AMERICAN
UNIVERSITY
OF BEIRUT

COURSEMATE

YOUR MSBA316 COMPANION

Olayan School of Business, American University of Beirut

MSBA 316: Text Analytics & Natural Language Processing

Presented by:

Fouad Trad
Lara Baltaji
Ali Hashem

Supervised by:

Dr. Ahmad El Hajj

Summer 2023

Abstract:

This project introduces **MSBA316 CourseMate,** an interactive chatbot used to answer questions related to the MSBA316-Text Analytics and Natural Language Processing course. This chatbot provides instant answers to course-related questions, replacing manual information searches with an efficient alternative. In addition, the chatbot can be used to offer students practice quizzes and assignments based on course lectures and concepts. Data for the chatbot is extracted from the PDF lectures and syllabus, which serve as the main information source. Using the LangChain framework, this data is integrated with a large language model, notably, the GPT-3.5-turbo, which improved cognitive abilities of the chatbot. An interface for the chatbot has been built and integrated into a website using Gradio, making it accessible to everyone interested in learning about NLP. By achieving these goals, the **MSBA316 CourseMate** project helps students better understand course material and improves their learning experience.

Introduction:

In the realm of modern education, interactive and technology-driven solutions play a vital role in enhancing the learning experience for students. As part of the mission, we present **MSBA316 CourseMate**, an advanced chatbot designed to revolutionize the learning process for students enrolled in the “MSBA316 - Text Analytics and Natural Language Processing” course at the American University of Beirut. The ultimate aim of this chatbot is to provide a personalized educational experience for every student.

The primary objective of the **MSBA316 CourseMate** is to provide students with an interactive platform that empowers them to seek instant answers to their course-related questions. By harnessing the power of natural language processing and text analytics, this chatbot will serve as a reliable and comprehensive source of information, ensuring that students can readily access the knowledge they seek without the need for manual searches. In addition to answering questions, the chatbot can extend its support by offering sample quizzes and assignments related to lecture concepts for practice. This feature consolidates students’ understanding of the course material and fosters a deeper grasp of the subject.

From a technical perspective, the foundation of the **MSBA316 CourseMate** lies in the utilization of the LangChain framework, which provides a robust backbone for data integration with language models. In fact, LangChain is a powerful open-source framework designed to help developers in building applications that combine Large Language Models (LLMs) with external data. This framework facilitates the integration of course-specific content extracted from PDF lectures, ensuring that the chatbot possesses a comprehensive repository of information relevant to the course.

To implement the chabot’s advanced language processing capabilities, we have chosen to leverage the power of GPT-3.5-turbo, a state-of-the-art Large Language Model. This integration enables our CourseMate to comprehend and address a wide array of queries, while referring to the course’s data, thereby enriching the learning experience of students. We have also decided to host the chatbot on Gradio, an interactive platform which provides a user-friendly interface, making it easier for students to access their CourseMate.

This project report provides a review of the related work regarding the use of chatbots for educational purposes and delves into the development process of **MSBA316 CourseMate**. The report outlines the main steps taken to construct the interactive chatbot, including: data collection and cleaning, model training, and the user interface implementation. It also discusses the results and the challenges faced during the development and some future recommendations.

Literature Review:

There has been an increased use of Artificial Intelligence (AI) assistants in different field areas, especially in the educational system. AI-Support chatbots leverage the use of large language models (LLMs) to provide personalized and informative responses to queries. The increasing use of AI, especially with the emergence of ChatGPT, has led to a growing body of research on the use of large language models and chatbots in the educational and academic systems. In this literature review, we will discuss the use of virtual assistants over time, the use of conversational AI with LLMs and generative AI models, and the use of LangChain and GPT-3.5 in the context of education and academics.

In the early 1980s, Artificial Linguistic Internet Computer Entity (ALICE) marked a major breakthrough in the development of Artificial Intelligence Markup Language (AIML). AIML was used to create chatbots, and ALICE's use of AIML helped to pave the way for more sophisticated chatbots (Lappalainen & Narayanan, 2023). With the development of new technologies and the emergence of AI, the capabilities of chatbots improved to encompass machine learning, data analytics and natural language processing (NLP) features. According to Hussein et al. (2019), AI chatbots can understand the context of a conversation, the meaning of words, and the nuances of a language. This allows them to handle more complex and varied queries than traditional chatbots. They can also learn and improve their performance over time based on user interactions, resulting in a more human-like and engaging conversational experience (Hussein et al., 2019). Over the past few years, several AI-based chatbots have emerged such as JABBERWACKY (1988), Watson (2006), Siri (2011), ALEXA (2015), Cortana (2015), Tay (2016) until the release of the two state-of-arts, ChatGPT (2022) and Google Bard (2023), which caught the attention of the public and raised a huge interest in chatbots.

