

事前学習モデルを用いた少量データに対する日本語抽象型要約

勝又智

株式会社レトリバ

satoru.katsumata@retrieva.jp

1 はじめに

近年、自動要約の技術として、Encoder-Decoder を用いた抽象型要約が盛んに研究されている [1, 2, 3, 4, 5, 6, 7]. この Encoder-Decoder モデルを用いた抽象型要約の特徴として、学習に大量のデータが必要となることが知られている。

事前学習モデルを使用することで、少量の要約データでも高い精度が得られることがいくつかの研究で報告されている。例えば Dong ら [8] は、Gigaword [1] と呼ばれる英語要約データセットの学習データ 10K を使用した場合、事前学習を行わないモデルでは ROUGE-L が 10.42 ポイントのところを、事前学習したモデルを用いることで 30.56 ポイントとなることを示した。しかしながら、日本語の抽象型要約については事前学習モデルの有効性は調べられていない。そこで本研究は、日本語抽象型要約タスクについて、事前学習モデルを用いることで、学習データが大量に必要な問題の緩和につながるか調査を行った。具体的には、UniLM [8], JASS [9], multilingual T5 [10] と呼ばれる、3つの事前学習モデルを用いて調査を行い、モデル毎の比較を行った。

日本語要約研究では、新聞に関する要約データ、Web ニュースに関する要約データが使用されている [11, 2, 3]. しかしながら、実世界ではニュース記事以外の分野についても要約を行いたいという需要が存在する。そこで本研究は、wikiHow¹⁾ という様々な分野に関するハウツーが記載された Web サイトから要約データを作成し、ニュース記事以外に対する抽象型要約についても調査した。

少量データに対する事前学習モデルを用いた抽象型要約実験として、本研究では Livedoor News と wikiHow データに対して抽象型要約を行っている。特に、Livedoor News については学習データとして 10K 使用した場合と 1K 使用した場合を比較し、事前学習モデルの少量データに対する精度を調査し

た。実験結果から、数千件の学習データを用いた日本語抽象型要約において、事前学習モデルは有効であることがわかった。また、データセットによって事前学習モデルの振る舞いが変わる場合が存在することがわかった。

本研究の貢献は以下の2つである。

1. 様々な事前学習モデルを用いて、学習データが少量の際の日本語抽象型要約を実験、分析し、事前学習モデルの有効性を確認した。
2. wikiHow を用いて日本語要約データセットを作成、公開²⁾した。このデータセットは既存のニュース要約データセットと比べて抽象度が高いことを確認した。

2 関連研究

現在、日本語の要約研究は様々な問題設定で行われている。具体的には、新聞記事に対して、適切な長さの見出しを生成するといった設定 [2], Web ニュース記事に対して、3行の要約を生成する設定 [11], 議会の議事録から対話構造を考慮した要約を生成する設定 [12] などが存在している。このように、出力の構造を考慮する問題設定が多く研究されている。一方で、本研究では、抽象型要約の学習データが大量に必要な問題に対して取り組む。

現在、日本語要約研究では主に2種類のデータセットが存在している。一つが、新聞記事から作成した要約データである。Japanese News Corpus (JNC) [2] は新聞記事の先頭3文と対応する見出しの対になっており、JAPANESE Multi-Length Summarization Corpus (JAMUL) や JAMUL2020 [3] は1記事に対して複数の要約が紐づいているデータセットである。もう一つが Livedoor News から作成した3行要約データセット [11] である。これは Livedoor News が記事と対応する3行の要約が紐づいて公開されていることを利用して作成したデータセットである。

1) <https://www.wikihow.jp>

2) https://github.com/Katsumata420/wikihow_japanese

表 1 事前学習モデル

モデル	構造	パラメータ数
UniLM [8]	Encoder	110M
JASS [9]	Encoder-Decoder	360M
mT5 small [10]	Encoder-Decoder	300M

