クエリ・出力長を考慮可能な文書要約モデル

斉藤いつみ　西田京介　大塚淳史　西田光甫　浅野久子　富田準二

NTT メディアインテリジェンス研究所

1 はじめに

ニューラルネットに基づく要約技術は進展しており，特に要約対象であるソーステキストから重要な部分を抽出するモデル（内容選択）と，ソーステキストに含まれない単語系列を生成するモデルを組み合わせた要約モデルが高い精度を実現することが示されている [9]．ここで，文書要約に対してユーザが持つ代表的なニーズとして，要約の焦点（クエリ）および出力長の指定が挙げられる．これらを制御することは従来から取り組まれてきた [8, 2] が，それぞれ個別のモデルとして提案されていた．特に，要約の焦点をクエリとして与えるクエリ依存要約において出力長の制御が可能なモデルは提案されていない．そこで，本研究では要約におけるクエリ・出力長を同時に考慮可能な文書要約モデルの提案を行う．本研究の貢献を下記に示す．

- **内容選択モデルにおけるクエリ依存性・非依存性の考慮．** クエリ依存性を内容選択モデルにおいて考慮し，生成モデルはクエリ依存・非依存に関わらず同じ構造とする．この構成により，クエリ依存性の有無を統一的なフレームワークにて説明可能となる．
- **内容選択モデルにおける出力長の制御．** 従来，内容選択モデルにおける出力長の制御は行われていなかった．内容選択モデルに出力長制御の仕組みを導入し，出力長をクエリ依存性と同時に制御可能であることを確認した．

2 関連研究

2.1 ニューラル要約

抽出と生成に基づく最も代表的なモデルは See ら [9] による pointer-generator モデルである．彼らは単語を生成するかソーステキストからのコピーに基づいて出力するのかを逐次的に振り分ける重みを学習した．Hsu ら [5] は，文レベルの抽出型要約の考え方を導入し，文レベルの重みを推定した上でソース文書からのコピー確率に重み付けする手法を提案した．Gehrmann ら [3] は，単語レベルでソース単語が要約に含まれるか否かの 1/0 モデルを推定し，ソース単語のコピー確率を重み付けることによって精度の向上を報告している．これらの研究は，クエリ非依存の設定でのみ検討されており，クエリ依存の設定では検討されていない．

2.2 クエリ依存要約

ソース文書から要約文を生成する通常の問題設定に加え，クエリが与えられた時の要約を求める問題設定をクエリ依存要約と呼ぶ．クエリ依存要約に関しては公開データの量が少ないため，ニューラルモデルに基づく研究は少ないが，クエリ, 文書の encoder を用いて

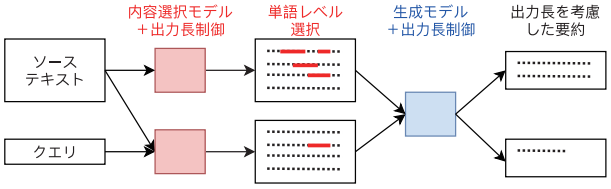

図 1: 提案フレームワーク．上：クエリ依存，下：非依存．内容選択と生成の両方で要約長を制御する．

要約文を生成する研究が複数存在する．[8, 4]．ただしこれらの研究ではクエリに基づくテキストの内容選択は明示的に考慮されていない．

2.3 出力長考慮型要約

要約の出力長を考慮するモデルはこれまでに複数提案されている [7]．これらのモデルは全て，デコーダのみに出力長の情報を考慮するモデルであり，内容選択は考慮していない．本研究では，デコーダに加え内容選択モデルにも出力長情報を考慮するモデルを提案する．

3 問題定義

本研究で取り組む問題について次のように定義する．

問題 1 クエリ非依存要約． ソーステキスト X^C，出力長 L を受け取り，出力長 L を考慮した要約 Y^L を生成する．

問題 2 クエリ依存要約． クエリ X^Q，ソーステキスト X^C，出力長 L を受け取り，出力長 L を考慮したクエリ依存要約 Y^L を生成する．

ここで，X^C，X^Q，Y^L，Y はそれぞれ単語列であり，$X^C = x_1^C, \ldots, x_n^C$，$X^Q = x_1^Q, \ldots, x_m^Q$，$Y^L = y_1^L, \ldots, y_k^L$，$Y = y_1, \ldots, y_k$ とする．

