書き手ごとの要約スタイルの分析と学習

Dolça Tellols† 狩野 竜示‡ 谷口 友紀‡ 大熊 智子‡
西川 仁† 徳永 健伸†
† 東京工業大学 情報理工学院
tellols.d.aa@m.titech.ac.jp，{hitoshi,take}@c.titech.ac.jp
‡ 富士ゼロックス株式会社
{kano.ryuji, tomoki.taniguchi, ohkuma.tomoko}@fujixerox.co.jp

1 はじめに

自動要約技術は，人が作成した要約との一致度を指標にして評価を行う [1, 2]．しかしながら，要約は作成者によってばらつくことが知られている．Kryscinski ら [3] は，CNN／Daily News の重要文をアノテーションさせたところ，アノテーター間で大きなばらつきがあることを解明した．こうしたばらつきを鑑み，従来の要約評価では，複数人が書いた正解の要約を複数取得することが一般的であった [1]．しかし，これまで書き手による要約スタイルのばらつきは分析されて来なかった．本稿では，書き手による要約スタイルの違いの分析と，書き手ごとの要約スタイルに合わせた要約モデルの構築が可能かの検証を行う．

Reddit[1][2]の TIFU スレッド [4] の投稿とタイトルを使い，三つの要約スタイルを代表する性質（要約の長さ，要約ソースの位置，抽象度）に着目し，書き手ごとの要約スタイルを分析した．分析の結果，同じ書き手の要約では，要約の長さ，抽象度，要約ソースの位置の順に一貫性が高いことが判明した．

更に，書き手ごとの要約スタイルが学習可能かを検証するため，書き手情報の埋め込みベクトルを使用した要約モデルと，使用していない要約モデルを学習させた．実験には，スタイルの分析同様，RedditTIFU データを使用し，同じユーザーの投稿を学習データ，開発データ，テストデータに分散させることで，書き手情報がどう学習されているか検証した．結果，書き手情報を加味したモデルはより高い ROUGE 値を記録したが，出力の性質においては，書き手情報を使用しないモデルよりも正解要約と異なる出力を生成していた．要約のスタイルを表す性質が要約精度の向上に寄与することを示すため，正解要約の性質を入力に加えたモデルを Oracle として学習すると，ROUGE 値が向上することを確認した．この結果は要約スタイルの性質自体は要約の生成に有効であることを示唆している．これらにより，書き手毎の性質を学習するには，書き手情報の埋め込み以外の機構が必要であることが判明したため，今後の課題としたい．

[1] https://www.reddit.com
[2] Reddit は、Reddit Inc. の登録商標です

2 関連研究

要約の性質に着目した先行研究はいくつか存在する．Kikuchi ら [5] は，要約の長さに着目し，Fan ら [6] はドメインのスタイルに合わせた要約を生成するモデルを提案した．Kim ら [4] は，要約対象となる本文の位置がドメインによって異なることを指摘した．Zhang ら [7] は，抽象度（本文に含まれない単語が生成要約に出現する度合い）に着目し，生成型要約モデルが本来生成すべき抽象度の高い要約を出力できていないことを指摘した．今回我々は，先行研究において言及されてきた 3 つの性質，すなわち要約の長さ，要約ソースの本文中の位置，抽象度の書き手ごとの違いを分析する．

要約以外のタスクにおいては、文章の書き手毎のスタイルを分析した関連研究が多く存在している．Preotiuc ら [8] は，単語の長さ，音節の数，単語の珍しさなどの文章のスタイルが，書き手のジェンダー，年齢とソーシャルクラスによって違うことを明らかにした．Author profiling に取り組んだ研究は文章の性質から書き手の性質が予測できる手法を提案している [9, 10]．

要約が書き手によってばらつくことは，多くの先行研究で指摘されてきた．Kryscinski ら [3] は，CNN／Daily News の重要文をアノテーションさせたところ，アノテーター間で大きなばらつきがあることを解明した．Owczarzak ら [11] は，要約の評価者がどの程度一

