テキスト変換モデルに基づく 様々な制約を用いたインタラクティブ要約

柴田 知秀† 山田 悠右‡ 小林 隼人† 田口 拓明† 奥村 学‡
†ヤフー株式会社 ‡東京工業大学
{tomshiba,hakobaya,htaguchi}@yahoo-corp.jp
{yuyamada@lr.,oku@}pi.titech.ac.jp

1 はじめに

Encoder-decoder モデル [1, 2] や事前学習モデル [3, 4] などの深層学習の進展により、テキスト自動要約の技術が進歩している。自動要約手法の精度を向上させる余地がある一方で [5]、そもそも要約の正解には様々な観点が考えられる場合も多いことから、ユーザからのフィードバックをもとに要約を生成・改良させる研究が進められている [6, 7]。本論文ではユーザが制約を与えてインタラクティブに要約を行う設定 (Mao らの研究) [6] に焦点をあてる。

制約付き要約が必要となる場面を、Yahoo!ニュース・トピックスを例として説明する。Yahoo!ニュース・トピックスでは、日々のニュース記事の中から重要なニュースを選択し、要点を押さえた短い見出しを編集者の手により作成している。そこでは、見出しの候補を自動生成し編集者に提示することによって、編集支援を行っている [8]。この発展として、自動生成された見出しに対して、編集者が含めたいキーワードやフレーズを制約として与えると、その条件を満たす見出しが生成されれば、さらなる編集支援を行えると考えられる。

先に述べた Mao らの研究 [6] やその他の制約付き生成の研究 [9, 10] ではキーワードやフレーズなど、入力文書に依らず同一の制約のタイプを想定している。しかし、制約のタイプは文書によって異なるのが自然である。例えば求める要約が抽出型であれば制約として文であることが望ましく、また、抽象型であればキーワードやフレーズであることが望ましい。また、Mao らの研究では制約を一回与えるのみであるが、数回与えながら徐々に要約を改善していくのが自然な設定である。

そこで本研究では様々なタイプの制約を与えるインタラクティブ要約の手法を提案する。提案手法の概要を図 1 に示す。本研究は自動処理の精度を向上させるという設定ではなく、入力の文書に対してユーザ (編集者) が生成したい要約が頭にあり、それをもとに制約を与えて要約を改善する、というインタラクティブな要約をシミュレートする設定となる。まず、制約なしで生成した要約がユーザに提示され、ユーザは制約を与える。この例ではキーワード制約として「新しい」という単語を与え、システムはこの制約のもとに要約を生成する。さらにフレーズ制約として「気象庁」を与える、という具合にインタラクティブに要約を生成していく。制約付きの学習・評価セットを構築するのはコストが高いことから、Mao らの研究と同じく、正解要約に含まれていてシステム出力には含まれていないキーワードやフレーズを疑似的に制約とみなす。

また、Mao らは制約を満たす生成を得るモデル [11] をデコード時に用いているが、このモデルでは様々なタイプの制約に対応することができない。そこで本論文ではテキスト変換モデルを用い、制約の単位にかかわらず、制約を入力側に入れることにより制約付き生成を実現する。本研究ではテキスト変換モデルとして T5 [3] を用いる。

標準的に用いられている評価セットを用いて、インタラクティブ要約がどのように実現されるかを検証した。

2 提案手法

2.1 枠組み

まず、入力文書 (*src*) と正解要約 (*tgt*) が与えられ、制約なしのベースモデル M_{base} を学習する。このモデル M_{base} を使って生成された要約を *pred* とする。制約 C は *tgt* にあって、*pred* に存在しないものを抽出する。

