

複数エンコーダを用いたヤフートピックス見出し候補生成

小林 健† 小林 隼人†† 村尾 一真† 増山 毅司†

† ヤフー株式会社 †† 理化学研究所 AIP センター

{kenkoba, hakobaya, kmurao, tamasuya}@yahoo-corp.jp

1 はじめに

スマートデバイスの普及に伴い、コンテンツを閲覧するデバイスはPCからスマートフォン等に移り変わってきている。しかし、これらのデバイスのディスプレイのサイズはPCと比較して小さく、ユーザーが様々なコンテンツを視認するために、コンテンツのタイトルを一定の長さ以内に抑える技術が求められている。

一方で、日々のニュースをピックアップする Yahoo! ニュース・トピックスでは、Yahoo! JAPAN のトップページの表示領域に合わせて、編集者の手により全角文字を 1 文字、半角文字を 0.5 文字として、合計 13.5 文字に収まるように、トピックスページのタイトルを作成してきた (図 1)。この作業を何らかの方法によって学習することができれば、限られた文字数でニュースの要点を押さえたトピックスページのタイトルの生成が可能になる。ここで、トピックスページは、あるニュースに対して記事詳細ページや関連リンクなどを編集したページであり、例えば、図 1 の右の例では最上部の黒字でトピックス見出し、青字で記事の見出し (詳細記事へのリンク)、グレーの文字で記事リード文が記載されている。以後、トピックスページのタイトルをトピックス見出し、記事の見出しを記事タイトルとして議論を行う。

トピックス見出しをニュース記事を凝縮したものだ考えると、トピックス見出しの生成は要約問題として捉えることができる。特に編集者がトピックス見出しを作成する際には、記事タイトル、記事の中身、関連記事等を考慮しながら適切な長さに収めているため、略語、体言止め、順序の入れ替え、といった処理が発生し、記事タイトルや記事中に含まれない文字が出現する場合もある。このため、トピックス見出しの生成を適切に学習するためには、抽象型要約として捉える必要がある。

近年では、エンコーダ・デコーダ [1] を用いたモデルが翻訳や対話などの文字列生成タスクで高い成果を出しており、記事タイトル生成 [2, 3] のような抽象型



記事提供: THE PAGE (2017年1月11日)

図 1: Yahoo!ニュース・トピックス

要約に関する研究も盛んに行われている。そのため、トピックスの見出し生成においても、エンコーダ・デコーダを用いたモデルが有効だと推測される。そこで本稿では、エンコーダ・デコーダの枠組みを利用して、トピックス見出し生成に取り組む。その際、トピックスページ中の記事タイトルと記事リード文を同時に利用するため、エンコーダ・デコーダの枠組みを拡張し、複数のエンコーダ出力を用いてトピックス見出しを生成する手法を提案する。

2 エンコーダ・デコーダモデル

Bahdanau らのエンコーダ・デコーダを用いたモデル [1] について説明する。このモデルでは、長さ S の入力文字列を $X = [x_1, x_2, \dots, x_S]$ 、長さ T の出力文字列を $Y = [y_1, y_2, \dots, y_T]$ として、以下の確率を最大にするように学習することで、 X から Y を生成するモデルを作成する。

