新聞記事からの都々逸生成のための訓練データの作成手法と生成アルゴリズムの改良

高昕 * 小坂想太朗 * 佐山龍之介 * 松崎拓也
東京理科大学 理学部第一部 応用数学科
1420038@ed.tus.ac.jp 1420039@ed.tus.ac.jp 1420046@ed.tus.ac.jp matuzaki@rs.tus.ac.jp

概要

新聞記事を入力とし，記事の内容を要約する都々逸を出力する齊藤ら [?] の手法の改良を試みた．具体的には，訓練データとなる都々逸の作成数を従来より増やし，候補の絞り込みを TF-IDF，BERT の回帰モデルを使用し改良した．また，生成アルゴリズムの制約を追加し，単語の重複を排除するように改良を行った．結果として，大幅な都々逸候補数の増加と生成する都々逸の質の向上に成功した．[1)]

1 はじめに

新聞記事の見出しは，本文の概要を正確に示す必要がある．また，簡潔で内容が伝わりやすいことが求められると同時にインパクトがあることが望ましい. 都々逸は語呂が良く, 頭に残りやすい七五調である．そこで，齋藤ら [?] は Text-toText 型のニューラル言語モデルを用いて新聞記事の内容を要約する都々逸を自動的に生成する手法を提案している．

都々逸とは，七・七・七・五のリズムに従う定型詩であり，始まりは江戸末期に寄席芸人が寄席で歌ったものが流行した事による．区切られた 4 つの部分をそれぞれ上七，中七，下七，座五と呼ぶ．さらに，それぞれの句は以下のようにより細かく区切られる (それぞれを以下では半句と呼ぶ).

- 上七 ··· 3＋4 モーラ または 4＋4 モーラ
- 中七 ··· 4＋3 モーラ または 2＋5 モーラ
- 下七 ··· 3＋4 モーラ または 4＋4 モーラ
- 座五 ··· 5 モーラ

半句の区切り (上記の"+"の部分) は形態素境界である．また，内容語と機能語が区切りで分かれてはいけない．例えば，上七で「クリス」+「マスイブ」や「東京」+「へ行こう」のように区切ることはできない．

1) 本研究の第 1〜第 3 著者は同等の貢献

図 1 見出しから都々逸

2 手法

2.1 全体の流れ

図 1 に斎藤らの手法のうち，訓練データとなる記事の見出しから生成した都々逸と記事本文のペアの作成までの流れを示す．まず，記事の見出しから，いくつかの半句の欠落を許しつつ，都々逸のモーラ数制約に従う形態素列（穴あき都々逸）を抽出する．次に，穴あき部分の数に基づくスコア付けに従って上位の穴あき都々逸を選ぶ．そして，選んだ穴空き都々逸に対して，穴の部分を T5 を用いた予測ですべて埋める．このようにして，得られた記事と都々逸のペアを訓練データとして T5 の学習を行う．実行時は，入力された記事本文に対して，生成される確率が最も大きい都々逸を選ぶ．

2.2 見出しの自動生成による改良

斎藤らは記事の見出しから穴あき都々逸を抽出していた．これに対し本研究では，穴あき都々逸の候補数を増やすために，自動生成した見出しからも穴あき都々逸を抽出した．具体的には，記事の見出しと本文を用いて T5 を見出し生成にファインチューニングし，各記事から 27 文字以上になるように見

出しを 10 個ずつ生成し，元の見出しを含めて各記事 11 個の見出しから穴あき都々逸を抽出した．生成した見出しを 27 文字以上とした理由は，都々逸は字余りを許すと 26〜28 文字であることと，予備実験において，見出しが 25 文字から 35 文字の間で抽出できた穴あき都々逸の数が多くなったためである．

2.3 穴あき都々逸抽出の改良

本研究では，意味が通る穴あき都々逸を抽出するために，斎藤らの手法に文法的制約と本文からの抽出を加え，改良を行なった．

2.3.1 文法的制約の考慮

形態素を単位とする穴あき都々逸の抽出では,「なかったので」→「たので」,「密猟規制しない」→「密猟しない」のように, 文法的・意味的な誤りを含む抽出結果となる場合がある. そこで, 抽出時に以下の二種類の制約を加えた.

