語彙と品質を考慮したデータ水増しの言語教育支援への適用

中町 礼文 [1]　西内 沙恵 [2]　浅原 正幸 [3] 佐藤 敏紀 [1]
[1]LINE 株式会社　[2] 北海道教育大学　[3] 国立国語研究所
{akifumi.nakamachi,toshinori.sato}@linecorp.com
nishiuchi.sae@a.hokkyodai.ac.jp
masayu-a@ninjal.ac.jp

概要

本研究では, テキスト生成のための基盤モデルを用いて, 語彙と品質を考慮したデータ水増しのシステムを提案した. 提案手法によるコーパス構築の応用可能性の検証として, 言語教育支援に向けたコーパス構築を行った. 本研究では, 提案手法を用いて, 語彙, 難易度, テキストの組からなる約 500 件程度の元データを, 4 倍以上に拡充した. また, 生成したデータのみによる文難易度推定器が, 人手により作成した推定器構築用の学習データを用いた推定器と同等以上の性能を持つことを確認し, 提案手法によるコーパスの自動構築の応用可能性を示した.

1　はじめに

自動採点や, 問題文の自動生成をはじめとした言語教育支援において, 良質な言語資源が必要不可欠である. しかし, 人手により, レベル・語彙が制御された例文データを作例する事は非常にコストが高い. そこで, 教科書や試験問題をはじめとする, 言語教育向けの高品質なリソースから文を抽出して, コーパスを作成することが考えられる. 高品質なリソースには限りがあり, また, 画一的なリソースからのデータ収集では, 個々の学習者に対しての適切なレベル・語彙を用いた例文の作例に対応できない. コーパス構築の効率化や, 個々の学習状況に応じた言語学習支援に向けて, 語彙や品質がある程度保証された言語資源の拡張手法の適用が考えられる.

言語資源の拡張手法として, データ水増し (Data Augumentation, DA) が広く研究されている. 特に, GPT-3 [1] をはじめとする, テキスト生成のための基盤モデルを用いた, スタイルを維持したテキスト生成でデータ水増しを行う手法 [2, 3] が存在する.

本研究では, GPT-3 と同等な形式の, 日本語のテキスト生成のための基盤モデルである HyperCLOVA[4] を用いて, 語彙と品質が考慮された文生成システムを提案する. 非母語話者の言語学習支援への適用を通じて, 提案手法のコーパス自動構築の応用可能性の検証を行う. テキストの品質については, 日本語能力試験 (JLPT) の 習熟度 (表 1) [1)] に基づく. まず, 語彙と品質がある程度保証されたコーパスの自動構築を行い, 語彙を適切に含めた例文が作成できることをを検証する. さらに, 作成したコーパスのみによる文難易度推定の性能を検証する. これらの検証を通じて, 記述文の自動採点や, 語彙・文法問題などの作問支援への応用可能性を確認する.

表 1: JLPT の習熟度の目安

習熟度	基準
N1	様々な話題の, 内容に深みのある読み物の内容や詳細な表現意図を理解できる. 自分の関心のある分野のレポートを記述できる.
N2	新聞や雑誌の記事、解説, 平易な評論など, 論旨が明解な文章が理解できる. 一般的な話題に関する読み物の話の流れや表現意図を理解できる. 理由を述べながら意見を書いたり, 学校, ホテルに問い合わせなどの連絡を記述できる.
N3	日常的な話題についての具体的な内容を表す文を理解できる. 新聞の見出しなどから情報の概要を理解できる. 難易度が高い文章でも, 言い換えられると要旨を理解できる. 知人に感謝・謝罪を伝える手紙を記述できる.
N4	基本的な語彙や漢字で書かれた, 日常生活で身近な話題の文章を理解できる. 日常の要件を伝える簡単なメモを記述できる. 依頼や誘いなどの簡単な文章が記述できる.
N5	平仮名, カタカナ, 基本的な漢字を理解できる. 定型的な語句, 文, 文章を理解できる. 書類に名前や国名などを記述できる. 簡単な自己紹介や短いお礼を記述できる.

