E-045

日本語 LLM の Prompt Tuning による著者識別
Authorship Attribution Using Prompt-Tuned Japanese LLM

櫻井　航[1)]　浅野　雅人[1)]　井元　大輔[1)]
Wataru Sakurai　Masato Asano　Daisuke Imoto
本間　正勝[1)]　黒沢　健至[1)]
Masakatsu Honma　Kenji Kurosawa

1　はじめに

1.1　Prompt Tuning

大規模言語モデル (LLM) のようなパラメータ数の膨大なモデルを下流のタスクに用いる場合は、大規模データで事前学習したものをモデルのパラメータの一部または全体を微調整する手法 (Fine Tuning) が用いられることが多い。また、パラメータの全てを更新することなく、効率よく Fine Tuning を行う LoRA などの手法が考案されてきた。しかしながら、言語モデルはパラメータ数、学習するデータ数を爆発的に増加させることで性能が急速に向上することが知られており（言語モデルのスケーリング則）[1]、実際提案される LLM のパラメータ数はますます大きくなっているため、LLM のパラメータを直接更新するアプローチはますます難しくなっているといえる。その一方で、近年提案されるモデルの多くは汎用化が進んでおり、適切なプロンプトを与えることで、パラメータの更新を行わずに様々な下流のタスクに対応できるようになってきている。問題を解かせる際に与えるプロンプトの設計に関する知見は、十分に得られているとは言えないものの、様々な分野で実応用の検討が行われている。図 1 に示される Prompt Tuning は、プロンプトを学習ベースで設計するものであるが、埋め込みベクトルの数値を直接学習によって設計する研究において、学習するパラメータ数を LLM のパラメータ数の 1 パーセント以下にすることができるのみならず、使用する LLM のパラメータ数が 10^{10} 程度の大規模なものになると、その性能はモデルのパラメータを学習する場合と同等となるものとされている [2]。また、この研究では、このような手法は Zero-shot 性能が高い傾向にあり、タスクによってはモデルのパラメータを学習で更新する場合より高い性能を示すことも示唆されており、この点で実応用に適した手法であると考えられる。

1.2　著者識別

文章を与え、候補者の中から当該文章を書いた人間を明らかにする著者識別は、学術的には Authorship Attribution と呼ばれ様々な手法が提案されている。識別に用いられる指標に関しては、計量文献学のアプローチから、単語の頻度や品詞の n-gram 表現などの有効性が示唆されているが、候補者のデータセットや、識別したいデータセットの数が少ない場合の適応が難しく、また文章の内容や人間が直感的に感じ取る書き癖と十分にリンクした特徴量となっていない。そのため、深層学習モデルが学習した知見の活用が有効であると考えられ、BERT のファインチューニングなどクラス分類モデルを用いてテキスト分類問題の一種として取り組まれることもある。前述の言語モデルのスケーリング則の観点から、この著者識別に関しても、よりパラメータ数の大きなモデルを活かした検討を行う必要があるものの、日本語の著者識別に関しては、あまり検討が進んでいるとはいえない。また、法科学の分野などでの実応用を見据えると、手法の信頼性の観点から、学習データと実際に適用するデータの著作や文章のトピック、フォーマットといったものが異なる場合への適応時に、それらが同一のデータを使用した際と比較して精度が落ちることは望ましくない。そのため、前述の Prompt Tuning が有効である可能性がある。今回は、大規模言語モデルのプロンプトを学習で設計するような Prompt Tuning によって日本語文章の著者識別を行い、学習データと同一の著作から抽出された文章の場合・異なる著作から抽出された文章の場合の識別性能を確認した。

