

大規模言語モデルを用いた 大学シラバスのための エージェントシステム

椎谷風香[†] 武田敦志^{††}

現在、多くの資料やマニュアルは膨大な情報量を含んでおり、必要な情報を見つけ出すためには相当な時間と労力を要することが多い。この問題は、大学生が履修登録を行う際にも共通しており、シラバスから自分に合った講義を見つけるのは容易ではない。そこで本研究では、大規模言語モデル (LLM) をエージェントとして活用し、利用者の回答に基づいて最適な講義を自動的に提案するシステムを開発した。実証実験を通じて、このシステムが利用者のニーズに応じた講義を適切に選定し、効率的に提案できることを確認した。今後は、大学シラバス以外にもマニュアルやその他の外部資料を対象とした応用展開を検討し、様々な分野での利用者支援に活用することを目指している。

Agent System for University Syllabus using Large Language Models

Fuka Shiya[†] and Atsushi Takeda^{††}

Currently, many materials and manuals contain a huge amount of information, and it often takes considerable time and effort to find the necessary information. This problem is also common when university students register for courses, and it is not easy for them to find lectures that suit them from the syllabus. In this study, we developed a system that uses a large-scale language model (LLM) as an agent to automatically suggest the most appropriate lecture based on the user's answers. Through demonstration experiments, we confirmed that the system can select lectures that meet the user's needs appropriately and propose them efficiently. In the future, we intend to consider applying the system to manuals and other external materials in addition to university syllabi, and to use the system to support users in a variety of fields.

[†] 東北学院大学 教養学部情報科学科
Faculty of Liberal Arts, Tohoku Gakuin University
^{††} 東北学院大学 情報学部データサイエンス学科
Faculty of Informatics, Tohoku Gakuin University

1. はじめに

現在、多くの資料やマニュアルは膨大な情報量を含んでおり、目的の情報を探し出すまでに時間と手間がかかることが多い。また、資料の形式や内容に依存し、特定知識やスキルを持たないと効率的に資料を閲覧、参照が出来ないことがある。このため、特に時間の制約がある場合や、専門的な知識が必要な場合、利用者の求める情報を迅速に取得することが難しく、資料活用のハードルが高くなっている。このような状況は、学生にとっても大きな問題となる。具体的には、資料として膨大な量のシラバスを見比べる必要のある履修登録が挙げられる。大学生活における履修登録は学生にとって重要なプロセスである。しかし、多くの講義を比較し、最終的には履修しない講義も含めたシラバスの内容をすべて確認するというのは手間のかかる作業である。また、学生が自身のニーズに合った講義を探すためには、複数の講義を比較し、選定する必要があるためさらに時間を要する。このような現状から、学生が効率的に資料の内容を確認し、求めている情報の提案や質問を通じた新たな発見を促進するような方法が必要とされている。

この方法を実現するため、近年身近な技術となった ChatGPT の基本技術である LLM(大規模言語モデル)に着目した。LLM は大量のデータとディープラーニング技術によって構築された言語モデルである。LLM の出力はモデルの持つ知識の範囲内で行われるため、非公開のデータや専門知識に対する性能が大幅に低下してしまうという課題点がある。この問題を解決するために、事前学習済みの LLM モデルに目的に合わせた学習データを与えて追加学習させる技術であるファインチューニングが主に行われている。ファインチューニングは LLM の機能向上に非常に効果的だが、ファインチューニングを行うためにはかなりの時間と技術が必要となる。そこで、ファインチューニングを行わずに LLM を用いることで情報システムを参照する上でのインターフェイスとして用いる仕組みが必要とされている。

そこで本稿では、外部資料を参照する際に LLM をエージェントとして利用することで、利用者の求める内容を理解しやすい形で抽出し、利用者が必要とする情報にアクセスしやすくするためのシステムを提案する。この目的のために Meta 社が発表した大規模言語モデルである llama3 を用いて外部システムの内容の要約とそれを基にした利用者への質問を生成、利用者の回答を基に提案を行うシステムの構築を行った。資料を参照する際に、LLM を用いて利用者が求める情報に対する提案や要約を行うことで資料を参照する際に効率よく求める情報を探し出すことが可能になると推測す