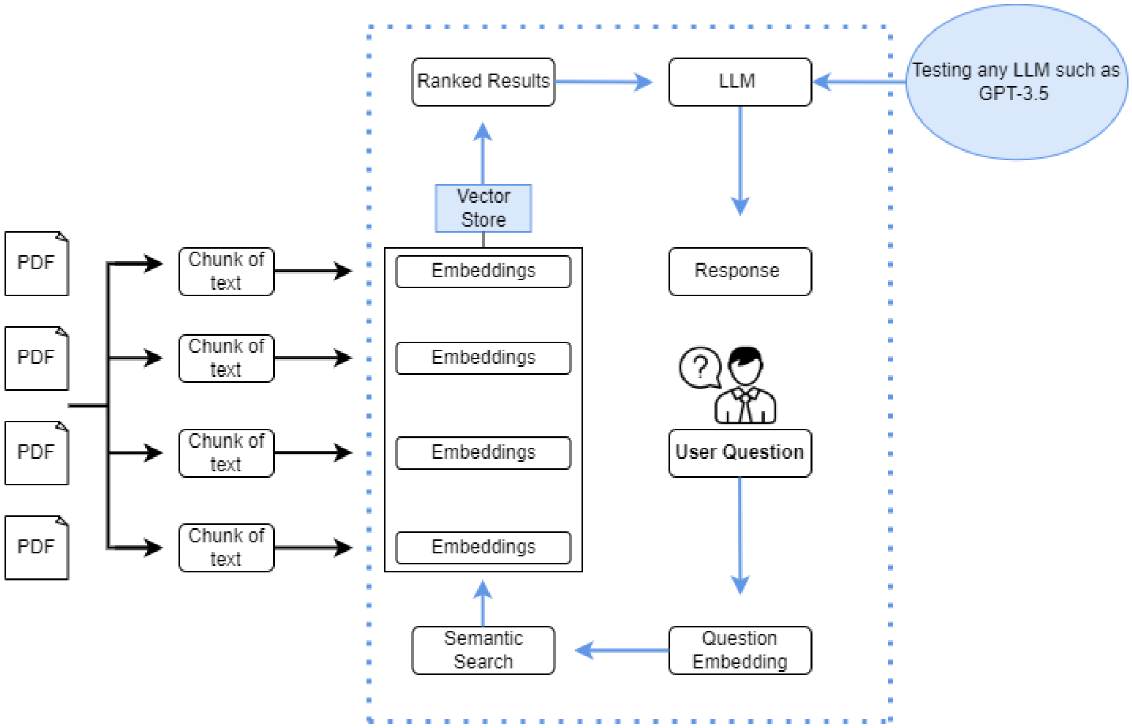
The last two years have marked a remarkable revolution in generative AI and large language models, especially with the introduction of ChatGPT and many other LLMs such as LAMBDA, DALL-E 2, Midjourney, Google Bard, and Stable Diffusion (Lappalainen & Narayanan, 2023). According to Sangzin Ahn (2023), large language models are machine learning models that can perform different natural language processing tasks, including translation, summarization, and grammar correction, by being trained on massive datasets of text using deep learning techniques like transformers. They do this by predicting the next word in a sequence of words in a self-supervised learning model. This allows them to generate long sequences of text that are coherent and grammatically correct, hence the name generative models (Ahn, 2023). For example, Google Bard uses a large language model named PaLM2 and utilizes data from the internet to provide updated and high quality responses to different queries (Lappalainen & Narayanan, 2023). On the other hand, ChatGPT which was developed by OpenAI is a conversational AI that has gained interest and admiration due to its ability to engage in human-like conversations in a very convincing manner. It is based on the GPT-3.5 (and GPT-4 for ChatGPT Plus) language model and has been trained on a huge text corpus which goes up until September 2021 to understand language and generate relevant responses to interact with humans (Lappalainen & Narayanan, 2023).

