異常検知に基づく文書のスタイル一貫性の改善

京野長彦 [1]　吉永直樹 [2]　佐藤翔悦 [2]
[1] 東京大学大学院 情報理工学系研究科　[2] 東京大学 生産技術研究所
{kyono,ynaga,shoetsu}@tkl.iis.u-tokyo.ac.jp

概要

文書のスタイルは文書全体で一貫していることが望ましいが，意図せず不適切なスタイルの文が混入することも多い．本研究では，与えられたテキストのスタイル一貫性を改善するタスクを提案し，スタイル分離手法と異常検知を用いてこれを解く手法を提案する．具体的には，入力中の各文のスタイルをベクトル表現として分離し，教師なし異常検知手法を用いて異質なスタイルで書かれた文を検出する．その上で，検出された文を，入力中の他の文のスタイルを考慮して変換する．予備実験として，4 種類の既存のスタイル変換データセットを用いて人工的な学習・評価データを構築し，提案手法の評価を行った．

1　はじめに

我々が文章を書く際は，読み手に合わせた適切なスタイルで，文章全体を統一することが望ましい．例えば，目上の人間に送るメールでうっかり乱暴な言葉遣いをしたり，異なる地域に住む人間が理解しない方言を混ぜてしまうと，結果として軋轢や誤解を生じることに繋がる（図 1 左）．また，計算機を用いて文書を処理する際にも，一貫したスタイルの文書から学習したモデルを用いる場合が多いため，この種のスタイルの乱れによって，性能の劣化が生じ得る [1]．

こうした問題を避けるため，広く特定のスタイル（例：話し言葉）から別のスタイル（例：書き言葉）への変換を目的としたテキストのスタイル変換技術が広く研究されている [2, 3, 4]．しかしながら，既存研究においては，入出力のスタイルを既知とした設定で文単位での変換を行う設定が多く，前述したスタイルが混在する文書のスタイル一貫性の改善に直接適用することが困難である．

そこで本研究では，一貫したスタイルの文書（以後，“多数派”）に異質なスタイルの文（以後，“少数派”）が混入した文書を想定した上で，文書中の少数派のスタイルを多数派のものへと変換するタスクを提案する．さらに，そのタスクを実現するための手法として，1) 入力中の各文から異常検知手法を用いて少数派の文を検出した後に，2) 教師なしスタイル変換手法を用いて動的に多数派のスタイルの特徴を検出し，少数派のスタイルを変換する手法を提案する（図 1）．提案手法により，スタイルを変換する対象の文や出力のスタイルを陽に指定することなく，入力テキスト全体のスタイル一貫性を改善することが可能になる．

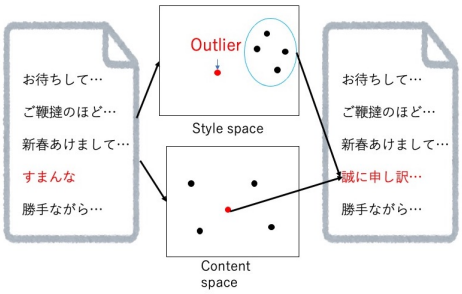

図 1　異常検知に基づく文書スタイル一貫性改善タスク．入力文書中のそれぞれの文に対しスタイル表現を計算し，異常検知によって少数派を検出した後，多数派のスタイル表現を用いてスタイル変換を行う．

本論文では予備実験として，既存のスタイル変換データセットである Yelp [5]，GYAFC [6]，Madaan らによる politeness データセット [7]，Davidson らによる hate speech and offensive language データセット [8] を用いて人工的な学習・評価データを構築し，提案手法のスタイル判定・変換精度に関する分析と考察を行う．

2　関連研究

スタイル変換に関する既存研究では，基本的に文単位でのスタイル変換に取り組んでおり，入出力のスタイルをそれぞれ文集合の形式で与えた上で，教師あり，または教師なしの設定で Encoder-Decoder モデルを学習するものがほとんどである [9, 10, 11]．