る。

本論文の構成は、次のとおりである。2 章では本研究で提案するシステムと関連研究の比較を行う。3 章では本システムの事前準備及びシステム動作の流れを提示し、4 章で開発に使用した環境、前処理、作成手順を述べ、5 章で本研究の実験の結果を示す。最後に、6 章で結果の考察路今後の課題について述べる。

2. 関連研究

講義提案における既存手法としては、個人の特性を考慮した履修登録支援システム [1] が提案されている。このシステムでは利用者の成績情報を利用し、分野ごとに利用者の GPA を用いて数値を算出している。この手法では利用者が GPA を基に得意とされる分野を分析し、その結果良い評価をとれる講義が提案される。そのため、利用者にとって最適な履修登録を促すことが可能になる。一方、興味関心の考慮は行われていないため、利用者が学びたい内容の講義を提案できるとは限らないということが考えられる。

LLM は、要約やテキストの生成、ユーザーとの対話といったタスクに優れている一方で、専門的な知識を必要とする作業や、最新のデータに基づいた推論に関しては限界がある。これは、LLM が過去のデータに基づいて学習しており、最新の情報や非公開のデータが学習範囲に含まれていない場合が多いためである。その結果、実際の事実と異なる内容を出力してしまう「ハルシネーション」と呼ばれる現象が発生することがある [2][3]。このような問題に対処するため、LLM の内部には存在しない情報を補完する手段として、外部資料を参照させられないかと考えた。例えば、大学のシラバスなど、LLM が学習していないデータを外部の資料として参照させることで、より正確な情報を提供することが可能になる。さらに、これにより、利用者が求める情報を外部資料から自動的に探索・抽出し、提示するためのエージェントとして LLM を活用できるのではないかと考えた。このアプローチにより、LLM が苦手とする専門的な分野や最新情報に対しても、適切に対応することが出来ないかと考えられる。

本論文で提案するシステムでは、LLM をエージェントとして利用し、外部システムを参照させることで利用者の興味関心に合わせた提案をすることが可能になる。また、講義の提案とともに生成した要約を提示することで利用者が大まかな内容を理解することが可能になる。また、日本語対応のモデルを利用することでプロンプトの指示を支援言語で行うことが出来るという点も利点として挙げられる。



図 1 LLM を用いたエージェントシステムの関連図

3. LLM を用いたエージェントシステムの提案

現状、利用者が資料を直接参照する際、資料の量が多い場合は必要な箇所を抽出するのに時間がかかる。また、資料の媒体によっては参照するために特定の知識が必要になるなどの問題がある。本研究では現状の問題を解決するため、LLM(大規模言語モデル)をエージェントとして用いることを提案する。本研究で提案するシステムと利用者、外部システムとの関係を図 1 に示す。本章では LLM をエージェントとして動作させるための事前処理について述べ、続いて提案するシステムの流れについて述べる。

3.1 システムの実行手順

本システムでは外部システムとなる資料をテキストファイルとして抽出し、その内容を利用して LLM をエージェントとして動作させる。抽出されたテキストファイルを処理せずに利用すると参照するデータ量が膨大になってしまうため、LLM を用いてテキストファイルの内容から 200 字程度の要約を作成し、出力されたものを外部システムとして用いる。複数の質問から利用者の求める情報を提案することを目的とするため、要約されたテキストファイルを基に利用者の求める情報を絞り込むための質問を複数作成する。作成された質問を利用者に提示し、回答を基に LLM を用いて求める情報を含む資料を探し出し、簡単な概要とともに提案する。

3.2 事前処理

本システムで利用者に対して質問、提案を行う前にデータ量の減少を目的として外部システムの文章の要約を行う。初めに外部システムの内容をテキストファイルとして抽出する。続いて、抽出されたテキストファイルを LLM で要約する。プロンプトは要点をまとめて出力するという指示を与える。最後に、要約を行ったテキストファイル名と要約された文章を基にテキストファイル名:要約という形で一行ずつテキストファイルに追記する。事前処理の流れを図 2 に示す。

