

Encoder-Decoder モデルにおける出力長制御

菊池悠太^{1,a)} Graham Neubig^{2,b)} 笹野遼平^{1,c)} 高村大也^{1,d)} 奥村学^{1,e)}

概要：本稿では，encoder-decoder モデルにその出力系列の長さを制御させるための取り組みについて報告する．これは，要約のような一部の応用タスクにおいては必須とも言える機能ではあるが，従来の encoder-decoder モデルでは明示的に取り組まれてこなかった．我々は，文要約タスクを対象に，出力系列の長さを制御するためのいくつかの手法を提案し比較を行う．実験の結果，提案した学習ベースの手法が，従来研究と比較してその精度を劣化させずに出力長の制御機能を獲得したことを確認した．

1. はじめに

機械翻訳に適用されて以降 [4], [18], [35], 画像キャプション生成 [38], [42], 構文解析 [37], 対話応答生成 [21], [34] や文要約 [5], [32] などを含む多くの系列生成タスクにおいて encoder-decoder モデルを適用した研究が報告されている．本稿においてはこのうち文要約に焦点を当てる．文要約は，文書要約 [29] やヘッドライン生成 [8] のように，与えられた原文（書）集合の内容を簡潔にまとめた短い文（書）を生成する，テキスト自動要約の一種である．Rush ら [32] がニュース記事とタイトルから大規模な訓練データを整備して以降，文要約はニューラルネットワークに基づく encoder-decoder モデルの新たな適用タスクとして盛んに研究され始めている [1], [5], [14], [15], [24], [26], [30]．

要約器が持つべき重要な能力の一つとして，出力する要約の長さを制御できることが挙げられる．ユーザが所望する要約の長さは，原文書から把握したい情報の粒度や要約を読むデバイスの大きさなど，利用する条件に強く依存する．そのため，ユーザにより入力された所望の長さに応じて柔軟に出力する要約長を制御できることは非常に重要であり，必須の能力である．実際，文書要約分野における実験設定では原文書と同時に要約の長さ制限を入力するのが標準となっている．しかし，その重要さに関わらず，encoder-decoder モデルによる文要約に取り組んだ従来の研究では，この出力長の制御という点について明確に取り組

まれてこなかった．

本研究では，encoder-decoder モデルにおいて出力長を制御するための 4 つの手法を提案する．そのうち 2 手法はビーム探索に変更を加える手法であり，標準的な encoder-decoder モデルを訓練したのち，探索時に長さの入力を受け取る．残りの 2 手法は学習に基づくものであり，出力系列の長さの指定を入力として受け取るようモデルの拡張を行い，学習を通して出力長の制御能力を獲得する．

提案手法の有効性を確かめるために，文要約における標準的な評価セットにおいて要約の長さ制限を変化させた時の評価値の変化を確認した．実験により，学習ベースの手法は長い制限長 (50 バイトおよび 75 バイト) のときに探索ベースの手法を上回る傾向にあることが確認できた．また，従来手法との比較により，我々の学習ベースの手法は出力長の制御能力を獲得しつつもその要約精度の劣化を避けていることを確認した．

2. 背景

2.1 関連研究

テキスト要約は自然言語処理において最も古くから研究されてきた分野のひとつである．その多くは文書を扱う文書要約であるが，文を対象とする文圧縮やヘッドライン生成といったタスクも古くから盛んに研究されてきた．与えられた文からより短い文を生成するというタスクにおいて，伝統的にはルール [8], [43] や統計 [9], [10], [11], [12], [20], [40] に基づく単語の削除を行う手法が盛んに研究されており，現在も主要なアプローチとなっている．また，単語の削除に限らずに原文にない表現を生成するアプローチも研究されている．それらのアプローチは統語的な変換 [6], [27] や句に基づく統計的機械翻訳の枠組みを利用したもの [3], [7], [41] などが報告されている．

¹ 東京工業大学

Tokyo Institute of Technology

² 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

^{a)} kikuchi@lr.pi.titech.ac.jp

^{b)} neubig@is.naist.jp

^{c)} sasano@pi.titech.ac.jp

^{d)} takamura@pi.titech.ac.jp

^{e)} oku@pi.titech.ac.jp

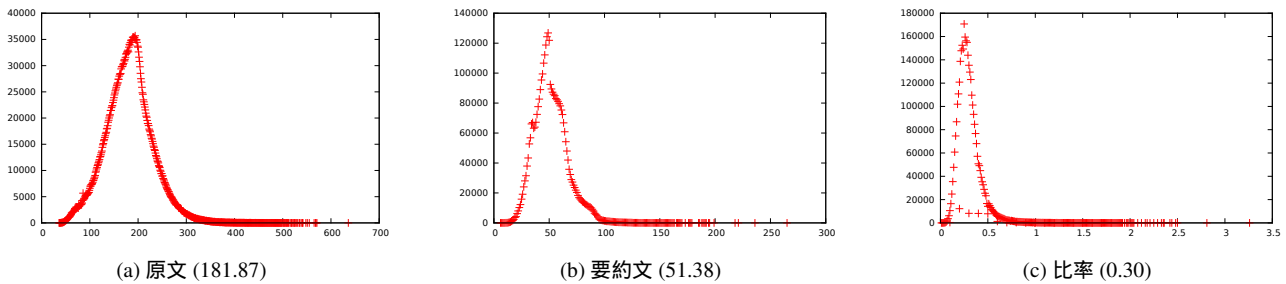


図1 Annotated English Gigaword Corpus における原文(本文の一文目), 要約文(タイトル)の長さ, またそれらの比率のヒストグラム. 括弧内の数値はコーパス全体での平均値である.

