SentencePiece の重複語入れ替えによる 日本語 T5 への言語モデル追加

高野 志歩 [1]　相馬 菜生 [2]　田村 みゆ [1]　梶浦 照乃 [1]　倉光 君郎 [2]
[1] 日本女子大学大学院 理学研究科　[2] 日本女子大学 理学部
m1816053ts@ug.jwu.ac.jp　kuramitsuk@fc.jwu.ac.jp

概要

SentencePiece は、テキストから教師なし学習によって語彙モデルを構築し、ニューラルネットワークに適した字句解析を提供する。現在の多くの大規模言語モデルの事前学習で採用され、SentencePiece は標準のひとつとなっている。本研究では、SentencePiece の語彙モデルから形態素に基づいて重複語を抽出し、新しい語彙と入れ替えることによる語彙の追加手法を提案する。これにより、大規模言語モデルに対して、数千語単位の語彙を追加する余地ができ、事前に学習されていない言語やドメインの語彙を含めた追加学習が可能になる。本論文では、2 つの日本語 T5 に対し、6000 語程度の Python 言語の語彙 (予約語/識別子) を追加した事前追加学習モデルを構築し、重複語入れ替えによる下流タスクの精度向上について論じる。

1　はじめに

大規模言語モデルは、事前に大量なテキストを学習することにより、機械翻訳や文書要約、感情分析など様々な下流タスクに適用できるモデルである。近年は、基盤モデルとも呼ばれ、自然言語処理タスクのみならず、様々な分野のニューラルネットワークに広く採用されている。

大規模言語モデルを下流タスクに適応させる際には、適応させたい下流タスクの少量のデータをモデルに与える。しかし、事前学習時に用いたデータのドメインと、下流タスクのドメインが大きく異なるとき、下流タスクにおいてモデルが十分な性能を発揮できない領域適応に課題がある。そこで、特定のドメインに特化した語彙を追加することにより、特定ドメインにおける下流タスクの精度が向上すると期待される。

しかし、大規模言語モデルは一般的に語彙数が固定である。そのため、語彙を追加するためには工夫が必要となる。

本研究では、大規模言語モデルの語彙を構築するとき、標準的に採用されている SentencePiece[1] に着目する。SentencePiece は、テキストから教師なし学習によって語彙モデルを構築し、構築された語彙が語彙数を減らし、学習時間を短縮する効果が知られている。一方、SentencePiece の語彙は、文字列の出現頻度に基づいて字句の区切りが決まるため、形態素的に重複した語彙が含まれる。我々は、このような重複語を新しい語彙と入れ替えることによる語彙の追加手法を提案する。

本論文では、提案手法を検証するため、SentencePiece の語彙モデルを採用した日本語 T5 に対し、3000 語程度の Python 言語の語彙 (予約語/識別子) を追加した。Python 言語の語彙は、日本語 T5 には含まれないため、このような語彙追加が可能になる。さらに、重複語の語彙入れ替えの下流タスクへの影響を調べるため、コード翻訳、コード要約という 2 種類の下流タスクの精度を比較した。

本論文では、大規模言語モデルにおける語彙の追加手法を提案し、実験結果とともに報告する。本論文の残りの構成は以下の通りである。2 節では、本研究で着目した重複語の予備調査について述べる。3 節では、提案する語彙の追加手法についてまとめる。4 節では、実験についてまとめる。5 節では、関連研究を概観し、6 節で本論文をまとめる。

2　SentencePiece と重複語

SentencePiece は、ニューラル言語処理向けのトークナイザである。本節では、SentencePiece について概説したのち、本研究で着目した SentencePiece の作成する語彙モデルに対し、どの程度重複語が含まれているか確認した予備調査の結果を述べる。

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

2.1 SentencePiece とは

SentencePiece は、言語モデルの学習データである生のテキストから最適な分割点を学習する教師なし単語分割システムである。MeCab[1] や janome[2] のような形態素に分割するトークナイザと異なり、サブワードを用いて分割を行う点が特徴的である。

サブワードは、その語が出現する頻度によって単語を分割する手法である。高頻度の単語は 1 単語として扱われ、低頻度の単語はより短い文字や文字列に分割される。日本語や中国語のように分かち書きをしない言語では、1 文を 1 単語として学習が行われる。