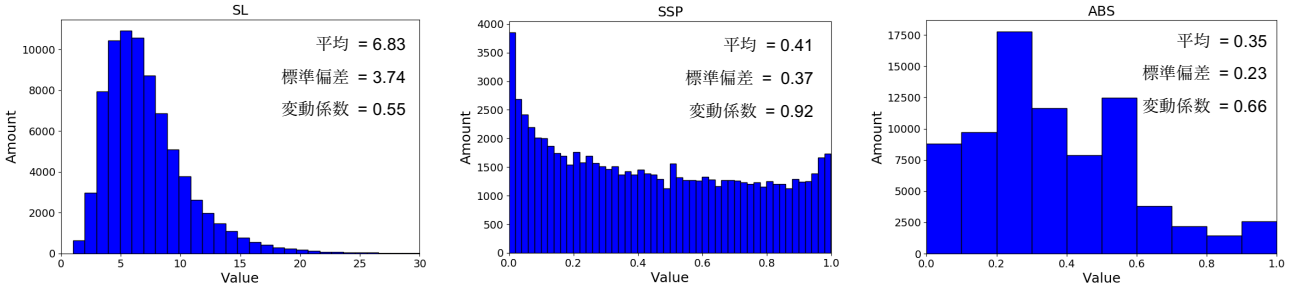

図 1: 性質ごとの分布ヒストグラム（SL，SSP と ABS）.

貫した評価を行うかを調査した．しかし，どのような要約の性質がばらつきやすいのかは分析されていない．

3　分析対象の要約の性質

関連研究で挙げた，要約の代表的な性質として，以下の三つの性質を分析する．

- 要約の長さ (Summary Length, SL) は要約の単語の数を表す．
- 要約ソースの位置 (Summary Source Position, SSP) は本文のどの箇所が要約に使われているかを表す．本文から要約の長さと同じ単語長の部分単語列を取得し，ROUGE-1 の F 値（ROUGE-1-F）を計算する．最も高い ROUGE-1-F の部分単語列が始まる位置を部分単語列の数で割ったものが SSP になる．該当箇所が 2 つ以上の場合は，ランダムに一つ選択する．全ての ROUGE-1-F が 0 である場合，その投稿を分析対象から外した．
- 抽象度 (Abstractedness, ABS) は要約にしか出ていない，すなわち本文に出ていない，ユニークな単語の数を要約のユニークな単語の数で割ったものである．

4　性質の分析

4.1　データセット

性質の分析対象として，Reddit の TIFU データセットを使用する [4]．各投稿にはタイトルがつけられており，これを要約とみなす．本データでは，ほぼ全ての投稿で，書き手のユーザー情報を取得可能であるため，ユーザーごとの分析が可能である．67,828 ユーザーが書いた 78,272 件の投稿とそのタイトルを扱う．このデータセットでは各投稿が予めトークンに分割されているため，本研究ではその分割済トークンを使用する．

4.2　データ分析

投稿ごとの性質の分布を図 1 に載せた．図が表すように，最も分散が大きい性質は SSP であった．ABS と SL の分布は両方とも，低い値に偏っているが，分散は ABS のほうが大きかった．しかし，同じユーザーが異なる性質の投稿を書いている場合，書き手による要約の性質の予測が難しくなると考えられる．

ユーザーごとの要約スタイルが存在するかを確認するため，各性質が同じユーザーの投稿間でどの程度一貫するかを分析する．表 1 に，投稿数ごとにユーザーの書いた要約の性質の平均変動係数を示す．投稿ごとの場合と同様に，SL，ABS，SSP の順に一貫性が高かった．

投稿数	ユーザー数	性質 SL	SSP	ABS
3 - 4	1,419	**0.315**	0.586	0.486
5 - 9	293	0.387	0.606	0.561
10 - 19	33	0.385	0.718	0.603
20 - 29	5	0.340	**0.567**	**0.466**

表 1: 各ユーザーの要約の性質の変動係数を，投稿数ごとのユーザーにまとめ，平均をとったもの．

投稿数を見ると 3-4 か、20-29 であるユーザーの方が一貫している．ユーザーの投稿の性質が一貫していると，各ユーザーの要約のスタイルの学習が容易となる．実際にスタイルの学習が可能であるかを 5 節で議論する．今回はデータセットに量の制限があったため今後の課題としてより多いデータセットで一貫性をまた確認したい．