Rush らは, Annotated English Gigaword Corpus に含まれるニュース記事から, 大量の訓練事例の自動構築を行った. これにより, 文要約タスクにおいても, encoder-decoder モデル [4], [18], [35] や attention 機構 [2], [25] といった近年注目されているニューラルネットワークに基づいた研究が増加している. いくつかの研究は, 機械翻訳などと共に系列生成タスクのベンチマークの一つとして利用している [1], [24], [30]. 一方で, 文要約というタスクの性質に着目し, 原文からのコピーといった人間による要約にも高頻度で出現する現象を捉えたモデルの拡張を行う研究も存在している [14], [15]. Nallapati らは, 原文に埋め込まれたキーワードを考慮する場合や複数文を入力する場合など, 要約における重要な側面について幅広い調査を行った [26].

言語モデルにおいて文の長さを考慮した研究として, Rosenfeld らによる whole-sentence language models [31] などが挙げられる. しかしながら, 冒頭で述べたように, encoder-decoder モデルやニューラル言語モデルにおいては, 出力文の長さの制御について明示的に取り組んだ研究は存在しない.

2.2 Encoder-Decoder モデルにおける出力長の制御

1 節で述べた通り, 文書要約における最も標準的な設定では, 要約対象である原文書と所望する要約の長さを同時に要約システムへの入力とする. そのため, 要約システムは任意の長さで要約を生成することが求められ, これは encoder-decoder モデルに基づく要約器においても同様である.

基本的に, encoder-decoder モデルが出力する系列の長さは, その訓練データに含まれる事例の統計値に依存する. たとえば, 本稿では Annotated English Gigaword Corpus から構築された訓練データを用いるが, 人手による要約文の平均長は 51.38byte である. 図 1 に, Annotated English Gigaword Corpus における文長の統計値を示す. 実際, 本コーパスで標準的な encoder-decoder モデルを訓練し, 標準的なビーム探索を行うと, その出力系列の長さは 40-50 バイトになる傾向にある. しかしながら, これまで述べたようにその出力長は用途に応じて柔軟に変化するべきであ

る. 例えば, 本タスクでよく用いられる評価セットである, DUC2004 の共通課題 1 は 75 バイトという要約長の制限が規定されている. 75 バイトよりも長い文に対しては 75 バイトに達するまで文末の文字が削除される. 一方で, 75 バイトよりも短い長さの要約に対する追加点は与えられないため, 要約システムはできるだけその制限長に近い長さの要約を出力することが求められる.

Rush らのモデルをはじめとする従来のニューラル文要約モデルでは, モデルそのものに出力長を制御する機能を持たないため, DUC2004 による評価時はアドホックな手法を用いている. 具体的には, モデルが文の生成を終える時に用いる文末記号に対応するシンボル (EOS タグ) の生起確率に $-\infty$ を割り当てる. モデルから EOS を出力する機能を排除する代わりに, 出力単語数がある閾値^{*1}に達したら出力を終える. 最終的に, 要約の長さが 75 バイトに収まるように文末から文字を切り取る. 本来であれば, モデルは指定された要約長に応じてその振る舞いを変化させ, 適切なタイミングで自ら EOS タグを出力できることが望まれる.

3. Attention 機構付き Encoder-Decoder モデル

本節では, 本研究で用いる encoder-decoder モデルについて説明する. 具体的には, encoder として両方向 Recurrent Neural Network (RNN) を利用する, attention 機構を備えたモデルである. 図 2 に我々のモデルの概要を示す.

モデルは, 単語の系列として表された原文 $x = (x_1, x_2, x_3, \dots, x_N)$ を要約対象として受け取り (原系列), 対応する要約文 $y = (y_1, y_2, y_3, \dots, y_M)$ を出力する (目的系列). モデルは原文が与えられたときの要約文の条件付き確率 $p(y|x)$ を推定するためのパラメータを大規模な訓練事例 (原文-要約文対) の集合から学習する. この条件付き確率は, これまで観測した履歴に条件付けられた各単語の生起確率の積に分解される:

^{*1} Rush らが公開している実装 (<https://github.com/facebook/NAMAS>) では, この閾値を表すデフォルト引数は 15 単語に設定されている. なお, DUC2004 の評価セットに含まれる平均の単語数は 10.43 である.

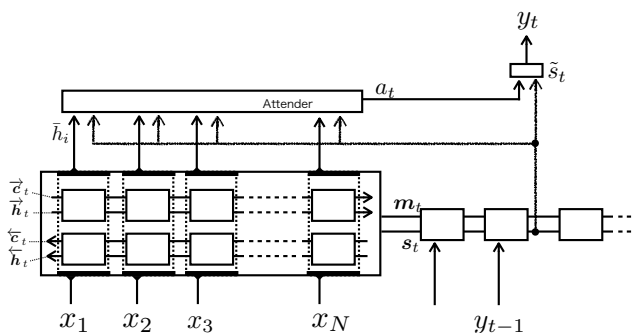


図2 本稿で用いる encoder-decoder モデルの概要図。

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^M p(y_t|\mathbf{y}_{<t}, \mathbf{x}),$$

ここで, $\mathbf{y}_{<t} = (y_1, y_2, y_3, \dots, y_{t-1})$. 以降では, この条件付き確率 $p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ の計算方法について説明する.