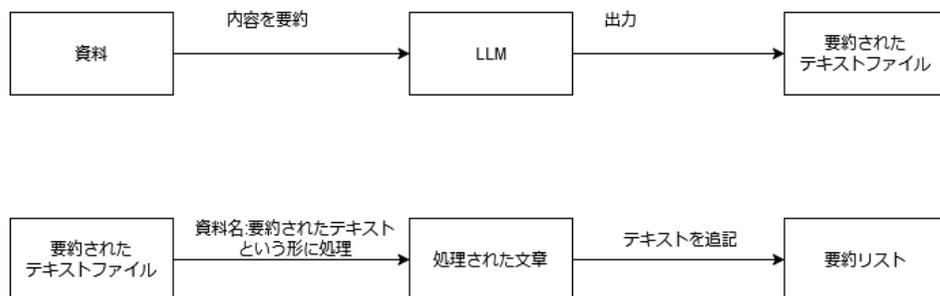


図 2 事前処理の流れ

3.3 システム

まず初めに、事前処理で出力されたテキストファイル名と要約を基に利用者にとって興味があると推測されるものを絞り込むための質問を LLM で出力させる。この際、プロンプト上では出力される質問の個数と質問の内容は利用者はいもしくはいいえで答えられるものと指定する。また、生成された文章から質問文のみを抽出するため質問の頭に番号を振るよう指示する。これは LLM による出力の内容に指定した質問以外の文章も同時に出力された場合にも質問だけを抽出するために行う。次に作成された文章内から質問の頭に振られた数字を基に質問のみを抽出し、リスト化する。この時、同時に要約の一覧から要約されたファイル名のみを抽出し、同様にリスト化する。この動作で作成された質問のリストを基に利用者には質問を行う。利用者は質問の内容に対してはいもしくはいいえの回答を入力し、その回答と要約のテキストファイル、ファイル名のみをリストを参照し、利用者が最も興味のあると推測されるものをリストから提案させる。ファイル名のみをリスト内から提案を行うよう指示することで、ハルシネーションの発生によって存在しないファイルを提案してしまうことを防止する。これらを参照して LLM にファイル名と提案する理由となった内容を簡単な概要として出力させる。システム実行の流れを図 3 に示す。

4. 実験と結果

LLM の中で GPT-4 に匹敵する性能を持つモデルである Llama3[4]を利用し、研究を行った。プロンプト上の指示は日本語で行うため、日本語対応モデルである elyza/Llama-3-ELYZA-JP-8B-GGUF を使用する。また、本研究では Python を用いて開発を行ったため、Python を用いて Llama を動作させる Llama-cpp というツールを用いる。さらに、外部システムとして東北学院大学の大学要覧内の科目別 pdf ファイルを