これらのデータセットは全てニュース記事に関するデータセットである。一方で、英語で記述された要約データはニュース記事 [1, 13], 特許 [14], 電子メール [15] など、様々な分野で存在する。その中の一つに、wikiHow³⁾ という様々な分野に関するハウツーが記載された Web サイトから作成した要約データ [16] が存在する。このデータはニュース記事で作成したデータセットと異なり、様々な分野について抽象度の高い要約データという特徴を持つことが報告されている。本研究では、この英語 wikiHow のデータセットを参考に、新たに日本語版 wikiHow から要約データセットを作成した。wikiHow は様々なトピックに関して記述されており、ニュース記事とは記事と要約、どちらも傾向が異なると考えられる。このデータセットを利用してニュース記事以外での抽象型要約の振る舞いを調査する。

3 事前学習モデルを用いた要約

本研究は抽出型要約について、いくつかの事前学習モデルを用いて実験した。表 1 に本研究で使用する事前学習モデルをまとめている。

3.1 UniLM

Dong ら [8] は BERT [17] の事前学習をより拡張した手法 (Unified pre-trained Language Model; UniLM) をいくつか提案した。その中の一つに Sequence-to-Sequence Language Model (seq2seq LM) が存在する。概要図を図 1 に示す。BERT は入力トークンの一部を [MASK] に置換し、置換前の単語を推定するように事前学習を行う (Masked Language Model; MLM)。この時、BERT は注意機構を双方向に用いている。一方で seq2seq LM は図のように、入力記事側は双方向なのだが、出力要約側は出力するものより以前のトークンのみに注意機構が張られる。この機構を使用することで要約などの系列変換が学習できることを報告している。

本研究では日本語データで学習された BERT を元に、次の fine-tuning と推論を行っている。抽出

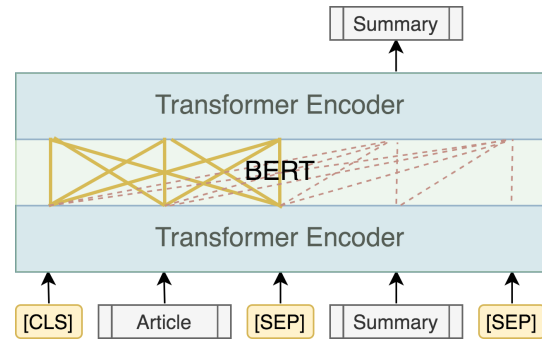


図 1 UniLM seq2seq LM の概要図

型要約の学習を行う際は、要約側の一部の単語を [MASK] に置換し、seq2seq LM を用いて元の単語を当てるように学習を行う。推論時には入力記事を原言語側に与え、要約側の入力として [MASK] を用意し、出力の終了記号 [EOS] が出るまで推論を行う。

3.2 JASS

JASS (JAPANESE-specific Sequence to Sequence) [9] は、MASS (MAsked Sequence to Sequence) [4] を日本語向けに拡張した目的関数を用いて Transformer [18] を事前学習したモデルである。MASS は大量の単言語データの各文中の単語をランダムにマスクトークンに置換し、このマスクトークンを Decoder を用いて推定するように事前学習されている。一方で、JASS はこのマスクトークンの置換区間として、文節単位で置換し、元の単語を推定する事前学習を行っている。さらに、入力文を文節単位でシャッフルし、元の語順を Decoder で推定する事前学習も行っている。本研究ではこの JASS モデルに対して、要約データで fine-tuning および推論を行った。

3.3 multilingual T5

Raffel ら [5] は事前学習 Encoder-Decoder として Text-to-Text Transfer Transformer (T5) を作成した。このモデルは、単言語コーパスの各文中の単語をランダムにマスクトークンに置換し、元の単語を当てるように Transformer を事前学習している。Xue ら [10] は様々な言語の単言語コーパスで T5 と同様の事前学習を行い、モデルを作成した (multilingual variant of T5; mT5)。本研究ではこの mT5 に対して、要約データを用いて fine-tuning, 推論を行った。