3.1 Encoder としての両方向 RNN

Encoder は入力された原文系列を後段の decoder を条件付けるための内部状態へ変換 (エンコード) するための機構である. 本研究では, encoder として, ニューラル機械翻訳 [2] や音声認識 [13], [33] において効果が確認されている両方向 RNN (BiRNN) を用いる.

BiRNN は, 原系列を順方向 (文頭から文末) と逆方向 (文末から文頭) で独立に処理する二種類の RNN を用いる. すなわち, エンコードを終えた時点で現系列を順方向に処理した履歴としての隠れ層のリスト $(\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N)$ と, 逆方向に処理した場合の隠れ層のリスト $(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_N)$ を以下のように計算する:

$$\begin{aligned} \vec{h}_t &= g(\vec{h}_{t-1}, x_t), \\ \overleftarrow{h}_t &= g(\overleftarrow{h}_{t+1}, x_t). \end{aligned}$$

ここで g は RNN で用いる任意の活性化関数であり, 本稿では Long short-term memory (LSTM) [16] を用いる. LSTM を利用する場合は, 隠れ層に加え両方向に対応するメモリーセル \vec{c}_t (あるいは \overleftarrow{c}_t) を持つ.

エンコード処理を終えると, 次節で説明する decoder の初期内部状態 (隠れ層 s_0 およびメモリーセル m_0) を以下のように初期化する.

$$\begin{aligned} s_0 &= \overleftarrow{h}_1, \\ m_0 &= \overleftarrow{c}_1. \end{aligned}$$

3.2 Decoder および Attention

Decoder は, encoder によりもたらされる原系列の情報に基づき対応する目的系列を出力 (デコード) する. 本研究では, decoder として encoder と同様に, LSTM をその活性化関数として持つ RNN を用いる.

加えて, Luong らによる attention 機構 [25] をモデルに組

み込む. Luong らの手法は, デコードの各ステップにおいて, その時刻における decoder の隠れ層 s_t を用いて以下の文脈ベクトル d_t を計算する. まず, 前節で求めた両方向の隠れ層を加算したベクトル $\bar{h}_i = \vec{h}_i + \overleftarrow{h}_i$ を原文の情報として保持する. 文脈ベクトル d_t は, これらの重み付き平均として計算される:

$$d_t = \sum_i a_{ti} \bar{h}_i,$$

ここで a_{ti} は t ステップ目における原系列中の i 番目の単語の重みであり, softmax 関数により計算される:

$$a_{ti} = \frac{\exp(s_t \cdot \bar{h}_i)}{\sum_{h'} \exp(s_t \cdot \bar{h}')}.$$

文脈ベクトルを計算した後, 文脈情報付き隠れ層 \tilde{s} と次ステップにおける単語の予測確率 $p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ を計算する:

$$\tilde{s}_t = \tanh(W_{hs}[s_t; d_t] + b_{hs}),$$

$$p(y_t|\mathbf{y}_{<t}, \mathbf{x}) = \text{softmax}(W_{so}\tilde{s}_t + b_{so}).$$

このとき確率が最大となった単語 y_t は, t ステップ目の出力となるとともに, $t+1$ ステップ目における LSTM への入力となる. また y_t に加えて, Luong らの input feeding 構造に基づき, \tilde{s}_t も同様に $t+1$ ステップ目の LSTM への入力に用いる.

3.3 目的関数

訓練における目的関数は訓練データ集合 D における対数尤度の最大化である:

$$\begin{aligned} L_t(\theta) &= \sum_{(\mathbf{x}, \mathbf{y}) \in D} \log p(\mathbf{y}|\mathbf{x}; \theta), \\ p(\mathbf{y}|\mathbf{x}; \theta) &= \prod_t p(y_t|\mathbf{y}_{<t}, \mathbf{x}). \end{aligned}$$

ここで θ はモデルが持つ訓練対象のパラメータである.

訓練後は, 入力される原文 x に対し, $p(\mathbf{y}|\mathbf{x}; \theta)$ を最大化する出力系列 y をビーム探索により求める.

4. 出力長制御手法

本節では, encoder-decoder モデルを用いて出力の長さを制御するための 4 つの手法を述べる. まず, はじめに訓練するモデルそのものには変更を加えず, ビーム探索時に出力系列の長さを考慮する 2 つの手法を説明する. その後, 出力長の制御を学習により獲得させるためにモデルそのものに変更を加える 2 つの手法について述べる. 本稿においては, 実験に用いる評価セットに合わせてバイト数を長さの単位として実験を行うが, 単語を長さの単位にすることも同様に可能である.

