自然言語推論と再現器を用いた Split and Rephrase における生成文の品質向上

塚越 駿 [1]　平尾 努 [2]　森下 睦 [2]　帖佐 克己 [2]
笹野 遼平 [1]　武田 浩一 [1]
[1] 名古屋大学大学院情報学研究科 [2]NTT コミュニケーション科学基礎研究所
tsukagoshi.hayato.r2@s.mail.nagoya-u.ac.jp
{tsutomu.hirao.kp,makoto.morishita.gr,katsuki.chousa.bg}@hco.ntt.co.jp
{sasano,takedasu}@i.nagoya-u.ac.jp

概要

複雑な文を同じ意味の単純な複数の文に分割する Split and Rephrase タスクは，可読性の向上や機械的なテキスト処理の性能向上に有用である．本研究では，Split and Rephrase の性能向上のため，訓練データセットに含まれている文分割前後の文意が一致しない事例を含意関係分類を用いて除去するフィルタリングと，モデルが入力文に対し忠実な文生成を行うよう出力から入力を再構成する再現器を用いた訓練手法を提案する．標準的なベンチマークデータセットを用いた実験の結果，提案手法が世界最高性能を達成したことを確認した．

1　はじめに

長く複雑な文を短く簡単な文に変換できると，可読性の向上や機械的なテキスト処理の性能向上が期待できる [1, 2, 3] ことから，テキスト平易化（text simplification）は人間とテキスト処理システムの双方にとって有用である．最近になり，テキスト平易化タスクの一種として，長く複雑な文を意味内容を保ったままより短く単純な複数の文に変換する Split and Rephrase[4] タスクが提案された．Split and Rephrase は他のテキスト平易化タスクと異なり，複雑な文の語彙や文意を可能な限り変更せず，単純な文に分割することに焦点を当てた文分割タスクである．Split and Rephrase では深層学習モデルを用いた手法 [4, 5, 6, 7] が高い性能を達成している．同タスクの学習や評価に利用される既存のデータセットは（1）自動構築された低品質だが大規模なデータセット [4, 5, 6, 8, 7]（2）人手で作成された高品質だが小規模なデータセット [9, 10] の二つに分類することができる．自動構築されたデータセットには複雑な文と機械的に生成された単純な文のペアが数十万事例含まれ，深層学習モデルの訓練には十分な量のデータが存在する一方，人間によって作成されたデータセットの事例数はいずれも 500 に満たない．したがって，多くの先行研究が自動構築されたデータセットを用いてモデルを訓練し，人手で作成されたデータセットで評価を行なっている．しかし，自動構築されたデータセットには文分割前後の文意が一致しない等の不適当な事例が多数含まれており，加えて，深層学習モデルには入力と整合しない出力をすることがあるという一般的な問題が知られている [11]．そのため，既存の Split and Rephrase モデルは入力に対して意味的に異なる文を出力してしまうことがあり，Split and Rephrase タスクの性能に悪影響を与えている．

この問題に対処するため，本研究では，Split and Rephrase タスクの性質に着目し，自然言語推論（NLI）モデルによるデータセットのフィルタリングと，出力から入力を再構成するモジュールである再現器 (Reconstructor)[12] を導入した訓練手法を提案する．まず，文分割前後の文意が一致しない事例を除くため，事前訓練済み NLI 分類モデルを用いて複雑な文と単純な文のペアを分類し，含意関係にある文ペアのみを残す．さらに，入力に対して意味的に異なる文をモデルが出力することを防ぐため，モデルの出力ベクトルから入力文を再構成する再現器をモデルに導入して訓練を行う．

標準的ベンチマークデータセットに対して，NLI 分類によるフィルタリングと再現器を用いた訓練の効果を調べたところ，それぞれ単独でも性能向上が得られたが，同時に利用することでさらなる性能向