5　書き手毎の要約スタイルの学習

書き手毎の要約スタイルが学習可能かを検証するため，生成型要約モデルをユーザー埋め込みなし（ベースライン）とありの 2 条件で学習し，それぞれのモデルが生成する要約の性質と ROUGE の F 値を分析した．さらに，正解要約の性質を入力に加えたモデルを学習し，性質自体が要約生成に有用であるか確認した．

5.1　モデル

実験には OpenNMT[12] を使用する．学習したモデルは Attention 付き Bidirectional LSTM Encoder-Decoder である．OpenNMT では，入力に離散的な情報を加えることが出来る．Kobus ら [13] は，Seq2seq モデルにおいて，ある性質の情報を加える際，単語ベクトルに concatenate することが効果的だと示している．そのため，我々もそれに倣い，ユーザーあるいは性質埋め込みベクトルを単語埋め込みベクトルに concatenate し，Seq2seq モデルに逐次的に入力した．

ユーザー名はユーザー埋め込みベクトルに変換した．ただし，4 回以下しか投稿していないユーザーのユーザー名は全て，同一のユーザー名 unknown に置換した．これは，データ数が少ないときにモデルが適切にユーザー埋め込みベクトルを学習できないことを防ぐためである．

正解要約の性質を正確に予測できた場合，結果がどうなるか確認するため，各投稿の性質を離散化した結果を埋め込みベクトルとして使用するモデルを学習した．この時，SL は 47，SSP は 999，ABS は 223 種類の値に離散化した．

5.2　実験設定

4 節で使用したデータを学習，開発，テストデータに分割し，実験に使用する．学習データとして，71,578 投稿を使用した．検証のため，二つの開発／テストデータセットを用意した．一つ目のデータセット（セット 1）では，ユーザー埋め込みベクトルの学習の効果を検証する．5 回以上投稿したユーザーの投稿の内，一つの投稿を開発データセットとし，もう一つをテストデータセットとした．残りの投稿は学習データに含まれているため，モデルは，各ユーザーの 3 以上の投稿からユーザー埋め込みベクトルを学習することとなる．開発データセット，およびテストデータセットはそれぞれ 347 投稿（5 投稿以上書いたユーザーの数と同数）となった．

二つ目のデータセット（セット 2）は，ユーザー埋め込みベクトルありの要約モデルが単に，パラメータの増加により ROUGE の F 値を向上させたことではないことの検証に使用する．投稿数が 4 回以下のユーザーの投稿からランダムに 3,000 投稿ずつ，開発データセットとテストデータセットを抽出した．節 5.2 で記したように，投稿数が 4 回以下のユーザーを unknown として置換してモデルに入力するため，モデルは，補助的なパラメータとして unknown ユーザー埋め込みベクトルを使用することとなる．

上記 2 種類の開発，およびテストデータに含まれないものを学習データとして使用している．ベースラインとユーザー埋め込みありモデルは 100k ステップ学習し，10k ステップごとに開発データで ROUGE-1, ROUGE-2 と ROUGE-L の F 値を確認した．

5.3　結果

表 2 に，テストセットの評価結果を示す．セット 1 ではベースラインに比べてユーザー埋め込みありのモデルの結果の方 ROUGE の F 値が高かった．すなわち，ユーザー情報を学習する事はモデルの精度向上に有用であった．セット 2 では，ベースラインとユーザー埋め込みありモデルには差は見られなかった．この結果は，ユーザー埋め込みベクトルによる精度の向上が，単にパラメータ数が増えたことに起因しないことを示している．

データ	R	BS	ユーザー埋込	性質埋込
	1	0.1863	0.2041	0.23917
セット 1	2	0.0646	0.0815	0.09360
(+5 投稿)	L	0.1804	0.1985	0.22404
	1	0.1880	0.1818	0.23348
セット 2	2	0.0647	0.0636	0.09091
(-4 投稿)	L	0.1821	0.1764	0.21976

表 2: 要約モデルの学習結果．ベースライン（BS、追加埋め込みベクトルなし）、ユーザー埋め込みあり（ユーザー埋込）と性質埋め込みあり（性質埋込）の ROUGE（R）の F 値の違い．

ユーザー埋め込みベクトルは，モデルの要約精度を向上させたが，それがどのような理由によるかは ROUGE 値からは，明らかでない．それがユーザーごとの要約スタイルを学習した結果であるかどうかを検