1.「なかっ」＋「た」のように, 片方だけを抽出した場合に文法的な誤りとなるような連続する形態素は結合し,「なかった」という一つの単位として扱う.
2. 一つの単位としてしまうと半句のモーラ数を超える場合は, 代わりに以下の制約を加える.
   - 「規制」＋「しない」のように, どちらか片方だけを抽出した場合に意味的あるいは文法的な誤りとなる場合, 一方を抽出したら他方も同じ句に含めるようにする. そのような例としては他に, ○○＋形容詞の「ない」, 動詞「し」＋助動詞「ない」などがある.
   - 「規制」＋「する」のように,「する」だけ抽出しても意味が通らないものについては, 両方抽出するか,「規制」だけを抽出する. 他の例としては, 内容語＋助動詞「ない」以外の助詞・助動詞や, 動詞＋「こと」などがある.

2.3.2 本文も利用した穴あき都々逸抽出

見出しから単語を抽出し，都々逸を生成する方法ではモーラ数制約のため，使う単語が限られてしまい，都々逸の穴が多くなってしまう．そこで，都々逸に使う単語のバリエーションを増やすため，見出しの単語のみを抽出するのではなく，本文の単語も同様に抽出できるように改良した．手順としては，まず見出しから単語を抽出し，次に見出しから抽出した単語に関連する単語を本文から抽出する．具体的には見出しから抽出した記号，助詞以外の単語に対して，本文中で係り受け関係にある単語がモーラ数の制約を満たす場合に抽出した．

2.4 穴あき都々逸のランキングと絞り込み

齊藤らの手法に比べ，我々の手法では一記事に対して抽出される穴あき都々逸の数が大幅に増える．穴埋めアルゴリズムは自己回帰型の生成モデルに基づくため計算コストが大きく，かつ候補数に比例して実行時間が増える．齊藤らは穴あき都々逸に対し穴の数と種類に従ってペナルティを付与し，それを用いて候補を絞り込むことで穴埋めのための時間を削減していたが，単純に穴の少ない都々逸が選ばれやすく，記事本文の要約として適切でないもの，日本語として不自然なものが選ばれる場合があった．そこで本研究では以下の 2 つの改良を行った．

2.4.1 TF-IDF に基づくランキング

TF-IDF を用いて穴あき都々逸にスコア付けし，絞り込みを行った．ある記事の本文 d 中のある単語 t の出現回数を $\mathrm{tf}(t,d)$，全記事数を N，全記事に対して記号を除いたある単語 t が出現する記事数を $\mathrm{df}(t)$ として,

$$\mathrm{idf}(t) = \log \frac{N}{\mathrm{df}(t)+1} \tag{1}$$

$$\text{tf-idf}(\mathrm{t},\mathrm{d}) = \mathrm{tf}(t,d) \times (\,\mathrm{idf}(t)+1\,) \tag{2}$$

とする．記事 d に対する穴あき都々逸 δ 中の（穴以外の）単語列を $t_1, t_2, ..., t_n$ とするとき δ のスコアを $\sum_{i=0}^{n} \text{tf-idf}(\mathrm{t_i},\mathrm{d})$ と定めた．$\mathrm{df}(t)$ の計算には，毎日新聞の 2,035,436 記事の本文を用いた．

2.4.2 穴埋め後のスコアの予測に基づくランキング

齊藤らは入力の記事に対し穴埋め後の都々逸の候補が見出しとして生成される確率を T5 に基づく見出し生成モデルによって得て，それが最大のものを記事とペアにして訓練データとしていた．本研究では穴あき都々逸を入力として BERT による回帰を用いて，穴埋め後のスコアを予測することで，穴埋め処理の対象とする候補数を絞り込むことにした．