4 提案手法

4.1 提案技術の全体像

提案モデルは大きく 2 つの構成要素（内容選択モデル，生成モデル）からなる．内容選択モデルは，要約に含めるべき単語を予測するモデルであり，生成モデルは内容選択モデルの予測値を考慮しながら単語を生成する．図 1 に提案モデルの枠組みを示した．提案技術の枠組みは，内容選択モデルを差し替えるだけでクエリ依存/非依存の双方に対応可能な形式となっている．

4.2 内容選択モデル

内容選択モデルは，ソーステキストの各単語 $x_1^C, \ldots, x_n^C$ に対して要約文に含まれるか否かの確率 $p_1^{\text{ext}}, \ldots, p_n^{\text{ext}} \in [0, 1]$ を予測する．p_t^{ext} が 1 に近いほ

ど t 番目の単語 x_t^C が要約文に含まれやすいことを表す．同様の問題設定は既存研究でも提案されており，クエリ非依存の場合は Gehrmann ら [3]，クエリ依存の場合は Wang ら [10] を参考とする．

4.2.1 クエリ依存

まず，ソーステキスト単語列 X^C とクエリ単語列 X^Q の単語 embedding 系列 $e_1^C, \ldots, e_n^C$ および $e_1^Q, \ldots, e_n^Q$ を隠れ状態 d 次元の BiGRU を用いてエンコードする．

$$u_t^Q = \mathrm{BiGRU}(u_{t-1}^Q, e_t^Q) \quad (1)$$

$$u_t^C = \mathrm{BiGRU}(u_{t-1}^C, e_t^C) \quad (2)$$

次に，上記の情報を用いてクエリとソーステキストのマッチングを行う．この際，マッチングは [6] を用いる．Hu ら [6] のモデルでは，主に interactive alignment，self alignment，evidence collection の 3 つのモジュールを用いてマッチングを表現している．

Interactive alignment 本モジュールは，入力として $(u^C \in \mathbb{R}^{2d\times n}, u^Q \in \mathbb{R}^{2d\times m})$ を受け取り，$o \in \mathbb{R}^{2d\times n}$ を出力する．説明を簡易にするため，以下，入力を (x, y) とする．本モジュールは attention match と semantic fusion の 2 つのサブモジュールで構成される．まず，**attention match**(x, y) は，それぞれのベクトルの attention を用いて，各位置 t での重みつきベクトル $h_t \in \mathbb{R}^{2d}$ を求める．

$$h_t = \textstyle\sum_j \alpha_{tj} y_j \quad (3)$$

$$\alpha_{tj} = \frac{\exp(x_t \cdot y_j)}{\sum_j \exp(x_t \cdot y_j)} \quad (4)$$

次に，**semantic fusion**(x_t, h_t) は，x_t, h_t の相互作用を考慮したベクトル $o_t \in \mathbb{R}^{2d}$ を計算する．

$$\tilde{x}_t = f(W_r[x_t; h_t; x_t \circ h_t; x_t - h_t]) \quad (5)$$

$$g_t = \sigma(W_g[x_t; h_t; x_t \circ h_t; x_t - h_t]) \quad (6)$$

$$o_t = g \circ \tilde{x}_t + (1 - g) \circ x_t \quad (7)$$

σ はシグモイド関数を表す．f は ReLU を表す．$W_r \in \mathbb{R}^{2d\times 8d}$, $W_g \in \mathbb{R}^{2d\times 8d}$ は学習パラメータである．$[;]$ はベクトル連結の演算子を表す．

Self alignment モデル構造は interactive alignment と共通である．本モジュールへの入力は，interactive alignment の出力を用いて $(x, y) = (o, o)$ とする．