[1)] 記述力の目安 https://www.jlpt.jp/about/candolist.html
読解力の基準 https://www.jlpt.jp/about/levelsummary.html

図 1: 難易度を考慮した例文生成システム

図 2: 語彙と難易度を考慮した例文生成システム

2 関連研究

言語処理において, データ水増しは広く研究されている. 単純な手法として, テキスト内の語彙を削除・置換・類義語辞書を用いた変換で水増しする EDA[5] がある. EDA は, 機械学習の精度改善のためのデータとしては利用できるが, 単純な単語操作のため, テキストの文法性が保障されない.

また, 基盤モデルを用いた判別タスク向けのデータ水増しとして, 小規模な生成型の基盤モデルをファインチューニングして例を作成し, 生成されたテキストを判別器でフィルタする手法 [6, 7, 8] が存在する. 小規模な生成型の基盤モデルのファインチューニングによる手法では, ファインチューニングのためにある程度コーパスが必要である.

ごく小規模な資源による文法性がある程度保たれたデータ水増しを行う手法として, 大規模な基盤モデルである GPT-3 を用いた Prompt ベースのデータ水増しがある. GPT3Mix[2] では, テキストとラベルからなる Prompt を与えて, テキストとラベルのペアを作成する. また, Sahu ら [3] は, あるラベルのテキストのみで構成した Prompt で, ラベルが考慮されたテキストを作成した. 既存の大規模な基盤モデルの Prompt に基づく手法は, 出力したい語彙を考慮することや, 出力したテキストの品質評価のフィルタリングなどは行われていない.

3 語彙と品質を考慮した例文生成

本研究では, 出力したい語彙を含む Prompt によるデータ生成と, 判別器を用いたフィルタリングを組み合わせることで, 語彙表現と品質がある程度保たれたデータ水増しのフレームワーク (Word-considered GPTDA) を提案する. 本研究では, GPT-3 と同様な形式の日本語の基盤モデルである HyperCLOVA[2] を用いる. システムの概要は, 図 2 に示したように, 出力したい難易度 N と単語を与え, 言語教育支援に向けた Prompt を作成する. 生成結果に対して, 難易度判別器や語彙の出力の有無によりテキストをフィルタすることで, 難易度や語彙の制約を満たしたテキストの生成を促す.

3.1 言語教育支援のための Prompt

本研究では, 2 種類の Prompt を提案する. まず, 文難易度判定に向けて, 図 3 で示したデータ水増しの Prompt を検証する. さらに, 語彙や文法問題の問題文の自動生成に向けて, ある語彙表現を含む例文を生成するため, 図 4 に示した, 出現語彙を考慮した Prompt を検証する. また, 図 3 の基礎 Prompt において, 最後の例文の先頭 4 文字を与える場合 Conditional 設定) と, 与えない場合を検証する.

3.2 Prompt のための例文データ

Prompt 内で用いる例文として, JLPT の 2018 年の試験問題[3] の全文を書き起こし, 例文データを作成した. さらに, 作成した難易度推定器の評価のために, 2012 年の試験問題 3) も同様に書き起こした. また, 単語を関連付けた Prompt に向けて, JLPT 向けの学習参考書に付属する単語リスト[4] から, 語彙難易度辞書を書き起こした. また, 語彙難易度辞書と 2018 年の試験問題を用いて, (難易度 N の語彙, 難易度 N の例文) となるような語彙と例文の組み合わせコーパスを作成した.

Prompt の例文の構成の比較として, 3 種類のサンプリングを検証する.

1. Random Level: 全ての難易度のテキスト集合からサンプリング.
2. Baseline: 難易度 N の Prompt の例文として, 難易度 N のテキスト集合からサンプリング.

[2] 本研究では 390 億パラメータからなる 39B モデルを用いた.

[3] 提供 日本語能力試験公式 ウェブサイト https://www.jlpt.jp/samples/sampleindex.html

[4] https://www.ask-books.com/jp/jlpt-try/try-wordlist/

```
例文を作成してください。

例文:{難易度 N の例文_1}
...
例文:{難易度 N の例文_n}
例文:{Optioanal: 難易度 N の例文の先頭 4 文字 (Conditional 設定)}
```

図 3: Base Prompt

```
与えられた単語を含む例文を作成してください。

単語:{難易度 N の単語 w_1}例文:{w_1 を含む難易度 N の例文_1}
...
単語:{難易度 N の単語 w_n}例文:{w_n を含む難易度 N_i の例文_n}
単語:{難易度 N の単語 w}例文:
```

図 4: Word-considered (W-c) Prompt

3. Word-considered: 語彙と難易度の組み合わせコーパスのうち, 難易度 N の事例からサンプリング.

それぞれのサンプリングでは, 4 件のデータを重複なくサンプリングする. 多様な表現を出力するため, 生成のたびにサンプリングを再実行する. 単語や文の件数を, 表 2 に示す.