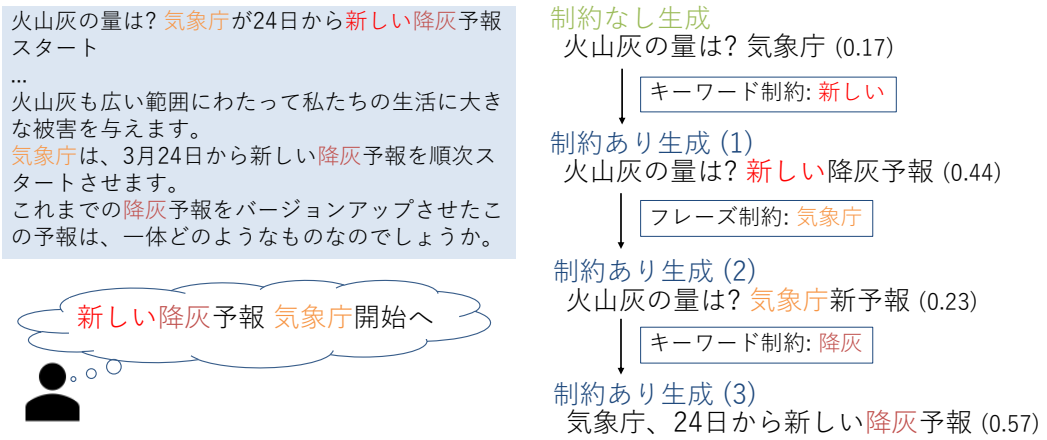

図 1　提案手法の概要 (括弧内の数字は ROUGE2 F の値を示す)

1. トランプ氏にサンダース氏 「異端」が躍進する米大統領選 2. 米大統領選の候補指名争いには、ある異変があります。 3. アイオワで勝利した共和党のクルーズ氏、今回のニューハンプシャーを制した共和党のトランプ氏、民主党のサンダース氏。 ..	tgt: 米大統領選「異端」なぜ躍進 pred: トランプ氏にサンダース氏 異端躍進 ・**文制約**: 1 文目 ・**フレーズ制約**: 米大統領選 ・**キーワード制約**: 大統領, 選, なぜ
1. <ソチ五輪>女子アイスホッケーの初戦に見えた課題と収穫 2. .. 16 年ぶりに五輪舞台に戻ってきたアイスホッケー女子日本代表「スマイルジャパン」が、9 日の 1 次リーグ初戦で世界ランク 6 位の強豪スウェーデンに 0 − 1 と惜敗。 3. スリリングな展開で多くの見せ場を作りながらも、..	tgt: スマイル J 格上相手に惜敗 pred: 女子アイスホッケー 初戦に見えた課題と収穫 ・**文制約**: 2 文目 ・**キーワード制約**: スマイル, 惜敗

図 2　様々なタイプの制約 (下線が引かれた制約はランダムに選ばれたものを示す)

2.2　様々なタイプの制約

制約として、入力によらず単一の制約のタイプを与えるというよりは様々なタイプの制約を与えるのが自然である。本研究では制約のタイプとして文、フレーズ、キーワードを考える。

文　*tgt* に含まれていて、*pred* に含まれない文を制約として抽出したいが、入力中のある文がそのまま *tgt* に含まれているとは限らないので、抽出型要約における正解文抽出の方法 [12, 13] を拡張し、*pred* と連結すると ROUGE F 値が高くなるような入力中の文を制約として抽出する。

キーワード　文単位で制約を考えると、特に抽象型要約の場合、不要な部分が含まれてしまう可能性が高い。そこで、キーワード単位での制約を考え、*tgt* に含まれ、*pred* には含まれないものを制約とする。

制約の候補を抽出する対象として 2 種類の問題設定を考える。1 つは *src* のみで、これはキーワードを入力文書から抜き出す形で設定することを想定する (この設定を *src* と呼ぶ)。もう 1 つの設定として、*tgt* も加えることを考え、これは入力文書には含まれていない言い換え表現などをキーワードを含めることができる (この設定を *src* + *tgt* と呼ぶ)。

フレーズ　キーワードよりも少し大きな単位としてフレーズを考える。要約に含むべきフレーズとして固有表現が多いことから、本研究では固有表現のみを対象とする。キーワードと同様に、*tgt* に含まれ、*pred* には含まれないものを制約とする。

ある文書について制約のタイプが複数抽出される場合、どの制約のタイプが最善であるかは分からないので、ランダムに制約のタイプを選ぶ。また、ある制約について複数ある場合はその中からランダムに一つ選ぶ。図 2 に制約の例を示す。一つ目の例では文・フレーズ・キーワード制約が得られており、ここからランダムに選んだ結果、キーワード制約が選ばれ、3 つのキーワードのうちからランダムに「大統領」が選ばれている。二つ目の例では文制約とキーワード制約が抽出されており、フレーズ制約は条件に合致するものがない。

2.3　テキスト変換モデルの利用

テキスト変換モデルを利用することにより、制約がどのような単位であっても入力に制約を加えるだけで、制約付き生成を実現することができる。制約なしのベースモデル M_{base} でシステム出力 *pred* を

