様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　６ 年　 ５ 月 ２３ 日現在

機関番号：３２６８６
研究種目：基盤研究(C)（一般）
研究期間：2021～2023
課題番号：２１Ｋ１２０１７

研究課題名（和文）コーパスの構成要素としての文書と単語列としての文書を架橋するトピックモデル

研究課題名（英文）Topic models bridging between documents as members composing a corpus and documents as sequences composed by words

研究代表者

正田　備也（Masada, Tomonari）

立教大学・人工知能科学研究科・教授

研究者番号：６０４１３９２８

交付決定額（研究期間全体）：（直接経費）　　3,100,000円

研究成果の概要（和文）：本研究の目的は、コーパスに特殊的なエンコーダとしてのトピックモデルに、汎用的なエンコーダとしての言語モデルを組み合わせ、トピック分析の質を向上させることだった。しかし、本研究開始後に急速に高性能化・高効率化した言語モデルをテキスト埋め込みに使い、様々なコーパスの分析を実施してみると、コーパスに特殊的なエンコーダは言語モデルのファインチューニングで十分実現できると分かった。トピックモデルに限らず、単語の出現頻度だけを基礎データとしてテキストマイニングを実現することにもはや技術的な意味はなく、今後は言語モデルの与える埋め込みをどう利用するかという課題に取り組むべきである。これが本研究の結論である。

研究成果の学術的意義や社会的意義
本研究の学術的意義は、従来ならミニバッチ式の変分推論で実践していたトピックモデリングを、事前学習済みの言語モデルを使ったテキスト埋め込みの利用により置き換える、定型的な手順を見つけた点にある。社会的意義は、変分推論の面倒を見なくてよい分、変分推論を十分に収束する前に止めてしまっている、ハイパーパラメータをチューニングしていない、等のミスが生じず、初心者でも失敗の可能性が低いトピック抽出を実現できる点にある。抽出されるトピックの質を上げるために言語モデルをファインチューニングする場合であっても、関連する技術情報がトピックモデルよりも豊富で見つけやすいため、初心者にも接近しやすい手順となっている。

研究成果の概要（英文）：The aim of our study was to combine a topic model as a domain-specific encoder with a deep learning-based language model as a general-purpose encoder, in order to improve the quality of topic analysis. However, after starting this research, language models have shown a remarkable progress in effectiveness and efficiency. By using them for text embeddings and analyzing various corpora, we have found that a domain-specific encoder can be realized only by fine-tuning a language model. Not only topic modeling but also any text mining based solely on word frequencies no longer have any technical importance. Here is the conclusion of our study: We should now focus on how to utilize the text embeddings provided by language models in order to improve the quality of topic analysis.

研究分野：機械学習

キーワード：機械学習　テキストマイニング　自然言語処理　トピックモデル　言語モデル

様　式　C－１９、F－１９－１（共通）

１．研究開始当初の背景

本研究の当初の目的は「コーパスに特殊的な文書エンコーダとしてのトピックモデルに、汎用性のある分散表現を与える文書エンコーダとしての言語モデルを組み合わせ、トピック抽出の質を向上させる」ことであった。この目的に設定したのは、この方向性の研究に当時は技術的意義があると考えたためである。

そして初年度には、トピックモデルの変分推論に Transformer 言語モデルを組み合わせる準備として、多層パーセプトロン(MLP)を用いた簡易的な単語埋め込みを使って変分推論を実現し、国際会議 SIMBig 2021 で発表した。この研究では、研究計画時に想定していなかった問題を解決しようとしていた。それは、変分オートエンコーダ(VAE)の枠組みをトピックモデルに利用すると component collapse と呼ばれる現象がどうしても邪魔をして事後分布のパラメータ推定に失敗するという問題だった。そこで、当該論文では VAE 自体を回避し、LDA の原論文[Blei+2003]にある変分下界(ELBO)をそのまま最大化する手法を提案した。その際、ELBO に現れるトークン毎の事後分布パラメータφを MLP の出力として得るという形で MLP を利用している。この研究の後には MLP を Transformer 言語モデルで置き換える予定だった。

ところが、この提案手法に問題が見つかった。トピック抽出に MLP が何も寄与していなかったのである。詳述すると、変分 EM アルゴリズムの E ステップ内の更新計算において MLP は初期値を与える役割を担っていたが、初期値は実は何でもよく、E ステップの更新をたった２回反復するだけで、変分推論全体としては十分うまくいくことが後から分かった。そのため、MLP はどのテキストにもほぼ同じ初期値を与えるようにだけ訓練されてしまっていた。この問題点は、すでに研究室の Web サイトで報告している(https://diversity-mining.jp/wp/?p=762)。