One way to use GPT-3.5 LLM for a personalized chatbot implementation is by integrating it with a framework for developing data-driven applications powered by language models, like LangChain. LangChain allows a language model to connect to other data sources, while interacting with its environment. It simplifies ingesting data, managing prompts, creating an embedding and parsing output. Mainly, it is used to create chains or patterns of several LLM calls and advanced agents that rely on LLMs to interact with other structures and features (Lappalainen & Narayanan, 2023). According to the official LangChain documentation, LangChain can be used with several LLM providers including OpenAI (GPT), HuggingFace, and Cohere.

We will mention two papers which used the GPT LLM and LangChain for developing chatbots for educational purposes. In an article written by Pesaru et. al (2023), the authors created an AI Assistant for document management using LangChain and Pinecode vectorstore. One aim of the chatbot is to provide tutoring and educational support for students by uploading PDF files and answering questions related to the uploaded files. The chatbot is trained on a dataset of PDF Files and can access and process information from the web through Google Search. The chatbot's architecture uses Pinecode for storing the text embeddings in a vector store, LangChain for searching for similar chunks and GPT-4 for generating responses.

Another paper by Joseph and Jose Ilagan (2023) uses LangChain and GPT-3.5 to create a prototype chatbot for generating and evaluating student business ideas. The chatbot is trained on articles and rubrics for evaluating startup pitches and gives feedback in three different perspectives, a harsh judge, a neutral expert and an optimistic investor.

System Model:



MSBA316 CourseMate employs a well-structured architecture comprising multiple components to facilitate seamless information retrieval from the "MSBA316 Text Analytics and Natural Language Processing" course lectures and syllabus. The diagram presented above illustrates the essential architectural elements of our prototype implementation.

Initially, we load the PDF files of the lectures and syllabus, then we extract the text from them and divide them into chunks, and generate their corresponding embeddings, which we save in a vector store. Then, whenever the user asks a question, we generate its embedding and accordingly search for the relevant documents/chunks in the vector store. Once these are found, they are sent along with the query to an LLM to give a final response to the user. More details about the technical implementation can be found in the next section

Code Breakdown:

The development process involved three Python scripts: index.py, conversation.py, and bot.py.

- **Data Collection and Embedding Generation (index.py):**

The project's first step involved gathering relevant course material, including the syllabus and lectures, required for the chatbot's knowledge. The PyPDFLoader was employed to load the PDF Files, enabling access to lecture content and essential information. In order to use the PDF documents as context for the LLM, the text should be divided into chunks. That is because, usually, LLMs have a finite maximum context length, and exceeding the limit of this maximum length will cause the model to fail. Therefore, the documents were further split into smaller chunks using the RecursiveCharacterTextSplitter by LangChain. This step ensured that CourseMate could provide accurate and contextually relevant responses to student queries.

For effective representation of the text chunks, OpenAIEmbeddings were utilized to generate embeddings for each chunk. These embeddings captured the semantic meaning of the text, enabling CourseMate to comprehend and analyze user queries effectively. The embeddings were stored in a Chroma vector store, ensuring easy access and retrieval during conversations with students.

- **Conversational Retrieval Chain (conversation.py):**

The heart of CourseMate lies in the ConversationalRetrievalChain from LangChain, implemented in the conversation.py script. This chain was designed to handle multi-turn conversations, making interactions with CourseMate feel natural and human-like. Multi-turn conversations are conversations that keep track of chat history and background through memory. To achieve this, a memory component was implemented using the ConversationBufferMemory class, which enables CourseMate to remember the history of the conversation, allowing for a more coherent and context-aware interaction.

The SelfQueryRetriever from LangChain was a critical component of the chain. It utilized the embeddings stored in the Chroma vector store to rank the text chunks and retrieve only relevant document chunks based on user queries. Additionally, metadata was employed to provide additional context. For example, the source of the document chunk (lecture number or syllabus) and the corresponding page number were used to improve the accuracy of CourseMate's responses, because these elements facilitated the process of knowing which PDF the information exists in. Finally, the question-answering process was based on the GPT-3.5

Turbo large language model to ensure that CourseMate provided precise and well-informed answers to student queries based on memory and specified context.

