Conversational Online Course Recommendation System Leveraging LSTM for Joint Intent and Slot Detection with LLM Retrieval-Based Response Generation

**Abstract**

This project presents a novel Conversational Course Recommendation System that integrates advanced techniques for intent and slot detection with sophisticated response generation. The system employs a Long Short-Term Memory (LSTM) network for joint intent classification and slot filling, enabling it to accurately interpret user queries and extract relevant information such as course names and subject areas. By leveraging LSTM's ability to manage sequential dependencies and context within user inputs, the model ensures precise extraction of intents and slots, which are crucial for retrieving appropriate course recommendations from a structured database. Subsequently, the retrieved courses are used to generate informative and contextually relevant responses through a Large Language Model (LLM), specifically the Microsoft Phi-3-mini-4k-instruct. This integration of LSTM and LLM facilitates the creation of responses that are not only relevant but also coherent and engaging, enhancing the user experience in querying and selecting courses.

The methodology encompasses rigorous evaluation of both the intent and slot detection accuracy and the quality of generated responses. Evaluation metrics include standard measures such as accuracy and F1-score for the detection phase, and BLEU and ROUGE scores alongside human evaluations for response quality. Despite its effectiveness, the system faces limitations including dependency on data quality, computational resource demands, and challenges in maintaining real-time performance. Future work will focus on improving data diversity, exploring more advanced model architectures, and optimizing system efficiency to better handle large-scale and real-time interactions. This approach promises to advance the state-of-the-art in conversational systems for personalized course recommendations, offering a robust framework for integrating natural language understanding and generation in educational contexts.

# Introduction

Online learning is any sort of study that occurs remotely via the internet. This mode of education enables students to study through educational resources in their own time without having to travel to a physical classroom. It enables more flexible learning and allows students to study while also juggling other responsibilities such as family and employment. Figure 1.1 highlights some of the advantages of online learning.

Online learning has transformed education over the last decades and has now become an integral part of the learning experience. The rise of the world wide web in the 90s which made the internet more accessible contibuted immensely to its adoption and spread, making it easier for people to get more access to quality education. Many universities and colleges have also adopted this method of teaching, and students can now gain knowledge and earn certificates without physically attending a classroom. Also, platforms like coursera, udacity, edx etc. have made educational resources available to people around the world at little or no cost. With advancement in technology, there is a great likelihood that online learning will continue to evolve and will redefine the way knowledge is acquired.A diagram of a person using a computer

Description automatically generated

Figure 1.1: Advantages of Online Learning (Source: EDUCBA)

The charts below illustrate the significant increase in the number of online courses and MOOCs over the past decade. Figure 1.2 shows the growth of MOOCs from 2012 to 2022, highlighting a steady and substantial rise in the number of courses offered, reaching around 20,000 courses by 2022. This reflects a global trend toward the adoption and availability of MOOCs globally, reflecting a growing trend towards online education.

Similarly, Figure 1.3 focuses on Udemy, a major online learning platform, detailing its growth from 2009 to 2022, culminating in 202,000 courses by 2022. Alongside this, Udemy saw a total of 662 million enrollments, with an average of 3,274 enrollments per course. This further emphasizes the increasing upward trend of online learning. Both charts demonstrates a significant rise in the number of online courses and MOOCs, indicating a strong and growing demand for accessible and flexible education options worldwide.

A graph of growth in the year

Description automatically generated with medium confidence

Figure 1.2: Growth of MOOCs over the years. (Source: https://www.classcentral.com/report/mooc-stats-2021)

A graph with red lines and numbers

Description automatically generated

Figure 1.3: Growth of Courses in Udemy. (Source: https://www.classcentral.com/report/udemy-by-the-numbers)

**1.1. Problem Statement**

This rapid expansion of online education has led to an overwhelming number of available courses across various domains, with an ever-expanding array of courses available to learners worldwide. However, with this abundance of options comes the challenge of navigating through the extensive catalogue to find courses that align with individual interests, learning goals, and proficiency levels.

With platforms like Udemy offering over 200,000 courses and MOOCs collectively reaching above 20,000, learners often face difficulties in navigating through such extensive catalogs. This can result in decision fatigue, where the sheer volume of choices hampers the ability to make effective decisions, ultimately leading to a less optimal learning experience. Resolving this issue is crucial for enhancing user satisfaction and can lead to better learning outcomes, increased user engagement, and more efficient course discovery. Without improvements, users may continue to face frustration and suboptimal learning experiences.

To address this challenge of guiding learners in making informed choices, various methods have been employed to enhance the course recommendation process. One widely adopted approach involves the use of recommendation algorithms, which draw on principles similar to those used in e-commerce and streaming services. These algorithms analyze a learner's past behavior, such as previously viewed or completed courses, along with their stated interests and preferences, to suggest courses that align with their learning patterns. By leveraging data-driven techniques such as collaborative filtering, content-based filtering, and hybrid models, these systems can deliver personalized course recommendations that cater to individual needs.