4.1 fixLen: EOS タグの排除による固定長ビーム探索

本手法は, 従来の多くのニューラル文要約手法で適用さ

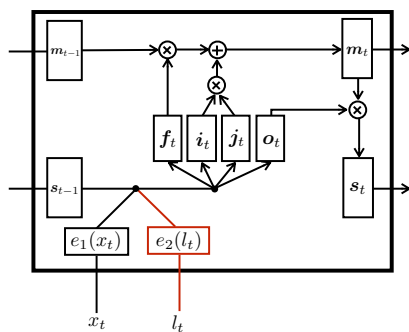


図3 LenEmb: 出力すべき残り長さを LSTM への追加的な入力とする。

れている方法に類似した方法である。具体的には、各ステップにおける EOS タグの生起確率を $-\infty$ に設定する。通常、encoder-decoder モデルは文末記号である EOS タグを生成することでその出力を終えるため、これはすなわちモデル自身では文の生成を終えること出来ないことを意味する。そのため本手法では、出力系列が指定の長さ達した時点で強制的に生成を止めることで指定した長さに応じた出力系列を得る。このとき、必ず指定長よりも短い系列を得るために、指定長を超えた最後の単語については EOS タグに置き換える。また生起確率についても同様に、最後の単語の確率の代わりに EOS タグ本来(すなわち、 $-\infty$ に置換する前)の生起確率を用いて文全体の生起確率を修正する。

4.2 fixRng: 出力長の範囲を規定するビーム探索

本手法では、fixLen のように探索時における EOS タグの排除は行わない。そのかわり出力系列の長さの許容範囲(最大値と最小値)を規定し、その範囲を超えた長さで EOS タグを出力した系列全体をビームから削除することで所望の長さの出力系列を得る。

具体的には、標準的なビーム探索の手続きに加え、以下の二つの規則を追加する:

- モデルが EOS タグを生成した時、その出力系列の長さが最低値未満であった場合、その系列をビームから削除する。
- モデルが EOS タグ以外の単語を生成したことでその系列の長さが最大値を超えた場合、その系列をビームから削除する。

言い換えると、規定された範囲内で EOS タグを出力した系列のみがビームに残る。本手法は decoder による文生成のプランニングを尊重することと、出力長を指定することのトレードオフを取った手法となっている。本手法はビーム探索中に系列を積極的に削除するため、指定範囲が訓練データの平均長と大きく異なる場合は他手法と比較して大きいビーム幅が必要になることに注意されたい。

4.3 LenEmb: 長さ埋め込みベクトルの導入

本節および 4.4 節の 2 種類の手法は、出力長の制御機能を訓練を通じて獲得するためにモデルに拡張を加える。近年、decoder への追加的な情報の入力が、その出力系列の特性を変化させることが確認されている [22], [39]。本手法はそれらの知見に基づき、長さの情報を decoder への追加入力とする。

具体的には、本手法では近年のニューラルネットワークを用いた自然言語処理において標準的に用いられている単語の埋め込み (word embedding) と同様に長さ埋め込み $W_{le} \in \mathbb{R}^{E \times L}$ をを導入する。 L は長さの異なり数であり、 W_{le} は訓練対象のパラメータである。デコードの各ステップにおいて、その時点における出力すべき残りの長さ l_t についての長さ埋め込みベクトル $e_2(l_t) \in \mathbb{R}^E$ を decoder の LSTM への追加的な入力とする。図 3 に本手法において用いる decoder の LSTM を示す。

l_t は encoder によるエンコード処理が終わったあとに初期化され、decoder によるデコード処理の過程で以下のように更新される:

$$l_1 = length,$$

$$l_{t+1} = \begin{cases} 0 & (l_t - byte(y_t) \leq 0) \\ l_t - byte(y_t) & (\text{otherwise}), \end{cases}$$

ここで、 $byte(y_t)$ は t ステップ目における出力単語 y_t の長さを返す関数であり、 $length$ はモデルが指定された出力長である。

各ステップにおいてモデルが出力すべき長さについての情報を入力することにより、decoder は残された長さに応じた文生成のプランニングが可能になる。

4.4 LenInit: 出力長に応じたメモリセルの初期化

LenEmb では decoder の各ステップにおいて出力すべき残り長さ l_t を入力していたが、本手法では LenInit をデコード処理の最初に一度だけ入力する。具体的には、指定された出力長の情報を用いて decoder の LSTM の状態(隠れ層 s_0 およびメモリセル m_0)を以下のように初期化する:

$$s_0 = \hat{h}_1,$$

$$m_0 = b_c * length. \quad (1)$$

ここで、ベクトル $b_c \in \mathbb{R}^H$ は訓練対象のパラメータであり、 $length$ は指定された出力長である。本手法を適用したモデル構造を図 4 に示す。

各ステップで残り長さの情報を明示的に入力していた LenEmb と異なり、本手法におけるモデルはその内部状態を用いて出力長の制御を行う。内部状態としてメモリーセルを備えた LSTM はこの目的に適していると考えられる。

出力長を制御する機構として、既存のメモリーセルを置き換えずに長さ専用のメモリーを追加するなど、ほかの方

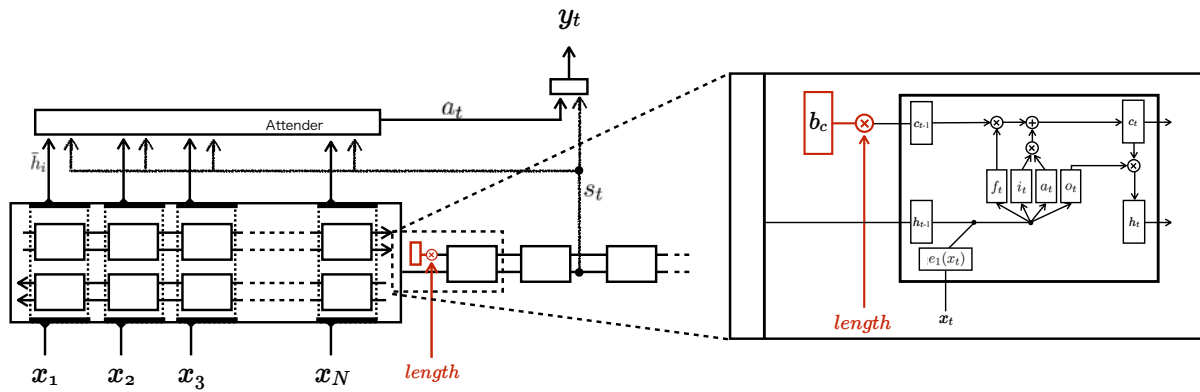


図4 LenInit: 長さに依存した decoder のメモリーセル m_0 の初期化.