2.2 重複語とは

SentencePiece では、出現頻度によって語彙が決まるため、言語文法の語彙とは切れ目が異なることが生じる。例えば、「日本語は」「日本語」「は」の 3 語が語彙モデルに存在するとき、名詞の「日本語」と助詞の「は」から構成される「日本語は」が重複した存在だと捉えることができる。

このように、ある語を形態素に基づいて分割したとき、形態素が既に語彙モデルの中に存在し、重複して存在すると捉えられる場合には、ある語を重複語と考える。

2.3 重複語の調査

我々は予備調査として、3 つの大規模言語モデルにおける各語彙モデルの作成した語彙を分析した。調査したモデルは、mT5[2] と、2 つの日本語 T5 である。まず、Google 社が公開する mT5 は、マルチタスクモデル T5 がベースであり、日本語と英語を含む 101 言語に対応する。次に、日本語 T5 の 1 つは、Megagon Labs 社が公開する t5-base-japanese-web[3]（以下、日本語 T5(Meg)）である。日本語の Web テキストで事前学習された T5 モデルであり、日本語の語彙に特化している。最後に、日本語 T5 の 2 つ目は、園部勲氏が公開する t5-base-japanese-adapt[4] (以下、日本語 T5(Sno)) である。

複数パターンから見た各モデルにおける語彙数の違いを表 1 に示す。漢字を含む語、ひらがなを含む語、カタカナを含む語の 3 パターンを日本語語彙とし結果に着目すると、101 言語に対応するマルチリンガルなモデルである mT5 よりも、2 つの日本語 T5 のほうが多い語彙を持つことがわかる。また、日本語語彙以外の重複語として数字を含む語、記号を含む語の各語彙数を確認した。このとき記号を含む語には、図 1 に示すような約 70 種類の記号が該当する。例として、mT5 に含まれる日本語語彙と、数字を含む語、記号を含む語を表 2 に示す。

表 1 3 つのモデルにおける各パターンの語彙数

	mT5	日本語 T5 (Meg)	日本語 T5 (Sno)
全語彙数	250,112	32,100	31,741
漢字を含む	21,485	21,587	20,007
ひらがなを含む	6,443	10,549	9,718
カタカナを含む	2,843	5,223	5,293
数字を含む	16,473	829	465
記号を含む	10,679	559	635

。、． ・〜 () ”“ 【】 「」 『』 [] {} ♪〜 〖〗 '+()[]{}!
#$%&=~|`;:,.@?<>¥*/_゜ ±¤©☒+−×÷£€¢¬●´ ,·¶«

図 1 記号として判定する字句

さらに、各語彙モデルの日本語語彙に対し、形態素解析器 janome を用いた品詞パターンの分析を行った。品詞パターンとは、その語を構成する品詞の組み合わせを指し、1 つの品詞から成る語と、2 つ以上の品詞から成る語の大きく 2 種類に分けられる。このとき、1 つの品詞パターンに含まれるのは、名詞や動詞といった基盤 10 個の品詞とフィラーを含めた 11 個の品詞である。

本研究のアイディアは、予備調査の結果から考えられる重複語を削除し、代わりに別の語彙を追加することである。

3 重複語と語彙入れ替え

我々は、SentecePiece の作成した語彙モデルに含まれる重複語を取り除いて、別の言語の語彙を追加する。本節は、Python 語彙（予約語/識別子）などを実例として述べるが、Python 語彙の依存する部分は

表 2 mT5 に含まれる語彙の一部

漢字	ひらがな	カタカナ	数字	記号
日本橋	こんにちは	スプーン	157	********
可愛く	下人は	ベースの	2.52	●—●
の中に	明日から	このコメント	[2][3]	%91%

1) https://taku910.github.io/mecab/
2) https://mocobeta.github.io/janome/
3) https://huggingface.co/megagonlabs/t5-base-japanese-web
4) https://huggingface.co/sonoisa/t5-base-japanese-adapt

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

名・名	さいたま市
名・名・名	東京都生まれ
名・名・名・名	故障者リスト入り
名・助動	様々な
名・助動・助動	重要である

動・動	される
動・動・助動	された
動・助動	させる
動・助動・助動	ありません

助・助	には
連体・助	どのくらい
助動・助動	である

図 2　2 つ以上の品詞パターンのうち、重複語であっても語彙モデルに残す品詞パターン

ないため、置き換え対象の語彙は別言語の語彙に置き換えて適用可能である。

3.1　重複語の判定

2.3 節の予備調査の結果をもとに、品詞パターンによる重複語の判定を行った。本研究では、全ての重複語を削除するのではなく、品詞の構成から決定した語彙モデルに残す品詞パターン以外の重複語のみを削除する。