このシステムをスタイル一貫性の改善に用いる場合，入力文書の中にスタイルが混在し，変換する必要のある文がごく一部であることと，入出力のスタイルが未知で，変換に適切な文のベクトル表現を取得できないことが問題となる．

本研究と同様に，複数文入力を想定したスタイル変換手法として，Cheng らの研究が存在する [12]．しかしこの手法は，変換対象の文が予め指定されることを前提としており，変換対象の周辺文脈を考慮することに主眼を置いたものである．一方，本研究は，入力の各文で用いられているスタイルのうち，多数派のスタイルを文書が想定するスタイルとみなすことで，既存のスタイル変換手法をスタイル一貫性の改善に利用することを目指す．

また本研究と同様に，複数のスタイルへの対応を前提として学習するモデルとしては，Hu らの研究 [13] や，Kang らの研究 [14] が存在する．しかしこれらの研究では学習データに含まれていたスタイルのみを生成や分類の対象としている．一方，本研究は，複数の入力文から動的に スタイル表現を分離するスタイル変換モデルを学習させることで，スタイルエンコーダにスタイル一般の特徴を学習させ，未知のスタイルに対してもスタイル一貫性の改善を可能とすることを目指す．

3 提案手法

本研究で提案するタスクは，大部分が一貫したスタイルで記述されている文書から，そのスタイルに該当しない文を自動で検出し，スタイルの統一を行うことを目標とする．本研究では入力となる k 文のうち 1 文のみが少数派スタイル，それ以外の $k-1$ 文が多数派スタイルであると設定した．そのため，提案手法は複数の文を入力として受け取り，変換対象の指定および入出力スタイルの明示的な指定無しに変換を可能とする仕組みが必要となる．

3.1 概要

提案手法の全体像を図 2 に示す．主に提案手法で行う処理は 1) スタイル変換の学習, 2) 文単位のスタイル表現の計算, 3) 少数派の文の検出, 4) 少数派のスタイル変換 の 4 ステップである．以下，各ステップについて説明する．

STEP1: スタイル変換モデルの学習　各文にスタイルタグが付与されたデータセットを用いてスタイル変換モデル StyIns [15]（3.2 節）の学習を行う．

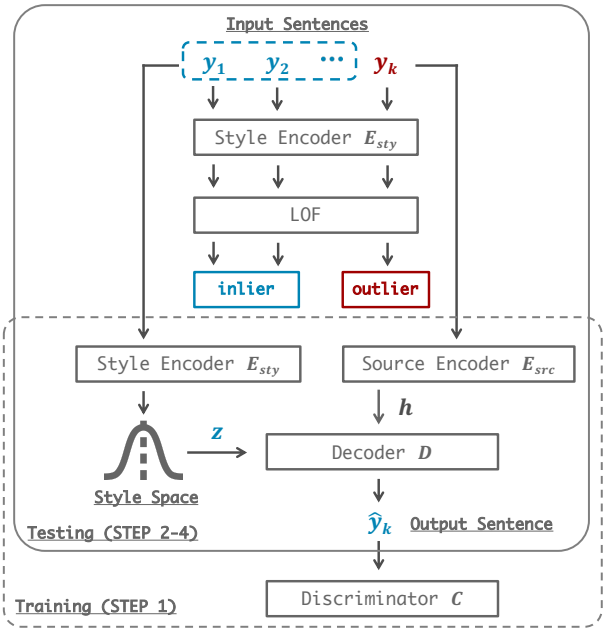

図 2　提案モデルの全体図．訓練データ中の各文にはスタイルタグが付与されていることを前提とし，StyIns の訓練時は LOF を用いずタグから少数派の検知を行う．

STEP2: スタイル表現の計算　STEP1 で学習した StyIns のスタイルエンコーダ E_{sty} を用いてテスト事例の各文からスタイル表現を計算する．