モデル	性質 SL	SSP	ABS
BS	**2.853**	0.240	**0.324**
ユーザー埋込	3.000	**0.236**	0.341

表 3: 各モデルが出力した要約の性質と，正解要約との差の絶対値の平均．データはセット 1 を使用．

証する．各モデル（ユーザー埋め込みありなし）が出力した要約の性質が実際の要約の性質とどれくらい異なるかを計算する．表 3 に差の絶対値の平均を載せる．結果を見るとユーザー埋め込みありモデル出力の性質は SSP 以外正解要約の性質とより異なっていることが判明した．ユーザー埋め込みベクトルは考慮した性質を学習していなかったと考えられる．セット 1 にて，モデルが生成した要約のユニークな単語の数は，ユーザー埋め込みありのモデルが 551，なしのモデルが 500 であった．すなわち，ユーザー埋め込みベクトルがありの場合，より多様な単語を出力していた．このため，ユーザー埋め込みベクトルは，要約の性質というよりも，語彙を学習していた可能性が高い．一方，正解要約の性質を入力に使用したモデルの ROUGE 値は他 2 つのモデルの ROUGE 値を大きく上回った，これは，求める要約の性質を予測するモデルが出しうる ROUGE 値の上限値を示している．モデルが要約の性質を高精度に予測できる場合．この値に近い結果が得られる事を示唆している．

6 おわりに

本稿では，書き手による要約スタイルの違いを代表する性質を分析した．本研究では，要約の長さ (Summary Length, SL)，要約ソースの位置 (Summary Source Position, SSP)，抽象度 (Abstractedness, ABS) の三つの性質を対象に分析を行った．分析した性質ごとに結果を見ると，SSP と ABS は分散が大きく，SL は小さいことが判明した．

書き手によって性質がどの程度一貫しているかを分析すると三つの性質の内，SL が最も一貫していた．また，一貫性はユーザーの投稿数に依存していることも確認できた．

Seq2seq モデルの実験により，ユーザー埋め込みが ROUGE の F 値を改善することが確認できた．今回は，投稿数が 5 以上であるユーザーに限定してユーザー埋め込みベクトルの学習を行ったが，ユーザー埋め込みベクトルの学習に必要なユーザー毎の投稿数の検証は行わなかったため，今後の課題としたい．

また，ユーザー埋め込みベクトルあり／なしのモデルが出力した要約と正解要約との違いを分析すると，ユーザー埋め込みベクトルは分析した性質を学習していないことが判明した．しかし，性質を要約モデルの入力に取り入れると，ROUGE の F 値が向上したため，性質自体は要約予測に有用であることが判明した．今後の課題として，性質を上手く予測するモデルを組み込むことでモデルの精度向上を計りたい．最終的に要約の性質の予測が，書き手のスタイルに合わせた自動要約生成に役立つと考えている．

参考文献

[1] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out 2004*.

[2] Satanjeev Banerjee et al. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization 2005*. ACL 2005.

[3] Wojciech Kryscinski et al. Neural text summarization: A critical evaluation. In *EMNLP-IJCNLP 2019*.

[4] Byeongchang Kim et al. Abstractive summarization of Reddit posts with multi-level memory networks. In *NAACL 2019*.

[5] Yuta Kikuchi et al. Controlling output length in neural encoder-decoders. In *EMNLP 2016*.

[6] Angela Fan et al. Controllable abstractive summarization. In *WNMT 2018*.

[7] Fangfang Zhang et al. On the abstractiveness of neural document summarization. In *EMNLP 2018*.

[8] Daniel Preotiuc-Pietro et al. Discovering user attribute stylistic differences via paraphrasing. In *AAAI 2016*.

[9] Ilia Markov et al. Language-and subtask-dependent feature selection and classifier parameter tuning for author profiling. In *CLEF (Working Notes)*, 2017.

[10] Roberto López-Santillán et al. Custom document embeddings via the centroids method: Gender classification in an author profiling task. In *CLEF 2018*.

[11] Karolina Owczarzak et al. Assessing the effect of inconsistent assessors on summarization evaluation. In *ACL 2012*.

[12] Guillaume Klein et al. OpenNMT: Open-source toolkit for neural machine translation. In *ACL 2017*.

[13] Catherine Kobus et al. Domain control for neural machine translation. In *RANLP 2017*.