In addition to recommendation algorithms, many online learning platforms incorporate user reviews and ratings as a means to provide additional insights into the quality and popularity of courses. Reviews offer qualitative feedback from other learners who have completed the courses, highlighting aspects such as course content, instructor effectiveness, and overall satisfaction. Ratings provide a quantitative measure of a course's reception and can help learners gauge the relative value of different options. By aggregating and displaying this feedback, platforms enable prospective students to make more informed decisions based on the experiences of their peers. This combination of algorithmic recommendations and user-generated content enhances the transparency of course offerings and supports learners in choosing courses that best meet their educational goals and preferences.

However, while these traditional approaches to course recommendations provide valuable insights, they often fall short in addressing the complexities of individual learner needs and preferences. Traditional recommendation systems typically rely heavily on historical data and predefined criteria, which can limit their ability to understand and respond to nuanced user queries. This is where conversational course recommendation systems, enhanced with machine learning (ML) and natural language processing (NLP) techniques, offer a significant advantage. By utilizing NLP, these systems can interpret and analyze user queries in natural language, allowing for a more nuanced understanding of learner intent and requirements. This capability enables the system to process complex, conversational inputs and tailor recommendations with a higher degree of precision.With advancements in natural language processing (NLP) and machine learning (ML), there is an opportunity to develop more sophisticated personalized, and flexible recommendation systems to help simplify the decision-making process and enhance the overall learning experience by ensuring that learners can more easily find and enrol in courses that are most relevant to their objectives. By developing a Conversational Course Recommendation System, this project aims to bridge the gap between user intent and course suggestions. Users can describe their learning objectives through natural language queries, which is processed using machine learning to accurately detect intent and extract relevant information. NLP techniques are then utilized to process this information and recommend courses that best align with the user's expressed needs. This approach not only enhances the precision of course recommendations but also provides a more intuitive and engaging user experience, reflecting a significant advancement over conventional systems that often lack the capability to fully understand and cater to individual learner preferences.

**1.2. Research question**

How does the integration of LSTM-based joint intent detection and slot filling with LLM-driven retrieval-based response generation enhance the accuracy, and relevance of course recommendations in a conversational recommendation system?

**1.3. Aims and Objectives**

Aim: To develop and evaluate a sophisticated conversational course recommendation system that leverages machine learning and natural language processing techniques to enhance the relevance and accuracy of course recommendations, thereby improving decision-making in online learning environments.

Objectives

1. Design and Implement an LSTM-Based Model for Joint Intent Detection and Slot Filling:

* Develop an LSTM architecture capable of understanding and processing user queries to identify the underlying intent and extract relevant information (slots).
* Train and validate the LSTM model using annotated datasets to ensure high accuracy in intent detection and slot filling.

1. Integrate NLP Techniques for Course Matching:

* Implement cosine similarity to match user queries with a database of courses based on extracted features.
* Optimize the matching algorithm to handle diverse and complex user queries, ensuring that the recommended courses are highly relevant to the user's needs.

1. Develop an LLM-Based Retrieval Mechanism for Generating Recommendations:

* Use a Large Language Model (LLM) to generate coherent and contextually appropriate responses based on the courses matched by the NLP techniques.
* Fine-tune the LLM to improve the coherence and relevance of the generated responses, enhancing the overall user experience.

1. Evaluate System Performance and User Satisfaction:

* Conduct performance testing to measure the accuracy and effectiveness of the recommendation system, focusing on how well it identifies relevant courses.
* Analyse the comparative results to identify strengths and areas for improvement, providing insights into the advantages of integrating LSTM, NLP, and LLM technologies in course recommendation systems.

# Literature Review

Online course recommendation systems have experienced substantial progress due to recent advancements in machine learning and natural language processing (NLP). Traditionally, these systems relied on Collaborative Filtering, which leverages user-item interactions to suggest courses based on the preferences and behaviors of similar users. Another approach was Content-based Filtering, which focuses on the attributes of the courses themselves, such as subject matter or duration, to recommend options aligned with the user’s past interests. Hybrid recommendation systems, which integrate both collaborative and content-based methods, emerged to combine the strengths of these approaches and provide more accurate and personalized recommendations. The advent of sophisticated machine learning algorithms and NLP techniques has further enhanced these systems, enabling more nuanced and context-aware recommendations that better meet individual learning needs and preferences.

Collaborative filtering is a popular recommendation approach that utilizes the preferences and behaviors of users to suggest items liked by similar users. This system generates recommendations based on the aggregate preferences of users with comparable tastes, and can be categorized into two main types: user-based and item-based collaborative filtering. A notable example of user-based collaborative filtering for course recommendations is the study by Obeidat et al. (2019) titled "A Collaborative Recommendation System for Online Courses Recommendations." In their research, the authors developed a collaborative recommender system that suggests online courses to students based on similarities in their course history. They applied data mining techniques to identify patterns between courses and demonstrated that clustering students with similar course selections significantly improves the quality of association rules compared to using the whole set of courses and students. Their research highlighted the strengths of user-based approaches in capturing the collective preferences of a user community.