以上の問題点に気づいたのが、2022 年前半である。そこで、研究の方向性を変更することにした。よって、現在から振り返ると、実質的にはここまでが「研究開始当初の背景」にあたる。

２．研究の目的

研究の方向性を変更するにあたり、出発点、つまり、「コーパスの構成要素としての文書と単語列としての文書を架橋するトピックモデル」というテーマに立ち戻った。単語列としての文書のモデリングは、今や、事前学習済み言語モデルだけで実現できる。ならば、LDA のように単語の出現頻度を基礎データとする古典的な BoW 型のモデルを、事前学習済み言語モデルに組み合わせようとするアプローチ自体を放棄するべきだと考えるに至った。そこで、

> 事前学習済み言語モデルが与えるテキスト埋め込みだけを利用して、コーパス内での各文書の位置付けを明らかにするようなトピック抽出を実現する手順を確立すること。

を新たな目的として設定した。

新たな研究目的の設定には、以下の二つの経験が影響を与えている。一つは、gensim や scikit-learn の実装を使ったトピックモデリングに、機械学習の初心者が失敗するという事例を、複数回、身近に観察したこと。もう一つは、Google Colab 無料版で気軽に動かせる規模の事前学習済み言語モデルを使っても、LDA レベルのトピック抽出を実現するに十分な質の埋め込み(embedding)が得られることに気づいたこと。これら二点について、以下詳述する。

（１）初学者には扱いにくい LDA のミニバッチ変分推論

gensim や scikit-learn の変分推論は、大規模なデータセットに対応するため、ミニバッチ方式で実装されている。だが、機械学習の初心者は、計算が収束しているか見極めないまま計算を止めてしまう。特に gensim は 1 epoch で計算が止まるのがデフォルトである。これでは良いトピック抽出を実現できない。トピック数を様々に変えても、少ない方が良いという結論しか出ない。また、ハイパーパラメータ（scikit-learn で言えば doc_topic_prior と topic_word_prior）のチューニングにも不案内なため、初学者は往々にして不十分なトピック分析しかできない。

LDA のミニバッチ式の変分推論をうまくいかせる作業は、BERT のような事前学習済み言語モデルのファインチューニングをうまくいかせる作業に比べると、実は簡単である。にもかかわらず、失敗する。それは、後者のノウハウに関する情報のほうが、LDA のそれに比べてずっとポピュラーで、Web からも入手しやすい情報だからである。また、自然言語処理のトレンドが事前学習済み言語モデルの利用にますます移行する中、LDA のような深層学習以前の機械学習モデルを丁寧にチューニングして使うこと自体に、あまり高いモチベーションを感じられなくなっていることもあるのではないか。

にもかかわらず、テキスト集合からトピックを抽出したいという需要は依然としてある。このギャップのため、身近にいる学生が gensim や scikit-learn の実装を使ったトピック抽出に失敗する事例を何度か観察した。これらの学生も、流行の言語モデルには興味があるだろうから、そのトレンドに乗りつつ LDA 風のトピック分析ができる手順を確立すれば、テキスト集合の EDA を、より積極的に実践してくれるのではないかと、考え始めた。

その頃、BERTopic [Grootendorst+ arXiv:2203.05794]というツールが現れた。これを使えばLDA のようなトピックモデルを置き換えられるとも思ったが、後述のように、使いやすくはないことが分かり、トピック抽出の手順を自前で確立する必要性を、一層感じることとなった。

（２）事前学習済み言語モデルの高性能化と高効率化

2022 年には、Hugging Face の各種ライブラリの充実とともに、高性能な言語モデルを気軽に利用できるようになってきた。学生がコストの心配をせず使える計算機環境が Google Colab だけだったため、BERT 規模の言語モデルを中心に、テキスト埋め込みの質を Google Colab 上で探ってみたところ、EDA の用途に耐える質の埋め込みが、手軽に得られることが分かった。

特に、Sentence Transformers (https://sbert.net/)というライブラリを使うと、テキストの埋め込みの計算を、非常に少ない行数の Python コードで実現できることがわかった。当然、GPU の利用が前提なので、計算も非常に高速で、GPU への対応が難しい LDA の変分推論に比べると、初学者でも気軽に高性能かつ高効率な分析が実施できるメリットは大きいと考えた。

ところで、上述の BERTopic は、内部でこの Sentence Transformers を利用できる。ならば、BERTopic をそのまま使えばよく、トピック抽出のための言語モデルの活用について研究をする必要などないのではないか？・・・いや、その必要はあった、ということを以下で説明する。