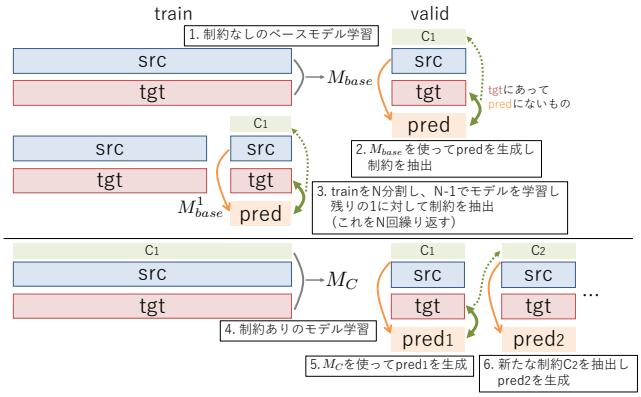

図 3　制約付きデータの構築とインタラクティブ生成の方法 (test は valid と同様の方法で構築)

生成することを以下のフォーマットで表現する。

summarize: $src \rightarrow pred$

制約ありモデル M_C、i 回目のシステム出力を $pred_i$ とすると、M_C で $pred_i$ を生成する場合、制約を入力の先頭に追加し、

C summarize: $src \rightarrow pred_i$

とする。制約 C は「(制約タイプ) constraint: (制約文字列)」とする。制約タイプは sentence、phrase、keyword のいずれかをとり、制約のタイプを区別するために付与している。図 2 の 1 つ目の例の入力は以下のようになる。

keyword constraint: 大統領 summarize: トランプ氏に.. → ..

インタラクティブに生成を繰り返し行うので、2 回目以降の生成では過去の制約も満たすように、以下のように過去の制約も入力に加える (C_i は i 回目の制約を表わす)。

C_1 C_2 .. C_i summarize: $src \rightarrow pred_i$

2.4　制約付きデータの構築とインタラクティブ生成

制約付き評価セットの構築ならびにインタラクティブ生成を図 3 に示す。まず、train の src と tgt を使ってベースモデル M_{base} を学習し、valid の $pred$ を生成する。そして、$pred$ と tgt を比較し、2.2 節で述べた方法で制約を抽出する。

制約ありモデルを学習するために、train についても制約を付与する必要があるので、train を N 分割し[1]、$N-1$ 個で制約なしモデルを学習し、残りに対して pred を生成し、同様に制約を抽出する。そしてこれを N 回繰り返すことにより、train のすべてに対して制約を得る。次に、train の制約付き src と tgt から制約ありモデル M_C を学習する。学習したモデルを用いて valid に対して $pred_1$ を生成し、インタラクション 1 回目の生成 $pred_1$ を得る。次に、$pred_1$ と tgt から次の制約 C_2 を得て、C_2 と src に対して M_C を適用することにより、2 回目の生成 $pred_2$ を得る。このようにして、インタラクティブな生成を実現することができる。

1) 実験では $N = 5$ とした。

	Train	Valid	Test
CNN/DM	287,111	13,367	11,489
XSum	204,017	11,327	11,333
Reddit	33,711	4,214	4,214
Topics	245,474	10,000	10,000

表 1　評価セットの統計データ

3　実験

3.1　実験設定

自動要約で標準的に用いられている英語の評価セット 3 つ (CNN/DM [14, 1], XSum [15], Reddit [16]) と、日本語のものとして Yahoo!ニュースのトピックス記事のデータセット (Topics [8]) を用いた。表 1 に各評価セットの統計データを示す。

CNN/DM は抽出度が高く、XSum と Reddit は抽出度が低いデータセットであり、抽出度によってどのようにシステムの振舞いが異なるかを実証する。

評価尺度は標準的に用いられている ROUGE-1,2,L の F 値 [17] を用いた。評価の基本単位として英語の場合は単語を、日本語の場合は文字を用いた。また、インタラクションの回数を 3 とした。

事前学習モデルとして、英語の場合は T5[2] を、日本語の場合は mT5[18][3] を用いた。モデルサイズはどちらも base を利用した。実装は Hugging Face 社が提供する transformers の seq2seq を使った[4]。

キーワード制約の抽出において、英語では TextRank[19, 20][5] を用い、日本語では形態素解析器 JUMAN[6] で単語に分割し、TFIDF が上位のものをキーワードとした。フレーズ制約の抽出において Spacy を用いて固有表現解析を行った。