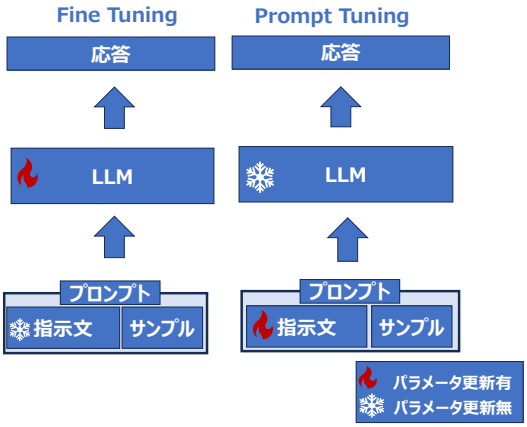

図 1　Fine Tuning と Prompt Tuning

2　手法

大規模言語モデルによるクラス分類を行う標準的な方法として、プロンプトとして与えたトークン列の入力の続きに、ラベルに対応する文字列を生成させるというものがある。今回は、これを応用して著者識別を行う。パラメータが θ の言語モデルに, プロンプトとして指示文のトークン列に対する埋め込みベクトルを $\boldsymbol{x} \in \mathbb{R}^{s \times d}$ (s, d はそれぞれ指示文のトークン数、各トークンの埋め込みベクトルの次元), サンプルのトークン列 $\boldsymbol{w} \in \mathbb{R}^{w}$ (w はサンプルのトークン数) を与え、プロンプトの次に生成される 1 トークンを $y([\boldsymbol{x};\boldsymbol{w}],\theta)$ とし、言語モデルによるこのトークンの生成確率を $P_\theta(y|[\boldsymbol{x};\boldsymbol{w}])$ と表記する。正解の著者 ID に対応するトークンを y_{ref}、生成される単語に対するターゲットとなる確率分布を $P(y)$ とすると, $P(y) = 1(if \quad y = y_{ref}), 0(if \quad y \neq y_{ref})$ となる。Prompt Tuning は、式 (1) で表される交差エントロピー誤差を最小化するような指示文の埋め込みベクトル $\boldsymbol{x}_{opt}$ を求め

1)　科学警察研究所
National Research Institute of Police Science

る問題であり、LLM を用いて、図 2 のようになる。

$$L = -\sum_{y \in V} P(y) ln(P_\theta(\boldsymbol{x}|\boldsymbol{w}, y)) = -\ln(P_\theta(\boldsymbol{x}|\boldsymbol{w}, y_{ref})) \quad (1)$$

また学習によって得られた $\boldsymbol{x_{opt}} \in \mathbb{R}^{s \times d}$ を用いた推論によって、確率 $P_\theta(y_{opt}|[\boldsymbol{x_{opt}}; \boldsymbol{w}])$ に従ってトークン y_{opt} を生成し著者 ID を決定する (図 2)。

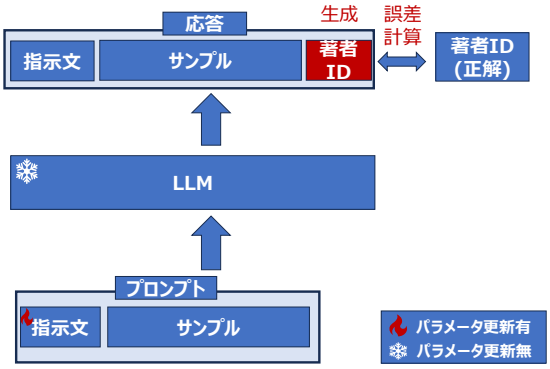

図 2　学習のイメージ

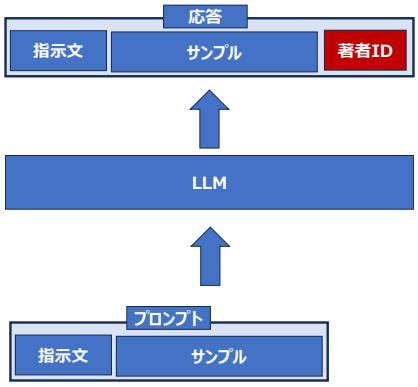

図 3　推論のイメージ

トークンの生成には確率に基づいて行われるため、その品質を向上させるためには適切なサンプリングを行う必要がある。そのために、各トークンの生成の際、温度付きソフトマックス関数を適用し確率分布における各トークンの生成確率の大小の差を小さくし、top-p サンプリングで確率が高い方からの累積確率が一定の値を超えた中から生成を行う。

3　実験

3.1　実験設定

実験に用いる LLM は、公開された 2 つのモデル (matsuo-weblab-10b[4],open-calm-7b[5]) を用いた。これらは、パラメータ数がそれぞれ 10^{10} および 10^7 規模の日本語対応モデルである。このモデルを 4bit 化して読み込み、計算時には 16bit にするなど計算の効率化を図った。また、プロンプトの指示文の初期値は” 以下の文章の著者 ID を出力せよ” の埋め込みベクトルにあたるものとし、図 4 のようにプロンプトの埋め込みベクトルを求める問題とした。

