要約の生成過程を考慮した弱教師あり学習による生成型要約のエラー検出

高塚雅人 [1] 小林 哲則 [1] 林 良彦 [1]
[1] 早稲田大学 理工学術院
takatsuka@pcl.cs.waseda.ac.jp

概要

近年，自動要約において，生成された要約に含まれている事実関係のエラーの検出が大きな課題となっている．既存研究では，人工的に要約のエラーを再現したデータセットを作成し，エラー検出モデルを学習する手法が提案されている．本研究では，文融合と文圧縮モデルを用いて，生成型要約モデルの文生成過程を模倣したデータセットの作成手法と，フレーズレベルのエラー検出と文レベルの原文書の根拠箇所の同定をマルチタスクで学習する SumPhrase モデルを提案する．複数のデータセットを用いた実験結果から，提案手法によるエラー検出と根拠文同定の精度向上を確認した．

1 はじめに

近年，生成型要約モデル [1, 2] の精度が向上し，より流暢で読みやすい文を生成することが可能になったが，生成された要約に事実関係のエラーが含まれており，原文書との整合性が取れていないという問題が指摘されている [3]．また，ROUGE[4] などの要約の評価指標として一般的に用いられている手法では，要約の事実関係の整合性を評価するのは困難であるため [5, 6]，要約の事実関係の整合性の自動評価手法が近年研究されている．

要約の事実関係の整合性の評価の既存手法として，人工的に要約のエラーを再現したデータセットを作成する手法 [7, 8] が提案されている．このアプローチでは，人工的に生成したエラーと生成型要約モデルが実際に生成するエラーの分布が似ていることが重要になるが，既存のエラーの生成手法では，生成型要約モデルが生成するエラーを再現できていないことが指摘されている [9]．そこで本研究では，人工的に生成するエラーを生成型要約モデルが生成するエラーに近づけるために，生成型要約モデルの文生成過程を模倣した人工的なデータセット作成手法を提案する．また，整合性の評価のモデルを実際に用いる上で，要約内のエラー箇所の提示と，原文書内の根拠箇所の提示は有益であると考えられるので [7]，要約のフレーズレベルのエラー検出と原文書の文レベルの根拠箇所の同定をマルチタスク学習として行うことを提案する．

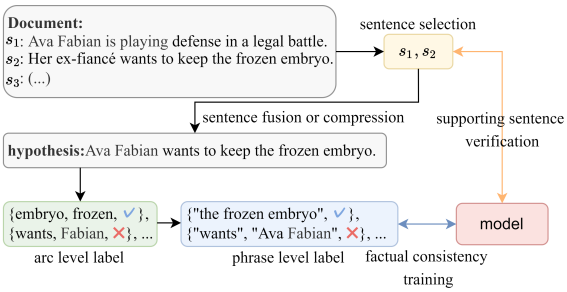

図 1 提案手法の概要図

提案手法の概要を図 1 に示す．要約のデータセットとして一般的に用いられている CNN-DailyMail データセット [10] では，参照要約のうち大部分が単一の文の文圧縮と二文の文融合で生成されている [11]．そこで本研究では，原文書から一文または二文を選択し，一文の場合は文圧縮，二文の場合は文融合モデルに入力して，要約の一部となりうる文 (以下，生成文と呼ぶ) を作成する．その後，生成文に対する弱教師ラベリングを行うことで，人工的なデータセットを構築することを提案する．また，弱教師ラベリングの際は，既存手法 [8] で提案された生成文に対する依存関係アークレベルの事実関係の整合性のラベリングを拡張し，フレーズレベルのラベリングを行う．さらに，原文書から選択した文 (文圧縮，文融合モデルへの入力文) を正解ラベルとして，原文書側の根拠箇所の学習を行う．

2 関連研究

要約の事実関係の整合性の評価の既存手法として，要約生成時のエラーを再現するように人工的に

エラーを生成し，作成した弱教師ラベルを用いてモデルを学習する手法が提案されている．エラーの生成手法には，ルールベースのテキストの変換 [7, 12, 13] や，生成型モデルを用いた手法 [8, 9] などが提案されている．Kryscinski ら [7] は，エンティティの置換や文否定によってエラーを生成した．Goyal ら [8] は，生成型モデルの出力のうち，事後確率が低い文 (beam search の下位の文) は，事後確率が高い文よりも品質が低く，エラーが出やすいと仮定した．そのうえで，事後確率が低い文に新しく出現した依存関係アークに対して，エラーであるとラベリングを行った．