3.2　実験結果・考察

制約を用いないベースラインと、制約として文、フレーズ、キーワードのうちの一つのみを採用した

2) https://huggingface.co/google/t5-v1_1-base
3) https://huggingface.co/google/mt5-base/tree/main
4) https://github.com/huggingface/transformers/tree/master/examples/seq2seq
5) https://github.com/summanlp/textrank
6) http://nlp.ist.i.kyoto-u.ac.jp/?JUMAN

制約					CNN/DM			XSum			Reddit			Topics		
S	P (src)	P (src+tgt)	K (src)	K (src+tgt)	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
					41.30	18.77	38.32	34.23	11.91	25.69	26.28	7.70	21.11	60.87	46.56	57.02
✓					**46.45**	**23.87**	**43.23**	38.54	15.33	29.55	29.55	10.09	23.86	62.77	48.34	58.71
	✓				43.22	20.68	40.17	37.74	14.84	28.87	24.49	7.05	20.02	58.34	44.73	55.16
		✓			43.81	21.22	40.75	**42.81**	**20.37**	**33.22**	26.05	7.92	21.13	67.98	55.42	64.28
			✓		43.86	20.92	40.72	38.80	15.31	29.54	17.89	4.62	15.47	75.39	60.42	69.59
				✓	43.28	20.49	40.18	41.02	17.26	31.75	28.08	8.67	22.65	78.62	63.43	72.44
✓	✓		✓		45.78	23.11	42.59	40.10	16.55	30.70	24.49	7.49	20.85	73.88	59.14	68.55
✓		✓		✓	46.14	23.39	42.95	41.98	19.01	31.84	**29.68**	**10.10**	**23.90**	**79.34**	**65.76**	**73.74**

表 2 実験結果 (制約の S, P, K はそれぞれ文、フレーズ、キーワードを示す)

		S	P (src)	P (src+tgt)	K (src)	K (src+tgt)	なし (src)	なし (src+tgt)
CNN/DM	1	0.983	0.572	0.726	0.849	0.930	0.007	0.003
	2	0.948	0.478	0.626	0.788	0.882	0.029	0.014
	3	0.895	0.396	0.544	0.712	0.829	0.070	0.040
XSum	1	0.394	0.115	0.570	0.480	0.734	0.354	0.112
	2	0.244	0.063	0.295	0.304	0.542	0.556	0.274
	3	0.144	0.028	0.247	0.152	0.353	0.762	0.503
Reddit	1	0.836	0.045	0.153	0.300	0.504	0.145	0.109
	2	0.791	0.029	0.096	0.194	0.348	0.195	0.222
	3	0.617	0.016	0.055	0.129	0.233	0.372	0.389
Topics	1	0.343	0.110	0.325	0.646	0.808	0.334	0.161
	2	0.194	0.082	0.200	0.503	0.654	0.483	0.326
	3	0.072	0.046	0.110	0.303	0.443	0.687	0.542

表 3 制約が抽出された割合 (「なし」はどのタイプの制約も抽出されなかったことを、2 カラム目の「1,2,3」はインタクラションの回数を示す)

		P (src)	P (src+tgt)	K (src)	K (src+tgt)
CNN/DM	1	0.682	0.658	0.610	0.655
	2	0.548	0.550	0.496	0.554
	3	0.417	0.491	0.475	0.484
XSum	1	0.815	0.870	0.796	0.781
	2	0.748	0.857	0.738	0.706
	3	0.746	0.805	0.680	0.645
Reddit	1	0.318	0.464	0.466	0.390
	2	0.297	0.373	0.281	0.331
	3	0.250	0.319	0.324	0.300
Topics	1	0.918	0.976	0.880	0.908
	2	0.829	0.908	0.803	0.794
	3	0.727	0.864	0.708	0.708

表 4 出力が制約を満たした割合

場合、全てを採用した場合、ならびに、フレーズとキーワード制約については抽出の対象として *src* のみと *src* + *tgt* の場合 (2.2 節参照) を比較した。表 2 に実験結果を示す。

CNN/DM は抽出度が高い評価セットであるので、文制約のみを用いた場合が最も精度が高く、また、XSum は *src* に含まれないフレーズが多いことからフレーズ制約 (*src* + *tgt*) のみを用いた場合が最も精度が高かったが、おおまかな傾向としてはすべての制約を用いた場合が精度が高く、様々な制約を用いることの有効性を示すことができている。