３．研究の方法

代表者は、2022 年度から、BERT 系言語モデルを Sentence Transformers 等を介して活用し、大学の講義でも BERT のファインチューニングによるテキスト分類を紹介するなどしていた。

また、指導学生の研究でも積極的に BERT 系のモデルを利用し、2023 年度の研究まで含めれば、文脈化トピックモデルによるゼロショット学習[小林 2021 年度修論]、エッセイの自動採点[Sasaki+ ICDM Workshop 2022]、論文タイトルからの研究トレンド抽出[Heo 2023 年度修論]、小説の感情曲線の典型的なパターンの抽出[冨名腰 情報処理学会全国大会 2024]、アーティストごとの歌詞内容の相違の分析[池ヶ谷 2023 年度修論]などによって、BERT 系のモデルの有用性を十分に確認できた。

以上の取り組みを通じて、次のことが明らかになった。（１）LDA と同等かそれ以上の分析が、事前学習済み言語モデルによって得た埋め込みを分析するだけで実現できること。（２）コーパスに特殊的なテキストマイニングを実施するには、言語モデルとは別に LDA のような古典的な機械学習モデルにその都度コーパスを学習させるのではなく、そのコーパスで言語モデル自体をファインチューニングするほうが作業としてシンプルで、かつ上手くいくこと。

しかし、2023 年に入って言語モデルの世界に新しいトレンドが現れる。パラメータ数が数 B（数 billion＝数十億）以上の autoregressive な言語モデルの、オープンソース化である。この変化により、BERTopic を介して BERT 系のモデルを使ってトピックモデリングを実践することの技術的価値が、色褪せ始めたように思った。というのも、数 B の autoregressive 言語モデルは、BERT 規模の言語モデルとは別の使い方をする必要があると考えたからである。

この新しいタイプの言語モデルの典型例は Meta 社の Llama だが、新しい言語モデルは、BERT 系のモデルとは性格を異にするため、以下の点を新たに探究する必要が生じた。

（A）テキスト生成向けに事前学習された autoregressive な言語モデルであるという点で、masked 言語モデルである BERT とは振る舞いが異なる。BERT であれば、トークン列の先頭にある[CLS]という特殊なトークンに対応する出力を使うか、あるいは全出力の平均(mean pooling)を求めれば、良質な埋め込みを得られる。だが、テキスト生成向けの autoregressive な言語モデルからはどのような方法でテキスト埋め込みを得ればよいのか。

（B）Llama 系のモデルは、最も小さいものでもパラメータ数が数 B で、BERT のように全パラメータを気軽にファインチューニングできない。特殊なドメインのテキスト群のトピック分析を、初学者でもアクセスできる計算資源上で、どのように低コストで実現するのか。

（C）BERT 系のモデルは すでに日本語対応が進んでいたが、Llama 系の言語モデルは、日本語テキストの埋め込み目的の利用で、その有効性を発揮できるのか。

以上の問題意識を踏まえて、最終年度の 2023 年度の研究方法は、次のように設定された。

①数 B 規模のモデルを、BERT 並みの気軽さでトピック抽出に使うには、量子化が必須となる。コンシューマ向け GPU で短時間に実行できる量子化手法の有効性を調査する。

②少数のパラメータをファインチューニングするだけで、特殊なドメインへと言語モデルを適応させることのできる効率的な追加学習の手法の有効性を調査する。

③Llama 系言語モデルを使ってテキスト分析をする際、英語データだけでなく、日本語データも使う。日本語テキストを対象とすることで生じる問題と、その解決方法を探る。

次のセクションで、以上３点について、研究の成果を報告する。

４．研究成果

最終年度の研究成果を一つにまとめた Python コードを、この報告書の末尾の資料 1 に示した。このコードに、上の①～③の方法により進めた研究の成果が盛り込まれている。

①言語モデルの量子化

大規模言語モデル(LLM; large language model)の量子化には、大きく分けて二つの手法がある。一つは、比較的シンプルなアルゴリズムに基づく on-the-fly の手法。もう一つは、リファレンスとなるデータセットを用意し、それに対して特定の尺度のもとで「良い」テキスト処理ができるよう事前に実行する手法。後者の例としては GPTQ[Frantar+ arXiv:2210.17323]や Georgi Gerganov 氏による GGUF がある。本研究では、GPTQ の方を試していた。