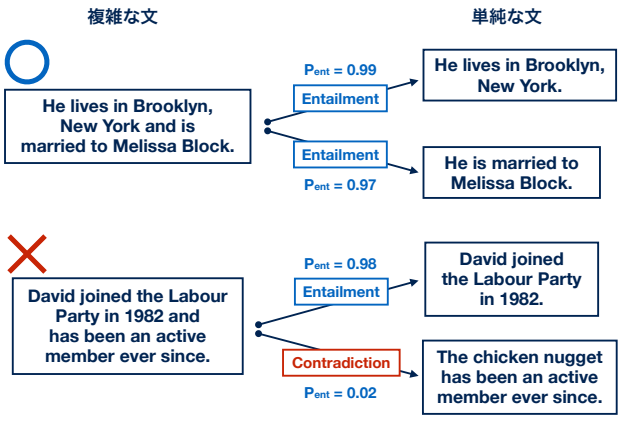

図 1　NLI 分類によるフィルタリングの概要図．図中の例文は WikiSplit の実際の事例．

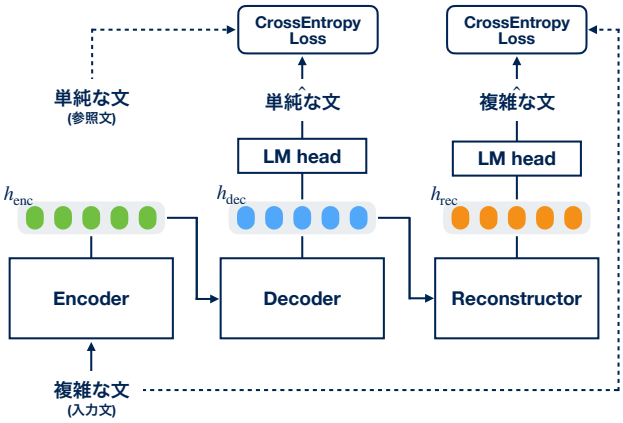

図 2　再現器を用いた訓練時の概要図．

上が得られ，世界最高性能を達成した．

2　関連研究

Split and Rephrase は Narayan ら [4] によって提案されたテキスト平易化タスクの一種である．Narayan ら [4] は既存のテキスト平易化データセットで文分割モデルを訓練することは困難であることを示し，Split and Rephrase 用データセットである WebSplit の提案とベースラインの設定を行った．Aharoni ら [13] は WebSplit のデータ分割の不備を指摘し，適切に分割したデータセットとコピー機構 [14] を用いたモデルを提案して新たなベースラインを設定した．また，Botha ら [5] は WebSplit の語彙や統語構造が多様性に乏しいことを指摘し，Wikipedia の編集履歴を用いたデータセットである WikiSplit を提案した．Niklaus ら [6] はルールベース文分割システムの DisSim[15] を WikiSplit に適用したデータセットである MinWikiSplit を提案した．Kim ら [7] は対訳コーパス中の翻訳前後の文数が 1 文と 2 文になっている事例を抽出して翻訳することで，1 文と 2 文が対応づいたデータセットである BiSECT を提案した．

自動構築された大規模なデータセットが整備される一方で，人手で作成された高品質な評価用データセットも整備されている．Sulem ら [9] は，Wikipedia の文を用いて，人手により作成された高品質な分割後の文を含むデータセットである HSplit を構築した．Zhang ら [10] らは人手により Wikipedia や契約書などの文をもとに，より多様な語彙と統語構造を含み，HSplit よりも難しい評価用データセットである Wiki-BM と Cont-BM を構築した．それらを用いた評価を通して既存手法では十分な性能を達成することは難しいことを示した．

3　提案手法

提案法の概要図を図 1 と図 2 に示す．既存データセットの各文に対して NLI 分類を行い，分割前後の文が含意関係にない，すなわち分割前後で文意が変化してしまっていると分類された文ペアを除去する．さらに，深層学習モデルで問題となる入力と整合しない出力を防ぐため，出力のベクトル表現から入力文を再構成する再現器をモデルに導入する．

3.1　NLI 分類によるフィルタリング

自然言語推論（Natural Language Inference: NLI）タスクは，前提文と仮説文が与えられた時に，前提文と仮説文の関係が含意（entailment），矛盾（contradiction），中立（neutral）のいずれであるかを分類するタスクである．NLI データセットとして代表的なものに Stanford NLI（SNLI）データセット [16] や Multi-Genre NLI（MNLI）データセットがあり，BERT[17] や RoBERTa[18] 等の事前訓練済み言語モデルが高い性能を達成している．