STEP3: 異常検知による少数派の検出　STEP2 で計算したスタイル表現に対して，教師なし異常検知手法である Local Outlier Factor (LOF) [16] を適用し，少数派の検出を行う．

STEP4: 少数派のスタイル変換　STEP3 で少数派と判定された文を StyIns による変換の対象，それ以外の文を多数派のスタイルを示す例としてそれぞれ StyIns に入力し，スタイル変換を行う．

3.2 StyIns

本研究ではスタイル変換モデルとして，敵対的学習に基づくスタイル変換モデル StyIns [15] を採用した（図 2 下部）．以下，StyIns における処理の概要を述べる．ここでは入力となる k 文のうち，目標スタイルを表す $k-1$ 文を $\boldsymbol{y} = \{y_1, y_2, \cdots, y_{k-1}\}$, 変換対象となる 1 文を y_k と表記する．

1. 変換対象 y_k をソースエンコーダ E_{src} に入力し文の内容を表すベクトル h を計算する．
2. 目標スタイルの文集合 $\boldsymbol{y}$ をスタイルエンコーダ E_{src} に入力し，スタイル表現ベクトルが従う正規分布の平均・分散を計算する．
3. h および分布からサンプルしたスタイル表現 z をデコーダ D に入力し，変換後の文 $\hat{y}_k$ を得る．

学習時は損失として，入力 y_k に対する reconstruction loss, $\hat{y}_k$ に対し逆変換を行い y_k との一致度から計算する cycle consistency loss, $\hat{y}_k$ と分類器 C から変換元のスタイルに対する敵対的学習を行う adversarial style loss の 3 つを用いて最適化を行う．

本研究では既存研究における StyIns の実装から若干の変更点が存在する．本来，StyIns のスタイルエンコーダは 2 文以上の入力を前提として分布を出力する仕様であった．しかし 3.1 節の STEP2 では 1 文の入力から単一のベクトルを計算する必要があるため，このエンコーダを拡張し，入力が 1 文のみの際に分布の平均をそのままスタイル表現ベクトルとして出力する実装とした．

4 実験設定

4.1 データ

本研究で提案するタスクの評価には異なるスタイルが混在する文書が必要となるが，人手によるアノテーションを用いてそうしたデータを構築するのはコストが大きい．そのため本研究では，既存のスタイル変換データセットに付与されたスタイルタグを利用し，人工的に異なるスタイルが混在した文書データを作成した上で実験を行う．

また，我々の目標は，多数派・少数派スタイルの性質を問わず入力文書のスタイルを統一することであるため，モデルが特定のデータセットのみに適応した学習をしてしまうことは望ましくない．そのため，複数のスタイル変換データセットを用いて本研究で用いるデータセットを構築する．具体的には，Yelp [5]，GYAFC [6]，politeness [7] を採用する．各データセットはそれぞれ 2 つのスタイルを持ち，データセット中の文はどちらかのスタイルタグが付与されている．具体的には，Yelp には肯定的・否定的なスタイルが，GYAFC には改まった・くだけたスタイルが，politeness には礼儀正しい・無礼なスタイルが存在する．

実験に用いたデータの加工手順は以下の通りである．まず，データ量の偏りを解消するため，各スタイル変換データセットから 200,000 文，4,000 文，1,000 文を訓練・検証・テストデータ構築のための文集合としてランダムサンプルした．各文集合内のスタイルの割合は 50%ずつである．次にサンプルした文集合から，片方のスタイルの文を $k-1$ 件，もう一方のスタイルの文を 1 件ランダムに選択する処理を交互に繰り返し，スタイルが混在する文書を擬似的に作成した．本実験では，$k = 10$ とした．その結果，訓練・検証・テストデータとして 100,000 件，2,000 件，500 件の文書を元となったスタイル変換データセットごとに作成し，訓練・検証の際はこれらを結合した計 300,000 件，6,000 件を用いる．

加えて，訓練および検証データに存在しない未知スタイルに対する性能を確認するため，hate speech and offensive language (以後，offensive) [8] を用いて同様の処理を行い，500 件のテストデータを作成した．