4 日本語 wikiHow 要約データセット

本研究では Koupaee and Wang [16] と同様の手法で wikiHow から日本語要約データセットを作成した。

3) <https://www.wikihow.com>

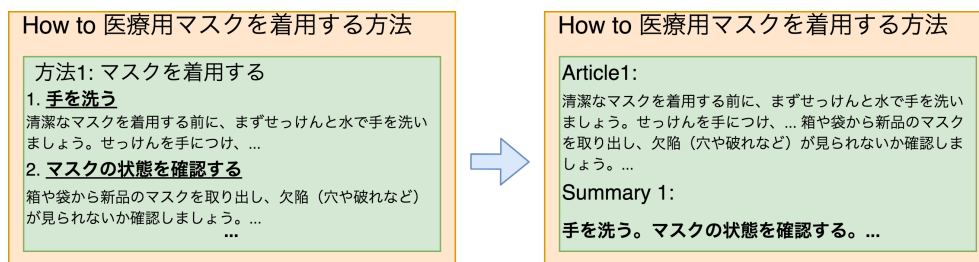


図2 wikiHow から要約データを作成する例

表2 wikiHow データの統計情報

データサイズ	5,066
1記事の平均単語数	566.8
1要約の平均単語数	48.8
1要約の平均文数	6.0

日本語 wikiHow は全 19 カテゴリについて、各ページが構造化して記述されている。具体的には、ページ内にいくつかの大段落が存在しており、この中に小段落として記事と対応した見出しが存在している。本研究ではこの組み合わせを利用して要約データを作成した。

具体例を図 2 に示す。この図のように、記事と要約はそれぞれ各小段落内の記事と見出しを繋げたものとして利用している。見出しとして‘手を洗う’、‘マスクの状態を確認する’とあれば作成する要約は‘手を洗う。マスクの状態を確認する。’となる。このようにして、2020 年 6 月時点で取得できた全てのページから要約データを作成した。

作成したデータセットの統計を表 2 に示す。英語 wikiHow データセットは数十万件の記事が含まれているが、日本語 wikiHow データセットは数千件しか含まれていない。

また、Livedoor News の要約データセットと要約の抽象度合いの観点で比較した。抽象度合いは記事側には存在せず、要約側のみに存在する単語 n-gram の割合で計測している。比較の結果、Livedoor News データと比べると wikiHow データの方が要約側にしか存在しない n-gram が多く、より抽象度が高いデータセットであることがわかった。n-gram の割合について、詳細な値は付録 A に記載した。

作成した wikiHow データセットの特徴として、要約側に同一の表現が複数回出現することを確認した。Livedoor News 5,066 件中、32.1 % 件の要約側で同一の 3-gram が複数回存在していたが、wikiHow には 62.2 % 同一の 3-gram が存在していた。このように、wikiHow データセットは既存のデータセットとは要約の記述スタイルが異なることがわかる。

5 実験

本研究では、事前学習モデルを用いて wikiHow データと Livedoor News の 1K, 10K データに対して入力が 1 記事の抽象型要約実験を行った。

5.1 実験設定

wikiHow については、得られたデータを 5 等分し、3:1:1 の割合で学習、開発、テストデータとした。そのため、学習データ、開発データ、テストデータはそれぞれは 3,040 件、1,013 件、1,013 件である。また、Livedoor News については、公開されている分割⁴⁾を元に学習、開発、テストデータを用意した。ただし、2020 年時点で取得できたデータのみを使用しているため、開発データとテストデータはそれぞれ 886 件、871 件である。学習データについては取得できたデータからランダムに 1K または 10K 抽出している。評価については出力分ち書きから空白を取り除いた後、MeCab⁵⁾で分割し、ROUGE-1 (R-1), 2 (R-2), L (R-L) の F 値を測定している。

UniLM として、日本語データで学習された BERT⁶⁾に対して、seq2seq LM を用いた fine-tuning を行った⁷⁾。単語分割は MeCab と WordPiece [19] を用いている。JASS は日英の単言語データで学習されたモデル⁸⁾に対して要約の fine-tuning を行った。単語分割は Juman++ [20] と SentencePiece [21] を、fine-tuning と推論には OpenNMT [22] を使用している。mT5 は他のモデルと同程度のパラメータ数のモデル⁹⁾に対して要約の fine-tuning を行った。単語分割は SentencePiece を使用し、fine-tuning と推論は