3.3 難易度推定器を用いたフィルタリング

水増しデータの文難易度フィルタリングに向けて, 中町ら [9] と同様の手法で, JLPT の 2018 年の試験問題の書き起こしデータによる, 文難易度の推定器を作成した. JLPT の 2012 年の試験問題の書き起こしデータを用いて, 作成した推定器を評価した. 評価結果は表 2 に示す.

3.4 生成の後制御

no-filter 以外の全ての設定について, 生成例の推定難易度と, 指定した難易度が一致しない水増しデータを破棄する事後フィルタリングを行う. また, より強く難易度の制約を与えるため, データ作成中にフィルタリングを行い, 与えた難易度 N と, 生成例の推定難易度が一致しない場合再生成を行う, in-filtering 設定を検証する. in-filtering 設定において, 与えた語が出力されない場合や, 意図していない難易度のテキストが生成される場合がある. その場合, 最大 10 回の再生成を行い, 与えた語彙が含まれている例文のうち, 推定難易度と難易度が最小な候補を出力する. あるいは, 与えた語彙が含まれない場合, 再生成したテキストのうち, 推定難易度と難易度が最小な候補を出力する. さらに, 出力の語彙の含有を保証するため, 指定した語彙が含まれない生成例を破棄する word-filtering (wf) の設定を検証する.

4 実験

語彙と難易度が考慮された例文生成の検証として, 提案手法による出力が通常の GPTDA による出力よりも, 語彙や文難易度を制御できることを確認する. また, 水増しデータのみを用いた難易度推定器を作成し性能評価を行うことで, 難易度推定への応用可能性を検討する.

4.1 データ水増しの実験

全てのデータ水増しにおいて, 1 時間以下で 5,000 件のデータを生成している. 表 2 のように, ほぼ全ての手法で, 40%以上 (5,000 件の生成のうち 2,000 件以上) の割合で, 生成したテキストの難易度が考慮されている. 特に, 出力の先頭 4 文字を与えた Conditional GPTDA では, 事後フィルタリングを行っても 5,000 件中 4,758 件の出力が意図した難易度であった. しかし, 出力の多様性の観点からは, 内容の重複が多くなる傾向があった.

また, 出力の難易度制御能力の検証として, 全ての難易度の例文からランダムにサンプリングした Random Level 設定と, 難易度を考慮した通常の GPTDA を比較した. それぞれの設定による水増しデータに対して, 難易度推定器によるフィルタリングを行うと, 表 2 に示すように Random Level GPTDA では, 与えた難易度以外のデータが多く作成されていることが確認された. Random Level 設定と, Word-considered 設定を組み合わせた手法では, 出力の難易度制御が行えないことが確認された.

さらに, 出力の語彙制御について, 提案手法が, 難易度と語彙をどの程度考慮できているかを確認した. Word-considered の手法において, 難易度が制御された出力の 8 割以上は語彙も制御されている. 特に, Word-considered GPTDA (in-filtering, wf) では, 元の語彙と例文のデータ 587 件から生成した 5,000 件のデータのうち, 2,636 件のサンプルでは語彙と難易度が制御された.

4.2 水増しデータを用いた文難易度推定

作成したデータのみで難易度推定器を作成し, 2012 年の JLPT の試験問題から作成したデータで性能評価を行った. 実験は, 中町ら [9] による文難易度判別器と同様の実験設定で行った.