Now, when a user inputs a question or a request, the user's prompt is turned into a vector embedding using OpenAI's embedding module. Here, LangChain plays a crucial role in searching for similar chunks and ranking the results according to their relevance whenever a question is asked. Now, the chosen LLM (GPT-3.5) processes the user's prompt, and returns a response to the user using only the most relevant chunk(s) and the metadata provided as additional context. The above steps are shown in the second part of the diagram. As a summary, the system goes as follows:

- The user inputs a question
- The question is embedded into a vector embedding
- LangChain semantically searches for chunks similar to the user's prompt, based on the estimated similarity with the vector space
- The embedded chunks are ranked based on their similarity score to the user's query
- The chosen LLM (GPT-3.5) processes the user's prompt and returns a response to the user using only the most relevant chunk(s) and the additionally specified metadata, such as the lecture name and the page number.

- **User Interface and Interaction (bot.py):**

The bot.py script was responsible for creating CourseMate's user interface. Using Gradio, the interface was designed to be user-friendly and accessible. Students could interact with CourseMate by entering their queries into a text box and submitting them for processing. They can also clear the chat history in order to start a new conversation.

Upon submission, the conversational retrieval chain, developed in conversation.py, came into action. It enabled CourseMate to understand the user's query, retrieve relevant information from the vector store, and deliver appropriate responses. The conversational nature of the responses, made possible by the PromptTemplate, created a seamless and engaging interaction experience for students.