Evidence collection 本モジュールは self alignment の出力 o に BiGRU により $v \in \mathbb{R}^{2d\times n}$ を得る．

$$v_t = \mathrm{BiGRU}(v_{t-1}, o_t) \quad (8)$$

最後に，2 層のフィードフォワードネットワークを作用させ，ソーステキストの各単語ごとに 1 次元の単語選択スコアを出力する．

$$p_t^{\mathrm{ext}} = \sigma(w_2^\top f(W_1 v_t + b_1) + b_2) \quad (9)$$

ここで，$W_1 \in \mathbb{R}^{d\times 2d}$, $b_1 \in \mathbb{R}^d$, $w_2 \in \mathbb{R}^{2d}$, $b_2 \in \mathbb{R}$ は学習パラメータである．

4.2.2 クエリ非依存

クエリ非依存の場合はクエリが存在しないが，ソーステキスト全体に対して重要な単語を選択する必要があるため，ソーステキスト全体の情報を考慮して各単語ごとのラベル予測を行う．まず，ソーステキストを BiLSTM を用いてエンコードする．

$$u_t^C = \mathrm{BiLSTM}(u_{t-1}^C, e_t^C) \quad (10)$$

クエリ依存における attention match の代わりに，u_t^C を用いて文書レベルのベクトル $u_t^{C_d}$ を求める．

$$u_t^{C_d} = \textstyle\sum_j \alpha_{tj} u_j^C \quad (11)$$

$$\alpha_{tj} = \frac{\exp(w_d \cdot u_t^C)}{\sum_j \exp(w_d \cdot u_j^C)} \quad (12)$$

ここで，$w_d \in \mathbb{R}^{2d}$ は学習パラメータである．次に，入力 $(x, y) = (u_t^C, u_t^{C_d})$ を semantic fusion へ作用させる．以降，式 10 を作用させず $v_t = o_t$ とする点を除きクエリ依存の場合と同様の処理を行う．

クエリ依存の場合はクエリとソーステキストの相互関係を考慮しながらモデル化を行っているのに対し，クエリ非依存の場合は文書レベルのベクトルと単語レベルのベクトルの双方を考慮しながらモデル化する．

4.2.3 出力長を考慮するモデル

出力長を制御することを考えるとき，内容選択の出力そのものが長さに依存して変化することが望ましい．例えば，10 単語出力するときと 30 単語出力するときでは内容選択モデルの出力分布自体を変化させたいため，出力単語数に依存したモデル化を行う．具体的には，式 1 の入力ベクトル e^C に，出力長 L に対応する embedding として e^L を次のように連結する．

$$u_t^C = \mathrm{BiGRU}(u_{t-1}^C, [e_t^C; e^L]) \quad (13)$$

このとき与える正解の出力長情報としては，参照要約の長さを与える．e^L は学習パラメータである．

4.2.4 学習データの作成

内容選択モデルに関しては正解データが存在しないため，既存研究 [3] や [5] に類似した方法で疑似的な正解データを自動生成する．具体的には，まず [5] と同様に文レベルの Rouge-L の Recall スコアを用いて疑似的な文正解ラベルをソーステキストの各文に付与する．そして，文ラベルが 1 となった文集合を疑似ソーステキストとして，参照要約と疑似ソーステキスト間で DP マッチングを用いて単語アライメントを計算し，単語が一致した場合には 1，それ以外の単語は 0 の単語選択ラベルを付与した．内容選択モデルの学習時には，単語選択ラベルを正解ラベル r として学習を行った．

4.3 内容選択モデルと生成モデルの結合

提案モデルでは，See らのベースモデル [9] の attention 確率を内容選択モデルの出力 p_t^{ext} にて重み付けする combined attention[5] を用いる．さらに，生成モデルでも出力長の制御を行う．

4.3.1 生成モデル

生成モデルのベースモデルとして，多くの要約モデルで用いられている See ら [9] の pointer-generator モデルを用いる．pointer-generator モデルでは，要約