$$p(Y | X) = \prod_{t=1}^{T-1} p(y_{t+1} | y_{\leq t}, X) \quad (1)$$

$p(y_{t+1} | y_{\leq t}, X)$ は X をベクトル化するエンコーダと Y を出力するデコーダを用いて、以下のように表さ

れる．

$$h_s = f_{\text{enc}}(x_s, h_{s-1}) \quad (2)$$

$$s_t = f_{\text{dec}}(y_t, s_{t-1}) \quad (3)$$

$$c_t = \sum_{s=1}^S \alpha_{t,s} h_s \quad (4)$$

$$p(y_{t+1} | y_{\leq t}, X) = g_{\text{dec}}(s_t, c_t) \quad (5)$$

ここで，式 (2) はエンコーダ，式 (3) から式 (5) はデコーダによる処理を表している．式 (2) の f_{enc} は入力 x_t と， $s-1$ 番目の状態ベクトル h_{s-1} を用いて s 番目の状態ベクトル h_s を計算する状態遷移関数である．式 (3) の f_{dec} も状態遷移関数であり， y_t と $t-1$ の状態ベクトル s_{t-1} を利用して s_t を計算している．なお， $s_0 = h_S$ である．式 (4) はアテンションと呼ばれる処理である． $\alpha_{t,s}$ は s_t と h_s に基づいて計算される重みで，各入力の h_s の重要度を表す．この重みつき和を計算することにより， s_t に関連する入力文字列のベクトルを計算することができる．そして，式 (5) の g_{dec} で c_t と s_t を入力として，次の出力の確率を計算する．

例えば，入力文字列を記事タイトル，出力文字列をトピックス見出しとすることで，記事タイトルからトピックス見出しを生成するモデルを作成することができる．

3 複数エンコーダを利用するモデル

編集者がトピックス見出しを作成する際は，記事タイトル，記事の内容，双方を参考にしており，どちらのデータも有用な情報源である．そこで，本研究では，記事タイトル，記事リード文をそれぞれエンコーダによってベクトル化し，その出力をデコーダで統合して利用する．3章では，3.1節と3.2節で複数のエンコーダの出力をデコーダで利用する関連研究について紹介し，3.3節で我々の提案する手法について述べる．

3.1 Query-based Attention モデル

複数のエンコーダの出力を利用して要約を行う研究として，Preksha らのクエリベースの要約に関する研究 [4] がある．質問文と回答文のペアから，回答文の要約を生成するタスクに取り組んでおり，式 (4) のアテンション重み $\alpha_{t,s}$ を計算する際に，質問文をベクトル化した情報を利用している．これにより，質問に関連する本文の部分を注目できるようになり，性能が向上したと報告している．本研究でも，記事リード文を

表 1: ハイパーパラメータ

パラメータ	値
層数 (RNN, CNN)	3
ユニット数 (Embedding)	200
ユニット数 (RNN, CNN)	400
CNN ウィンドウ幅	7
ドロップアウト率	0.3
学習率	0.05
モメンタム率	0.8
減衰率	0.85
エポック数	20
バッチサイズ	64

$\alpha_{t,s}$ の計算時に利用することで，適切なベクトルが計算できるようになると考えられる．

3.2 Multimodal Attention モデル

Hori らによる動画の説明文生成に関する研究 [5] でも複数のエンコーダの出力を組み合わせる利用している．彼らは，映像用のエンコーダ，音声用のエンコーダ等を利用して動画の説明文をエンコーダ・デコーダの枠組みで生成しているが，その際，エンコーダごとにスカラーの重みを動的に計算し，その重み付き和ベクトルを利用することで，精度が向上したと報告している．我々の問題でも，記事タイトルのベクトルを d_t ，記事リード文のベクトルを d'_t ， β をスカラーの重みとすると， $c_t = \beta d_t + (1 - \beta) d'_t$ のようにして，出力を統合する．

3.3 提案手法

Multimodal Attention モデルでは，重みをスカラー値で計算し，これを掛けることによって重み付き和を計算した．しかし， c_t の各要素がそれぞれ，特徴を表現する役割を担っていると考えると，要素ごとに重みを計算できれば，より適切な特徴を表現できると予想される．そこで，以下のようにして c_t を計算し，複数エンコーダの出力を統合する手法を提案する．

$$w_t = \sigma(W[d_t; d'_t; s_t]) \quad (6)$$

$$w'_t = \sigma(W'[d_t; d'_t; s_t]) \quad (7)$$

$$c_t = w_t \odot d_t + w'_t \odot d'_t \quad (8)$$

ここで， \odot は要素ごとの積， σ はシグモイド関数， W と W' は重み行列を表している．このように，要素ご

表 2: ROUGE 値による評価結果 (下線は上段のモデルで比較した場合の最良の数値で, 太字は全モデルを対象とした時の最良の数値)