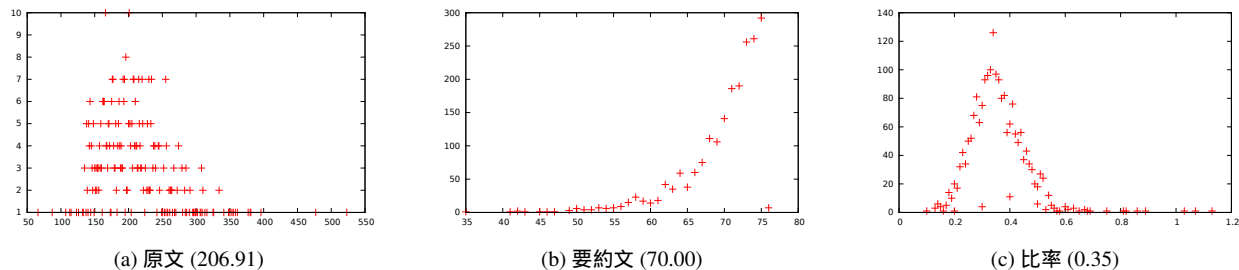


図5 DUC2004 における原文 (本文の一文目), 要約文 (タイトル) の長さ, またそれらの比率のヒストグラム. 括弧内の数値はコーパス全体での平均.

法も考えられるが, このようなシンプルな拡張でも効果的に働くことを実験を通して確認した.

5. 実験

5.1 データセット

訓練データとして, Rush ら [32] がニューラル文要約のために構築した Annotated English Gigaword Corpus[28]の一部を利用した. 前処理として彼らの公開しているスクリプトを利用した^{*2}.

本データセットは約 360 万対のニュース記事本文の 1 文目とそのタイトルの組で構成されている. 本データセットの長さに関する統計値は図 1 に示したとおりである. また, 文頭 (BOS), 文末 (EOS), 未知語 (UNK) タグを含めた原文の語彙サイズは 116,875 であり, 要約文の語彙サイズは 67,564 であった. LenEmb および LenInit については, 訓練時にはその訓練事例の要約文の長さを入力する. それぞれ長さに応じた独立したモデルを訓練するのではなく, 指定した長さに柔軟に対応する一つのモデルを訓練することに注意されたい.

モデルの評価は, DUC2004 における共通課題 1 (generating very short single-document summaries) にて行う. 本評価タスクでは, 要約システムは入力された文書に対して 75 バイト未満という非常に短い長さの要約を出力することが求められる. 75 バイトを超えた分のテキストは強制的に切り取られ, 75 バイトよりも短い文を生成したとしても加

点は与えられない. 本タスクには合計 500 の原文書に対し最大 4 つの人手で書かれた要約 (参照要約) が付与されている. 図 5 に, 本評価セットにおける長さに関する統計値を示す.

本研究では出力長の制御能力を評価するため, 75 バイトの他に制限長を 30 バイト, 50 バイトと変化させたときの評価値の変化を確認する. これにより各々のモデルの出力系列は短くなるが, 全ての条件において参照要約については 75 バイトのまま評価値を計算することに注意されたい.

自動評価尺度として ROUGE[23] を用いた. 評価を行う言語単位としてはユニグラム (ROUGE-1), バイグラム (ROUGE-2) 及び最長共通部分列 (ROUGE-L) を用いた.

5.2 実装の詳細

パラメータの学習は Adam[19]($\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$, $\text{eps}=10^{-8}$) により行い, ミニバッチのサイズは 80 とした. 計算量の効率化を図るため, 同じバッチ内の事例の長さを出来るだけ揃えるようにした. 具体的には, まず 10,000 バッチ (800,000 事例) 分の訓練事例を読み込み, ソートにより原文の長さが同じ 80 事例ずつのグループを作成し, その 10,000 個のグループを無作為に並べ替えたものを順に読み込みパラメータの更新を行った.

単語埋め込みベクトル E の次元は 100, 隠れ層の次元 H は 200 とした. Jozefowicz らの報告 [17] に倣い, LSTM の forget gate のバイアスのみを 1.0 に初期化し, 他の gate のバイアスの初期値は 0.0 とした. モデルの実装には Chainer[36]

^{*2} <https://github.com/facebook/NAMAS>

model	30 byte			50 byte			75 byte		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
<i>fixLen</i>	14.35	3.10	13.24	20.01*	5.99	18.27*	25.88*	7.93	23.07*
<i>fixRng</i>	14.32	3.13	13.23	20.08*	5.75	18.19*	26.02	7.69*	22.78*
<i>LenEmb</i>	14.24	3.22	13.02	20.79	5.98	18.57	26.73	8.40	23.88
<i>LenInit</i>	14.31	3.28	13.20	20.88	6.17	19.01	25.87*	8.28	23.25