Despite the effectiveness of collaborative filtering in capturing the collective preferences of users, it has notable shortcomings. One major limitation is the "cold start" problem, which occurs when there is insufficient data for new users or items, making it difficult to generate accurate recommendations. For new users, the system struggles to make accurate recommendations due to a lack of interaction history. Similarly, new items face challenges in gaining traction until they accumulate sufficient user interactions. Additionally, collaborative filtering can suffer from scalability issues as the number of users and items grows. This can lead to slower response times and increased system complexity. Furthermore, it may lead to recommendation biases, where popular items receive more attention while niche or less popular items are overlooked, as it tends to suggest items that are popular among similar users, potentially limiting exposure to new or less popular options. These challenges highlight the need for incorporating complementary techniques to enhance recommendation accuracy and coverage, since it is crucial to ensure that recommendations are diverse, inclusive, and do not reinforce biases or restrict student's options.

In contrast, content-based filtering approaches generate recommendations by focusing on the attributes of courses and the user’s historical interactions. Unlike collaborative filtering, which relies on user-item interactions, content-based filtering focuses on recommending items based on their specific features and the user’s historical preferences. This method builds a detailed user profile based on past behavior and compares it with course features to suggest relevant options. Neamah et al. (2018) demonstrated how to design an effective content-based course recommender system using k-Nearest Neighbors (kNN) and Naïve Bayes classification algorithms. Their proposed system builds a detailed user profile based on prior interactions, such as enrolling in or rating courses, and then compares this profile with course attributes to generate personalized recommendations.

While content-based filtering offers several advantages, such as the ability to handle new items without requiring user interaction data, it also has its own set of limitations. One significant shortcoming is its reliance on the quality and completeness of item attributes. If the descriptive features of courses are insufficiently detailed or inaccurate, the recommendations may be less relevant. Additionally, content-based systems can lead to a "narrowing" effect, where recommendations are limited to items similar to those the user has already engaged with, potentially reducing exposure to novel or diverse options. Furthermore, the initial user profile can be limited if the user’s past interactions are sparse, impacting the system's ability to generate accurate recommendations.

Hybrid recommendation systems combine the strengths of collaborative filtering and content-based filtering, to deliver more accurate and diverse recommendations. By integrating these approaches, hybrid systems can leverage user interaction data and item attributes to overcome the limitations inherent in each individual method. For instance, while collaborative filtering excels in capturing the collective preferences of users, it may struggle with new or less popular items. Content-based filtering, on the other hand, can effectively recommend new items based on their features but may produce recommendations that lack novelty. Hybrid systems address these issues by blending collaborative and content-based techniques, thereby providing more comprehensive and personalized recommendations. This integration allows the system to benefit from the detailed user profiles created through content-based methods while also utilizing the broader context of user preferences found in collaborative filtering. For example, Estrela et al, (2017) developed a system that combines content-based, collaborative filtering, and hybrid techniques to suggest courses based on user profiles and interests. Despite these advantages, hybrid systems still face challenges, such as the potential for cold start problems for new users or items, although to a lesser extent than single-method approaches. All the aforementioned recommendation approaches, also raises concerns about data privacy. To provide personalized recommendations, these systems collect and analyse data on student's interactions, which necessitates, transparent data collection and privacy policies to ensure the security student data.

As recommendation systems evolved, the incorporation of machine learning (ML) and natural language processing (NLP) techniques became increasingly prevalent. The integration of advanced ML models allowed for more nuanced and personalized recommendations. Algorithms began to incorporate contextual information and user behavior patterns to enhance prediction accuracy. For instance, deep learning models, such as autoencoders and recurrent neural networks (RNNs), introduced new possibilities for capturing complex user preferences, interactions or queries, thereby significantly improving the relevance of recommendations.

Conversational recommendation systems represent a more recent and innovative development in the field and offers a compelling alternative by focusing on the user's immediate needs and context. These systems aim to enhance user interaction by utilizing dialogue-based interactions to understand user needs more effectively and provide tailored suggestions in real-time. The evolution of conversational recommendation systems began with simple rule-based chatbots that offered static responses based on predefined scripts. Over time, these systems have evolved to incorporate sophisticated NLP techniques, including intent recognition and entity extraction, to understand user queries more effectively. The integration of conversational agents with recommendation algorithms has enabled a more interactive and personalized user experience. For example, modern conversational recommendation systems can engage users in natural dialogue, asking clarifying questions to refine recommendations based on user responses. This evolution has been applied across various domains, including e-commerce, and entertainment, where systems like Amazon's Alexa or Netflix's recommendation engine use conversational interfaces to enhance user engagement and satisfaction. This method can effectively bypass the cold start problem since it does not rely on historical interaction data but rather on the user's current needs and preferences.