本研究では，生成型モデルを用いた手法 [8, 9] を踏襲して人工的なデータを作成するが，パラフレーズモデルのみを用いていた既存手法 [9] に加えて，文融合，文圧縮モデルを用いて，文を生成し，弱教師ラベリングを行う．また，文または依存関係アーク単位で整合性の評価を行っていた既存研究とは異なり，フレーズレベルの整合性の評価を提案する．要約の整合性の評価と原文書の根拠箇所のマルチタスク学習は既存研究 [7] でも同様に行われていたが，単一のスパンしか学習できない既存手法とは異なり，提案手法では，文レベルのマルチラベル問題として解くことで，原文書中の複数箇所を根拠として学習することができる．

3 提案手法

本章では，3.1 節で人工的に学習データを生成する手法を説明し，3.2 節でフレーズレベルの整合性の学習と根拠文同定のマルチタスク学習を行うモデルを説明する．

3.1 人工的なデータセットの構築

提案手法では，フレーズレベルの要約の整合性の弱教師ラベルと文レベルの原文書の根拠文ラベルを作成する．提案手法では，まず，文選択において，生成型モデル (文融合，文圧縮モデル) への入力となる文を選択する．その後，選択した文を生成型モデルに入力し，生成文を作成する．この時，生成型モデルへの入力文を生成文の根拠箇所とみなし，根拠文ラベルを作成する．また Goyal らの手法 [8] を用いて，生成文に対し，依存関係アークレベルで事実関係の整合性の弱教師のラベルを付与する．既存研究では，このラベルを用いて，アークレベルで整合性の評価を行っていたが，本研究では，より広い文脈を明示的に利用できるフレーズレベルの整合性の評価を行うために，アークレベルのラベルからフレーズレベルのラベルを構成し，学習することを提案する．

文選択

文選択では，n 文からなる原文書 $D = [s_1, s_2, ..., s_n]$ から生成型モデルへの入力文を選択する．具体的には，CNN/DM データセットを基にした文融合のデータセット [14] と文圧縮のデータセット [15] を使用した．各データセットは参照要約文に対して最も ROUGE スコアが高い原文書の文を選択する (文圧縮の場合は一文，文融合の場合は二文) ことで作成されている．また，原文書から選択した文を根拠文として，原文書の各文に根拠文ラベルを振る．

依存関係アークレベルのラベリング

文選択において選択した文を各モデルに入力し，生成文を作成する (文融合モデルと文圧縮モデルの詳しい設定は後述する)．その後，生成文に対して，依存関係アークレベルの整合性のラベリングを行う．まず，CoreNLP[16] を用いて，生成文に対する依存構造木を得る．

文融合モデルの生成文の依存関係アークに対しては，Goyal らの手法 [8] を用いて，ラベリングを行う．Goyal らの手法では，生成型モデルの出力のうち，事後確率が低い文 (beam search の下位の文) は，事後確率が高い文 (beam search の上位の文) よりも事実関係のエラーが出やすく，そのような文に新しく出現した情報は，エラーである可能性が高いと仮定し，inconsistent ラベルを振る．同様に，事後確率が低い文のアークに対して，入力文や参照要約に同一のアークが存在する場合，そのアークは入力文によって含意されるとして consistent ラベルを振る．

文圧縮では，要約の文法性と整合性を維持したまま，文中の重要でない節を取り除くことが可能であり，事実関係のエラーが発生しづらいと考えられる．そこで本研究では，文圧縮モデルの生成文の依存関係アークに対して，入力文や参照要約に同一のアークが存在する場合，そのアークは入力文によって含意されるとして consistent ラベルのみを作成した．

フレーズレベルのラベリング

次に，作成したアークレベルの事実の整合性のラベルからフレーズレベルのラベルを作成する．まず，生成文の依存構造木に基づいて，ルールベースでフレーズを作成する．具体的には，各アークの依