本研究で提案する NLI 分類によるフィルタリングでは，図 1 に示すように，複雑な文 c とそれに対応する単純な文の系列 $s_1, ..., s_n$ が与えられた時に，複雑な文 c と各単純な文 $s_i (1 \leq i \leq n)$ を NLI 分類モデルに入力する．この時得られる複雑な文 c と単純な文 s_i が含意関係にある確率 P_{ent} が，矛盾や中立と分類される確率よりも高かった場合に c と s_i を含意と分類する．そして，複雑な文 c とすべての s_i とが含意関係にあると分類された事例のみを残す．

松丸ら [19] が提案した見出し生成のデータフィルタ法では，記事と見出しの含意関係分類のために事前訓練済み NLI 分類モデルをさらに fine-tuning した

が，本研究では，入力となる複雑な文とそれに対応する各単純な文がそれぞれ単一の文であることから，既存の NLI データセットで学習したモデルで十分適切に含意関係を分類できると考えられるため，事前訓練済み NLI 分類モデルをそのまま用いる．

3.2 再現器を用いた訓練

Tu ら [12] によって提案された再現器は，エンコーダ・デコーダモデルによる機械翻訳システムにおいて，デコーダの出力ベクトルから翻訳元の入力文を再構成する再現器を追加することで，出力ベクトルが入力文の情報をできるだけ含むようモデルに強制し，翻訳の妥当性を向上させた．Split and Rephrase は単言語の翻訳タスクとして考えることができることから，再現器を用いることで分割文の妥当性を向上させることが可能であると考えた．

モデルの訓練時は，複雑な文を入力した時のエンコーダとデコーダの出力ベクトルをそれぞれ $\mathbf{h}_{\text{enc}}, \mathbf{h}_{\text{dec}}$ とすると，まず $\mathbf{h}_{\text{dec}}$ から単語の出現確率を計算し，正解の単純な文に対する交差エントロピー誤差を計算する．再現器を用いる場合はこれに加えて，デコーダの出力ベクトル $\mathbf{h}_{\text{dec}}$ を再現器に入力して得られた出力ベクトル $\mathbf{h}_{\text{rec}}$ から単語の出現確率を計算し，エンコーダへの入力である複雑な文に対する交差エントロピー誤差を計算し，これらを同時に用いることでパラメータの更新を行う．本研究では，Tu ら [12] と同様，二つの損失の重み付き和をとることで，マルチタスク学習として全体を訓練する．具体的には，学習に用いるデータの事例数を N，$n(\leq N)$ 番目の入力，出力，対応するデコーダの出力をそれぞれ $\mathbf{x}^n, \mathbf{y}^n, \mathbf{h}^n_{\text{dec}}$，エンコーダ・デコーダと再現器のパラメータを θ, γ，再現器の損失の重みを α としたとき，目的関数は以下のようになる．

$$J(\theta, \gamma) = \sum_{n=1}^{N} \left\{ \log P(\mathbf{y}^n | \mathbf{x}^n; \theta) + \alpha \log P(\mathbf{x}^n | \mathbf{h}^n_{\text{dec}}; \gamma) \right\}$$

4 評価実験

提案手法の有用性を検証するため，標準的なベンチマークデータセットを用いて実験を行った．

4.1 実験設定

モデルの訓練には，人間の編集履歴をもとに作成されている WikiSplit[5] を利用した．これは，単純な文が機械的に作成されている MinWikiSplit[6] や BiSECT[7] よりも，流暢な文が多いと考えたためである．フィルタリングに用いる事前訓練済み NLI 分類モデルとして，MNLI で訓練された RoBERTa を用いた[1)]．WikiSplit に対して提案法によるフィルタリングを行ったところ，全体の事例数が 999,944 から 652,548 となり，データセット全体の約 34.7%にあたる 347,396 事例が除去された．参考までに，実際に除去された事例を付録 A に示す．その後，データセットを 8:1:1 の割合で訓練/開発/テストセットに分割した．フィルタリング前後のデータセットの事例数を表 1 に示す．

表 1 実験に用いたデータセットの事例数

フィルタリング	前	後
全体	999,944	652,548
訓練セット	799,955	522,038
開発セット	99,994	65,255
テストセット	99,995	65,255