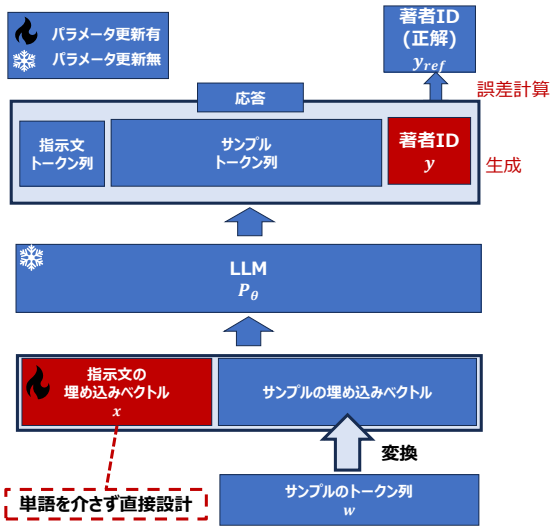

図 4　実験設定

実験には、国立国語研究所の現代日本語書き言葉均衡コーパス（BCCWJ）の 50 人の著者の作品から抽出した合計 10839 個のサンプルからなるデータセットを用いた（1 サンプル 1 パラグラフ）。トークン化の際には最大 150 トークンとなるように適宜 padding や truncation を行い、前述の指示文のトークンや著者 ID に相当するトークンと合わせて利用した。学習データに含まれる著作と同じ作品から抽出したデータ（以下、データセット S とする。1815 サンプル。）と、著者は同じであるが異なる著作から抽出したデータ（以下、データセット D とする。2159 サンプル。）の 2 つのデータを用いて性能比較を行った。それぞれのデータから識別したい人数 (クラス数) 分のサンプルを抜き出して、クラス数を変化させながら多クラス分類形式の著者識別を行った。学習データにおける識別する人数とサンプル数はそれぞれ図のとおりとなる。計算機には Quadro RTX 8000 を 4 枚搭載したワークステーションを利用した。その他、学習のハイパーパラメータは表 1 のとおりである。

表 1　ハイパーパラメータ

ハイパーパラメータ	値
batch size	32
training step	20
optimizer	AdamW(lr=1e-3)
scheduler	cosine annealing
warmup step	2

また、BERT によるクラス分類ベースのモデルに関しても、同じデータセットで学習・正答率の比較を行った。4 種類の事前学習済みモデル (bert-base-japanese-v3[6],xlm-roberta-base[7],bert-base-japanese-char-v3[8], bert-large-japanese-v2[9]) を LoRA でファインチューニングを行った。

3.2　実験結果

モデルの種類、学習する指示文のトークン数 (v)、識別する人数を変化させ、データセット S 及びデータセット D に対して、その識別性能の変化を確認したところ、図 5 及び図 6 のようになった。

どちらのモデルも識別する人数（クラス数）が 2 人の場合に正答率が最大となり、50 人の場合はほとんど識別できていないことが確認された。指示文のトークン数については、weblab-10b で 10、open-calm-7b で 2 の場合に最大となるなど、その傾向に違いがあることが確認された。その値は weblab-10b で 87.15% (データセット S),75.8% (データセット D),open-calm-7b で 71.2% (データセット S),68.1% (データセット D) となり、パラメータ数の大きいモデルである matsuo-weblab-10b のほうが高い正答率となった。これは、モデルのパラメータ数が大きいほど性能が高くなるという既往研究 [2] の結果と同じ傾向であった。データセットに応じた性能の違いについては、識別人数などの設定によって傾向に若干の違いはあるものの、異なる著作からなるデータセット D と同じ著作からなるデータセット S での性能に大きな違いはなかった。また、BERT ベースのモデルについては図 7 のとおりとなり、データセットごとに大きな違いは見られず、Prompt Tuning と比較して高い正答率となった。