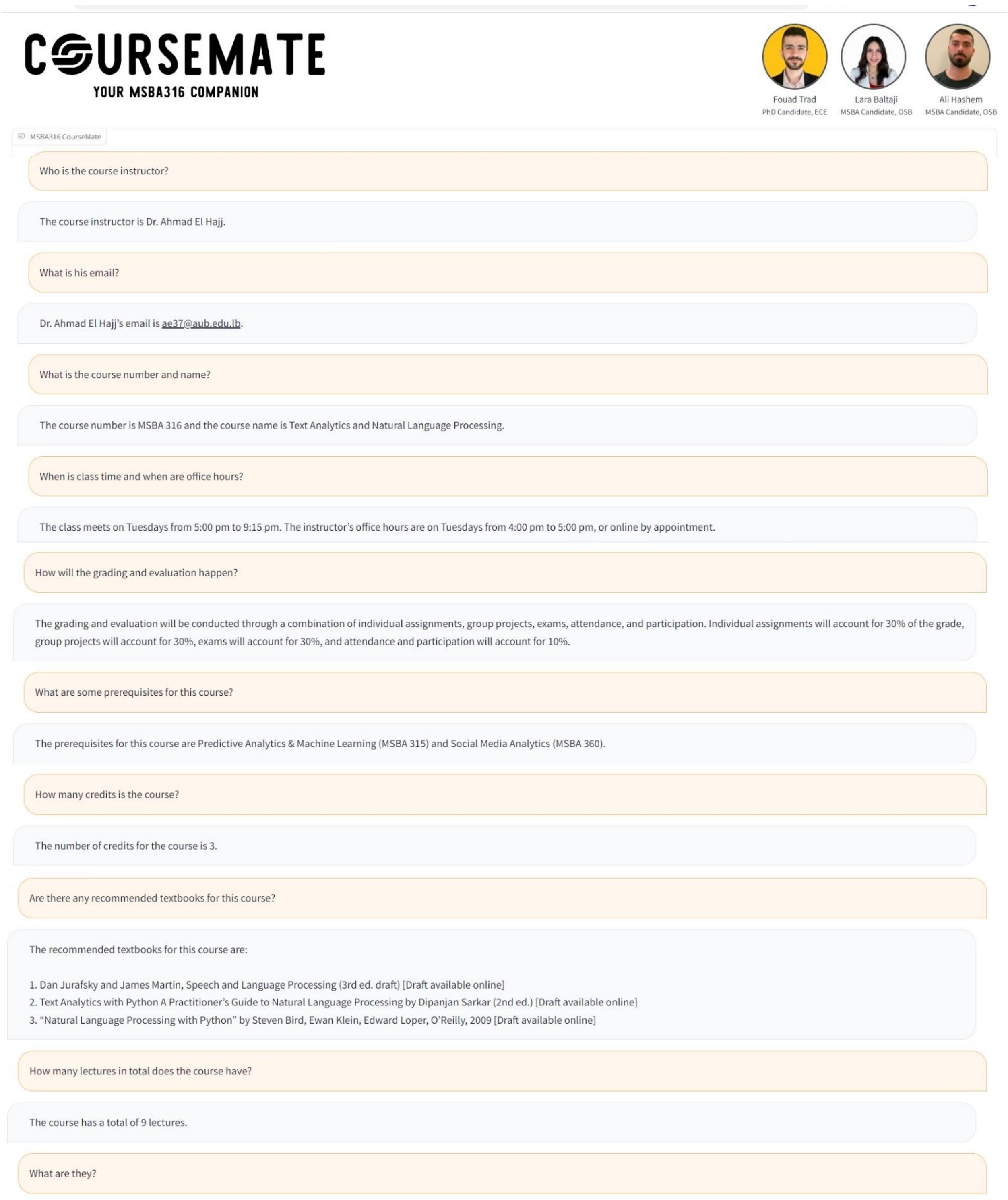
- **Steps to run the code:**

Several resources have been investigated to write the code, and they are cited in the references section. In order to run the final code, several steps need to be done.

- Install the libraries listed in the requirements.txt file
- Run the index.py file. This should be done once, as the goal is to divide the lectures into chunks, embed them, and save them in a vector store. If you see the folder named "db", there would be no need for this step at all.
- Run the bot.py file. This will generate a link for the interface, and you can access it to communicate with the bot.

Results:

The chatbot was tested through a series of queries about the MSBA316 course. The questions included information about the course syllabus and the different lecture concepts. One prompt asked the CourseMate to generate multiple choice questions to practice some concepts. We present below the MSBA316 interface and the generated results. More queries and explanations can be found in the demo.



COURSEMATE

YOUR MSBA316 COMPANION

Fouad Trad
PhD Candidate, ECE

Lara Baltaji
MSBA Candidate, OSB

Ali Hashem
MSBA Candidate, OSB

MSBA316 CourseMate

Who is the course instructor?

The course instructor is Dr. Ahmad El Hajj.

What is his email?

Dr. Ahmad El Hajj's email is ae37@aub.edu.lb.

What is the course number and name?

The course number is MSBA 316 and the course name is Text Analytics and Natural Language Processing.

When is class time and when are office hours?

The class meets on Tuesdays from 5:00 pm to 9:15 pm. The instructor's office hours are on Tuesdays from 4:00 pm to 5:00 pm, or online by appointment.

How will the grading and evaluation happen?

The grading and evaluation will be conducted through a combination of individual assignments, group projects, exams, attendance, and participation. Individual assignments will account for 30% of the grade, group projects will account for 30%, exams will account for 30%, and attendance and participation will account for 10%.

What are some prerequisites for this course?

The prerequisites for this course are Predictive Analytics & Machine Learning (MSBA 315) and Social Media Analytics (MSBA 360).

How many credits is the course?

The number of credits for the course is 3.

Are there any recommended textbooks for this course?

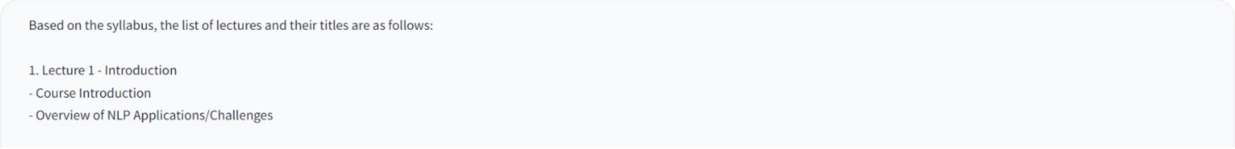
The recommended textbooks for this course are:

1. Dan Jurafsky and James Martin, Speech and Language Processing (3rd ed. draft) [Draft available online]
2. Text Analytics with Python A Practitioner's Guide to Natural Language Processing by Dipanjan Sarkar (2nd ed.) [Draft available online]
3. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, Edward Loper, O'Reilly, 2009 [Draft available online]

How many lectures in total does the course have?

The course has a total of 9 lectures.