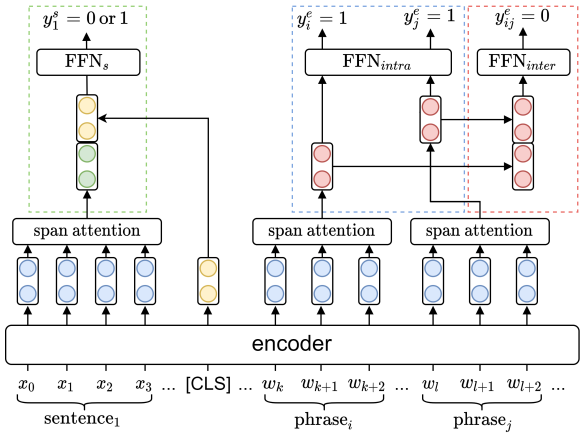

図 2　提案モデルの概要図: 緑枠が根拠文同定，青枠がフレーズ内の事実の整合性，赤枠がフレーズ間の事実の整合性を学習する部分である

存関係ラベルに基づいて，その係り元と係り先の単語をマージするかを決定し，フレーズを作成する．その後，各アークの整合性のラベルに対して，アークの係り元と係り先の単語が同じフレーズに存在する場合はフレーズ内ラベルとして，別のフレーズに存在する場合はフレーズ間ラベルとしてフレーズレベルの整合性のラベルを作成する．フレーズ内/間に複数のアークレベルのラベルが存在する場合，一つでも inconsistent ラベルがある場合はフレーズレベルを inconsistent ラベルとし，全てのアークラベルが consistent ラベルの場合のみ，フレーズレベルを consistent ラベルとした．

追加データ

追加のデータとして，既存研究 [9] で作成されたパラフレーズモデルを利用したデータを用いた．本研究では，他のデータと同様に，アークレベルのラベルをフレーズレベルのラベルに変換して利用した．また，consistent ラベルのフレーズの表現を増強するため，参照要約文の各フレーズに対して，consistent ラベルを振ったデータを追加した．

3.2　提案モデル (SumPhrase)

図 2 に提案する事実関係の整合性の評価モデル (SumPhrase) を示す．モデルは，原文書 D と生成文 h を入力とし，フレーズレベルの事実関係の整合性のラベル y^e と文レベルの根拠文ラベル y^s を学習する．まず，原文書と生成文の間に特殊トークン [CLS] で挟んで結合する．結合した文章を事前学習済みの encoder に入力し，文脈を考慮した各トークンのベクトルと生成文の表現を表す [CLS] トークンの出力を得る．その後，各フレーズの表現と原文書の各文の表現を span attention[17] を用いて計算する．計算したフレーズの表現 h^p を用いて，フレーズ内とフレーズ間の事実の整合性の確率を計算する．フレーズ内の場合は，各フレーズの表現 h_i^p をそのまま全結合層に通して，フレーズ内にエラーがある確率 $p(y_i^e|p_i)$ を計算する．

$$p(y_i^e|p_i) = \text{softmax}(\text{FFN}_{intra}(h_i^p)) \quad (1)$$

フレーズ間の場合は，二つのフレーズの表現 h_i^p, h_j^p を結合して，全結合層に入力し，フレーズ間にエラーがある確率 $p(y_{ij}^e|p_i, p_j)$ を計算する．

$$p(y_{ij}^e|p_i, p_j) = \text{softmax}(\text{FFN}_{inter}([h_i^p; h_j^p])) \quad (2)$$

根拠文の同定は，原文書の各文の表現 h_i^s と [CLS] トークンの表現 h^{cls} と結合して，各文が根拠文である確率 $p(y_i^s|s_i)$ を計算する．

$$p(y_i^s|s_i) = \text{softmax}(\text{FFN}_s([h_i^s; h^{cls}])) \quad (3)$$

最終的に各出力から binary cross-entropy loss を計算し，モデルの目的関数とする．

$$\text{Loss} = \text{Loss}_{intra} + \text{Loss}_{inter} + \alpha * \text{Loss}_s \quad (4)$$

Loss_{intra} がフレーズ内の事実の整合性の目的関数，Loss_{inter} がフレーズ間の事実の整合性の目的関数，Loss_s が根拠文同定の目的関数である．α は，根拠文同定タスクの目的関数の重みを決定するハイパーパラメータである．

4　実験

4.1　実験設定

要約のベンチマークのデータセットとして用いられている CNN/DailyMail データセット [10] を用いて実験を行った．3.1 節の手法を用いて人工的なデータセットを作成し，モデルを学習した．文融合モデルは，Trans-LINKING[18] を，文圧縮モデルは，CUPS[15] を用いた．それぞれ筆者らが公開している事前学習済みモデルを用いた．作成したデータセットの統計情報を AppendixA に示す．

テストデータには，人手でアノテーションされている K2020[7] と Reranking Summary task[5] を用いた．K2020 のテストデータには，ポジティブサンプルが 441 個，ネガティブサンプルが 62 個含まれている．Reranking Summary task のデータセットには，原文書と 2 つの要約文のペアが 373 ペア含まれてい