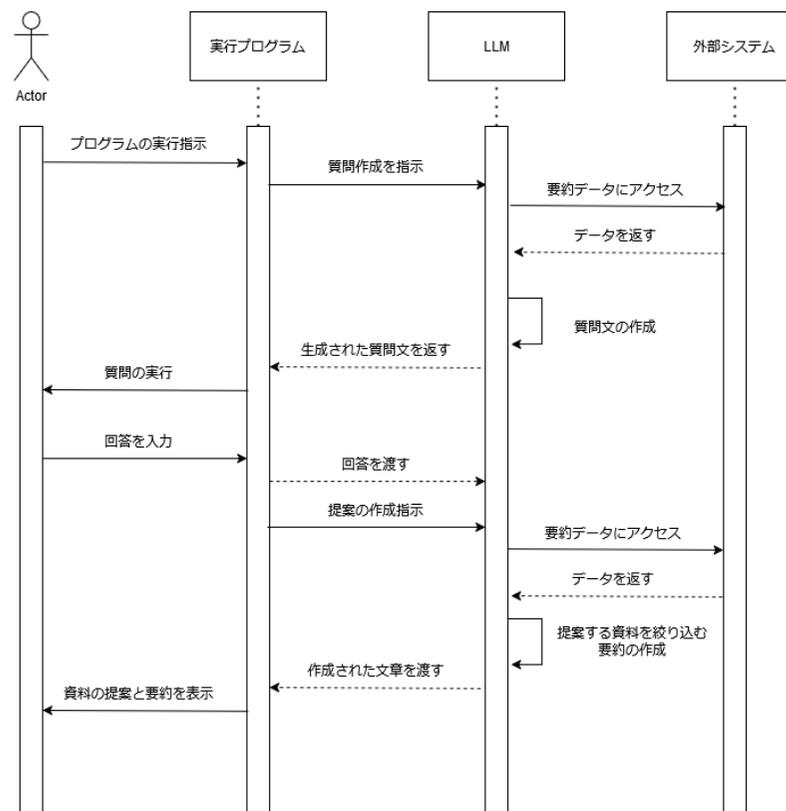


図 3 システム実行の流れ

用いる。事前処理として pdf ファイル内の文章の要約を行うために PDFplumber という Python ライブラリを用いる。PDFplumber は PDF ファイル内から図形や文字列などを抽出するためのライブラリである。これらを用いて学生が利用することを想定し、利用者が学びたい内容のシラバスを探し出し、要約とともに提案するシステムを開発する。

4.1 事前処理の設定

本研究では使用する外部システムとして東北学院大学のシラバスを用いる。このシラバスは古い pdf の形式であるため、文章内の強調をする際に文字をずらして二重打ちをすることで強調を表現する手法がとられている。そのため、PDFplumber を用いて抽出を行った際に文章内で同じ文字が連続で抽出された場合は重複した文字を削除するという処理を行った。

4.2 プロンプト上の制約

システムを動作させる際に求めた内容の出力が行われるよう、プロンプト上でいくつかの制約を設定した。具体的に設定した制約は以下のとおりである。

(1) 出力の文字数制限

LLM はテキストデータをトークン化して処理を行う。トークンごとに情報を理解し、予測と生成を行っている。英語は 1 単語あたり 1 トークンであるのに対して、日本語は 1 文字あたり 1~3 トークンと英語と比べてトークン数が大きくなる傾向がある。トークンによって LLM がより詳細に情報を理解できるようになるが、トークン数が大きくなりすぎると処理にかなりの時間を要する場合や最大トークン数を超過してしまい正しい処理が行えなくなってしまうという問題点がある。そこで、外部システムの要約を行う際に 200 字以内という制限を設けた。

(2) 禁止事項の設定

本システムでは、外部システムとして大学のシラバスを利用し、LLM をエージェントとして要約を行った。実験を通して、要約結果に授業名ではなく授業コードのみが出力されるケースが発生したため、適切な要約を得るためにプロンプト上で制約を設けた。具体的には、授業コードの出力を防ぎ、授業名を明示的に出力させるよう指示を追加した。この調整により、シラバスの内容をより正確に反映した要約を得ることが可能となった。さらに、シラバスの要約から質問を作成する際、具体的な講義名を提示し、この講義を受けたいかといった直接的な質問が生成されることがあった。このような直接的な提案を避け、より柔軟な提案を行うため講義名をそのまま質問に含めることをプロンプト上で禁止した。これによって利用者の意図に基づいた講義選択がより自然に行えるようになった。

5. 実証結果

Llama-cpp-python を用いて Llama3 の動作環境を実装し、検証を行った。OS は Windows 11, 言語は python, ライブラリとして PDFplumber を用いて実装した。また、LLM として Meta 社が開発した Llama3 をベースに追加学習を行い開発された日本語 LLM である elyza/Llama-3-ELYZA-JP-8B-GGUF を使用する。今回の実証では自然言語

	元の文字数	①要約指示無し	②要約を指示	③要約・文字数を指示
資料 1	2761	544	487	221
資料 2	2012	595	466	230
資料 3	2273	561	433	219