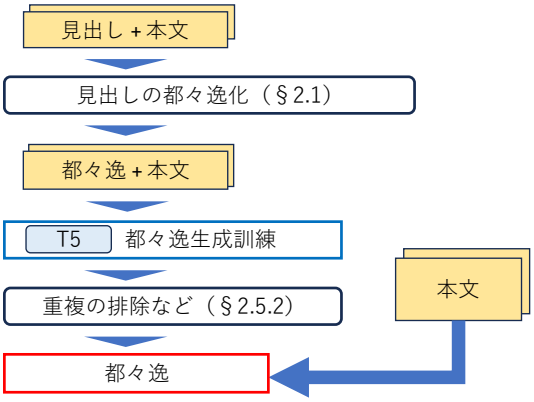

図 2　本文から都々逸

手順としては，まず穴あき都々逸および記事本文を入力とし T5 による穴埋め後の都々逸のスコアを出力とする回帰モデルを BERT を用いて学習した．次に，回帰モデルで穴あき都々逸候補の穴埋め後のスコアを予測し，スコア上位の候補のみを選ぶことで絞り込みを行った．

2.5　都々逸生成アルゴリズムの改良

斎藤らの方法では，入力を記事本文，出力を穴埋めした都々逸として T5 を訓練し，実行時は，Encoder に記事本文を入力し，Decoder で自己回帰的に次のトークンを予測していき，モーラ数制約を満たす生成結果のみを保持しつつビームサーチを行うことによって，最終的な都々逸を生成していた（図 2）．これに対し本研究では，このアルゴリズムの 2 通りの改良を行なった．

2.5.1　重複の排除などのビームサーチの工夫

ビームサーチで生成した都々逸には，同じ単語が複数回出現することで不自然あるいは意味不明になることが多いという問題があった．この問題を解決するために，ビームサーチで保持する生成結果に以下の制約を付け加えた．

1. 漢字，カタカナを含む 3 文字以上の単語の重複を禁止する
2. 文字種に無関係に文字数 2 以上かつモーラ数 3 以上の単語を重複の禁止する
3. 数字が連続することを禁止する

1, 2 の制約は「漁協 漁船 が 海岸 座礁 2 人 救出 伊豆諸島」のように同一の単語を意味なく複数都々逸に含むことを防ぐ．3 の制約は「1994 年」のような連続した数字を含む単語を禁止しない．

2.5.2　句ごとの都々逸生成

斎藤らのビームサーチをすると似た都々逸の候補ばかり残りやすく，句ごとでの意味を考慮していない．これに対し本研究では，生成を句ごとにランダム性を持たせて行なった．まず，入力を記事本文とし，出力を穴埋めされた都々逸全体，あるいは，その都々逸の上七・中七・下七・座五のそれぞれとして T5 を訓練し，5 つのモデルを作成した．実行時は，句ごとに訓練したモデル，記事本文を入力し，ビームサーチによって、各句の候補を複数生成した．その際，助詞や接続詞を句の始めに入れないなどの文法的制約を満たすようにした．そして，都々逸全体を出力として訓練したモデルを用いて，生成した句を結合した結果をスコアリングし，スコアが最大の都々逸を得た．

3　実験

記事の元の見出しを基準として，4 つの手法で生成した都々逸を比較するため，評価に ROUGE を用いて実験を行なった．

3.1　実験設定

3.1.1　使用データ

1994 年から 2018 年の毎日新聞データの記事 2,035,436 件を使用した．見出しが 11 文字以上かつ本文 31 文字以上の記事のみを使い，著作権上の問題で本文が閲覧できない記事や新聞の休み報告をする記事は使用していない．これらのうち，訓練データとして 1995 年から 2003 年の記事からそれぞれ 10,000 記事ずつ合計 90,000 記事，テストデータとして 2004 年の記事から 500 記事を使用した．

3.1.2　比較手法

都々逸生成の 4 つの異なる手法を，テストデータの 500 記事に適用し，生成成功率を調べるとともに，生成した都々逸と元の記事の見出しの一致度を ROUGE で測り評価し比較した．その 4 つの手法を説明する。1 つ目は，訓練データから自動生成した 10 個の見出しに元の見出しを加えた計 11 個から穴あき都々逸を抽出して，TF-IDF によって各記事あたり最大 100 候補まで絞り込んだ後，BERT を用いたランキングでさらに 10 個まで絞り込み，穴埋め処理の後見出し生成モデルを用いたスコアリングの結果が最も良いものをその記事に対する都々逸とする