Moreover, a significant advantage of user query-based recommendations is that it does not require access to or storage of user data, addressing privacy concerns by providing recommendations based on the query at hand rather than accumulating and analyzing personal data. By avoiding the need to track or store user behavior, this method enhances user privacy and security while still delivering relevant suggestions. It also address the issue of recommendation biases by providing a more dynamic and interactive experience. Users can refine their queries to get more tailored suggestions, thus reducing the risk of being confined to popular or similar items and enhancing the system's adaptability to evolving user interests and emerging trends, as it can quickly adjust recommendations based on the latest queries and user inputs.

In the 2021 publication "Advances and Challenges in Conversational Recommender Systems: A Survey," Gao et al. provide an extensive review of the progress and ongoing issues in the field of conversational recommender systems. The paper systematically explores the evolution of conversational recommendation technologies, covering key advancements in natural language processing (NLP), dialogue management, and recommendation algorithms. The authors highlight significant strides made in creating more interactive and user-centric recommendation systems that leverage conversational interfaces to enhance user engagement and satisfaction. The survey delves into the various approaches and techniques employed in conversational recommender systems, including deep learning models, reinforcement learning, and hybrid methods that integrate multiple sources of information to refine recommendations. They also discussed the challenges that persist in this area, such as managing dialogue coherence, handling diverse and dynamic user queries, and effectively incorporating contextual information into the recommendation process.

Several studies have explored conversational recommendation systems, highlighting their potential and applications. For instance, Zhang et al. (2018) in their publication “Towards Conversational Search and Recommendation: System Ask, User Respond” made significant contributions to the field of conversational recommendations by introducing a novel framework that enhances interactive dialogue between users and recommendation systems in e-commerce. To accomplish this, they proposed the Multi-Memory Network (MMN) architecture, which can be trained based on large scale collections of user reviews in e-commerce. Their work focuses on the innovative concept of "System Ask, User Respond," where the system proactively asks users clarifying questions to refine and personalize recommendations based on real-time feedback. This approach allows the system to better understand user preferences and context, leading to more accurate and relevant suggestions. By leveraging natural language processing (NLP) techniques and dialogue management strategies, Zhang et al. demonstrated how a conversational agent can dynamically engage with users, adapt to their responses, and provide tailored recommendations that go beyond static, one-size-fits-all solutions. Their research highlights the potential of conversational systems to improve user interaction and satisfaction, setting a precedent for more sophisticated and responsive recommendation technologies. This framework represents a crucial step forward in making conversational recommendations more interactive and contextually aware, paving the way for further exploration and application in various domains, including online course recommendations.

Also, Habib et al. (2018) developed a conversational recommender system for movie recommendations, focusing on enhancing user experience through dialogue-based interactions. They introduce IAI MovieBot, a conversational recommender system for movies. It features a task-specific dialogue flow, a multi-modal chat interface, and an effective way to deal with dynamically changing user preferences. The system architecture comprised natural language understander (NLU) which converts the natural language response from the user into a dialogue act. This process, comprising of intent detection and slot filling, is performed based on the current dialogue state. The dialogue state tracker (DST) in the dialogue manager (DM) updates the dialogue state (DS) and dialogue context (DC) based on the dialogue acts by both the agent and the user. For the response generation, multiple templates were designed for each intent and its parameters to select a response randomly.

Despite these advancements, the application of conversational recommendation systems to the domain of online course recommendations remains relatively underexplored. The existing research has predominantly focused on domains such as movies and retail, leaving a gap in understanding how conversational approaches can be applied to the educational sector. The integration of conversational interfaces into course recommendations could facilitate more personalized and context-aware suggestions by allowing users to articulate their learning goals, preferences, and constraints in natural language. This approach aligns with the work of researchers like Goudar et al. who described an educational recommendation system (ERS) that uses cosine similarity (CS) and content-based filtering (CBF) to suggest online courses to users based on their search terms. They proposed a system that utilizes the titles of the courses present in the dataset to calculate cosine similarity scores between the user query and each course in the dataset. Subsequently, the system sorts the results in descending order and then filters them against various parameters such as price, reviews, and other relevant factors that may influence the user's decision to select a particular course. Plaza et al, (2023) also demonstrated the effectiveness of NLP techniques in recommending online courses based on user input. Plaza's content-based recommender system uses key phrases to match user queries with relevant forum threads and reinforcement activitie. The gap in research presents an opportunity for further exploration, as integrating conversational systems into online course recommendations could enhance user satisfaction and improve the relevance of suggestions by leveraging interactive dialogue to capture and address individual learning needs and preferences.