評価用データセットとして HSplit と Wiki-BM を用いた．HSplit と Wiki-BM は同様に Wikipedia の文を用いて人手評価により構築されたデータセットであるが，Wiki-BM の方が多様な文を含んでおり比較的難しいベンチマークであるといえる．提案手法の有用性を検証するため，NLI 分類によるデータセットのフィルタリングの実施の是非，及び訓練時の再現器の有無によって，4 通りの設定で実験と評価を行った．モデルにはエンコーダ・デコーダアーキテクチャの事前訓練済み言語モデルである T5[20] を用い，これを Split and Rephrase タスクで fine-tuning した．比較対象として，原文をそのまま出力するシステム，ルールベースシステムである DisSim[15]，および，Kim ら [7] の事前訓練済みモデルを用いた．実験の詳細は付録 B を参照されたい．

従来研究に従い，評価指標として BLEU[21]，BERTScore[22]，Flesch-Kincaid Grade Level（FKGL）[23] を用いた．また，生成文の品質評価において，BERTScore よりも人間評価との相関が高いと報告されている BLEURT[24] も用いた．評価実験に用いた評価指標のうち，BLEU，BERTScore，BLEURT は正解の文に対する生成文の表層的あるいは意味的な品質を測る指標であり，高い方が良い．一方で，FKGL はシステムの生成文のみを用いて可読性を測る指標であり，小さい方が良い．既存の評価指標に加えて，本研究でフィルタリングに用いた NLI 分類は，システムの出力文の意味的妥当性を評価するた

1） https://huggingface.co/roberta-large-mnli

表 2　評価実験の結果．“WikiSplit-NLI” は NLI 分類によりフィルタリングした後の WikiSplit を表す．

システム	データセット	BLEU↑	BERTScore↑	BLEURT↑	含意割合↑	FKGL↓	分割数
HSplit[9]							
原文出力	N/A	88.04	99.23	88.51	100.00	12.68	1.00
DisSim[15]	N/A	62.95	96.69	77.82	94.15	**7.68**	2.98
Kim ら [7]	BiSECT+WikiSplit	85.29	98.44	84.52	95.54	8.45	2.00
T5-small	WikiSplit	85.06	98.69	86.08	95.82	8.99	1.88
T5-small	WikiSplit-NLI	86.46	98.74	86.52	97.49	8.90	1.89
T5-small +再現器	WikiSplit	87.34	98.82	86.43	96.66	9.18	1.77
T5-small +再現器	WikiSplit-NLI	**87.88**	**98.87**	**86.70**	**97.77**	9.01	1.82
参照文	N/A	100.00	100.00	97.99	98.33	8.65	1.94
Wiki-BM[10]							
原文出力	N/A	69.96	97.90	83.07	100.00	14.75	1.03
DisSim[15]	N/A	55.28	95.76	74.10	90.57	**7.22**	4.19
Kim ら [7]	BiSECT+WikiSplit	76.72	98.31	85.18	98.76	9.52	2.02
T5-small	WikiSplit	77.95	98.39	85.53	98.76	9.58	2.02
T5-small	WikiSplit-NLI	78.15	**98.40**	**85.54**	**99.50**	9.57	2.02
T5-small +再現器	WikiSplit	**78.16**	98.39	85.47	98.75	9.59	2.01
T5-small +再現器	WikiSplit-NLI	78.03	98.39	85.48	99.26	9.58	2.01
参照文	N/A	100.00	100.00	97.74	98.59	7.79	3.09

めにも有用であると考えられることから，入力文に対してシステムが出力した各文がすべて含意と分類される割合も評価指標として用いた．この含意と分類される割合を含意割合と呼称する．さらに，システムが出力した単純な文の数（分割数）も計算した．

4.2　実験結果

実験の結果を表 2 に示す．HSplit ではフィルタリングによって，BLEU，BERTScore，BLEURT，含意割合のすべてが向上した．また，再現器を導入することでも BLEU，BERTScore，BLEURT，含意割合のすべてが向上した．さらに，双方を用いた場合には，BLEU，BERTScore，BLEURT，含意割合において最も良い性能を達成し，現在の世界最高性能を達成したシステムである Kim らの手法を上回った．一方，FKGL および文の分割数について注目すると，再現器を用いることでスコアが劣化している．これは再現器がモデルの出力に対して入力の情報を保持するように強制するため，分割数の少ない文が生成される割合が増加してしまったためだと考えられる．FKGL の評価値は文の分割数に大きく影響を受けるため，FKGL の値が悪化した要因も分割数が減少したことによるものと考えられる．