語彙モデルに残す品詞パターンは、計 23 個である。1 つの品詞パターンに含まれるのは、名詞や動詞といった基盤 10 個の品詞とフィラーを含めた 11 個の品詞と、図 2 に示す 12 パターンである。23 個の品詞パターンに該当しない語のうち、形態素的に重複した語彙が含まれるとき、該当の語を無意味な文字列に置き換えた。

3.2　数字や記号の重複除去

SentencePiece の作成する語彙モデルには、2.3 節の表 1 に示した通り、数字や記号を含む語彙が多く含まれている。

我々は、日本語語彙の重複語とは別に数値や記号の重複も取り除いた。例えば、数字は全て 1 文字を字句の単位として、2021 のような字句は、2 0 2 1 というように 4 つの字句からみなすようにした。　記号も 1 文字も字句の単位とした。

なお、数字の扱いに関しては、議論の余地があるが、最新の大規模言語モデル PaLM [3] の語彙構成に準じている。

3.3　Python 語彙の追加

最後に、語彙モデルへの語彙追加について簡単に述べる。

追加する語彙は、原理的には、ドメイン特化辞書などの任意の語彙を追加することができる。ただし、SentencePiece の語彙モデルは、出現頻度を持っているので、出現頻度に関する情報を持っていることが望ましい。

我々は、Python 言語の語彙（予約語/識別子）の追加を行う際は、Python ソースコードに対し、SentencePiece を使って語彙モデルを構築し、そこから追加する語彙を選定した。追加の先語彙モデルと新しい語彙の間では出現頻度の調整は行わず、それぞれの語彙が独立して出現すると仮定して、それぞれの出現頻度順序だけが保証されるように追加を行った。

4　実験

4.1　概要

我々は、重複語入れ替え手法の影響と効果を確かめるため、次の手順で実験を行う。

まず、異なる種類の SentencePiece の語彙モデルから構築された事前学習済み言語モデルを用意してベースラインとする。用意した言語モデルは、以下の通りである。

- **Sno**: 園部勲氏が公開する日本語 T5 モデル
- **Meg**: Megagon Lab 社が公開する日本語 T5 モデル
- **mT5**: Google 社が公開する多言語 T5 モデル [2](small)

これらのモデルから前節で述べた手法で重複語のみを削除したモデルを作る。続いて、重複語を削除して、その代わりにドメイン語彙を追加したモデルを作る。本実験では、前節で述べた通り、Python 語彙をドメイン語彙として追加した。最後に、下流タスクのドメインに合わせた追加学習 [4] を行ったモデルも用意した。追加学習で学習させたデータは、CodeSearchNet などの公開データセットから収集した Python コード (1GB) である。学習時のハイパーパラメータは、学習時のハイパーパラメータは、ピーク学習率 3e-4、エポック数を 5 回とした。

以上、我々の用意したモデルは次の 4 モデルとなる。

- ベースライン
- 重複語削除
- ドメイン語彙追加
- ドメイン追加学習

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 3　各タスクごとの実験結果

		コード生成			コード修正			エラー診断	
		EM	SynM	BLEU	EM	SynM	BLEU	BLEU	Leven
日本語 T5(Sno)	ベースライン	23.85	96.64	61.44	**65.15**	31.12	**86.41**	82.53	**90.30**
	＋重複語削除	**24.16**	96.94	**63.12**	54.36	26.56	78.46	74.39	83.20
	＋ドメイン語彙追加	22.63	**97.86**	61.48	58.51	**31.54**	79.98	**82.69**	90.26
	＋ドメイン追加学習	22.63	**97.86**	61.48	**65.15**	31.12	**86.41**	82.11	89.35
日本語 T5(Meg)	ベースライン	23.24	96.02	58.59	**62.66**	**34.44**	**87.13**	81.29	88.83
	＋重複語削除	**25.69**	**97.55**	**97.55**	59.34	29.05	78.79	80.85	**89.51**
	＋ドメイン語彙追加	16.51	96.94	56.46	62.24	30.29	79.34	**81.60**	89.22
mT5	ベースライン	8.26	92.36	46.66	53.53	19.09	82.36	80.65	88.38
	＋重複語削除	8.87	96.33	47.24	49.79	17.01	78.64	81.36	88.81
	＋ドメイン語彙追加	7.03	95.72	45.85	52.28	17.84	84.04	74.39	83.20
	＋ドメイン追加学習	**19.88**	**98.47**	**60.34**	**81.12**	**60.71**	**83.86**	**84.67**	**91.99**