Reddit の P(src), K(src) ならびに Topics の P(src) は制約を用いないベースラインよりも低くなってしまっている。この原因を明らかにするために、valid セットにおいてそれぞれの制約が抽出された割合を表 3 に示す。Reddit の P(src),K(src) と Topics の P(src) は他に比べて制約が抽出された割合が低いことがわかる。これはこれらの評価セットでは src にない多くのキーワード・フレーズが要約に使用されていることに起因する。

表 4 にシステム出力が与えた制約を満たした割合を示す[7)]。インタラクションが進むにつれて、制約を満たすのが次第に難しくなっていることがわかる。また、Reddit は他の評価セットに比べて制約を満たした割合が低い。これは難しい評価セットであることと、制約が抽出された割合が他の評価セットに比べて低い (表 3 参照) ことに起因する。

以下の例では Topics 評価セットにおいて、制約付き出力が制約「夏」を満たしておらず、制約なし出力と同じ出力となってしまっている。この問題に対しては N-best を生成し、制約を満たす出力を採用することによって対処することが考えられる。

tgt: 梅雨の熱中症 真夏並みに注意
pred: 熱中症の季節到来 気をつけたい落とし穴
キーワード制約: 夏
制約付き pred: 熱中症の季節到来 気をつけたい落とし穴

4 おわりに

本論文ではテキスト変換モデルに基づき、文・フレーズ・キーワードの様々な制約を用いてインタラクティブに要約を行う手法を提案した。今後の課題としてはキーワードやフレーズを削除する制約の導入や出力長の制約を加えることなどがあげられる。

7) 文制約の場合、制約が満たされたかどうかを判断するのが難しいので、フレーズ制約とキーワード制約についてのみ算出した。

参考文献

[1] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of SIGNLL2016*, pp. 280–290, 2016.

[2] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL2017*, pp. 1073–1083, 2017.

[3] 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*, Vol. 21, No. 140, pp. 1–67, 2020.

[4] 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 ACL2020*, pp. 7871–7880, 2020.

[5] Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. What have we achieved on text summarization? In *Proceedings of EMNLP2020*, pp. 446–469, 2020.

[6] Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. Constrained abstractive summarization: Preserving factual consistency with constrained generation, 2020.

[7] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *CoRR*, Vol. abs/2009.01325, , 2020.

[8] Kazuma Murao, Ken Kobayashi, Hayato Kobayashi, Taichi Yatsuka, Takeshi Masuyama, Tatsuru Higurashi, and Yoshimune Tabuchi. A case study on neural headline generation for editing support. In *Proceedings of NAACL2019 (Industry Papers)*, pp. 73–82, 2019.

[9] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization, 2020.

[10] Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models, 2020.

[11] Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of NAACL2018*, pp. 1314–1324, 2018.

[12] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of AAAI2017*, pp. 3075–3081, 2017.

[13] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of EMNLP-IJCNLP2019*, pp. 3730–3740, 2019.

[14] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 28, pp. 1693–1701. Curran Associates, Inc., 2015.

[15] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP2018*, pp. 1797–1807, 2018.

[16] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proceedings of NAACL2019*, pp. 2519–2531, 2019.

[17] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004.

[18] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer, 2020.

[19] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of EMNLP2004*, pp. 404–411, 2004.

[20] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *CoRR*, Vol. abs/1602.03606, , 2016.

付録

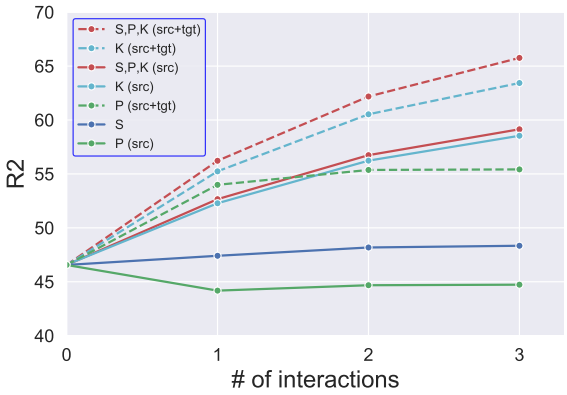

図 4　各インタラクションでの精度 (Topics)

図 4 に評価セット Topics における各インタラクションでの精度 (R2) を示す。全般的な傾向として、最初の上がり幅が最も大きく、ゆるやかに向上していくことがわかる。