$Y = y_1, \ldots, y_k$ の生成確率を次のように定義する．

$$\begin{aligned} & p(y_j|y_{1:j-1}, x) \\ = \;& p(z_j = 1|y_{1:j-1}, x) \times p(y_j|z_j = 1, y_{1:j-1}, x) \\ + \;& p(z_j = 0|y_{1:j-1}, x) \times p(y_j|z_j = 0, y_{1:j-1}, x) \end{aligned}$$

ここで，$p(z)$ は y_j がソーステキストからコピーされるか否かの確率を表す．また，$z = 1$ の y_j がコピーされる場合の確率 $p(y_j|z_j = 1, y_{1:j-1}, x)$ を時刻 j の attention 分布 a_j に基づき，

$$p(y_j|z_j = 1, y_{1:j-1}, x) = \sum_{t:x_t^C = y_j} a_{jt} \tag{14}$$

と定義する．時刻 j のデコーダ状態を s_j としたとき，

$$a_{jt} = \exp(g_{jt})/(\textstyle\sum_{t'} \exp(g_{jt'})) \tag{15}$$

$$g_{jt} = w_a^\top \tanh(W_v v_t + W_s s_j + b_a) \tag{16}$$

である．$w_a \in \mathbb{R}^d$，$W_v \in \mathbb{R}^{d \times 2d}$, $W_s \in \mathbb{R}^{d \times 2d}$ $b_a \in \mathbb{R}^d$ は学習パラメータである．

4.3.2 Combined Attention

Combined attention は，ベースモデルの attention 確率 $p(a_j|y_{1:j-1}, x)$ を内容選択モデルの出力 p^{ext} を用いて以下の様にアテンション分布を再定義する．

$$\tilde{a_j} = \frac{a_j p_j^{\text{ext}}}{\sum_j a_j p_j^{\text{ext}}} \tag{17}$$

4.3.3 出力長を考慮するモデル

デコーダ側で出力長を考慮するモデルは，既存研究でも扱われているように，長さ embedding を用いて定式化する．本研究では，出力すべき残存長さ embedding をデコーダの各入力に連結する方法を用いる．具体的には，デコーダの LSTM の入力に下記の様に長さ embedding ベクトル $\tilde{e}^l$ を結合する．

$$h_t = \text{LSTM}(h_{t-1}, [e_t^x; \tilde{e}^l]) \tag{18}$$

ここで，$l = max((L - t), 0)$ とする．なお，$\tilde{e}^l$ は内容選択で用いる e^L とは別の学習パラメータである．

4.4 学習

4.4.1 事前学習と fine tuning

内容選択モデルと生成モデルは，事前にそれぞれの目的関数で事前学習を行う．その後，内容選択モデルを固定し結合生成モデルを学習する．

4.4.2 目的関数

内容選択モデル，生成モデルの目的関数をそれぞれ L_{ext}，L_{gen} とする．内容選択モデルは各単語に対する二値識別問題であるためバイナリクロスエントロピー関数とし，生成モデルは一般的な言語モデルと同様に定義する．

$$L_{\text{ext}} = -\frac{1}{N}\sum_{n=1}^{N}\{r \log p^{\text{ext}} + (1 - r)\log(1 - p^{\text{ext}})\}$$

$$L_{\text{gen}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J}\log p(y_j|y_{1:j-1}, x)$$

	CNN/DM	MS-MARCO
train	287,113	300,000
validation	13,368	10,000
test	11,490	-

表 1: 実験に利用したデータ数.

5 評価実験

5.1 実験データ

クエリ非依存データとして要約タスクで標準的に用いられる CNN/DM を用いた．クエリ依存データとしては大規模なデータが存在しないため MS-MARCO 2.1[1] の Q&A タスクデータを用いた．Q&A タスクデータは，クエリに対して対象となるソーステキスト（Bing の検索結果 10 件）をベースに回答を人手で作成したデータである．回答中の単語の多くがソーステキストからの抽出で構成されているため，クエリ依存要約の設定と考えることが出来る．本実験では上記 10 件のテキストを連結したものを要約の対象とする．MS-MARCO は，オリジナルのデータは大規模であるため，今回の実験では学習データとして 30 万件，評価データとして 1 万件をそれぞれランダムサンプリングして用いた．各データの概要を表 1 に示した．

