文圧縮を活用したヘッドライン生成

長谷川 駿[1]，平尾 努[2]，奥村 学[1]，永田 昌明[2]
[1] 東京工業大学，[2]NTT コミュニケーション科学基礎研究所
{hasegawa.s, oku}@lr.pi.titech.ac.jp[1]
{hirao.tsutomu, nagata.masaaki}@lab.ntt.co.jp[2]

1 はじめに

ヘッドライン生成とは新聞記事の内容を端的に表す 1 文を生成する研究課題であり，文書要約の一種である．これまで提案されたいくつかの手法 [2，3，4] は，記事そのものを入力とはせず，記事の一文目（以後，本稿では原文と表す）を入力としてヘッドラインを生成する問題として課題を定義している．近年では，原文とそのヘッドラインを大量に用意し，原文からヘッドラインを生成するモデルを sequence-to-sequence モデル [8]（以後，seq-to-seq モデルと表す）を用いて学習する手法 [3，4] が注目を浴びている．本稿ではまず，seq-to-seq モデルを本課題に適用する場合，モデルへの入力文長が一定の長さを越えるとヘッドライン生成の性能が大きく劣化することを示す．そして，この問題は入力となる原文を文圧縮技術によりあらかじめ短くしておくことで解決できることを示す．

2 seq-to-seq モデルによるヘッドライン生成の問題点

本稿では多くのこれまでの手法同様，原文からヘッドラインを生成する際 seq-to-seq モデルを適用するが，実際のモデルには attention 付き encoder-decoder モデル[1][5] を用い，語彙中から単語を順に選んでいくことでヘッドラインを生成する．

まず，いわゆる seq-to-seq モデルにおいてモデルへの入力文長がヘッドライン生成の性能にどう影響するかを調べるため，原文の文長（単語数）ごとの ROUGE-2 スコアを表 1 に示す．4 節で詳述するように，attention 付き encoder-decoder モデルは encoder に片方向の Long short-term memory (LSTM) を，最適化に SGD を，出力時には beam-search を用いた．訓練データには約 218 万ペアの日本語新聞記事の原文とヘッドラインを用いた．

表 1: 原文の文長ごとの ROUGE スコア

文長	ROUGE-2	文数
-9	16.7	1
10-19	35.5	37
20-29	32.6	176
30-39	31.7	328
40-49	28.3	348
50-59	27.4	289
60-69	29.0	186
70-75	27.2	108
全文	29.6	1473

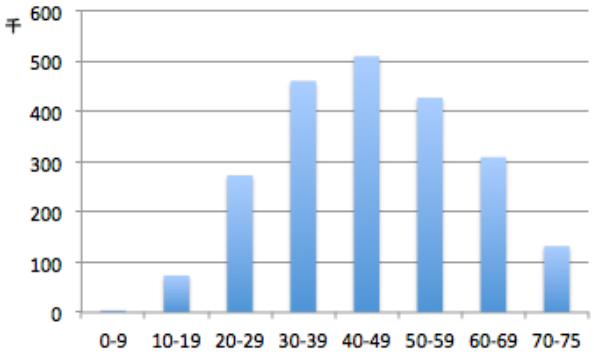

図 1: 訓練データにおける原文の文長の分布

表 1 をみると，原文の文長が 40 単語以上の場合に ROUGE スコアが明らかに低下している．訓練データにおける原文の文長の分布を示した図 1 を見ると，文長が 40 単語以上である原文の割合は約 63%もあり，大部分を占めている．よって，訓練データとテストデータとの分布の違いが原因となって ROUGE スコアが低下したわけではない．そのため原文の文長が 40 単語以上になった場合の ROUGE スコアの低下は，seq-to-seq モデルをヘッドライン生成に適用する場合，モデルへ

[1]Luong ら [5] の論文における global-attention-model を Input-feeding approach で用いている．

の入力文長が長いとヘッドライン生成の性能が劣化するという問題点を示唆している．これを解決するための一案として，原文をあらかじめ短くするということが考えられる．ヘッドラインは原文のすべての情報を含んでいるわけではないため，ヘッドライン生成に必要な情報を残しつつ文長を短くする，つまり不要な単語を削除することができれば，性能を向上させることが可能であると考える．そこで本稿では原文が 40 単語以上である場合を対象として，文圧縮技術によりあらかじめ文長を短くすることで性能の向上を図る．