表 1　要約の整合性評価の実験結果

	K2020		Reranking
モデル	BA	F 値	% Correct
FactCC [7]	72.7	0.706	70.0%
SumFC [12]	80.4	-	78.7%
FactAdv [20]	73.3	0.701	-
Electra-DAE [9]	72.1	-	-
Electra-DAE (ours)	82.7	0.754	85.9%
SumPhrase	**85.3**	**0.765**	**86.0%**
-Multi task	85.2	0.759	84.7%

る．2 つの要約文は似たような表現でありながら、一方はポジティブサンプルで，もう一方はネガティブサンプルとなっている．評価指標として，K2020 には balanced accuracy (BA) と macro F1 を，Reranking Summary task にはモデルがポジティブサンプルを正しく選択できた割合を用いた．

提案モデルである SumPhrase の encoder には，事前学習済みの Electra-base[19] を用いた．

ベースライン

人工的に弱教師ラベルを作成し，要約の事実関係の整合性の評価モデルを学習する手法を比較対象として用いた．FactCC[7]，SumFC[12]，FactAdv[20] では，ルールベースのテキスト変換でネガティブサンプルを作成し，文レベルのラベルで学習を行う．Electra-DAE[9] では，本研究でも追加データとして用いた，パラフレーズモデルを利用したデータセットで学習を行う．Electra-DAE は，文レベルではなく，依存関係アークレベルのラベルで学習を行う．

4.2　実験結果

実験結果を表 1 に示す．表中の Electra-DAE (ours) は，提案したデータセットにおいて，アークレベルのラベルで学習した場合の結果を示す．表 1 から，既存の人工的なデータセットで学習した手法と比較して，提案したデータセットで学習したモデルは，アークレベルとフレーズレベルの両方で精度が向上しており，提案手法の有効性が確認された．文融合や文圧縮時のデータを追加することでより多様な要約のエラーを学習することが可能になったと考えられる．また，アークレベルからフレーズレベルにラベルを拡張することで，Reranking summary task 以外の精度が向上した．このことから，明示的に広い文脈を用いることができるフレーズレベルの整合性の判定の有効性が確認された．さらに，根拠文同定のタスクを追加することで，どちらのデータセットでも精度向上が見られた．

4.3　根拠文同定の精度

補助タスクとして追加した根拠文同定タスクの精度を確認するために，Reranking Summary task のデータを用いた．ベースラインとして，tf-idx を用いて根拠文を判定する SumFC[12] と根拠スパンを予測する FactCCX[7] を用いた．評価指標には balanced accuracy (BA) と macro F1 を用いた．実験結果を表 2 に示す．

表 2　根拠同定の実験結果

	reranking	根拠文同定	
model	%Correct	BA	F 値
FactCCX	61.1%	95.9	0.945
SumFC (tf-idx)	78.7%	97.0	0.970
SumPhrase+multi	**86.0%**	**99.2**	**0.980**

表 2 から，提案モデルがベースラインよりも高い精度で根拠文を同定できていることがわかる．また，根拠文の同定では提案モデルとベースラインの差は小さいが (BA で SumFC と 2.1%の差)，reranking の精度の差が非常に大きくなっている (SumFC と 7.3%の差)．このことから，ベースラインでは，要約文に対応する原文書の文を正しく推定できているのにもかかわらず，要約の事実関係の整合性の評価が正しくできていないことがわかる．

5　まとめ

要約の事実関係の整合性の評価のための人工的なデータセットの構築手法と，フレーズレベルの整合性の学習と原文書の根拠文同定のマルチタスク学習を提案した．文融合，文圧縮モデルを用いて人工的に生成したフレーズレベルのラベルを用いることで，弱教師ラベルを用いた既存手法を上回る精度を達成した．また，提案手法である文レベルの根拠箇所の同定によって，他の既存手法よりも高い精度で根拠箇所を予測することが可能となった．今後の課題として，今回提案した人工的なエラーの作成手法を，対照学習を用いた要約生成手法 [21] に適応し，より原文書に忠実な要約生成を行うことなどが考えられる．

参考文献

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy,

Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th ACL**, 2020.

[2] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In **International Conference on Machine Learning**. PMLR, 2020.

[3] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In **Proceedings of the AAAI**, 2018.

[4] Chin-Yew Lin. Rouge: A package for auto- matic evaluation of summaries. In **Text summarization branches out: ACL workshop**, 2004.

[5] Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In **Proceedings of the 57th ACL**, July 2019.