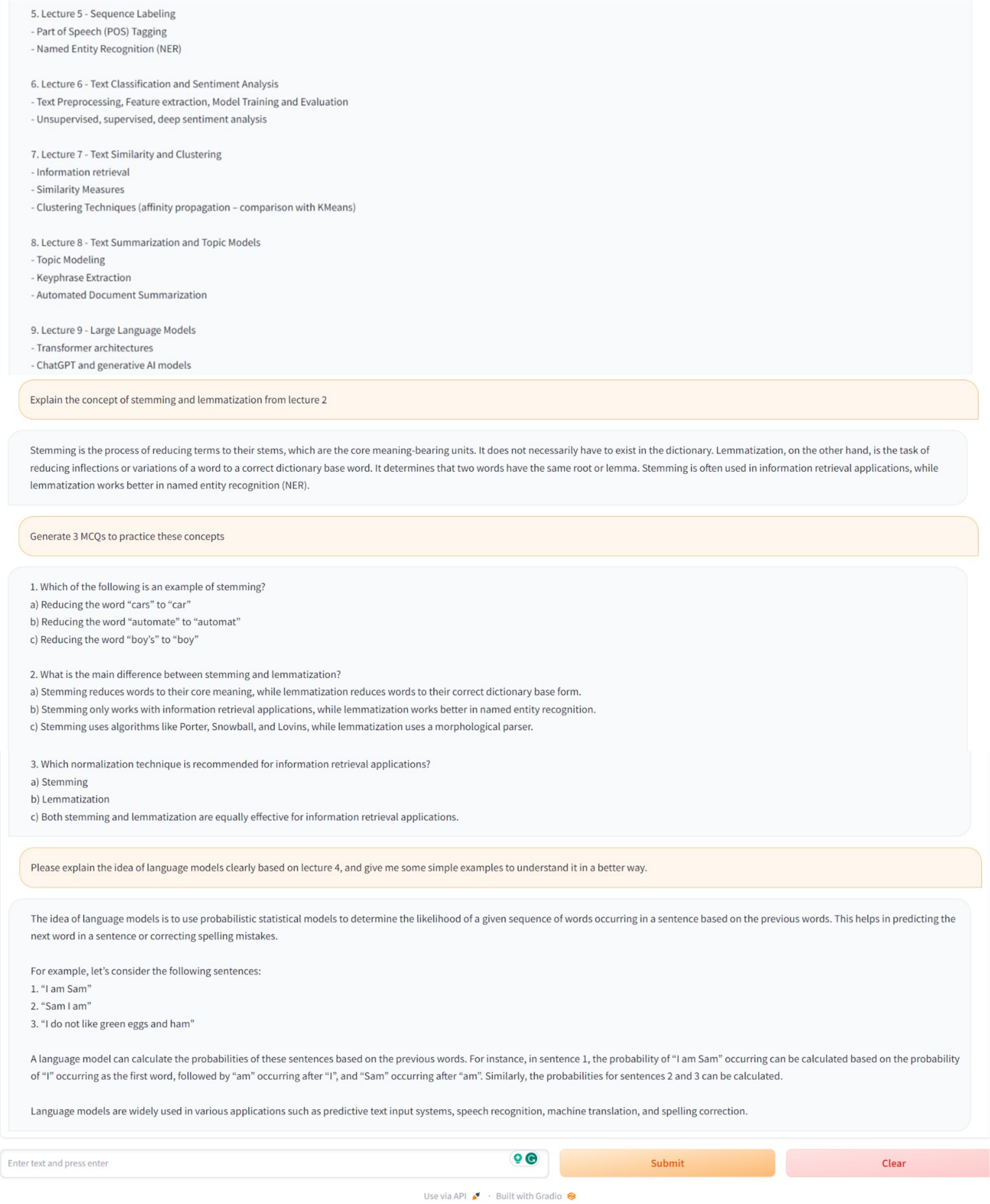
What are they?



Based on the syllabus, the list of lectures and their titles are as follows:

1. Lecture 1 - Introduction
- Course Introduction
- Overview of NLP Applications/Challenges

. . .