3 前処理としての文圧縮

ヘッドライン生成に必要な情報を残しつつ原文の文長を短くするため，文圧縮を行う．文圧縮とは任意の一文から，文の大意を捉えた文法的な部分単語列を選択するタスクである．本稿では手始めに，長谷川ら [7] で示されている，大規模データを用いた日本語文圧縮手法を前処理として用い，ヘッドライン生成モデルへの入力文を短くすることを試みる．

3.1 抽出型ヘッドライン

近年，原文と圧縮文のペアを大量に用意し，機械学習を用いて単語削除のモデルを学習することにより高い性能の文圧縮器が実現されている．ただし，正解データである圧縮文を人手で大量に用意することは困難なため，Filippova ら [6] は新聞記事から自動的に文圧縮器の訓練データを構築する手法を提案した．彼女らはヘッドラインを手掛かりにして作成した原文の圧縮文を抽出型ヘッドライン（extractive headline）と呼び，原文と抽出型ヘッドラインのペアを訓練データとして文圧縮器を学習する．

この手法ではヘッドラインと原文で共有されている内容語 [2] を全て含む原文の部分依存木を抽出型ヘッドラインとしている．ただし，本稿では日本語を対象としているため Filippova らの訓練データ自動構築手法 [6] を日本語の特徴に合わせて改良した長谷川らの手法 [7] で訓練データを作成する．また，この手法を適用するためにはある程度ヘッドラインと原文が似ている必要があるため，ヘッドライン中の内容語の 1 割以上を含む原文のみを対象として抽出型ヘッドラインを得た．

重要な点は，抽出型ヘッドラインがヘッドライン生成に必要なほぼ全ての情報を含んだ文法的な圧縮文であることである．よって，本稿ではこの抽出型ヘッドラインを圧縮文の上限とみなす．

3.2 LSTM に基づく文圧縮

Filippova ら [1] が提案した LSTM に基づく文圧縮器を利用して任意の原文に対する圧縮文を得る．Filippova らの手法 [1] では大量の訓練データを用いて LSTM に基づく encoder-decoder モデルを学習し，原文の単語列から原文中の各単語を保持するか削除するかのラベル列を推定することで文圧縮を行う．訓練データには長谷川らの手法 [7] で作成した原文と抽出型ヘッドラインのペアを用いた．また，モデルには attention の付いていない encoder-decoder モデルを使用した．実験の設定として，encoder には片方向 LSTM[3] を，最適化には adam[4] を，出力時には beam-search[5] を用いた．

4 実験

圧縮文の上限の性質を持つ抽出型ヘッドラインと LSTM に基づく文圧縮器を用いて生成した自動圧縮文の 2 種の圧縮文を用いて，生成されるヘッドラインの質の変化を調べる．

そのため，抽出型ヘッドラインからヘッドラインを推定するよう学習したヘッドライン生成モデル（from_exth）を用いて，抽出型ヘッドラインから生成したヘッドライン（upper），自動圧縮文から生成したヘッドライン（auto）を ROUGE で評価する．また，原文からヘッドラインを推定するよう学習したヘッドライン生成モデル（from_sent）を用いて，原文から生成したヘッドライン（base）とも比較する．

ヘッドライン生成器の訓練データには新聞記事 3 社 [6] 合計 35 年分から作成した約 218 万ペアを用いた．訓練時間の短縮のため，75 単語より長い文は除いてある．実験の設定として，encoder には片方向 LSTM[7] を，最適化には SGD[8] を，出力時には beam-search[9] を用いた．ただし，再現率に基づく評価指標である ROUGE は評価対象が長いほど有利となる．そこで，生成されたヘッドラインが長いことにより不当に高い評価値となるこ

[2] 名詞，動詞，形容詞，副詞

[3] LSTM は 3 層で dropout probability として 0.2 を用いている．

[4] 各変数は以下の通り．alpha:0.001，beta1:0.9，beta2:0.999，eps:1e-8，batchsize:180.

[5] beam 幅は 20.