表1 いくつかの要約長制限による各手法の ROUGE 値。* は、各列について太字で示された最高値と比較して、並べ替え検定 ($p \leq 0.05$) により有意な差が確認できた値を示している。

source	five-time world champion michelle kwan withdrew from the ##### us figure skating championships on wednesday , but will petition us skating officials for the chance to compete at the ##### turin olympics .
reference	injury leaves kwan 's olympic hopes in limbo
<i>fixLen</i> (30)	kwan withdraws from us gp
(50)	kwan withdraws from us skating championships
(75)	kwan pulls out of us figure skating championships for turin olympics
<i>fixRng</i> (30)	kwan withdraws from us gp
(50)	kwan withdraws from figure skating championships
(75)	kwan pulls out of us figure skating championships for turin olympics bid
<i>LenEmb</i> (30)	kwan withdraws from us figure
(50)	kwan withdraws from us figure skating championships
(75)	world champion kwan withdraws from ##### olympic figure skating championships
<i>LenInit</i> (30)	kwan quits us figure skating
(50)	kwan withdraws from ##### us figure skating worlds
(75)	kwan withdraws from ##### us figure skating championships for ##### olympics

表2 各手法が実際に出力した文の例。

を利用した。

LenEmb において、 L は 300 とした。これは本研究で用いるデータセットにおける要約文の最大長よりも大きな数である (図 1-(b) および図 5-(b) を参照されたい)。ビーム探索時のビーム幅は *fixRng* 以外のモデルでは 10 とした。前述の通り *fixRng* では多くの事例がビームから削除されるため、ビーム幅を 100 にして実験を行った。また、*fixRng* における範囲の最大値は指定された長さ、最小値は最大値から 5 を減じた値を用いた。

6. 実験結果

6.1 ROUGE による評価

表 1 に、各長さ制限における ROUGE 値を示す。ここで、*fixLen* および *fixRng* はその性質上、出力系列の長さに関するハードな制約があるため必ず制限長以下の長さの系列が出力されるが、*LenEmb* と *LenInit* はハードな制約として長さの制御を学習しているわけではないため、ビーム探索の終了時には制限長を微妙に超える系列も存在してしまう。比較を公平に行うため、ROUGE 値を計算するときは *LenEmb* および *LenInit* にも長さをハードに制約したビーム探索を行う。具体的には、*fixRng* における最小値を 0 にした設定でビーム探索を行う。

ただし、以降の節で詳しく述べる表 2、表 3 および表 4-(a) については長さに関する制約は一切用いずにビーム探索を行った結果を示す。これは、学習ベースの手法に基づいた

モデルが獲得した出力長の制御の能力を示すためである。

表 1 を見ると、学習に基づく手法 (*LenEmb* および *LenInit*) は、50 バイトおよび 75 バイト設定において探索に基づく手法 (*fixLen* および *fixRng*) を上回る傾向にあることがわかる。しかしながら、30 バイト設定において両者に有意な差は見られなかった。これは、30 バイト設定においてモデルが強いられる平均圧縮率は 15.38% であり、訓練時に観測した平均圧縮率である 30% (図 1-(c)) との乖離が大きいことが原因として考えられる。

6.2 各手法の出力例

表 2 および表 3 に、訓練データセットの検証セットにおいて各モデルが出力した要約を示す。表より、4 つの手法のいずれの例においても、多くの場合で定性的に破綻の無い文を生成できていることが確認できる。“##### us figure championships”^{*3} や “withdrew” に対する多様な言い換えなどが観測できることは従来のニューラル文要約タスクでも報告されていたが、本稿の提案手法でも同様に観測できた。また、表 3 における *LenEmb* および *LenInit* の 30 バイト設定のように一つの名詞句としてタイトルを生成することは、短い要約文が必要なケースにおいて有効な能力であるといえる。

^{*3} “#” は、数字を置き換えたタグであり、“us” は “US” (United States) を意味している。

source	at least two people have tested positive for the bird flu virus in eastern turkey , health minister recep akdag told a news conference wednesday .
reference	two test positive for bird flu virus in turkey
<i>fixLen</i> (30)	two people tested positive for
(50)	two people tested positive for bird flu in eastern
(75)	two people tested positive for bird flu in eastern turkey says health minister
<i>fixRng</i> (30)	two infected in bird flu case
(50)	two tests positive for bird flu in eastern turkey
(75)	two people tested positive for bird flu in eastern turkey says minister
<i>LenEmb</i> (30)	two bird flu cases in turkey
(50)	two confirmed positive for bird flu in eastern turkey
(75)	at least two bird flu patients test positive for bird flu in eastern turkey
<i>LenInit</i> (30)	two cases of bird flu in turkey
(50)	two people tested positive for bird flu in turkey
(75)	two people tested positive for bird flu in eastern turkey health conference

表 3 各手法による別の事例 .

$\log p(\mathbf{y} \mathbf{x})$	byte	sentence	$\log p(\mathbf{y} \mathbf{x})$	byte	sentence
-4.27	31	two cases of bird flu in turkey	-5.05	57	two people tested positive for bird flu in eastern turkey
-4.41	28	two bird flu cases in turkey	-5.13	50	two tested positive for bird flu in eastern turkey
-4.65	30	two people tested for bird flu	-5.30	39	two people tested positive for bird flu
-5.25	30	two people tested in e. turkey	-5.49	51	two people infected with bird flu in eastern turkey
-5.27	31	two bird flu cases in e. turkey	-5.52	32	two tested positive for bird flu
-5.51	29	two bird flu cases in eastern	-5.55	44	two infected with bird flu in eastern turkey
-5.55	32	two people tested in east turkey	-6.00	49	two more infected with bird flu in eastern turkey
-5.72	30	two bird flu cases in turkey :	-6.04	54	two more confirmed cases of bird flu in eastern turkey
-6.04	30	two people fail bird flu virus	-6.50	49	two people tested positive for bird flu in turkey

(a) *LenInit* のビームの最終状態 .