The proposed conversational course recommendation system in this project aims to advance this field by integrating LSTM-based joint intent detection and slot filling with LLM-driven retrieval augmented response generation. Unlike traditional systems, which rely on static profiles and simplistic matching algorithms, this approach leverages advanced NLP techniques to dynamically interpret and respond to user queries. The use of LSTM networks for intent detection and slot filling allows for a deeper understanding of user queries, enabling the system to capture complex user intents and extract relevant information accurately. Additionally, the incorporation of LLMs for response generation provides contextually appropriate and coherent recommendations based on the extracted information. This novel approach promises to address the limitations of previous course recommendation research by offering more personalized, context-aware recommendations and enhancing user satisfaction through interactive dialogue.

# Methodology

**3.1. System Overview**

The Conversational Online Course Recommendation System is designed with a modular architecture that integrates two core components: the LSTM-based intent and slot detection module and the LLM-based response generation module. The system begins with the data preprocessing phase, where user queries and course data are prepared for modeling, then we carry out some data exploration to get some insights on the data.

Initially, the system explored various TFIDF, sentence embeddings, and Word2Vec embedding techniques to create course embeddings and compare them with user query embeddings. These traditional methods were found to be inadequate in capturing the nuanced semantic relationships needed for accurate course recommendations. To address these inefficiencies, the system adopted a more advanced approach using LSTM networks to accurately detect user intents and extract relevant slots. The extracted intents and slots are then used to guide the response generation process.

For generating contextually relevant and coherent responses, the system uses the Microsoft/Phi-3-mini-4k-instruct LLM. This model was chosen over other LLMs for its optimal balance between computational efficiency and response quality, as well as its effectiveness in handling specific prompts crafted through prompt engineering techniques. Prompt engineering involves crafting specific inputs to guide the LLM in generating the desired responses without altering the model's underlying parameters. This approach not only mitigates the need for extensive training data but also allows for rapid experimentation and iteration with different prompts to achieve optimal performance. By leveraging prompt engineering, the system effectively addresses the challenge of data scarcity for fine-tuning, while still benefiting from the advanced capabilities of the Microsoft/Phi-3-mini-4k-instruct LLM.

Each of these processes will be discussed in detail in the subsequent sections of this chapter, providing a comprehensive overview of the methodologies and their implementation within the system.

**3.2. Data Overview**

The data used in this project was ethically scraped from Class Central website, a popular online platform for discovering and reviewing online courses. The scraping process was conducted using Scrapy, a powerful and versatile web scraping framework, ensuring that the data collection adhered to ethical guidelines and terms of service. The dataset focuses on courses within the domains of programming, computer science, and data science, all of which are taught in English. This targeted approach ensured that the dataset remained relevant to the scope of the project and manageable in size. I aim to leverage this rich dataset to build and evaluate a recommendation system that can suggest relevant courses to users based on their input queries.

The dataset includes comprehensive details about 35,188 courses in CSV format, with a total size of

53.9MB. The columns that capture important aspects of each course in the dataset are:

* **additional\_course\_detail**: Provides supplementary information about the course, including unique aspects such as prerequisites, key learning outcomes, or special features not detailed in other columns.
* **avg\_rating**: Represents the average rating given to the course by users, calculated from individual reviews, reflecting the overall user satisfaction and quality of the course.
* **has\_certificate**: Indicates whether the course awards a certificate upon successful completion. This can be a distinguishing feature for users seeking formal recognition of their learning.
* **institution**: Identifies the educational institution or organization responsible for creating and delivering the course content, such as a university, college, or specialized training provider.
* **is\_university**: A boolean field specifying whether the course is offered by a university (Yes) or by another type of institution (No), providing insight into the type of provider.
* **level**: Categorizes the course based on its difficulty and intended audience, such as beginner, intermediate, or advanced, helping users select courses that match their skill level.
* **link**: Contains the URL to the course’s webpage, where users can find additional information, enroll, or access course materials. It provides direct access to the course for users who wish to obtain more information or enroll.
* **course\_name**: The official title of the course as listed on the platform, which provides a concise identifier for the course.
* **num\_rating**: The total number of ratings or reviews the course has received, indicating the level of feedback and engagement from users.
* **provider**: Details the platform or organization that hosts and delivers the course, such as Coursera, Udemy, or edX, giving context on where the course is available.
* **subject**: Describes the primary subject area or field of study covered by the course, such as Python Programming, Data Science, or Machine Learning, which helps in categorizing the course content.
* **type**: Specifies the format or category of the course, such as Micro-Credential, Certification, Professional Certificate, Specialization, or Single Course. This column provides insight into the nature of the course, including whether it offers a comprehensive learning track, a single module, or a specialized credential.
* **description**: Provides a detailed overview of the course content, including key topics covered, objectives, and what learners can expect to achieve by the end of the course.
* **duration**: Specifies the estimated time required to complete the course. It helps users gauge the time commitment needed for the course.
* **pricing**: Indicates the cost associated with the course or specifies if the course is free.
* **start\_date**: The scheduled start date of the course, informing users when the course will begin and allowing them to plan their enrollment accordingly.