4) <https://github.com/KodairaTomonori/ThreeLineSummaryDataset>

5) <https://taku910.github.io/mecab>

6) <https://github.com/cl-tohoku/bert-japanese> の bert-base-japanese-whole-word-masking を使用。

7) <https://github.com/microsoft/unilm/tree/master/s2s-ft>

8) <https://github.com/Mao-KU/JASS> の JASS-jaen を使用。

9) <https://github.com/google-research/multilingual-t5> の mT5-Small を使用。

表 3 日本語抽象型要約実験結果

	Livedoor News 1K			Livedoor News 10K			日本語 wikiHow		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	43.2	18.9	39.6	43.2	18.9	39.6	31.5	8.6	29.9
Pointer-Generator	25.5	5.0	24.2	41.1	18.0	38.2	24.0	7.8	23.7
UniLM	46.8	21.8	43.3	49.4	23.9	45.8	36.7	13.7	35.8
JASS	43.4	17.2	39.8	43.5	17.9	39.8	34.6	11.9	33.7
mT5	43.6	20.2	40.1	45.5	21.6	41.8	29.0	9.3	27.9

Mesh Tensorflow [23] を使用している。また、比較として事前学習なしの手法を2つ、実験した。1つが LEAD-3 と呼ばれる、記事から最初の3文を要約として使用する手法で、もう1つが Pointer-Generator [6] と呼ばれる Encoder-Decoder を用いた手法である。Pointer-Generator は OpenNMT の実装を使用、単語分割は MeCab を用いている。

mT5, LEAD-3 以外のモデルは推論時に 3-gram の繰り返しの削除 (Trigram Block) [7] を行っており、全てのモデルはビーム幅は4で実験している。上記4モデルのハイパーパラメータの詳細は付録 B に記載している。

5.2 実験結果

実験結果を表 3 に示す。各評価指標において最も高いスコアを太字にしている。全ての実験設定において、UniLM を用いた手法が最も良い結果であった。また、学習データが数千件の設定では、事前学習なしの手法より事前学習を行っている手法の方が良いことがわかる。

Livedoor News については UniLM に次いで mT5 の精度が高く、これら2つのモデルは LEAD-3, Pointer-Generator よりもどの評価指標においても優れている結果となった。一方で JASS は LEAD-3 と同程度または低い精度であり、特に 10K の場合において、ROUGE-2 について Pointer-Generator と同程度の精度であることがわかる。wikiHow においては Livedoor News の場合と異なり、JASS の方が mT5 より高い精度であることがわかる。

6 分析と考察

要約実験結果から、mT5 が wikiHow に対して精度が低いことがわかる。調査の結果、mT5 のテストデータ全体に対して 88.3 % について同一の文の繰り返しが発生していた。実際の出力例は付録の C に示す。一方で、Livedoor News 1K に対しては mT5 は 10.0 % 繰り返している。UniLM は wikiHow,

Livedoor News 1K のどちらも 0.1 % 繰り返していた。このように、mT5 を用いた場合、文単位での繰り返しが発生しやすく、データセットによって繰り返し度合いが大きく変わっていることがわかる。この理由の1つとして、mT5 は推論時に Trigram Block を行っていないからだと考えられる。また、wikiHow については、4章で述べたように、要約側に同一の表現が複数回現れることを確認している。そのため、wikiHow で学習することにより、同一の表現を出力するように学習していき、推論時にもそのような出力をする傾向にあると考えられる。

事前学習を行ったモデルと行っていないモデルの定性的な比較を行う。Pointer-Generator の wikiHow に対する出力は平均単語数が 12.6 で、平均語彙は 8.9 であった。一方で、UniLM の出力は平均単語数が 26.2 で平均語彙が 18.6 となり、2倍以上の長さ、語彙であった。この結果と ROUGE スコアから、事前学習を行うことでより入力記事を説明している要約が生成できていると考えられる。これは、学習データが数千件の状況で事前学習を行わなかった場合は学習が安定せず、単純な要約しか学習できない一方で、事前学習を行うことで、学習が安定し、より複雑な要約も可能になると考えられる。