また現実の状況では，既にスタイルが一貫している文書が入力されることも考えられる．そうした入力に対してどの程度誤検出が発生し得るかも検証するため，前述した疑似文書の作成の際に同一スタイルの文のみを k 件選択した，少数派が存在しない例も各データセットにつき 500 件ずつ作成した．

入力文に対する前処理としては，GYAFC, offensive には nltk (v3.6.2) の word_tokenize 関数[1] による単語分割を行い，Yelp, politenes には元のデータセットの時点で行われた単語分割結果をそのまま用いた．

4.2 ハイパーパラメタ

提案手法の実装にあたり，スタイル変換モデルである StyIns は著者の Yi らによる実装を改変して用いた．[2] 少数派の検知のために用いた LOF は scikit-learn (v1.0.1) の実装を用いた．[3]

StyIns に関する Yi らの設定からのハイパーパラメタの変更点としては，バッチサイズを 128 とし，語彙数を 50,000 に制限した．また元はクラス数 $M = 2$ であったが，3 種類のデータセットを学習させるため，$M = 6$ とした．LOF については nearest neighbors の数は $\frac{k}{2}$，contamination の割合は自動判定とした．

4.3 評価

本論文では，提案手法の評価は LOF による少数派の検出（3.1 節，STEP3）に対するものとスタイル変換結果（3.1 節，STEP4）に対するものの 2 段階に分けて行う．その理由としては，STEP3 での検出に失敗した場合，誤検出された例に対するスタイル変換は意味を為さないためである．そのため，STEP4 に対する評価は，正しく少数派を選択出来ていた例に対してのみ行う．

STEP3 の評価について，4.1 節で述べたように

1) https://www.nltk.org/
2) https://github.com/XiaoyuanYi/StyIns
3) https://scikit-learn.org/

表 1　少数派スタイルの検出精度

	少数派あり (%)	少数派なし (%)
Yelp	86.2	0.0
GYAFC	11.8	5.2
politeness	12.0	1.0
offensive	11.3	0.6

表 2　スタイル変換の評価結果

	データ数	BLEU	分類精度 (%)
Yelp	431	51.8	93.0
GYAFC	59	54.5	18.6
politeness	60	54.1	60.0
offensive	56	53.8	23.2

我々は少数派スタイルの文が存在する場合と存在しない場合の 2 種類のテストデータを作成した．各文書について，前者は文書中の少数派スタイルのものを正しく検知出来れば正答とし，後者は検知した少数派の件数が 0 件であれば正答とする．

STEP4 の評価は BLEU スコアとスタイル分類器によって行う．前者については，入力文と出力文の間で nltk.corpus_bleu を用いてスコアを計算し，内容が大きく変化していないかを検証する．後者に関しては，変換後の文がスタイル分類器によって文書中の多数派のスタイルであると分類されれば正答とする．Yi らの実験に倣い，スタイル分類器は事前学習済み BERT [17] を本実験で利用した各スタイル変換データセットに対し fine-tuning したものを用いる．事前実験におけるこの分類器の分類精度は Yelp で 96%，GYAFC で 87%，politeness で 88%，offensive で 96%であった．

5　結果と考察

5.1　少数派スタイルの検出

表 1 に LOF による少数派スタイルの文の検出精度を示す．まず，少数派スタイルを含む例については Yelp のみ 86.2%と非常に高精度であったのに対し，他のデータセットでは 11.3% ~ 12.0%と，少数派の検知の段階で精度が不十分であることが分かった．また，少数派を含まないテストデータについてはどのドメインも非常に精度が悪く，ほとんどの場合に実際には存在しない外れ値を検出してしまう結果となった．この理由としては LOF によって外れ値検出を行う際に入力文数が 10 件という少数であること，LOF に入力したスタイル表現ベクトルの次元の大きさによって，LOF の計算に悪影響が出たためだと考えている．