The columns included in the dataset offer a comprehensive view of each course, supporting various aspects of the recommendation process. In conducting this data collection, the following ethical considerations were observed:

* **Compliance with Terms of Service:** The scraping process was conducted in accordance with Class Central’s terms of service to ensure that no rules were violated during data collection.
* **Transparency and Fair Use:** The data is used solely for educational and research purposes, aiming to enhance the user experience by providing more personalized course recommendations.
* **Attribution**: Proper attribution to the source (Class Central) is maintained, acknowledging the origin of the dataset.

In addition to the primary dataset, data for intent detection and slot filling was sourced from the CLINC150 dataset, which is available through Hugging Face. The CLINC150 dataset is a notable resource for intent classification and conversational modeling. Developed for the CLINC (Conversational Language Understanding in Context) challenge, it features a diverse array of utterances categorized into 150 distinct intents, covering scenarios such as greetings, thanks, out-of-scope queries (OOS), and goodbyes. By incorporating the CLINC150 dataset, the project leverages pre-existing, well-structured data to enhance the system’s ability to recognize and respond to common conversational intents. This integration not only enriches the training data but also ensures that the system can manage routine conversational elements. While the dataset excels in providing examples for a wide range of conversational intents, it does not include intents related to course recommendations or domain-specific queries pertinent to online education. As a result, additional domain-specific data was necessary to effectively train the model for the specific task of course recommendation. Hence, for the recommendation aspect, manually generated data and augmentation techniques were employed to ensure comprehensive coverage of course-related intents and slot filling in such a way that they have as much variability and robustness as possible.

**3.3. Data Preprocessing**

The data preprocessing phase involved several critical steps to prepare both the course data and user queries data for effective use in the Conversational Online Course Recommendation System. This process ensured that the data was clean, consistent, and suitable for model training and evaluation.

Preprocessing of Course Data

1. **Creation of Combined Details Column:** A new column, combined\_details, was created to consolidate all relevant information from the original columns into a single text field. This column aggregated data from course\_name, description, additional\_course\_detail, and other relevant columns to provide a comprehensive overview of each course.
2. **Text Cleaning:**
   1. **Punctuation Removal:** Punctuations were removed from the combined\_details column to standardize the text. However, the dot (.) was retained in numerical values (e.g., 4.5 rating) to preserve their integrity.
   2. **Lemmatization:** The combined\_details column underwent lemmatization using the spaCy library. Lemmatization reduces words to their base or root form, aiding in more accurate text analysis. The spaCy tokenizer was configured to avoid splitting words at characters like underscores (\_), hyphens (-), and slashes (/), ensuring that multi-word terms and hyphenated phrases were preserved correctly.

Preprocessing of Intent and Slot Data

1. **Lemmatization of User Queries:** User queries were converted to lower case and lemmatized to reduce words to their root form, which helps in normalizing the text and improving the model’s ability to generalize across different variations of the same word.
2. **Data Splitting:** The dataset was divided into training and test sets, with 80% of the data used as the training data and the remaining 20% as the test data. This split ensures that the model can be trained on one subset of the data and evaluated on another, helping to assess its performance and generalization ability.
3. **Text Vectorization and Label Encoding:**
   1. **Text Vectorization:** After splitting the lemmatised data, the training and test data were transformed separately into numerical vectors using keras TextVectorization module, which is a preprocessing layer in TensorFlow designed to convert raw text data into numerical representations suitable for machine learning models. It does this by first tokenizing the text into manageable units, such as words in our case, then standardizes the text through operations like converting to lowercase and stripping punctuation (I didn’t apply standardization as the text have already being converted to lowercase during lemmatization and I didn’t want to remove the punctuations). Next, it builds a vocabulary from the training data, mapping each unique token to an integer index. This numerical representation allows the layer to convert text into sequences of integers. It also handles variable-length sequences by padding or truncating them to a specified length.
   2. **Label Encoding:** The intents and slots tags were encoded using label encoding techniques. This process converts categorical labels into numerical values, allowing the machine learning models to interpret and process these labels effectively.

By implementing these preprocessing steps, both the course data and the intent and slot data were prepared in a manner that supports effective model training and accurate prediction. The preprocessing ensured that the text data was clean, consistent, and appropriately formatted for further analysis and model development.