Wiki-BM でもフィルタリングによって BLEU と含意割合が向上することを確認でき，再現器を用いることで BLEU 値の改善も確認できた．一方，フィルタリングと再現器の双方を用いた場合の改善は HSplit ほど顕著ではなかった．ただし，HSplit の場合と同様に Kim らの手法を上回った．各モデルの出力の分割数を見ると，正解の平均分割数が 3.09 であるのに対し，提案手法はすべて 2.01 程度になっており，これは訓練に WikiSplit を用いている Kim らの手法も同様である．WikiSplit 中に含まれる一つの複雑な文あたりの単純な文の数の平均は 2.03 程度であることから，訓練に用いたデータセット中の分割数にモデルが影響を受けていると考えられる．

HSplit での実験結果から，NLI 分類によるフィルタリングと再現器を共に用いることでシステムの性能が向上し，既存のシステムよりも性能が上回ることを確認した．一方で，Wiki-BM のように訓練データとは傾向が異なるデータで評価を行う場合には，既存システムからの改善幅が小さいので，改善の余地が残されていると考えられる．

5　おわりに

本研究では，Split and Rephrase タスクにおいて，モデルが入力に対して意味的に異なる文を出力してしまう問題に対して，NLI 分類モデルを用いたフィルタリング手法と，再現器を用いて入力に対し意味的に異なる文をモデルが出力することを防ぐ手法を提案した．評価実験の結果，訓練データと似た傾向の HSplit において，提案法は一貫してシステムの性能を向上させ，既存手法を上回る性能を達成した．特にデータセットのフィルタリングと再現器を共に用いた場合に最も高い性能を達成し，その有効性が示された．一方で，訓練データと傾向が異なるは Wkiki-BM では性能の改善幅が小さくシステムに改善の余地が残されていることが示唆された．

参考文献

[1] Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. Overcoming the Curse of Sentence Length for Neural Machine Translation using Automatic Segmentation. In **Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation**, pp. 78–85, 2014.

[2] Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In **Proceedings of the First Workshop on Neural Machine Translation (WMT)**, pp. 28–39. Association for Computational Linguistics, 2017.

[3] R. Chandrasekar, Christine Doran, and B. Srinivas. Motivations and Methods for Text Simplification. In **The 16th International Conference on Computational Linguistics (COLING)**, 1996.

[4] Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. Split and Rephrase. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 606–616, 2017.

[5] Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. Learning To Split and Rephrase From Wikipedia Edit History. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 732–737, 2018.

[6] Christina Niklaus, André Freitas, and Siegfried Handschuh. MinWikiSplit: A Sentence Splitting Corpus with Minimal Propositions. In **Proceedings of the 12th International Conference on Natural Language Generation (INLG)**, pp. 118–123, 2019.

[7] Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. BiSECT: Learning to Split and Rephrase Sentences with Bitexts. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6193–6209, 2021.

[8] Yinuo Guo, Tao Ge, and Furu Wei. Fact-aware Sentence Split and Rephrase with Permutation Invariant Training. 2020.

[9] Elior Sulem, Omri Abend, and Ari Rappoport. BLEU is Not Suitable for the Evaluation of Text Simplification. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 738–744, 2018.

[10] Li Zhang, Huaiyu Zhu, Siddhartha Brahma, and Yunyao Li. Small but Mighty: New Benchmarks for Split and Rephrase. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1198–1205, 2020.

[11] Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in Neural Machine Translation. 2018.

[12] Zhaopeng Tu, Yang P. Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural Machine Translation with Reconstruction. In **Proceedings of the 31th AAAI Conference on Artificial Intelligence**, 2017.

[13] Roee Aharoni and Yoav Goldberg. Split and Rephrase: Better Evaluation and Stronger Baselines. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 719–724, 2018.

[14] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 1631–1640, 2016.

[15] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. Transforming Complex Sentences into a Semantic Hierarchy. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 3415–3427, 2019.

[16] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 632–642, 2015.

[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 (NAACL)**, pp. 4171–4186, 2019.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **arXiv:1907.11692**, 2019.

[19] Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. Improving Truthfulness of Headline Generation. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 1335–1346, 2020.

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **Journal of Machine Learning Research (JMLR)**, Vol. 21, No. 140, pp. 1–67, 2020.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 311–318, July 2002.

[22] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In **International Conference on Learning Representations (ICLR)**, 2020.

[23] J. Peter Kincaid, Robert P. Fishburne, R L Rogers, and Brad S. Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. 1975.