(b) 出力系列の長さを考慮しなかった場合のビームの最終状態 .

表 4 *LenInit* と長さの考慮を伴わない encoder-decoder モデルにおけるビームの最終状態 .

6.3 学習ベース手法の出力長制御能力

表 4-(a) は、表 3 の 30 バイト設定において *LenInit* が要約文を出力した際の最終的なビームの状態である . 長さについて一切制約を加えない標準的なビーム探索であるにも関わらず、ビーム内のすべての系列が指定した 30 バイトに近い長さの文になっており、学習に基づく *LenInit* のモデルがその出力長の制御能力を獲得していることがわかる . なお、この傾向は *LenEmb* においても同様であった .

比較として、表 4-(b) に明示的に長さについて考慮しない標準的な設定で訓練した encoder-decoder モデル (すなわち、*fixLen* および *fixRng* で用いたもの) に対して標準的なビーム探索を行った際の最終的なビームの状態を示す . この例においても要約文自体は破綻のない原文に沿ったものになっているものの、その長さは多様であることがわかる .

6.4 既存手法との比較

最後に本節では、DUC2004 の共通課題 1 における標準的な 75 バイト設定における従来手法との比較を行う . 本稿の目的は本評価セットにおける最高精度を獲得することではないものの、出力長の制御機能を獲得した我々のモデルが ROUGE 値の観点で従来手法とどう異なるかを確認す

model	ROUGE-1	ROUGE-2	ROUGE-L
<i>fixLen</i>	25.88	7.93	23.07
<i>fixRng</i>	26.02	7.69	22.78
<i>LenEmb</i>	26.73	8.40	23.88
<i>LenInit</i>	25.87	8.28	23.25
ABS _[32]	26.55	7.06	22.05
ABS+ _[32]	28.18	8.49	23.81
RAS-Elman _[5]	28.97	8.26	24.06
RAS-LSTM _[5]	27.41	7.69	23.06

表 5 従来手法で報告されている ROUGE 値との比較 . 上 4 行は表 1 からの再掲 .

ることは興味深い .

表 5 に、表 1 から再掲した我々の手法の ROUGE 値と、主な従来手法による ROUGE 値を示す . 提案手法のうち *fixLen* が、モデルそのものの複雑度を無視した場合に、ABS[32] を始めとした他の従来手法と最も似た設定での実験を行っている . 表 5 に示す結果から、*fixLen* と従来手法との間には精度の面で大きな差異がないことが確認できる . また、学習ベースである *LenEmb* および *LenInit* が、ROUGE 値の劣化を避けつつ出力長の制御能力を獲得していることも分かる .

7. おわりに

本研究では、文要約タスクを対象に、encoder-decoder モデルにおいてその出力系列の長さを制御させるための取り組みを行った。本稿では探索ベースや学習ベースの4つの手法を提案し、定量的および定性的な実験を行った。実験の結果、学習ベースの手法が探索ベースの手法を上回る傾向にあることが分かり、学習ベースの手法はその出力系列の長さを制御する機能を、ROUGE 値の劣化を伴うことなく獲得できていることが分かった。

謝辞

本研究は JSPS 科研費 JP26280080 の助成を受けたものです。また、本研究を進めるにあたり株式会社ドワンゴ様から GPU サーバファーム紅莉栖の一部貸与を受けました。ここに記して謝意を表します。