7 おわりに

本研究では、学習データが数千、数万件の日本語抽象型要約タスクにおいて、事前学習モデルの有効性を調査した。また、既存のニュース記事の分野とは異なる、ハウツーが記述されたデータから要約データ日本語 wikiHow を作成、公開した。この日本語 wikiHow および Livedoor News について、学習データが数千件の場合、事前学習モデルである UniLM を用いることで、事前学習なしのモデルと比べて大きく精度が向上することを確認した。今後は別の分野に関する要約データセットの作成、事前学習モデルを元に要約精度を改善する手法の研究に取り組みたい。

参考文献

- [1] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proc. of EMNLP*, pp. 379–389. Association for Computational Linguistics, 2015.
- [2] Yuta Hitomi, Yuya Taguchi, Hideaki Tamori, Ko Kikuta, Jiro Nishitoba, Naoaki Okazaki, Kentaro Inui, and Manabu Okumura. A large-scale multi-length headline corpus for analyzing length-constrained headline generation model evaluation. In *Proc. of the 12th International Conference on Natural Language Generation*, pp. 333–343. Association for Computational Linguistics, 2019.
- [3] 人見雄太, 田口雄哉, 田森秀明, 岡崎直観, 乾健太郎. 小規模リソースにおける生成型要約のためのスタイル転移. 言語処理学会第 26 回年次大会, pp. 929–932, 2020.
- [4] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pp. 5926–5936, 2019.
- [5] 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.
- [6] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*, pp. 1073–1083. Association for Computational Linguistics, 2017.
- [7] Romain Paulus, Caiming Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *ArXiv*, Vol. abs/1705.04304, , 2018.
- [8] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proc. of NeurIPS*, Vol. 32, pp. 13063–13075. Curran Associates, Inc., 2019.
- [9] Zhuoyuan Mao, Fabien Cromieres, Raj Dabre, Haiyue Song, and Sadao Kurohashi. JASS: Japanese-specific sequence to sequence pre-training for neural machine translation. In *Proc. of LREC*, pp. 3683–3691. European Language Resources Association, 2020.
- [10] Linting Xue, Noah Constant, A. Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, A. Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. *ArXiv*, Vol. abs/2010.11934, , 2020.
- [11] 小平知範, 小町守. TL;DR 3 行要約に着目したニューラル文書要約 (データ工学). 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, Vol. 117, No. 212, pp. 193–198, 2017.
- [12] 木村泰知, 渋木英潔, 高丸圭一, 秋葉友良, 石下円香, 内田ゆず, 小川泰弘, 乙武北斗, 佐々木稔, 三田村照子, 横手健一, 吉岡真治, 神門典子. NTCIR-15 QA Lab-PoliInfo2 のタスク設計. 言語処理学会第 26 回年次大会, pp. 625–628, 2020.
- [13] 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, *Proc. of NeurIPS*, Vol. 28, pp. 1693–1701. Curran Associates, Inc., 2015.
- [14] Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proc. of ACL*, pp. 2204–2213. Association for Computational Linguistics, 2019.
- [15] Rui Zhang and Joel Tetreault. This email could save your life: Introducing the task of email subject line generation. In *Proc. of ACL*, pp. 446–456. Association for Computational Linguistics, 2019.
- [16] M. Koupae and William Yang Wang. Wikihow: A large scale text summarization dataset. *ArXiv*, Vol. abs/1810.09305, , 2018.
- [17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pp. 4171–4186. Association for Computational Linguistics, 2019.
- [18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proc. of NeurIPS*, Vol. 30, pp. 5998–6008. Curran Associates, Inc., 2017.
- [19] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152. IEEE, 2012.
- [20] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proc. of EMNLP*, pp. 2292–2297. Association for Computational Linguistics, 2015.
- [21] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP*, pp. 66–71. Association for Computational Linguistics, 2018.
- [22] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. of ACL*, pp. 67–72. Association for Computational Linguistics, 2017.
- [23] Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, and Blake Hechtman. Mesh-TensorFlow: Deep learning for supercomputers. In *Proc. of NeurIPS*, 2018.