5.2 実験設定

クエリ非依存データに関しては，要約の従来手法と同様に，学習時はソーステキストの長さを 400 単語，要約テキストの長さを 100 単語に truncate して学習を行った．テスト時の単語数上限は 120 とした．クエリ依存データに関しては学習時のソーステキストの最大長を 800 単語とした．最適化手法は Adagrad を用い，事前学習時の初期学習率は 0.15，fine tuning 時の初期学習率は 0.05 に設定した．また initial accumulator value は既存研究と同様に 0.1 とした．word embedding は glove の 100 次元ベクトルを用い固定した．GRU, LSTM の隠れ層は 256 に設定した．長さ embedding の e^L, $\tilde{e}^l$ は 100 次元とした．また，dropout は 0.3 とした．

5.3 評価方法

提案モデルは内容選択モデルにおけるクエリ依存性および出力長の制御を行う点が貢献である．まず，5.4.1 節にて，これらの制御を行った場合の内容選択と生成モデルを結合した際の精度について Rouge-L により評価する．次に，5.4.2 節にて，紙面の制約上，クエリ依存要約に絞って出力長制御の分析結果を示す．

5.4 結果と考察

5.4.1 出力長を制御した要約精度に関する評価

内容選択＋生成モデルは指定した出力長に応じて重要な情報を要約出来ているか？ 表 2 に提案モデル（内容選択＋生成）全体の出力長を制御した場合の Rouge-L スコアを示す．まず，クエリ依存/非依存のいずれの場合も出力長を短くすると識別率が上昇し，長くすると再現率が上昇していることから，より重要な情報を要約に含められていることが分かる．この結果より，クエリ依存の設定においても内容選択と長さのコントロールが有効に機能していることがわかる．また，CNN/DM の場合は出力長が長くなるほど Rouge の F 値が向上している．一方で MS-MARCO の場合は出

	CNN/DM			MS-MARCO		
出力長	識別率	再現率	F 値	識別率	再現率	F 値
10	58.6	8.43	8.62	36.3	20.9	19.4
20	50.3	17.5	18.9	27.6	31.0	24.2
30	47.8	23.3	25.5	23.2	36.2	22.9
gold	41.9	32.4	34.2	41.7	38.8	39.5

表 2: クエリ非依存（CNN/DM）/非依存（MS-MARCO）の Rouge-L に関する評価結果.

内容選択	R-1	R-2	R-L
出力長制御無し	38.4	16.5	33.9
出力長制御有り	38.7	17.1	34.2

表 3: 内容選択での出力長制御の効果（CNN/DM）

指定単語数	10	20	30
平均出力長	7.0	17.0	27.0

表 4: 指定単語数とシステムの平均出力長

力長制約が 20 の場合に最も F 値が高くなっている．これは，元々のデータの要約文の長さに起因しており，CNN/DM の場合は要約の平均長が長く MS-MARCO の場合は要約の平均長が短いため，このような挙動となっている．

内容選択モデルでの出力長制御は有効か？ 表 3 に，クエリ非依存要約（CNN/DM）において，生成モデルに加えて内容選択モデルでも出力長の制御（gold 条件）を考慮する場合としない場合の結果を示す．内容選択モデルで出力長の制御を行うことで Rouge スコアが改善していることが分かる．

5.4.2 クエリ依存要約の出力長制御に関する分析

システムは指定した出力長に沿って要約を出力しているか？ 表 4 に，指定した出力長と実際の出力長を示した．システムは指定された出力長付近の長さを出力出来ていることがわかり，クエリ依存の設定でも適切に出力長を制御出来ることがわかった．特に，MS-MARCO は 1 文程度の短い出力が多いため，短い長さのバリエーションが多く出力長情報をうまく学習出来たと考えられる．

