元，各 Attention 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．

各モデルは，先行研究である比較モデルを参考に，最適化手法として Adam [8]（α=0.0004，$\beta_1 = 0.9$，$\beta_2 = 0.999$，$eps = 10^{-8}$）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[6] [20]，および，ChainerMN[7] [1] を用いた．また，形態素解析器として，The Stanford CoreNLP [14] を用いた．

5.4 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 ROUGE-N [12]（N = 1,2,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 SumEval[8]を用い，オプションについては，DUC 2004 の設定[9]を適用した．実験結果を表 1 に示す．

6：Chainer: A flexible framework for neural networks : https://chainer.org/

7：ChainerMN: Scalable distributed deep learning with Chainer : https://github.com/chainer/chainermn

8：Well tested & Multi-language evaluation framework for text summarization : https://github.com/chakki-works/sumeval

9：An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems : https://duc.nist.gov/pubs/2004slides/duc2004.intro.pdf

表 2　人手により整合性が確認された正解データに対する，比較手法と各提案手法の ROUGE-N(N = 1,2,L) による比較.

モデル名	ROUGE-1	ROUGE-2	ROUGE-L
比較モデル	24.23	0.88	24.23
モデル 1	27.63	2.09	27.63
モデル 2	28.59	2.78	27.59

6 考　　察

表 1 より，比較手法と比べ，各提案モデルの ROUGE-N の値が上回ることを確認できる．特に，文単位のコンテキストベクトルを適応的に用いるモデル 2 の精度は，ほぼ全ての条件において，最高精度を達成した．この結果より，より適切な要約生成する上で，文単位のベクトル，および，適応的な文単位の注意機構を導入することが有効であると考えられる．また，各ミニバッチサイズ間の精度から，全てのモデルにおいて，ミニバッチのサイズが小さくなるにつれて，精度が向上す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，全てのモデルにおいて，原文やクエリ中に存在しないが，意味がよく似た別の単語を，要約として生成しやすいことが確認できた．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，クエリ，および，要約中の数単語について，単語埋め込み次元上で距離が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，人手により，原文や要約間の整合性を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（以下，GS と呼ぶ）として，新たな正解データとした．表 1 より，特に精度が高かったミニバッチサイズ 16 における，GS による評価結果について，表 2 に示す．表 2 より，整合性が確認された GS においても，比較手法と比べ，各提案モデルの精度が上回ることが確認できた.

7 ま と め

本論文では，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．また，文単位の注意機構の出力ベクトルを適応的に用いる Sentence Adaptive Attention を導入したモデルを提案した.

実験により，提案手法によって，ROUGE おける精度が改善したことを確認できた．特に，デコード時に，アテンションした文単位のベクトルを，適応的に用いるモデルの精度が安定して高いことがわかった．一方，本稿で用いた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，他のクエリ指向性要約のデータセットに対しても，精度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.

文　　献

[1] 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.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[3] 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*, pages 547–556. The COLING 2016 Organizing Committee, 2016.

[4] K. Cho, B. van Merrienboer, 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)*, pages 1724–1734. Association for Computational Linguistics, 2014.

[5] 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*, pages 670–680. Association for Computational Linguistics, 2017.

[6] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[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. In *International Conference on Learning Representations (ICLR)*, 2015.

[9] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc., 2015.

[10] P. Koehn and R. Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics, 2017.

[11] 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, pages II–1188–II–1196. JMLR.org, 2014.

[12] 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.

[13] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[14] 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.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[16] 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)*, pages 1063–1072. Association for Computational Linguistics, 2017.

[17] A. Nie, E. D. Bennett, and N. D. Goodman. Dissent: Sentence representation learning from explicit discourse relations. *CoRR*, abs/1710.04334, 2017.

[18] 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*, pages 379–389. Association for Computational Linguistics, 2015.

[19] 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.

[20] 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.

[21] 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.