	Simple			Complex			Total		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
タイトルのみ (5 モデル)	75.52	63.74	72.05	56.64	41.00	<u>52.07</u>	<u>60.50</u>	<u>45.65</u>	<u>56.15</u>
Query-based(5 モデル)	75.13	63.12	71.73	<u>56.67</u>	<u>41.07</u>	52.04	60.45	45.57	56.07
Multimodal(5 モデル)	75.45	63.46	72.01	56.38	40.75	51.81	60.28	45.39	55.94
提案手法 (5 モデル)	<u>75.66</u>	<u>63.96</u>	<u>72.35</u>	56.35	40.82	51.79	60.30	45.55	55.99
タイトルのみ (10 モデル)	75.52	63.64	72.13	56.60	40.97	51.99	60.47	45.61	56.10
提案+Query-based(10 モデル)	75.91	64.12	72.58	56.83	41.25	52.24	60.73	45.93	56.39

表 3: クラウドソーシング結果 (太字は下段のモデルで比較して最良の数値であり, * はタイトルのみ手法に対して対応のある t 検定の結果 $p < 0.01$ となる項目)

手法	可読性	有用性
編集者	3.623	3.179
先頭 13.5 文字	2.715	2.380
タイトルのみ	3.534	3.157
Multimodal	3.511	3.121
Query-based	3.524	3.112
提案手法	3.496	3.219*
提案手法+Query-based	3.553	3.224*

とに重みを計算することで, より詳細な重みつき和を計算できる.

4 実験

4.1 モデルの学習

モデルの学習は, Yahoo!ニュース・トピックスで掲載されたニュース記事 (トピックス見出し, 記事タイトル, 記事リード文) の 3 つ組を用いて行った. また, 学習時には形態素単位ではなく文字単位による学習を行った. これは, 事前実験を行った際, 形態素単位よりも文字単位の方がより高い精度を出すことができたためである. なお, 記事タイトルのエンコーダには双方向 LSTM(BiLSTM) を, 記事リード文のエンコーダには速度等を考慮し, CNN を利用した.

モデルのハイパーパラメータについて表 1 で示す. 各パラメータは埋め込み表現も含め, $(-0.1, 0.1)$ の一様分布よりサンプリングした値を用いて初期化した. パラメータの最適化では, SGD に Nesterov モメンタ

ムを用いた手法を利用し, 4 エポック経過後からは学習率を 15% づつ減衰させつつ学習を行った.

4.2 見出しの生成と評価

評価には, 次の 2 種類のデータを利用した.

- Simple: 記事タイトルに含まれる文字のみで見出しを生成可能なデータ
- Complex: Simple 以外のデータ (トピックス見出しに記事タイトルに無い文字が含まれる)

評価指標としては, 要約タスクの評価指標として利用されている ROUGE-1(R-1), ROUGE-2(R2), ROUGE-L(R-L) を用いた. また, 人手による評価として, クラウドソーシングを利用した見出しの可読性, 有用性の評価も行った. 可読性は内容の意味は考慮せず, 日本語として自然かを, 有用性は生成された見出しの内容が適切かを 1~4 の範囲で評価する. これを 1 記事あたり 10 人が行い, その平均値を利用する.

評価対象として, 2 章で説明した記事タイトルのみを用いたシンプルな手法 (タイトルのみ), 3 章で説明した, Query-based Attention(Query-based), Multimodal Attention(Multimodal), 提案手法の 4 手法を利用し, 比較実験を行なった. なお, モデルの学習結果は, ばらつきを考慮し, アンサンブルを行った結果で評価する.