**3.4. Implementation Details**

In the development of the conversational course recommendation system, I used a sophisticated Retrieval-Based Response Generation methodology that leverages the Microsoft/Phi-3-mini-4k-instruct LLM to generate tailored responses based on the detected user intent and extracted slots. Below is a breakdown of the process

**3.4.1. LSTM-Based Intent Detection and Slot Extraction**

The process begins with the system detecting the user’s intent and extracting relevant slots from the user query. This intent and slot detection step is critical, as it is crucial for understanding the user's needs and determining the courses that will be retrieved from the courses dataset and passed to the LLM. The slots are specific pieces of information, such as the topic, course rating, duration and other attributes that help refine the recommendations or the details provided. The intents considered in this system are:

* **recommend**: The user is seeking course recommendations.
* **details**: The user is requesting specific details about a course.
* **oos (out of scope)**: The user's query is outside the scope of the recommendation system.
* **courtesy**: The user is engaging in polite conversation, such as greetings, thank-yous, or farewells.

I used a LSTM-based joint intent and slot detection model to classify the user intents and extract relevant slots from the user queries. Combining intent detection and slot filling into a single model allows for shared learning and improved performance by leveraging the correlations between intents and slots. Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) designed to address the limitations of traditional RNNs, particularly their struggles with learning long-term dependencies and avoiding vanishing gradients. LSTM networks feature a unique architecture that includes memory cells and gating mechanisms to regulate the flow of information. Each LSTM unit comprises three gates: the input gate, which controls the addition of new information to the memory cell; the forget gate, which determines which information to discard; and the output gate, which decides what information to output based on the current memory state. This structure allows LSTMs to maintain and manipulate information over long sequences, making them particularly effective for tasks involving sequential data such as time series prediction and natural language processing. By overcoming the limitations of standard RNNs, LSTMs have become a cornerstone in modeling complex dependencies and temporal patterns in various machine learning applications. This section describes the model architecture, hyperparameter tuning, and evaluation process used to develop and optimize the intent and slot detection system.

#### **Model Architecture**

The model is built using a sequence of layers tailored to handle text input and perform both intent classification and slot detection.

1. **Embedding Layer**: The architecture begins with an Embedding layer that converts input text sequences into dense vector representations. The dimensionality of these embeddings is determined by a hyperparameter that is optimized during the tuning process.
2. **Bidirectional LSTM Layer**: Following the embedding layer, a Bidirectional LSTM layer processes the text sequences to capture context from both directions in the sequence. The number of units in this layer is another hyperparameter that is tuned.
3. **Stacked LSTM Layers**: The model includes additional LSTM layers, with the number of layers and units being dynamically set based on hyperparameters. Each LSTM layer returns sequences, allowing the model to learn complex temporal dependencies in the data.
4. **Dense Layers**: The output from the LSTM layers is then passed through one or more Dense layers with optional dropout for regularization. The number of dense layers, units, activation functions, and dropout rates are all tuned to optimize model performance.
5. **Intent Output**: A final Dense layer with a softmax activation function is used to predict the intent of the input text. This layer is designed to output probabilities for each possible intent class.
6. **Slot Output**: For slot detection, a TimeDistributed layer applies a Dense layer to each time step of the sequence, predicting slot labels for each token in the input. The slot output is also processed through a softmax activation to classify slot tags.

#### **Hyperparameter Tuning**

Hyperparameter tuning was performed using Keras Tuner, focusing on optimizing both the intent classification accuracy and slot detection F1 score. The tuning process involved:

* **Search Space**: Hyperparameters such as the number of LSTM and Dense layers, units in each layer, dropout rates, activation functions, and optimizers were explored. The embedding\_dim and batch size were also optimized.
* **Callbacks**: The ReduceLROnPlateau and EarlyStopping callbacks were used to adjust the learning rate and prevent overfitting, respectively.
* Validation: A validation split of 20% of the training data was used to monitor performance on a hold-out set. The validation data was evaluated based on the intent output accuracy and slot output F1 score. F1 score provides a balance between precision (the proportion of correctly identified slots relative to all identified slots) and recall (the proportion of correctly identified slots relative to all actual slots). It is particularly useful for tasks with imbalanced classes or where false positives and false negatives are both significant. The F1 score for slot detection was computed using a custom macro F1 score metric.

After hyperparameter tuning, the model was re-trained to capture the training history, using the best hyperparameters returned by keras tuner, which was then used to plot performance metrics.

#### **Model Evaluation**

Post-tuning, the optimized model was evaluated on a test dataset. The evaluation involved:

* **Performance Metrics**: The model's performance was assessed using classification reports, which provided detailed metrics such as precision, recall, and F1 score for both intent classification and slot detection.
* **Analysis**: The classification report and performance metrics were analysed to gauge the model's effectiveness in accurately predicting user intents and extracting slots.

Overall, the LSTM-based intent and slot detection model demonstrates a robust architecture capable of handling complex conversational data, with optimized hyperparameters ensuring high accuracy and effective slot extraction. Once the intent and slots are identified, the system retrieves the most relevant courses from the database, based on the detected intent and slots. These retrieved courses are then passed to the LLM for response generation.