A 要約側のみに存在する n-gram 統計

日本語 wikiHow 要約データと Livedoor News 1K について、記事側には存在せず、要約側のみに存在する単語 n-gram の割合を表 4 に示す。なお、ここでの単語は MeCab の解析結果の形態素を指している。この表から、wikiHow の方が要約側にしか含まれない n-gram が多く、Livedoor News と比べて抽象度の高いデータセットであることがわかる。

表 4 事前学習モデル

データセット	1-gram	2-gram	3-gram	4-gram
Livedoor News 1K	11.4	39.1	57.5	68.3
wikiHow	25.7	60.8	81.3	91.2

B 詳細な実験設定

UniLM, JASS, mT5, Pointer-Generator について詳細な実験設定を表 5 に示す。

表 5 実験設定

	UniLM	JASS	mT5	Pointer-Generator
単語分割	MeCab + WordPiece	Juman++ + SentencePiece	SentencePiece	MeCab
事前学習データ	Ja Wikipedia	Ja Common Crawl + En News Crawl	mC4	-
入力文長/出力文長	400/100	400/100	512/64	400/100
語彙数	32,000	60,000	250,100	10,182
ネットワーク構造	BERT _{BASE}	Transformer-big	8ヘッド, FF1024次元 8層 Transformer	注意機構付き 1層 biLSTM Enc, 1層 LSTM Dec
最適化/学習率	AdamW/7e-5	Adam/3e-3	AdaFactor/1e-3	AdaGrad/0.15
Dropout	0.1	0.3	0.1	0.0
バッチサイズ	10 samples	2,048 tokens	2,048 tokens	32 samples
Gradient Accumulation	6	5	1	1
学習数	150 epochs	30,000 steps	10,000 steps	10,000 steps
GPU			GTX 1080Ti 1枚	

C 出力例

wikiHow に対する UniLM, JASS, mT5, Pointer-Generator の出力例を表 6 に示す。

表 6 wikiHow に対する出力例

入力記事	<p>落ち着いた気持ちになって、辛抱強い精神力を持ちましょう。何を学ぼうとする時でも、時間がかかるのが普通です。夜遅くまで起きてする勉強は、うまい方法ではありません。睡眠不足は、集中力の低下を引き起こし、覚えたものが定着しません。生徒の皆さんは、授業の時間割や宿題の提出日などの大事な予定を、時間通りに当たり前のようになさっています。このような予定の一環として、勉強や課題をするために一定の時間を確保しましょう。そうすれば、土壇場で課題を終わらせるために慌てふためいたり、大事なテストの前夜に詰め込み勉強をする事態は減ってきます。また、時に、学校とは関係のない活動、たとえばスポーツのような時間を確保しましょう。実際に、予定が広範囲にわたっている方が、能率的に学業を達成できます。社会人の皆さんの場合は、仕事にノートをとることもできます。常用語を略語で書く、重要な情報（またはキーワード）のみを記録する、情報をまとめるためにわかりやすい見出しを使う、実例を示すために写真や図表を使うというような方法を用いるとノートを取りやすくなります。資料中のキーポイントは蛍光マーカーをつけるか、下線を引きましょう。グループ内でクイズを出し合ったり、仲間同士で学び合うことができます。散歩に行く、自転車をこぐ、家族と一緒に過ごすといった気分転換をしましょう。適度な休憩を取ると、宿題のストレスは恐れるほどのものではないとわかり、すぐに取り組もうというやる気が湧いてきます。脳をリラックスさせる音楽を聞きましょう。</p>
正解要約	<p>落ち着いた精神状態になる。十分な睡眠をしっかり取る。予定を立てる。授業を受ける際に必ずノートをとる。勉強のグループを作る。適度に休憩を取る。</p>
UniLM	<p>落ち着いた気持ちになる。気晴らしの時間を設ける。休憩時には特別点を与える。</p>
JASS	<p>睡眠不足を引き起こす要因を理解しましょう。学校とは関係のない活動をしましょう。</p>
mT5	<p>勉強や課題をこなす。勉強や課題をこなす。</p>
Pointer-Generator	<p>宿題を持ちます。</p>