表 2 のように, 後処理を行わない GPTDA (no-filtered) や, 難易度を考慮しない Random Level GPTDA

表 2: データセットと難易度判定器の評価結果

Resource	#Samples	N1	N2	N3	N4	N5	Accuracy	F1	Kappa	Pearson
JLPT2012 年試験	1,154	300	259	248	200	147	-	-	-	-
JLPT2018 年試験	1,227	367	280	227	196	157	0.522	0.527	0.787	0.788
単語辞書	5,886	2,501	1,183	1,013	638	551	-	-	-	-
単語を含む 18 年試験問題	587	131	96	116	95	149	-	-	-	-
Random Level GPTDA	996	336	140	225	197	98	0.465	0.475	0.779	0.783
Baseline GPTDA (no-filter)	5,000	1,000	1,000	1,000	1,000	1,000	0.487	0.489	0.758	0.770
Baseline GPTDA	2396	609	393	320	593	481	0.529	0.529	0.783	0.785
Conditional Baseline GPTDA	4,758	936	953	949	950	970	**0.537**	**0.545**	0.794	0.794
Random Level Word-considered GPTDA	997	266	168	214	239	110	0.500	0.515	0.789	0.792
Random Level Word-considered GPTDA (wf)	834	236	148	177	193	80	0.508	0.512	0.802	0.777
Word-considered GPTDA	2,336	598	242	228	560	708	0.491	0.501	0.795	0.809
Word-considered GPTDA (wf)	1,940	517	189	188	449	597	0.522	0.540	**0.812**	**0.818**
Word-considered GPTDA (in-filtering)	3,017	731	354	380	690	862	0.483	0.492	0.806	0.816
Word considered GPTDA (in-filtering, wf)	2,636	646	326	333	577	754	0.506	0.544	0.808	0.817

※ no-filter 以外は, 5,000 件の水増しデータを難易度推定器でフィルタ.

表 3: 単語を考慮した GPTDA の生成例

Level	Word	Text
N1	現場	その場しのぎの対応ばかりする現場責任者への怒りから、部下たちはストライキを起こした。
N2	過去	この地域は以前から水害が多いため、過去に何度も堤防工事が行われてきた。
N3	利用者	この図書館はとても利用者が多いです。
N4	ロボット	私はロボットではありません。
N5	シャワー	シャワーをあびます。

表 4: 単語を考慮しない GPTDA の生成例

Method	Level	Text
GPTDA	N1	人々に木が身近な存在であることを意識させる。
	N2	公衆トイレの入り口に書かれている男女の絵のマーク。
	N3	前に読んだことを忘れて、同じ本をまた買ってしまったこと
	N4	まいばんテレビでニュースをけんがくしています。
	N5	これはそふとそぼのしゃしんです。
Conditional	N1	この物質は常温で固体なので、通常「氷」と呼ばれている。
	N2	ここは桜の名所である。
	N3	帰国することになりました。
	N4	「また駅前でビラ配りをしているよ。まったくしょうがないなあ。」
	N5	日本のいちばんとおいところはどこですか。

は, 全ての指標において試験問題で作成した推定器より性能が低かった. 水増しデータのみを使用したモデルの大部分が, 水増し元のデータのみを使用したものと同等かそれ以上の予測精度であった. 難易度の完全一致の観点においては, Accuracy や F1 の指標より, Conditional GPTDA が最も高精度であった. 一方で, 単語を考慮した GPTDA は, 誤判別した際でも, 大きく推定を誤る比率が減っているため, Kappa や Pearson などが高い傾向があった. また, 語彙が制御されているため, 語彙や文法表現などのための問題文生成としての活用も可能である.

5 終わりに

本研究では, HyperCLOVA を用いて生成的なデータ水増しを行い, 水増しデータをフィルタリングすることで, 言語教育支援に向けた語彙と品質を考慮できるデータ水増し手法を検証した. 実験の結果, 少数の元データからでも, 語彙と品質がある程度制御されたデータの自動生成を高速に行えた. 今後, 水増しデータを用いたフィルタを用いることで, フィルタのためのコーパスを必要としない手法の検証や, 生成・フィルタを再帰的に繰り返すことによる精度改善などを検証していく. また, 作成した言語教育向けコーパスに対して, 専門家による難易度のアノテーションを行い, 言語資源の公開を目指す. さらに, 言語教育支援に向けて, 語彙・文法問題などの演習問題の自動作成に取り組む.

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In **Advances in Neural Information**

Processing Systems, pp. 1877–1901, 2020.

[2] Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. In **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 2225–2239, 2021.