[6] 読売新聞，毎日新聞，産経新聞

[7] LSTM は 2 層で dropout probability として 0.3 を用いている．

[8] 学習率は 1，開発データにおいてロスが下がらなくなった場合に 0.5 をかけていく．

[9] beam 幅は 5.

表 2: 実験結果

評価対象	モデル	R-1	R-2	R-L
base	from_sent	49.3	27.8	27.5
auto	from_exth	46.3	25.1	25.5
upper	from_exth	**55.3**	**33.1**	**30.0**

とを防ぐため文長制限を満たさない beam-search 中の候補は除外した．文長制限には正解ヘッドラインの単語数を用いた．また，テストデータにはあらかじめ訓練データから取り分けておいた無作為に選んだ 1500 文のうち，入力文長が 40 単語以上である 930 文を用いた [10]．評価には ROUGE-1, 2, L（R-1，2，L）を用いる．結果を表 2 に示す．

表 2 をみると upper は base と比べてすべての ROUGE スコアが高い．これは，ヘッドライン生成に不要な単語を削除しモデルへの入力文長を短くすることで性能が向上することを示唆している．一方，auto は base よりも ROUGE スコアが低い．自動圧縮文は抽出型ヘッドラインを再現するよう学習した文圧縮器により生成されているが，文圧縮器の性能は完全なものではない．つまり，圧縮の性能がヘッドライン生成の性能に強く影響することを示している．

5 考察

実験の結果から，前処理として文圧縮器を用いることで生成されるヘッドラインの質を向上可能であることがわかった．しかし，自動で圧縮を行った場合には質を向上させることができなかった．そこで，ヘッドライン生成のための文圧縮器にはどのような性質が求められるのかを考察する．

5.1 自動圧縮文の質

自動圧縮文の質を調べるため，文圧縮器が生成することを目的としている抽出型ヘッドラインを参照文として自動圧縮文を ROUGE で評価し，結果をヘッドライン生成モデルへの入力文長ごとに表 3 に示す．

自動圧縮文は，本研究で対象とするモデルへの入力文長が 40 単語以上の場合で ROUGE-2 が 74.4 でしかなく，ヘッドライン生成に必要な情報が失われていることがわかる．よって原文を文圧縮器で圧縮した際にヘッドライン生成に必要な情報が失われ，その結果として生成されるヘッドラインの質も低下したと考えられる．ただし，この自動圧縮文の生成には seq-to-seq モデルに基づく文圧縮器を用いたため，表 3 のようにモデルへの入力文長が長い場合に圧縮の質が悪くなることも強く影響している．

[10]様々な文への適応力を見るため，文数が多いが定型文に近い訃報を取り除く．具体的には「死去」という単語が含まれる文を除外している．

表 3: 文圧縮：原文の文長ごとの ROUGE スコア

文長	R-2	圧縮率	文数
-9	100.0	55.6	1
10-19	93.9	82.6	37
20-29	84.9	67.7	176
30-39	81.4	59.5	328
40-49	78.8	55.1	348
50-59	72.6	49.7	289
60-69	70.7	45.5	186
70-75	71.4	45.5	108
40-75	74.4	50.4	932
全文	77.7	55.3	1473

5.2 ヘッドライン生成モデルへの入力文の情報量と文長が与える影響

抽出型ヘッドラインを基準として，モデルへの入力文の情報量と文長の変化がヘッドライン生成の性能にどの程度の影響を調べる．

具体的には，抽出型ヘッドラインに無作為に 1 文節追加した文 [11] から from_exth を用いて生成したヘッドライン（upper_add）と，抽出型ヘッドラインから無作為に 1 文節削除した文 [11] から from_exth を用いて生成したヘッドライン（upper_rm）を ROUGE で評価し upper と比較する．upper_add と upper_rm はそれぞれ 1 度ずつ抽出型ヘッドラインの各文から作成している．抽出型ヘッドラインに 1 文節追加した文は必要な情報が失われてしまう影響がなく，抽出型ヘッドラインから 1 文節削除した文は文長が長くなってしまう影響がないため，これらから生成したヘッドラインを upper と比較することでモデルへの入力文の情報量と文長の変化が与える影響を調べることができる．結果を表 4 に示す．