表 1　ROUGE による評価

尺度	見出しから生成 P/R/F	斎藤ら P/R/F	ビーム＋制約 P/R/F	句ごとに生成 P/R/F
ROUGE_1	26.7/19.3/21.7	24.9/15.6/18.5	23.0/14.7/17.3	19.0/14.8/16.1
ROUGE_2	6.1/ 4.0/ 4.7	5.2/ 3.3/ 3.9	4.0/ 2.6/ 3.0	2.26/ 2.0/ 2.2
ROUGE_S	10.6/ 6.2/ 6.9	9.0/ 4.0/ 4.8	7.5/ 3.6/ 3.9	5.7/ 3.2/ 3.2
ROUGE_SU	14.4/9.2/10.5	12.8/ 6.8/ 8.2	11.3/ 6.1/ 7.3	8.8/ 5.9/ 6.4
生成成功率	99.4	95.2	96.6	100

手法である．２つ目以降の手法では，訓練データ用の 90,000 記事に対して，1 つ目と同じ方法で生成した都々逸を T5 の訓練データとして用いる．２つ目は，都々逸生成の訓練を行なった T5 を用いてビームサーチによって生成する斎藤らの生成アルゴリズムを用いる手法である。３つ目は，斎藤らの生成アルゴリズムに重複の排除などを加えた§ 2.5.1 で述べた手法である（表 1 の「ビーム＋制約」）．4 つ目の手法は，句ごとに生成したあと，それらの組み合わせを都々逸生成モデルによるスコアリングに従って選ぶ§ 2.5.2 で述べた手法である（表 1 の「句ごとに生成」）．

3.2　生成成功率と ROUGE による評価

4 つの手法によって生成した都々逸を ROUGE によって評価結果及び各手法で記事本文から都々逸が得られた割合（生成成功率）を表 1 に示す．生成した都々逸の評価は，全ての手法で生成が成功した記事を対象とした．

4　分析

4.1　穴あきランキング

穴あき都々逸のランキングにおいて，斎藤らの穴あき部分の数に基づくスコア付けと，本研究の TFIDF と BERT の回帰を使ったランキングで，穴埋め後の本文と一致度を測るスコアリングで各記事から最も良いスコアの平均の比較と，本研究の方がスコアが良くなった記事数を示す．図 1 の「スコアリング」をした後に，生成が成功している 5000 記事を使用した．スコアは低い方が良いとして，平均値は斎藤らが 7.05，本研究が 6.89 で本研究の方が少し良い結果を得られた．また，本研究の方がスコアが良い記事は 54.1%（2706 件）であった．

4.2　穴空き都々逸抽出手法の比較評価

4.2.1　文法的制約の追加前後での穴あき都々逸の候補数の変化

1994 年から 2018 年の記事からそれぞれ 1,000 記事ずつ計 25,000 記事を使用して, 文法的制約（§ 2.3.1）による穴あき都々逸の候補数の変化を調べた. その結果, 制約がない場合と比べて一記事あたり平均 46.9% 候補数が減少した. この改良により, 文法的に誤っている候補を排除し， より質の良い候補を抽出することができる.

4.2.2　本文からの抽出による都々逸の穴の変化

1994 年の 50,000 記事に対して，本文も利用した抽出方法 (§ 2.3.2) による都々逸の穴の変化を調べた. 改良前は，抽出段階でひとつも穴を含まない都々逸が 1 記事に対して平均 0.15 個であったのに対して，改良後は，平均 1.74 個となっていた．これは本文抽出により，改良前よりも都々逸の穴が埋まっていることを示している．抽出段階で穴が埋まっていることで，穴埋めのコストを削減することにつながる.

5　終わりに

新聞記事の内容を要約し都々逸を生成する際の, 訓練データと生成アルゴリズムの改良を行なった. その結果, 各記事に対する平均の都々逸生成率と, 本文との一致度両方において, 斎藤らの手法より良い結果を示すことができた. その一方で, 文章としてのわかりやすさという面ではまだ改善の余地がある.