5. Lecture 5 - Sequence Labeling
- Part of Speech (POS) Tagging
- Named Entity Recognition (NER)

6. Lecture 6 - Text Classification and Sentiment Analysis
- Text Preprocessing, Feature extraction, Model Training and Evaluation
- Unsupervised, supervised, deep sentiment analysis

7. Lecture 7 - Text Similarity and Clustering
- Information retrieval
- Similarity Measures
- Clustering Techniques (affinity propagation – comparison with KMeans)

8. Lecture 8 - Text Summarization and Topic Models
- Topic Modeling
- Keyphrase Extraction
- Automated Document Summarization

9. Lecture 9 - Large Language Models
- Transformer architectures
- ChatGPT and generative AI models

Explain the concept of stemming and lemmatization from lecture 2

Stemming is the process of reducing terms to their stems, which are the core meaning-bearing units. It does not necessarily have to exist in the dictionary. Lemmatization, on the other hand, is the task of reducing inflections or variations of a word to a correct dictionary base word. It determines that two words have the same root or lemma. Stemming is often used in information retrieval applications, while lemmatization works better in named entity recognition (NER).

Generate 3 MCQs to practice these concepts

1. Which of the following is an example of stemming?
a) Reducing the word “cars” to “car”
b) Reducing the word “automate” to “automat”
c) Reducing the word “boy’s” to “boy”

2. What is the main difference between stemming and lemmatization?
a) Stemming reduces words to their core meaning, while lemmatization reduces words to their correct dictionary base form.
b) Stemming only works with information retrieval applications, while lemmatization works better in named entity recognition.
c) Stemming uses algorithms like Porter, Snowball, and Lovins, while lemmatization uses a morphological parser.

3. Which normalization technique is recommended for information retrieval applications?
a) Stemming
b) Lemmatization
c) Both stemming and lemmatization are equally effective for information retrieval applications.

Please explain the idea of language models clearly based on lecture 4, and give me some simple examples to understand it in a better way.

The idea of language models is to use probabilistic statistical models to determine the likelihood of a given sequence of words occurring in a sentence based on the previous words. This helps in predicting the next word in a sentence or correcting spelling mistakes.

For example, let’s consider the following sentences:
1. “I am Sam”
2. “Sam I am”
3. “I do not like green eggs and ham”

A language model can calculate the probabilities of these sentences based on the previous words. For instance, in sentence 1, the probability of “I am Sam” occurring can be calculated based on the probability of “I” occurring as the first word, followed by “am” occurring after “I”, and “Sam” occurring after “am”. Similarly, the probabilities for sentences 2 and 3 can be calculated.

Language models are widely used in various applications such as predictive text input systems, speech recognition, machine translation, and spelling correction.

Discussion:

The results obtained from the interactions with the **MSBA316 CourseMate** chatbot reveal a highly impressive performance in multiple key aspects, significantly enhancing the learning experience of students enrolled in this course. In this section, we discuss the remarkable capabilities demonstrated by the chatbot, including its proficiency in explaining concepts, creating practice exercises, engaging in human-like conversations, generating illustrative examples and retrieving specific information tailored to the user's desires.

The CourseMate chatbot effectively fulfills its primary goal of providing clear and concise explanations of complex concepts related to “MSBA316 - Text Analytics and Machine Learning”. Its responses are thoughtfully crafted, offering students a thorough understanding of the course material. Moreover, the chatbot’s capability to create practice exercises tailored to the course lectures allows students to actively apply their knowledge, thus enhancing their grasp of the subject matter.

In addition to explanations, the chatbot goes above and beyond by generating Multiple Choice Questions (MCQs) and creative examples for students to practice the course concepts. This additional practice component empowers students to assess their knowledge, identify areas of improvement and actively reinforce their understanding of topics.

One of the chatbot’s most impressive features is its ability to engage in human-like conversational style. By effectively recalling previous chat interactions, the chatbot creates a personalized experience for each student. This aspect of context retention contributes to coherent conversations, as the chatbot accurately uses pronouns to refer to relevant entities. This human-like conversational approach fosters an engaging and interactive learning experience, allowing users to seamlessly follow-up on their previous questions.

In addition, CourseMate showcases its flexibility by promptly retrieving specific information according to the user’s desires. When prompted to explain concepts like stemming from lecture 2 and language models from lecture 4, the chatbot successfully accesses and delivers the relevant content. This customized information retrieval allows students to obtain targeted insights aligned with their learning needs.