※各10回ずつ生成を行い、出力された文章の文字数から平均を算出

【プロンプトの内容】

①要約指示無し	以下の内容から、講義内容を知りたいです。 講義の特徴や内容を箇条書きでなく文章で教えてください。講義内容：{pdf_text}
②要約を指示	以下の内容から、講義内容の要約を知りたいです。 講義の特徴や内容を箇条書きでなく文章で教えてください。講義内容：{pdf_text}
③要約・字数を指示	以下の内容から、講義内容の要約を知りたいです。要約は200文字前後にしてください。 講義の特徴や内容を箇条書きでなく文章で教えてください。講義内容：{pdf_text}

※{pdf_text}で読み取りを行ったテキストファイルを参照

図 4 プロンプトごとの出力結果

を用いてプロンプトの内容を指定するため、前述した制約を用いて出力の結果を比較した。図 4 にプロンプト上の生成される文章の文字数に対する制約の違いによる量の差を示す。内容の違うプロンプトごとに 10 回生成を行い、生成された文章の平均文字数を示す。また、利用したプロンプトも同時に示す。

プロンプトの変更によって出力文字数を制御できるか検証するため、内容を出力するという指示と文字数に制限をかけずに要約を指示し、要約を 200 字前後で行うといった指示の三種類プロンプトを作成し、各 10 回の出力を比較した。その結果、図 4 よりプロンプトの指示によって一定程度文字数の制御が可能であることが確認できた。しかし、各資料の 200 字前後での出力において、最小文字数は 180 字、最大文字数は 289 字であり、出力に 100 字程度の差が生じた。この結果から、LLM はプロンプトの指示に基づいて大まかな範囲で文字数を調整することは可能であるが、正確な文字数制限を実現するには限界があることがわかった。

次に、要約された内容を基に利用者に対する質問を作成した。作成を行うプロンプトでは質問の個数と出力の形式に指示を与えた。質問の個数を 8 個、出力される質問はいはいはいえで答えられるもの、質問の前に番号を振るという指示を与えた結果、以下のような結果が得られた。質問の作成に利用した要約の一部を図 5 に、生成された質問を図 6 に示す。

要約リスト.txt (文章一部省略)
プログラミング初級: この講義「プログラミング初級」は、テキスト型プログラミング言語の Python を使用して、基本的なプログラミングの理解を深める実習形式の授業です。…成績評価は、毎回のプログラム作成課題とレポートの提出をもとに評価されます。
共生社会と倫理: この講義では、共生社会と倫理を学ぶことができます。…
哲学: この講義は「哲学」を学ぶ授業で、人間の「生」のあり方について考えることが目標です。…メディア授業科目としてオンデマンドで行うため、自宅で受講することができます。…成績評価は、確認テストの総合点で行われます。
文学: この講義では、異界を描いた日本の近代文学を読み解く。…
キリスト教と倫理: 今回の講義は、キリスト教と倫理をテーマに、現代社会が直面する様々な倫理的問題をキリスト教の視点から考えることを目指します。…

図 5 シラバスの要約一覧 (一部省略)

質問リスト.txt
1. 文系の講義を受講したいと思いますか？
2. 自宅で受講できる講義を希望していますか？
3. プログラミングやコードを書くことには興味はありますか？
4. 小説や文学作品に興味はありますか？
5. キリスト教や宗教に関する内容を学ぶことに抵抗はありませんか？
6. 社会や文化に関する内容を学ぶことに興味はありますか？
7. 古典的な思想や哲学に触れてみたいと思いますか？
8. 小テストや期末テストのある講義を希望していますか？