しかし、おそらく実装技術向上のためと思われるが、例えば elyza/ELYZA-japanese-Llama-2-7b という Llama 系の 7B の LLM を on the fly で 4bit 量子化しても約 10 秒で済むようになった。初学者にとっての利用の手軽さも考慮すると、on-the-fly の量子化に軍配が上がる。自分が使いたい LLM を、誰かが GPTQ や GGUF で量子化してくれるのを待っていられないからである。末尾に添付した Python コードでも、NF4 量子化を on the fly で実行している。モデルを量子化された状態で読み込む時間は、パラメータがローカルな環境にダウンロードされていれば、RTX3080 搭載のパソコン（同等性能のパソコンは現在約 20 万円）上でも 10 秒以内である。

②言語モデルのファインチューニング

数 B 規模の LLM について、全パラメータをファインチューニングすることは、初学者が利用できる計算資源では不可能である。そこで、PEFT (https://huggingface.co/docs/peft)を利用することにする。これは Hugging Face で公開されているライブラリであるが、内容がますます充実してきている。本研究では利用実績の多い LoRA [Hu+ arXiv:2106.09685]を用いた。

例えば、Hugging Face のデータセット・ハブから取得できる shunk031/livedoor-news-corpus というニュース記事データセットに含まれる記事のタイトル約 6,000 件について、日本語対応 LLM である elyza/ELYZA-japanese-Llama-2-7b をファインチューニングするのに必要な時間は、添付の Python コードで計測して、RTX4090 上で約 5 分間、RTX3080 上でも約 12 分間である。

③言語モデルによる日本語テキストの埋め込み

日本語テキストを埋め込む際、まず問題になるのは、日本語であれ英語であれ、埋め込みベクトルをどのように取得するか、である。これについては、Hugging Face の Transformers ライブラリのなかにある AutoModelForSequenceClassification の実装を読むことにした。特に、今回使ったのが Llama 系のモデルであるので、LlamaForSequenceClassification の実装を読んだ。以下が forward メソッドの当該箇所である。

```
                sequence_lengths = torch.eq(input_ids, self.config.pad_token_id).int().argmax(-1) - 1
                sequence_lengths = sequence_lengths % input_ids.shape[-1]
                                                ... ...
        pooled_logits = logits[torch.arange(batch_size, device=logits.device), sequence_lengths]
```

これを読む限り、最後のトークンに対応する出力だけを利用していることが分かる。そのため、とりあえずこの方式を採用することにした。mean pooling も併用するとどうなるかは気になるが、本研究の期間内には調査できなかったので、今後の課題とする。

次に、対象とするテキストが日本語である場合に何か問題が発生するかについてであるが、埋め込みに関しては、日本語対応の elyza/ELYZA-japanese-Llama-2-7b をトピック抽出に使っていて、特に問題は感じなかった。しかし、同モデルでテキストを埋め込み、得られた埋め込みベクトルをクラスタリングした後で、各クラスタにラベル付けをしようとしたとき、問題が発生した。

ラベルに使う日本語の単語は、例えば spaCy の ja_core_news_sm モデルの形態素解析器を使って、分析対象のコーパスから抽出できる。しかし、この単語を単独で言語モデルに入力して、テキストと同じ空間に埋め込もうとすると、良い embedding が得られないことが分かった。英語データ（DBLP から取得した論文タイトル）で同様のトピック抽出を行ったときは、個々の英単語単独で言語モデルに入力して embedding を得ても、うまく EDA が実現できていた。

そこで、分析対象のコーパス上で TF-IDF を計算し、個々の単語について TF-IDF を規格化、これを重みとしてテキストの埋め込みを線形結合することで、単語毎の埋め込みを得た。そして、テキスト埋め込みのベクトル群を k-平均法でクラスタリングした後、クラスタの重心に近い単語を選ぶことで、クラスタにラベル付けができるようになった。（もちろん、ラベリングには GPT-4 などのプロプライエタリな言語モデルも使えるが、作業量に比例するコストがかかる。）

なお、テキスト埋め込みをクラスタリングする手法については、k-平均法がシンプルであるし、EDA 目的には十分な働きをすると考えられる。BERTopic では HDBSCAN を使っているが、埋め込みの質が高ければ、クラスタリングに関してそれほど特別な配慮は必要ないと考える。

以上が、本研究の成果である。ライブドアニュース・コーパスのタイトル部分だけを使ったトピック抽出の実行例を、この報告書の末尾の資料 2 に示す。この結果を得るのに RTX4090 上では 15 分程度、RTX3080 だと 30 分程度である。