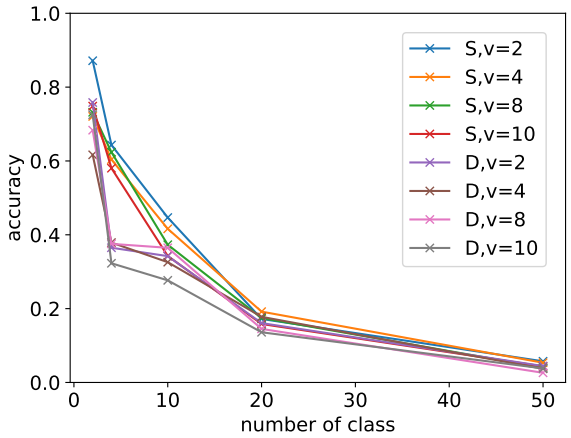

図 5　識別精度（weblab-10b）

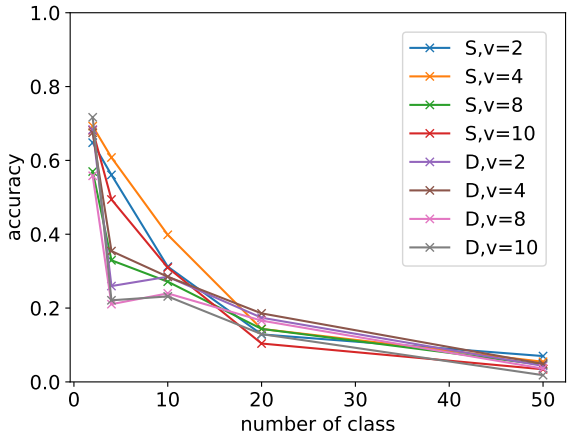

図 6　識別精度（open-calm-7b）

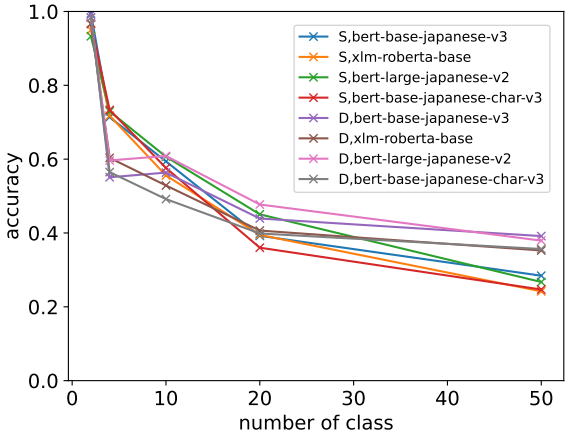

図 7　識別精度（BERT ベースのモデル）

4　まとめ・今後の発展

今回は著者識別問題への Prompt Tuning の適用に関する検討を行った。指示文の初期値やトークン数の検討が必要であるものの、クラス分類ベースの著者識別では精度は高くない結果となったが、識別人数やトークン数の設定によっては著作が違う場合にもその精度の低下があまり起こらなかった。また、BERT によるクラス分類ベースの著者識別と比較して、学習データと同じ著作・異なる著作で構成されるデータセットによる識別性能はいずれも高くない結果となった。そのため、今後は精度向上や問題の設計をより精緻に行うとともに、このモデルや学習後の埋め込みベクトルから得られる知見を深めるための解析を行う必要がある。また、LLM ベースのモデルにも、言語と画像を同時に扱うことのできるモデルが増加しているため、画像と文章を投稿できる SNS の投稿者の識別や、様々な問題に応用できる可能性がある。

謝辞

本研究は JSPS 科研費 JP22K21320 の助成を受けた。

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[2] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

[3] https://huggingface.co/matsuo-lab/weblab-10b. Accessed: 2024-06-13.

[4] https://huggingface.co/cyberagent/open-calm-7b. Accessed: 2024-06-13.

[5] https://huggingface.co/tohoku-nlp/bert-base-japanese-v3. Accessed: 2024-06-13.

[6] https://huggingface.co/FacebookAI/xlm-roberta-base. Accessed: 2024-06-13.

[7] https://huggingface.co/tohoku-nlp/bert-base-japanese-char-v3. Accessed: 2024-06-13.

[8] https://huggingface.co/tohoku-nlp/bert-large-japanese-v2. Accessed: 2024-06-13.