[24] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning Robust Metrics for Text Generation. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 7881–7892, 2020.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In **Advances in Neural Information Processing Systems (NeurIPS)**, pp. 8024–8035. 2019.

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations**, pp. 38–45, 2020.

[27] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In **International Conference on Learning Representations (ICLR)**, 2019.

[28] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT) Demonstrations**, pp. 48–53, 2019.

表 3　NLI 分類によるフィルタリングにより除去された WikiSplit[5] 中の実際の事例.

複雑な文	単純な文	
Her father was a physician and she was raised in a secular environment.	Her father was a physician, and she followed in his footsteps.	She was raised in a secular environment.
It debuted at number 24 on the US "Billboard" 200, and at number 70 in Canada.	It debuted at number 24 on the "Billboard" 200, one of the top debuts of that week.	The album debuted at number 70 in Canada.
A pink Hippo-like diplodorian, he can produce bubbles from his mouth.	A pink Hippo-like diplodorian.	A blue diplodorian who can produce staples from his mouth.

表 4　HSplit[9] の実際の事例と，各システムごとの生成文．“+NLI” はフィルタリングを行ったデータセットで訓練したモデル，“+再現器” は再現器を用いたモデル，“+NLI+再現器” はその両方を用いて訓練したモデルを表す.

入力文（複雑な文）	Below are some useful links to facilitate your involvement.
参照文（単純な文）	Below are some useful links. They will help facilitate your involvement.
システム	生成文
DisSim[15]	Below are some useful links to facilitate your involvement.
Kim ら [7]	Below are some useful links to facilitate your participation. For more information, click here.
T5-small	Below are some useful links to facilitate your participation. For more information, click here.
T5-small + NLI	Below are some useful links to facilitate your participation. For more information, click here.
T5-small + 再現器	Below are some useful links.
T5-small + NLI + 再現器	Below are some useful links. These links are useful to facilitate your involvement.

A　フィルタされた実際の事例

NLI 分類によるフィルタリングによって実際に除去された WikiSplit[5] 中の事例を表 3 に示す.

B　評価実験の詳細

実験のためのエンコーダ・デコーダモデルとして，T5[20] の small サイズ（T5-small）を用いた．T5-small のパラメータ数は 6000 万である．実験には PyTorch[25] を用い，T5 の事前訓練済みモデルとして Transformers[26] で公開されている事前訓練済み重み[2] を用いた．T5 は事前訓練時に task prefix と呼ばれる入力文に対する接頭辞を付加してマルチタスク学習を行っているが，単一のタスクで fine-tuning する際は prefix を用いないのが一般的であるため[3][4]，prefix を入力文に付加せずに訓練を行った．fine-tuning の設定として，バッチサイズを 16，最適化手法に AdamW[27] を用い，訓練ステップ数を Kim ら [7] の設定と同様の 10 万とした．また，学習率を 10^{-5} として，訓練開始時点での学習率を 0 とし，全訓練ステップのうち 10%で設定した値まで線形に学習率を増加させる学習率スケジューリング手法の warm-up を用いた．再現器の損失にかける重みである α は 1 に設定した．1 万ステップごとに開発セットでの評価を行い，最も開発セットでの損失が小さいチェックポイントを最終的な評価に用いた．評価時の文生成には，最も基本的な文生成の性能を評価するため，貪欲法を用いて行った．Tu ら [12] は再現器を文生成時のビームサーチにおける各候補文のスコアリングのためにも用いたが，今回は貪欲法で文生成を行うため，再現器は訓練にしか使用せず，評価時にはエンコーダ・デコーダのみを用いて文生成を行った.

比較対象である Kim ら [7] のモデルとして，著者らが公開する実装[5] から，fairseq[28] で実装された事前訓練済みモデルを用いた．評価時の文生成は，Kim ら [7] の設定と同様，トライグラムが連続しないように制約を加えた上で，窓幅 10 のビームサーチをすることで行った.

C　システムごとの実際の生成文

表 2 で用いた各システムごとの，複雑な文が入力された時の実際の生成文を表 4 に示す.

2）https://huggingface.co/t5-small
3）https://github.com/huggingface/transformers/issues/6007#issuecomment-663627849
4）https://discuss.huggingface.co/t/does-task-specific-prefix-matters-for-t5-fine-tuning/501/9
5）https://github.com/mounicam/BiSECT