4.2　下流タスクへの影響

我々は、下流タスクとして、Python コードのコード生成、コード修正、エラー診断を用意した。

- **コード生成**: プログラム意図を表す自然言語文を入力として受け取り、対応するコードを生成するタスク [5] である。
- **コード修正**: バグのあるコードを入力として、出力に正常なコードを生成するタスク [6] である。入出力はともにコードである。
- **エラー診断**: 英文のエラーメッセージとソースコードから日本語のエラー解決策 [7] を提示するタスクである。

これらのタスクを用意したモデルにおいて、学習させた結果を表 3 に示す。

5　関連研究

近年、Transformer をベースとした事前学習済みモデルが数多く提案され，転移学習やファインチューニングによって下流タスクに応用したときに優れた精度の向上を見せている。事前学習済みモデルは、大規模データセットを用いて事前学習され、下流タスクを定めず汎用的なモデルとして構築されることが多い。しかし、下流タスクのドメインが事前学習に用いたデータのドメインと大きく異なる場合、下流タスクでモデルが十分な性能を発揮できない領域適応の問題がある。

事前学習に用いるデータセットをドメインデータに絞った、専門領域に特化した言語モデルが提案され、汎用モデルと比較した時により高い精度でドメインタスクが解けることが確認された [8]。

Gururangan ら [9] は、事前学習済みモデルを特定ドメインに適応させるために、下流タスクドメインのデータで追加学習することで高い精度でドメインタスクを解く DAPT 手法を提案している。ここで、事前学習済みモデルが保持する語彙の中に、専門用語などドメインの語彙が含まれないため、語彙を拡張する必要がある。Yao ら [10] は、語彙を拡張しながらドメイン特化モデルを構築する手法として Adapt-and-Distill 手法を提案している。

6　むすびに

本研究は、大規模言語モデルの語彙に含まれる重複語を新しい語彙と入れ替えることによる語彙の追加を提案する。提案手法を検証するため、複数のモデルに対して (1) 不要語除去 (2) 不要語を Python 語彙に入れ替え (3)Python 語彙に入れ替え+追加学習と、3 つの条件を変えて比較による評価をおこなった。モデルに Python 語彙を追加することで、入出力で Python を用いるコードタスクで精度が上がるのではないかと考えていたが、実験結果として語彙の追加による精度向上は見られなかった。

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

謝辞

本研究を進めるにあたり，有意義なコメントをいただきました秋信有花氏 (NTT) と小原百々雅氏 (日本女子大学) と佐藤美唯氏 (日本女子大学) と高橋舞衣氏 (日本女子大学) に感謝いたします.

参考文献

[1] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[2] 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. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pages 483–498, Online, June 2021. Association for Computational Linguistics.

[3] Aakanksha Chowdhery et. al. Palm: Scaling language modeling with pathways. **CoRR**, abs/2204.02311, 2022.

[4] 梶浦 照乃, 小原 百々雅, 秋信 有花, and 倉光 君郎. 多言語 t5 への追加事前学習による python 言語モデルの構築. In **The 6th cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming（xSIG2022)**, 2022.

[5] Momoka Obara, Yuka Akinobu, Teruno Kajiura, Shiho Takano, and Kimio Kuramitsu. A preliminary report on novice programming with natural language translation. In **IFIP WCCE 2022: World Conference on Computers in Education**, 2022.

[6] 相馬 菜生, 梶浦 照乃, 高橋 舞衣, and 倉光 君郎. 大規模言語モデルへの事前追加学習による誤り訂正モデルのコードへの適用. In **第 21 回日本データベース学会年次大会（DEIM2023)**, 2023.

[7] 高橋 舞衣, 小原 百々雅, 相馬 菜生, and 倉光 君郎. 大規模言語モデルを応用した初学者に親切なエラーメッセージの実現. In **第 21 回日本データベース学会年次大会（DEIM2023)**, 2023.

[8] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, 36(4):1234–1240, 2020.

[9] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[10] Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. **arXiv preprint arXiv:2106.13474**, 2021.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).