表 4 より，必要な情報が失われてしまうと大きく性能が劣化してしまうことがわかる（upper_rm を参照）．これは圧縮の際にヘッドライン生成に必要な情報が失われてしまうと，生成時にはそれを復元できないことを示唆している．一方，表 4 の upper_add をみると 1

[11]文法性を高めるための制約は抽出型ヘッドラインと同じである．

表 4: 入力文の情報量と文長の影響

モデル	R-1	R-2	R-L
upper	55.3	33.1	30.0
upper_add	53.8	31.1	28.6
upper_rm	48.5	27.1	26.7

表 5: 一文節削除によるヘッドライン生成への効果

評価対象	文長	R-1	R-2	R-L
base	40-75	**49.3**	**27.8**	**27.5**
sent_rm	40-75	48.0	27.2	27.3
base	50-75	**48.7**	27.4	26.9
sent_rm	50-75	48.5	**27.7**	**27.5**

文節分文長が長くなることによる性能劣化は確実にあるものの，情報が足りない upper_rm の場合と比較すると劣化が小さい．つまり，必要な情報をほぼ失わずに少しでも文長を短くすることができればヘッドラインの質の向上が期待できる．よって，ヘッドライン生成の前処理としての文圧縮には，多少圧縮率が低かろうともヘッドライン生成に必要な単語を削除しないことが求められる．

5.3 ルールによる文圧縮

文圧縮器が先に述べた要件を満たすようにするためには、圧縮率を下げるということが考えられる。圧縮率を下げると削除する単語数が減るため，長さはさして短くならないが，重要な単語を削除する危険性は減る．そこで，文節依存木を刈り込むことで圧縮文を生成する手法で，原文から一文節短い圧縮文を作成することを試みる．この際，文節依存木の葉にあたるノードを 1 つ選び削除することでヘッドライン生成に不要な単語を削除し，かつ文長を 1 文節分短くすることが期待できる．削除するノードは，根から一番深いノードのうち最も左に位置しているノードとしている．

この文圧縮法は，原文の 1 文節しか削除しないため圧縮文の長さは原文とさほど変わらない．そこでヘッドライン生成のためのモデルには from_sent を用いる．この圧縮文から生成されたヘッドラインを sent_rm とし ROUGE で評価し base と比較する．結果を表 5 に示す．

表 5 をみると 40 単語以上の文では ROUGE スコアの向上はみられないが，50 単語以上の文であれば ROUGE-2, L においてスコアの向上が確認できる．よって，1 文節分だけでも高精度な単語の削除ができれば性能の向上が可能ということが実際に確認できた．

6 まとめ

本稿では，seq-to-seq モデルをヘッドライン生成に適用する場合，モデルへの入力文長が長いと生成の性能が劣化するという問題点を明らかにした．また，原文に前処理として文圧縮を行いモデルへの入力文長を短くすることで，生成の性能を向上できることを示した．その際，ヘッドライン生成の前処理としての文圧縮は，多少圧縮率が低かろうともヘッドライン生成に必要な単語を削除してはならないということがわかった．今後、ヘッドラインに必要な単語を残しつつ、いかに原文を圧縮するかが大きな課題である

参考文献

[1] K. Filippova, E. Alfonseca, C. A. Colmenares L. Kaiser and O. Vinyals, Sentence Compression by Deletion with LSTMs. In *EMNLP*, pp 360-368, 2015.

[2] A. M. Rush, S. Chopra and J. Weston, A Neural Attention Model for Sentence Summarization. In *ACL*, pp 379-389, 2015.

[3] R. Nallapati, B. Zhou, C. D. Santos, C. Gulcehre and B. Xiang, Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *CoNLL*, pp 280-290, 2016.

[4] S. Chopra, M. Auli and A. M. Rush, Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *NAACL-HLT*, pp 93-98, 2016.

[5] M. T. Luong, H. Pham and C. D. Manning, Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*, pp 1412-1421, 2015.

[6] K. Filippova and Y. Altun, Overcoming the Lack of Parallel Data in Sentence Compression. In *EMNLP*, pp 1481-1491, 2013.

[7] 長谷川 駿，菊池 悠太，高村 大也，奥村 学，大規模データを用いた日本語文圧縮. 言語処理学会第 23 回年次大会, 2017.

[8] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to Sequence Learning with Neural Networks. In *NIPS*, 2014.