参考文献

- [1] Ayana, Shen, S., Liu, Z. and Sun, M.: Neural Headline Generation with Minimum Risk Training, *CoRR*, Vol. abs/1604.01904 (2016).
- [2] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proceedings of ICLR15* (2015).
- [3] Banko, M., Mittal, V. O. and Witbrock, M. J.: Headline Generation Based on Statistical Translation, *Proceedings of ACL00*, pp. 318–325 (2000).
- [4] 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, *Proceedings of the EMNLP14*, pp. 1724–1734 (2014).
- [5] Chopra, S., Rush, A. M. and Auli, M.: Abstractive Summarization with Attentive Recurrent Neural Networks, *Proceedings of the NAACL-HLT16* (2016).
- [6] Cohn, T. and Lapata, M.: Sentence Compression Beyond Word Deletion, *Proceedings of COLING08*, pp. 137–144 (2008).
- [7] Cohn, T. and Lapata, M.: An Abstractive Approach to Sentence Compression, *ACM TIST13*, Vol. 4, No. 3, pp. 41:1–41:35 (2013).
- [8] Dorr, B., Zajic, D. and Schwartz, R.: Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation, *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pp. 1–8 (2003).
- [9] Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L. and Vinyals, O.: Sentence Compression by Deletion with LSTMs, *Proceedings of EMNLP15*, pp. 360–368 (2015).
- [10] Filippova, K. and Altun, Y.: Overcoming the Lack of Parallel Data in Sentence Compression, *Proceedings of EMNLP13*, pp. 1481–1491 (2013).
- [11] Filippova, K. and Strube, M.: Dependency Tree Based Sentence Compression, *Proceedings of INLG08*, pp. 25–32 (2008).
- [12] Galanis, D. and Androutsopoulos, I.: An extractive supervised two-stage method for sentence compression, *Proceedings of NAACL-HLT10*, pp. 885–893 (2010).
- [13] Graves, A., Jaitly, N. and Mohamed, A.: Hybrid speech recognition with Deep Bidirectional LSTM, *Proceedings of IEEE Workshop on ASRU13*, pp. 273–278 (2013).
- [14] Gu, J., Lu, Z., Li, H. and Li, V. O. K.: Incorporating Copying Mechanism in Sequence-to-Sequence Learning, *Proceedings of ACL16 (to appear)* (2016).
- [15] Gülçehre, Ç., Ahn, S., Nallapati, R., Zhou, B. and Bengio, Y.: Pointing the Unknown Words, *Proceedings of ACL16 (to appear)* (2016).
- [16] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).
- [17] Józefowicz, R., Zaremba, W. and Sutskever, I.: An Empirical Exploration of Recurrent Network Architectures, *Proceedings of ICML15*, pp. 2342–2350 (2015).
- [18] Kalchbrenner, N. and Blunsom, P.: Recurrent Continuous Translation Models, *Proceedings of EMNLP13*, pp. 1700–1709 (2013).
- [19] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *Proceedings of ICLR15* (2015).
- [20] Klerke, S., Goldberg, Y. and Søgaard, A.: Improving sentence compression by learning to predict gaze, *Proceedings of NAACL-HLT16*, pp. 1528–1533 (2016).
- [21] Li, J., Galley, M., Brockett, C., Gao, J. and Dolan, B.: A Diversity-Promoting Objective Function for Neural Conversation Models, *Proceedings of the NAACL-HLT16* (2016).
- [22] Li, J., Galley, M., Brockett, C., Gao, J. and Dolan, B.: A Persona-Based Neural Conversation Model, *Proceedings of ACL16 (to appear)* (2016).
- [23] Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81 (2004).
- [24] Lopyrev, K.: Generating News Headlines with Recurrent Neural Networks, *CoRR*, Vol. abs/1512.01712 (online), available from <http://arxiv.org/abs/1512.01712> (2015).
- [25] Luong, T., Pham, H. and Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation, *Proceedings of EMNLP15*, pp. 1412–1421 (2015).
- [26] Nallapati, R., Xiang, B. and Zhou, B.: Sequence-to-Sequence RNNs for Text Summarization, *CoRR*, Vol. abs/1602.06023 (online), available from <http://arxiv.org/abs/1602.06023> (2016).
- [27] Napoles, C., Callison-Burch, C., Ganitkevitch, J. and Van Durme, B.: Paraphrastic Sentence Compression with a Character-based Metric: Tightening without Deletion, *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pp. 84–90 (2011).
- [28] Napoles, C., Gormley, M. and Van Durme, B.: Annotated Gigaword, *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pp. 95–100 (2012).
- [29] Nenkova, A. and McKeown, K.: Automatic Summarization, *Foundations and Trends® in Information Retrieval*, Vol. 2-3, pp. 103–233 (2011).
- [30] Ranzato, M., Chopra, S., Auli, M. and Zaremba, W.: Sequence Level Training with Recurrent Neural Networks, *CoRR*, Vol. abs/1511.06732 (online), available from <http://arxiv.org/abs/1511.06732> (2015).
- [31] Rosenfeld, R., Chen, S. F. and Zhu, X.: Whole-sentence exponential language models: a vehicle for linguistic-statistical integration., *Computer Speech & Language*, Vol. 15, No. 1, pp. 55–73 (2001).
- [32] Rush, A. M., Chopra, S. and Weston, J.: A Neural Attention Model for Abstractive Sentence Summarization, *Proceedings of EMNLP15*, pp. 379–389 (2015).
- [33] Schuster, M. and Paliwal, K.: Bidirectional Recurrent Neural

- Networks, *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2673–2681 (1997).
- [34] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C. and Pineau, J.: Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models, *Proceedings of AAAI16*, pp. 3776–3784 (2016).
- [35] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Proceedings of NIPS14*, pp. 3104–3112 (2014).
- [36] Tokui, S., Oono, K., Hido, S. and Clayton, J.: Chainer: a Next-Generation Open Source Framework for Deep Learning, *Proceedings of NIPS15 Workshop on LearningSys* (2015).
- [37] Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. and Hinton, G. E.: Grammar as a Foreign Language, *Proceedings of NIPS15*, pp. 2773–2781 (2015).
- [38] Vinyals, O., Toshev, A., Bengio, S. and Erhan, D.: Show and tell: A neural image caption generator, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164 (2015).
- [39] Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D. and Young, S.: Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems, *Proceedings of EMNLP15*, pp. 1711–1721 (2015).
- [40] Woodsend, K., Feng, Y. and Lapata, M.: Title Generation with Quasi-Synchronous Grammar, *Proceedings of the EMNLP10*, pp. 513–523 (2010).
- [41] Wubben, S., van den Bosch, A. and Kraemer, E.: Sentence Simplification by Monolingual Machine Translation, *Proceedings of ACL12*, pp. 1015–1024 (2012).
- [42] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, *Proceedings of ICML15* (Blei, D. and Bach, F., eds.), pp. 2048–2057 (2015).
- [43] Zajic, D., Dorr, B. J. and Schwartz, R.: Bbn/umd at duc-2004: Topiary, *Proceedings of NAACL-HLT04 Document Understanding Workshop*, pp. 112 – 119 (2004).