以上が本研究の成果である。要約すれば、LLM を使ったトピックモデリングに関して

数十億パラメータの言語モデルを PEFT でファインチューニングする作業を含めても、短いテキストが１万件程度なら、１時間弱でトピック抽出を実行できる

ことが分かった。しかし、いくつか課題も残った。

（１）autoregressive な言語モデルから埋め込みベクトルを取得する方法は、末尾のトークンに対応する出力から得る以外にも方法はある。特に、mean pooling を併用することでテキストマイニングの質を上げることができるか、検討する余地がある。

（２）本研究では、トークン数が 512 トークン以下の比較的短いテキストでしか検証実験を実施できていない。長いテキスト（例えば、青空文庫の小説など）の場合、たとえ言語モデルの側で受け付けるトークン列の長さが今後より長くなるとしても、self-attention の性質上、計算コストが高くなり、EDA には向かなくなる。かといって、短い文へ分割し、文の埋め込みの平均を求めるのでは、元のテキストの内容を正確に反映できないだろう。長いテキストを bag of sentences とみなしてトピック抽出をするトピックモデリングの手法の提案が必要だろう。

（３）末尾に示した Python コードでは、テキストにあらかじめ付与されているカテゴリの情報を使って分類問題を解くことで、LLM をファインチューニングしている。トピックモデルでは、通常、カテゴリのレベルよりも細かい粒度で内容の多様性を分析するため、このファインチューニングの方法自体に問題はない。しかし、カテゴリのような上位レベルの意味情報がないテキスト集合に対して、ファインチューニングの際にどのような損失を利用するかは、検討の余地がある。素直には、次トークンの予測におけるクロスエントロピー損失を利用するのだろうが、例えば、個々のテキストがより長い文脈の一部を構成するようなテキストであれば、その文脈での近傍の文を正例、遠く離れた文を負例とする contrastive learning を行うのも一つの方法だろう。

（４）LLM はトークン列レベルのテキストのモデリングにおいて、BoW モデルである LDA よりもはるかに優れている。しかし、解釈性は低い。例えば、埋め込みをクラスタリングして得られたクラスタの一つに特定のテキストが属することに対して、そのテキストに含まれる個々のトークンがどの程度寄与しているかを示すことで、より深いトピック分析が可能となるだろう。この種の分析を可能にする XAI の手法として Integrated Gradients [Sundararajan+ arXiv:1703.01365]があるが、ハイパーパラメータの調整ができないので使いものにならないという報告もある[牧野+ 言語処理学会 2024]。適切な XAI 手法を見つける必要がある。

添付資料

資料1 事前学習済み言語モデルを LoRA でファインチューニングしてトピック抽出に使う Python コード

https://github.com/tomonari-masada/21K12017/blob/main/topic_modeling_with_LLM.ipynb

資料２ ライブドアニュース・コーパスのタイトル群から抽出した 20 のトピック

https://github.com/tomonari-masada/21K12017/blob/main/extracted_topics.txt

５．主な発表論文等

〔雑誌論文〕　計2件（うち査読付論文　2件／うち国際共著　0件／うちオープンアクセス　0件）

１．著者名 Sasaki Toru、Masada Tomonari	４．巻 1
２．論文標題 Sentence-BERT Distinguishes Good and Bad Essays in Cross-prompt Automated Essay Scoring	５．発行年 2022年
３．雑誌名 Proceedings of 2022 IEEE International Conference on Data Mining Workshops (ICDMW)	６．最初と最後の頁 274-281
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/ICDMW58026.2022.00045	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Tomonari MASADA	４．巻 1577
２．論文標題 AmLDA: A Non-VAE Neural Topic Model	５．発行年 2022年
３．雑誌名 Springer Communications in Computer and Information Science	６．最初と最後の頁 281～295
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-031-04447-2_19	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計3件（うち招待講演　0件／うち国際学会　2件）

１．発表者名 冨名腰哲，正田備也
２．発表標題 言語モデルを使用した日本文学の感情展開と分類
３．学会等名 情報処理学会 第86回全国大会
４．発表年 2024年

１．発表者名 Toru Sasaki
２．発表標題 Sentence-BERT Distinguishes Good and Bad Essays in Cross-prompt Automated Essay Scoring
３．学会等名 The 1st Workshop on Data Mining in Learning Science (at the 22nd IEEE International Conference on Data Mining, ICDM2022)（国際学会）
４．発表年 2022年

１．発表者名 正田備也
２．発表標題 AmLDA: A Non-VAE Neural Topic Model
３．学会等名 8th International Conference on Information Management and Big Data (SIMBig 2021)（国際学会）
４．発表年 2021年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関