5.2　スタイル変換

表 2 にスタイル変換の評価結果を示す．4.3 節で述べた理由から，これらは表 1（左）において検出に成功した例のみをスタイル変換した結果である．

まず BLEU スコアに注目すると，どのデータセットについても高水準を保っており，変換によって内容そのものが大きく変わってしまう場合は少ないと考えられる．一方，分類精度については表 1 の結果と同様，データセットによって大きく差が開く結果となった．特に Yelp は LOF による検出・変換ともに非常に高い精度を誇っている．

この理由としては，スタイルの中でも検出・変換が容易なものとそうでないものが存在し，スタイル表現の抽出の過程で前者による影響が大きくなりすぎてしまっているためであると考えられる．例えば，GYAFC データセットに存在したテストケースに，“the best of luck to ya !” という文が存在し，これは本来くだけたスタイル (informal) から改まったスタイル (formal) への変換を行うべき例であった．しかし実際には出力は “the worst of luck to ya !” であり，本来 Yelp で行うべき肯定的なスタイル (positive) から否定的なスタイル (negative) への変換を行ってしまっていた．この問題を解決するため，複数スタイルを同時に扱う際のより効果的な学習方法の考案が今後の課題である．

6　おわりに

本研究では，一貫したスタイルのテキスト集合に異質なスタイルの文が混入した文書を想定した上で，少数派のスタイルを多数派のものへと変換する新たなタスクを提案した．そのタスクを実現する手法として，入力テキスト集合の各文から異常検知手法を用いて少数派のスタイルで書かれた文を検出し，教師なしスタイル変換手法を用いて動的に多数派スタイルの特徴を捉えた上で変換する手法を提案した．今回報告した人工データを用いた予備実験では，一部のスタイルに対してのみ特異的に高い分類精度と変換精度が得られ，各スタイルの特徴の動的な抽出が十分に出来ているとは言い難い結果となった．今後は多数のスタイルをより扱いやすい学習手法およびスタイルの表現手法を考案し，提案タスクにおける精度改善を図る．

謝辞

この研究は国立情報学研究所（NII）CRIS と LINE 株式会社とが推進する NII CRIS 共同研究の助成を受けています。

参考文献

[1] Gina Neff and Peter Nagy. Talking to bots: Symbiotic agency and the case of tay. **International Journal of Communication**, Vol. 10, pp. 4915–4931, 2016.

[2] Eduard Hovy. Generating natural language under pragmatic constraints. **Journal of Pragmatics**, Vol. 11, pp. 689–719, 1987.

[3] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 866–876, July 2018.

[4] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. Fighting offensive language on social media with unsupervised text style transfer. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 189–194, 2018.

[5] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1865–1874, 2018.

[6] Sudha Rao and Joel Tetreault. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 129–140, 2018.

[7] Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 1869–1881, 2020.

[8] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. **Proceedings of the International AAAI Conference on Web and Social Media**, Vol. 11, No. 1, pp. 512–515, 2017.

[9] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence to sequence models. In **Proceedings of the Workshop on Stylistic Variation**, pp. 10–19, 2017.

[10] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In **Proceedings of the 31st International Conference on Neural Information Processing Systems**, pp. 6833–6844, 2017.

[11] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 424–434, 2019.

[12] Yu Cheng, Zhe Gan, Yizhe Zhang, Oussama Elachqar, Dianqi Li, and Jingjing Liu. Contextual text style transfer. In **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 2915–2924, 2020.

[13] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In **Proceedings of the 34th International Conference on Machine Learning**, Vol. 70, pp. 1587–1596, 2017.

[14] Dongyeop Kang and Eduard Hovy. Style is NOT a single variable: Case studies for cross-stylistic language understanding. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 2376–2387, 2021.

[15] Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. Text style transfer via learning style instance supported latent space. In **Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20**, pp. 3801–3807, 2020.

[16] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. **ACM Sigmod Record**, Vol. 29, No. 2, pp. 93–104, 2000.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, 2019.