図 6 要約から作成された質問

生成された質問に対して回答を行い、もっとも関連のある講義として出力された内容を図 7、図 8 に示す。実行結果から、回答に基づいて適切な講義が提案できることが確認された。例えば、プログラミングやコードに興味がある利用者には「プログラミング初級」が提案され、古典的な思想や哲学に関心がある利用者には「哲学」講義が推奨された。このように、各質問への回答と講義内容の関連性を適切に反映した提案が行われ、講義選択をサポートするシステムの有用性が示された。ただし、現段階では、回答に応じて講義の関連性が提示されるのみであり、講義の具体的な内容や詳細な特徴までは十分に提示されていないため、利用者が講義の全体像を把握しにくい

「はい」と回答した講義
3. プログラミングやコードを書くことには興味はありますか？
8. 小テストや期末テストのある講義を希望していますか？



出力された内容
最も関連性の高い講義は「プログラミング初級」です。なぜなら、質問者はプログラミングやコードを書くことに対して興味を示しているからです。講義「プログラミング初級」は、Python を使用した基本的なプログラミングを学ぶことができるため、質問者に最も関連性の高い講義と言えます。

図 7 出力内容 (1)

「はい」と回答した講義
1. 文系の講義を受講したいと思いますか？
2. 自宅で受講できる講義を希望していますか？
7. 古典的な思想や哲学に触れてみたいと思いますか？
8. 小テストや期末テストのある講義を希望していますか？



出力された内容
講義リストから提案する最も関連性の高い講義は「古典的な思想や哲学に触れてみたいと思いますか？」の回答が「はい」であることから、「哲学」講義です。古典的な思想や哲学に触れることができるこの講義は、自宅で受講することが可能で、メディア授業科目としてオンデマンドで行うため、受講しやすい環境が整えられています。小テストや期末テストのある講義を希望していることから、成績評価が確認テストの総合点で行われる「哲学」講義は、希望に合致しています。以上の点から、「哲学」講義が最も関連性の高い講義と言えます。

図 8 出力内容 (2)

可能性がある。例えば、講義で取り扱うトピックの深さや、授業の進行方法、使用教材の詳細など、より具体的な情報が提示されることで、利用者はさらに講義を選択しやすくなるだろう。このため、今後はプロンプトを改良し、提案された講義の具体的

な内容や特徴も併せて提示できるような出力を目指す必要があると考えられる。

6. まとめ

本論文では、LLM エージェントとして用いることで外部システムとなる資料内から利用者の求める内容をわかりやすい形で抽出し、求める情報にアクセスしやすくするためのシステムを提案した。今回の実証結果から、LLM を用いて適切な講義提案を行うことができた。ユーザーの回答に基づいて、関連性の高い講義を効率的に絞り込むことが確認された。また、プロンプトの改善により、さらにわかりやすく具体的な講義提案を行うことが可能であることが示唆された。今後は、利用者が講義の選択を行う際に、講義の内容や特徴をより明確に提示することができるようにプロンプトの改良を進めることが求められるだろう。

さらに、今回の実験では大学のシラバスを用いたが、今後の発展として、マニュアルやガイドラインなど、他の資料を外部システムとして利用することができないかという点にも注目したい。たとえば、企業向けの業務マニュアルや製品の取扱説明書など、特定の分野に関する情報を外部資料として LLM に参照させることができれば、さまざまな分野での有用性を高め、より広範な知識を活用した提案が可能となるだろう。このようなシステムの拡張により、LLM を用いた情報抽出の適用範囲はさらに広がり、より多様なユーザーに対応できるようになると期待される。

参考文献

- 1) 甲斐大輝、鈴木孝幸、清原良三「個人の特性を考慮した履修登録支援システムの提案」、情報処理学会第 84 回全国大会、2022
- 2) NEC ソリューションイノベータ「LLM（大規模言語モデル）とは？生成 AI との違いや仕組みを解説」
https://www.nec-solutioninnovators.co.jp/sp/contents/column/20240229_llm.html
- 3) NII Today 「第 103 号 LLM 構築における透明性・信頼性・安全性」
<https://www.nii.ac.jp/today/103/3.html>
- 4) Meta「Llama 3 の紹介：これまでで最も高性能でオープンな大規模言語モデル」
<https://about.fb.com/ja/news/2024/04/meta-ai-assistant-built-with-llama-3>