[6] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In **Proceedings of the EMNLP-IJCNLP**, 2019.

[7] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In **Proceedings of the EMNLP**, 2020.

[8] Tanya Goyal and Greg Durrett. Evaluating factuality in generation with dependency-level entailment. In **Findings of the EMNLP**, 2020.

[9] Tanya Goyal and Greg Durrett. Annotating and modeling fine-grained factuality in summarization. In **Proceedings of the NAACL**, 2021.

[10] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In **Proceedings of The 20th SIGNLL**, 2016.

[11] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. Scoring sentence singletons and pairs for abstractive summarization. In **Proceedings of the 57th ACL**, 2019.

[12] Sen Zhang, Jianwei Niu, and Chuyuan Wei. Fine-grained factual consistency assessment for abstractive summarization models. In **Proceedings of the EMNLP**, 2021.

[13] Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. Factual error correction for abstractive summarization models. In **Proceedings of the EMNLP**, 2020.

[14] Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Lidan Wang, Walter Chang, and Fei Liu. Understanding points of correspondence between sentences for abstractive summarization. In **Proceedings of the ACL: Student Research Workshop**, 2020.

[15] Shrey Desai, Jiacheng Xu, and Greg Durrett. Compressive summarization with plausibility and salience modeling. In **Proceedings of the EMNLP**, 2020.

[16] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In **Proceedings of 52nd ACL: System Demonstrations**, 2014.

[17] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In **Proceedings of the EMNLP**, 2017.

[18] Logan Lebanoff, Franck Dernoncourt, Doo Soon Kim, Lidan Wang, Walter Chang, and Fei Liu. Learning to fuse sentences with transformers for summarization. In **Proceedings of the EMNLP**, 2020.

[19] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In **ICLR**, 2020.

[20] Zhiyuan Zeng, Jiaze Chen, Weiran Xu, and Lei Li. Gradient-based adversarial factual consistency evaluation for abstractive summarization. In **Proceedings of the EMNLP**, 2021.

[21] Shuyang Cao and Lu Wang. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In **Proceedings of the EMNLP**, 2021.

A　データセットの統計情報

作成したデータセットの統計情報を表 3 に示す．表中の fusion は文融合，comp は文圧縮，para はパラフレーズ，ref は参照要約のデータセットを示す．このデータセットにはフレーズレベルの consistent ラベルが 2,021,592 個，inconsistent ラベルが 191,553 個含まれている．また根拠文同定のラベルは根拠文ラベルが 186,028 個，それ以外が 3,389,275 個含まれている．

表 3　作成したデータセット

データの type	モデルの入力	データ数
para[9]	参照要約文	46,925
fusion	原文書の二文	72,093
comp	原文書の一文	47,296
ref	-	107,278

B　詳細分析

各データセットの貢献度の検証

作成した各データセットがどの程度精度に貢献しているかを確認するために，文融合のデータセットをベースラインとして，表 3 の各データセットを追加していき，精度の変化を調べた．モデルには，提案モデルである SumPhrase を用いて，マルチタスク学習は行わなかった．実験結果を表 4 に示す．

表 4 から，文融合のみのデータを用いた場合でも，K2020 の baranced accuracy と reranking summary task において，既存の人工的なデータセットで学習した手法を上回る精度を達成している．また，文圧縮，パラフレーズ，参照要約のデータを追加することで，参照要約を追加したときの reranking summary task 以外の全ての評価指標が向上しており，提案手法の有効性が確認された．さらに，参照要約のデータを追加した際に，K2020 における balanced accuracy と F 値が大きく向上した．このことから，consistent ラベルのフレーズの表現を拡充することも重要であることが分かる．

表 4　各データセットの精度への影響

	K2020		Reranking
data type	BA	F 値	%Correct
fusion	81.1	0.663	83.4%
fusion+comp	82.5	0.687	84.3%
fusion+comp+para	83.7	0.700	**85.1%**
fusion+comp+para+ref	**85.2**	**0.759**	84.7%

span attention の影響

span attention の影響を調査するために，span attention の代わりに各トークンのベクトルの平均値をフレーズレベルの表現として用いた場合の結果を示す．実験結果を表 5 に示す．表 5 から，span attention を用いることによって，フレーズ内の意味的に重要な語を重視することができるようになり，各トークンの平均を用いるよりも，balanced accuracy，F 値共に向上した．

表 5　span attention の影響

	K2020	
model	BA	F 値
SumPhrase	**85.2**	**0.759**
SumPhrase (average)	83.8	0.746