4.3 実験結果

モデルを 5 個作成し, アンサンブルによる出力を評価した結果を表 2 の上段に示す. Simple データを対象にした場合には, 提案手法が最も良い性能を示したが, Complex データを対象にした場合の精度は最も

小さくなった。これは、記事タイトルと記事リード文双方で出現する文字の重みが大きくなり、結果的に、元タイトルの文字を優先的に出力するようになったと推察できる。Multimodal も、重み付き和を行うため、Complex を苦手とするが、より詳細に重みを計算する提案手法の方が Simple では良い性能となった。

Complex を対象にした場合には、Query-based が R-1, R-2 の値で若干その他手法を上回ったが、Simple データを対象とした場合の精度は最も小さくなった。この結果は、Query-based が積極的に記事タイトル以外に出現する文字を利用していることを示唆している。

Total(Complex データ+Simple データ)で見ると、タイトルのみを利用する手法が最も良い値を示した。今回検証した手法がそれぞれ異なった特徴を持ち、全体で平均すると悪い結果が良い結果を打ち消してしまうため、このような結果になったと予想される。

ここまでの議論により、Query-based と提案手法ではそれぞれ異なる特徴が明らかになった。これらの手法を組み合わせることにより、より良い結果が得られると予想できる。そこで、これら 2 手法のモデルをそれぞれ 5 個ずつ (計 10 個) 利用しアンサンブルを行った場合の結果と、タイトルだけのモデルを 10 個使ったアンサンブルを行った場合の結果を表 2 下段に示す。

表 2 下段を見ると、組み合わせたモデルでは Simple, Complex 共に精度が向上し、タイトルだけのモデルと比較しても良い結果となった。これは、双方の得意な部分をうまく取り入れることができたことを示している。

次に、クラウドソーシングの結果を表 3 に示す。この結果を見ると、提案手法が可読性は低いものの、有用性はタイトルのみ手法と比較しても有意に高くなっている。つまり、提案手法では読みやすさを犠牲にして、内容に関係のある語を利用するようになったことを示唆している。また、提案手法+Query-based では可読性、有用性、共に最も良い値になっており、この弱点を克服することができた。表 4 の実際の出力例を見てみても、提案手法+Query-based のモデルでは、読みやすさを犠牲にせず、必要な語を利用できていることがわかる。

5 おわりに

本論文では、記事のタイトルとリード文とを組み合わせたトピック見出し生成について取り組み、既存の複数エンコーダ利用手法と、提案手法の比較を行った。各手法、5 個ずつモデルを作成し、アンサンブル

表 4: 出力例 (I:記事タイトル E:編集者 T:タイトルのみモデル H:提案手法+Query-based モデル)

I:10 年ぶりの減益でローソンが岐路に ローソンに秘策はあるのか E:10 年ぶり減益 岐路のローソン T:ローソン ローソンに秘策ある? H:ローソン 10 年ぶり減益で岐路
I:逆境をチャンスに変えたダルビッシュの進化 E:術前より進化 ダルの肉體改造 T:逆境をチャンスに変えた進化 H:ダル 逆境をチャンスに変えた

記事提供 (上): THE PAGE (2016 年 10 月 17 日)
記事提供 (下): THE PAGE (2016 年 5 月 30 日)

で評価した結果、提案手法は Simple データに対して、Query-based は Complex データに対して強い傾向があり、全てのデータを利用した場合にはタイトルのみを利用した手法に劣ることがわかった。そこで、異なる特徴があると想定される Query-based と提案手法を組み合わせた結果、ROUGE 値、クラウドソーシングの評価結果ともに、タイトルのみ手法を上回る最も良い性能を出すことが明らかになった。

今後は、より適切な手法の開発や、強化学習を取り入れた手法の検討を行っていく予定である。

参考文献

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.
- [2] Alexander M. Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Sentence Summarization. *EMNLP*, 2015.
- [3] Konstantin Lopyrev. Generating News Headlines with Recurrent Neural Networks. *CoRR*, Vol. abs/1512.01712, , 2015.
- [4] Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. Diversity driven attention model for query-based abstractive summarization. In *ACL*, pp. 1063–1072, 2017.
- [5] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi. Attention-Based Multimodal Fusion for Video Description. In *ICCV*, 2017.