Current Limitations:

While the **MSBA316 CourseMate** has demonstrated exceptional performance in various aspects, it is essential to acknowledge its limitations to gain a comprehensive understanding of its capabilities and to encourage improvement. One significant limitation of the chatbot lies in its inability to effectively access large contexts. For example,when asked to summarize an entire lecture, like lecture 2, the chatbot fails at including all the main concepts. This limitation arises from the architectural model of the chatbot, which restricts its access to a limited number of chunks of information.

The chatbot's architecture, designed to prioritize and rank information chunks, constrains its capacity to summarize extensive content adequately. When faced with a request to summarize a lengthy lecture, such as lecture 2, the chatbot's access is limited to only the top four highest-ranked chunks. As a result, it may omit crucial details and fail to provide a comprehensive and coherent summary of the entire lecture.

Addressing this limitation may require exploring novel architectural approaches, potentially leveraging methods to identify and prioritize key segments of large content to generate more informative summaries.

Conclusion:

MSBA316 CourseMate marks a significant advancement in the area of educational technology, aiming to modernize the learning experience for students enrolled in the course "MSBA316 - Text Analytics and Natural Language Processing" at the American University of Beirut. Throughout the development and evaluation of the chatbot, several remarkable achievements have been observed, along with valuable insights into its capabilities and potential areas of improvement.

The CourseMate chatbot has shown outstanding performance in various aspects. Its proficiency in explaining complex concepts, generating practice exercises, and engaging in human-like conversations, enhances the learning journey for students. By providing instant answers to queries and personalized support, the chatbot has proven to be an interactive and continuously accessible platform for students to consolidate their understanding and deepen their knowledge. Additionally, the chatbot's ability to retrieve targeted information based on user requests has introduced a new dimension of customization to the learning experience. This customized information retrieval empowers students to delve into specific topics of interest, promoting self-directed learning and fostering a deeper connection with the course material.

However, the chatbot faces a challenge in summarizing large context information. Despite its impressive capabilities, the chatbot's restriction to access only a limited number of ranked chunks hinders its comprehensive understanding and representation of extensive content.

In conclusion, **MSBA316 CourseMate** represents a revolutionary step towards creating an innovative and engaging learning environment. As technology continues to progress, the integration of AI-driven chatbots like the CourseMate chatbot is set to reshape the educational environment, delivering personalized and interactive learning experiences to students worldwide. By leveraging the power of natural language processing and intelligent data integration, such educational tools hold the potential to become integral companions in the academic journey of learners across different fields. With a commitment to continuous monitoring and improvement, this project sets the stage for the future of education, where innovative technology and education merge to empower students and inspire a generation of lifelong learners.

References:

Ahn S. (2023). The impending impacts of large language models on medical education. Korean J Med Educ, 35(1), 103-107. DOI:10.3946/kjme.2023.253. Epub 2023 Feb 28. PMID: 36858381; PMCID: PMC10020064.

Chase H. (2023). Short Course on LangChain: Chat with Your Data. DeepLearning.AI. https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data/

Hussain, S., Sianaki, O. A., & Ababneh, N. (2019). A survey on conversational agents/chatbots classification and design techniques. In Web, Artificial Intelligence And Network Applications: Proceedings Of The Workshops Of The 33Rd International Conference On Advanced Information Networking And Applications (Waina-2019), Matsue, Japan, 27-29 March 2019 (pp. 946-956). DOI: 10.1007/978-3-030-15035-8_93

Ilagan, J. R., & Ilagan, J. R. (2023, June 3). A prototype of a chatbot for evaluating and refining student startup ideas using a large language model. doi:10.35542/osf.io/azhf9

Lappalainen Y. & Narayanan N. (2023). Aisha: A Custom AI Library Chatbot Using the ChatGPT API. Journal of Web Librarianship. DOI:10.1080/19322909.2023.2221477

Ottley L. (2023, May 2). Create Your Own ChatGPT with PDF Data in 5 Minutes (LangChain Tutorial) [Video]. YouTube. https://www.youtube.com/watch?v=au2WVVGUvc8

Pesaru, A., Gill, T. S., & Tangella, A. R. (2023). AI Assistant For Document Management Using LangChain and Pinecone. International Research Journal of Modernization in Engineering Technology and Science.

Prompt Engineering. (2023, April 14). Working with MULTIPLE PDF Files in LangChain: ChatGPT for your Data [Video]. YouTube. https://www.youtube.com/watch?v=s5LhRdh5fu4