[3] Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data Augmentation for Intent Classification with Off-the-shelf Large Language Models. In **Proceedings of the 4th Workshop on NLP for Conversational AI**, pp. 47–57, 2022.

[4] Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, Na-Hyeon Ryu, Kang Min Yoo, Minsuk Chang, Soobin Suh, Sookyo In, Jinseong Park, Kyungduk Kim, Hiun Kim, Jisu Jeong, Yong Goo Yeo, Donghoon Ham, Dongju Park, Min Young Lee, Jaewook Kang, Inho Kang, Jung-Woo Ha, Woomyoung Park, and Nako Sung. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3405–3424, 2021.

[5] Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks". In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing**, pp. 6382–6388, 2019.

[6] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do Not Have Enough Data? Deep Learning to the Rescue! **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, No. 05, pp. 7383–7390, 2020.

[7] Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, pp. 4242–4255, 2022.

[8] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 9031–9041, 2020.

[9] 中町礼文, 佐藤俊紀, 西内沙恵, 浅原正幸, 奥村学. 日本語能力試験に基づく日本語文の難易度推定. 言語処理学会第 28 回年次大会発表論文集, pp. 658–663, 2022.

A GPTDA (no-filtered) の生成例

表 5: GPTDA (no-filtered) の生成例

Level	Pred	Text
N1	1	彼らは今まで見たこともないものを見るために旅に出たのだ。
	1	私たちはいつも彼ら (彼女ら) より一歩先んじている。
	1	彼は自分の意見を押し通そうとしたため、会議室からつまみ出された。
	2	お支払い方法 クレジットカード決済 (VISA/MasterCard/JCB/American Express)・銀行振込・コンビニ払い・Pay-easy
	1	彼は自分で決めた目標に向かって努力し続けています。
	2	「するべきだ」などの表現よりも「したほうがよい」や「すべきだ」などの表現の方が柔らかい印象を与える。
N2	1	私もいつかそんな風に旅をしてみたいものだ。
	2	このような状況では、「している」よりも「しているところだ」のほうが自然です。
	3	この商品は 10 個単位でのみ販売しております。
	2	この人たちは、みんなそれぞれ個性や才能を持っています。
	1	A : 1.In Japan, we call our daily life ”busy” because we have many (good) qualities that keep us “busy”.
	2	彼はそのことを知らないようだった。 A :「たり・たり」は並列ですから、二つ以上の物事を並べます。
N3	1	彼の行動力にはいつも驚かされる。
	1	私たちはこのような状況の中で、どのようにすれば良いかわからなかった。
	3	私はこの前、母から電話があった時、思わず「もしもし。」と言ってしまった。
	2	日本では多くの種類の野菜や果物が作られている。
	3	私はその時初めて自分の国以外にもたくさん国があることを知った。
	3	私はこの本を読んで、自分でもやってみようと思った。
N4	5	ぼくはサッカー選手になりたいです。
	3	この部屋の中で、私たちだけが知っていることだよ。
	4	その時はまだ日本語があまり上手じゃなかったので、よくわかりませんでした。
	5	きのうのよる、わたしはテレビを見ていました。
	4	この本はとてもおもしろいからおすすめです。
	5	私たちは明日から夏休みです。
N5	5	「はい、そうです。」
	4	私たちは明日、何時に起きますか。
	3	きのうの試合で、彼は 3 点シュートを決めました。
	5	日本語がじょうずだね。
	5	日本語の勉強はむずかしいですか。
	5	この本はとてもおもしろいです。

語彙や難易度推定器によるフィルタリングを全く行わない GPTDA の生成例を表 5 に示す. Level は, システムに入力で与えた出力したい難易度で, Pred は 2018 年試験問題で作成した難易度推定器による難易度フィルタリングの予測結果である.

事後フィルタリングでは, 難易度推定器によって, Level と Pred が一致している 16 件のみを出力する. 難易度を考慮する Prompt により, 一定程度, 通常の GPTDA でも, ある程度出力の難易度が考慮されており, さらに事後フィルタリングを行うことで, 難易度が揃っている.

また, N1 の”お支払い方法...”の出力や, N2 で”A: 1.In Japan...”のように, 試験問題として不適な生成が見られる. Conditional 設定で実際の試験問題の冒頭 4 文字を与えることでこのような生成を防ぐことができる.