出力例 表 5 に提案モデルの出力例を示す．提案手法は，指定した単語数に応じて，パッセージの広範囲な部分から適切な部分を抽出かつ不足した情報を生成し，ユーザの情報要求に適合した出力が出来ていることが分かる．特に，正解の長さに近い単語数制約 25 の出力においては，正解の単語列との一致度が高い．ただし，関連する情報を長さに応じて出力することは出来ているが，特に重要な単語やフレーズを特定してそれを維持しながら文を生成するなどの細かい制御は出来ておらず，関連しそうな文をまとめてコピーする挙動が多くみられた．内容選択において，特に重要な単語をより高精度に予測し，それらを出来る限り要約に含める生成モデルの実現が重要と考える．

6 おわりに

本研究では，内容選択モデルと生成モデルを組み合わせる方法をクエリ依存/非依存のそれぞれに適用可能なモデルに拡張し，出力長制御の組み合わせについても検討を行った．本研究の貢献は大きく 2 つ存在する．

質問	what is the pepita seeds
ソース	Pumpkin seeds after shelling, roasting, and salting. Pumpkin seed, also known as pepita (from Mexican Spanish: pepita de calabaza, little seed of squash), are the edible seeds of a pumpkin or certain other cultivars of squash. (...) Pumpkin seeds contain healthy fats as well as protein, fiber, potassium, iron and zinc. Pumpkin seeds are a good source of zinc, a mineral present in many foods. (...) Pumpkins, and their seeds, are native to the Americas, and indigenous species are found across North America, South America, and Central America.
正解	Pumpkin seed, also known as pepita are the edible seeds of a pumpkin or certain other cultivars of squash.
出力 (10)	it means little seed of squash .
出力 (15)	a good source of zinc , a mineral present in many foods
出力 (20)	it contain healthy fats as well as protein , fiber , potassium , iron and zinc .
出力 (25)	pumpkin seed , also known as pepita (from mexican spanish : pepita de calabaza , little seed of squash) .
出力 (30)	pumpkin seeds are a good source of zinc , a mineral present in many foods . pumpkins , and their seeds , are native to the americas

表 5: MS-MARCO データでの出力長制御例．括弧内の数字は指定した出力長である．

まず，内容選択モデルについてクエリ依存に拡張し，クエリ非依存と統一的に記述可能な定式化を行った点である．今回はクエリに対して内容選択モデルを定義したが，クエリ以外のキーワードや対話的な文脈など，様々な条件付きの内容選択モデルに拡張可能である．

さらに，これらの枠組みに出力長制御の概念を導入し，クエリ依存の設定でも要約の出力長を適切に制御可能であることを確認した．要約の観点（クエリ）および出力長を 1 つのモデルで制御可能にした提案モデルは様々なユーザのニーズに合わせた柔軟な要約の実現に貢献すると考える．今後は，内容選択と生成の組み合わせをより高度にしていくことで，高品質かつ柔軟に制御可能な要約生成手法について検討を行う．

参考文献

[1] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268v3, 2018.

[2] A. Fan, D. Grangier, and M. Auli. Controllable abstractive summarization. In *NMT@ACL*, pages 45–54, 2018.

[3] S. Gehrmann, Y. Deng, and A. Rush. Bottom-up abstractive summarization. In *EMNLP*, pages 4098–4109, 2018.

[4] J. Hasselqvist, N. Helmertz, and M. Kågebäck. Query-based abstractive summarization using neural networks. *CoRR*, abs/1712.06100, 2017.

[5] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In *ACL (1)*, pages 132–141, 2018.

[6] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou. Reinforced mnemonic reader for machine reading comprehension. In *IJCAI*, pages 4099–4106, 2018.

[7] Y. Liu, Z. Luo, and K. Zhu. Controlling length in abstractive summarization using a convolutional neural network. In *EMNLP*, pages 4110–4119, 2018.

[8] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran. Diversity driven attention model for query-based abstractive summarization. In *ACL (1)*, pages 1063–1072, 2017.

[9] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL (1)*, pages 1073–1083, 2017.

[10] Y. Wang, K. Liu, J. Liu, W. He, Y. Lyu, H. Wu, S. Li, and H. Wang. Multi-passage machine reading comprehension with cross-passage answer verification. In *ACL (1)*, pages 1918–1927, 2018.