* + 1. **Retrieval of Matching Courses**

After the intent and slots are identified, the system queries the courses dataset to retrieve those that match the user’s intent and provided slots. This retrieval step ensures that the LLM receives a set of relevant courses that align with the user's needs. Matching courses are retrieved using a combination of natural language processing techniques, including word embeddings and vector-based similarity methods. The courses are filtered data based on the slots detected from the user query, using different filtering criterion for each slot. For textual attributes like course name, topic, institution, provider, and others, the function applies either TF-IDF or Word2Vec embeddings to transform these attributes into vectors. These vectors are then used to calculate the similarity between the user's query and the courses in the database.

After transforming the course attributes into vectors, the function calculates the cosine similarity between the vector representing the user’s query and each course vector. Cosine similarity is a measure of similarity between two vectors, considering their orientation in a multi-dimensional space. This similarity score helps rank the courses, with higher scores indicating more relevant matches. Finally, the top n courses with the highest similarity scores are selected as the final recommendations and then passed to the LLM to generate a user-friendly response.

* + 1. **LLM Retrieval-Based Response Generation**

Large Language Models (LLMs) are pre-trained on vast amounts of text data and are capable of generating human-like text. They understand context and can generate coherent and contextually appropriate responses. The Phi-3-Mini-4K-Instruct is a 3.8B parameters, lightweight, state-of-the-art open instruction tuned model trained with the Phi-3 datasets that includes both synthetic data and the filtered publicly available websites data with a focus on high-quality and reasoning dense properties. The model belongs to the Phi-3 family with the Mini version in two variants 4K and 128K which is the context length (in tokens) that it can support. It is great for general purpose AI systems and applications which require memory constrained environments, latency bound scenarios, and strong reasoning. The model's architecture includes several transformer layers that utilize self-attention mechanisms to capture relationships between tokens in a sequence, allowing it to generate nuanced and context-aware responses. Its optimization for instruction-following tasks makes it suitable for conversational AI systems and other applications where precise, contextually relevant text generation is crucial.

The key advantage of using retrieval-based text generation is its ability to combine specific, up-to-date information with the advanced language understanding of large language models (LLMs). This approach not only helps prevent hallucinations (where the model might generate plausible but incorrect information) but also tailors responses directly to our specific dataset and use case. By dynamically integrating relevant data into its responses, the system ensures that the generated content is contextually accurate. Unlike fine-tuning, which can be resource-intensive and may require frequent retraining to stay current, retrieval-based generation provides flexibility and efficiency, enabling the system to adapt quickly to changes in the underlying data while maintaining high-quality, use-case-specific responses.

**Instruction Prompt Formulation**

To generate a contextually appropriate and coherent response, the retrieved courses and chat history are passed to the LLM via an instruction prompt. This prompt is carefully crafted to guide the model in generating responses that are aligned with the identified intent. The prompt is structured as follows:

**System Content Setup:** The prompt begins by establishing the role of the LLM as an intelligent course recommendation assistant. The system content section outlines how the model should handle different types of intents:

**Recommend:** When the intent is to recommend courses, the model is instructed to use the provided context to suggest relevant courses and provide brief descriptions for each.

**Details**: If the intent is to provide details, the model should use the context or previous answers to give the user the required information.

**Out of Scope (OOS):** For queries that fall outside the scope of course recommendations or related details, the model is guided to politely inform the user that the query is outside the system's expertise.

**Courtesy:** For interactions involving greetings, thank-yous, or goodbyes, the model is instructed to respond politely.

**User Content Setup:** The second part of the prompt integrates the chat history and current user query. The chat history, including previous user queries and the model's answers, is formatted and included in the prompt. This ensures that the model is aware of the ongoing conversation and can generate consistent and follow-up responses.

After the instruction prompt is fully assembled, it is passed to the Microsoft/Phi-3-mini-4k-instruct LLM. The model processes the prompt and generates a response that aligns with the user’s intent, using the provided context and considering the chat history to maintain the conversation's continuity and handle follow-up questions or clarifications seamlessly, ensuring that the user receives coherent and contextually appropriate information throughout the conversation. The generated responses are assessed using human evaluation.

**References**

**Obeidat, R., Duwairi, R. and Al-Aiad, A.** A Collaborative Recommendation System for Online Courses Recommendations. 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). 10.1109/Deep-ML.2019.00018

Goudar, S., and Shidaganti G. Multi-Parameter Based Educational Recommendation System for Online Courses: A Content Based Filtering Approach. 2023 International Conference on Network, Multimedia and Information Technology (NMITCON). 10.1109/NMITCON58196.2023.10276153

Estrela, D., Batista, S., Martinho, D., Marreiros, G. (2017) ‘A Recommendation System for Online Courses’, Advances in Intelligent Systems and Computing, vol 569. Springer, Cham. (Available at: <https://doi.org/10.1007/978-3-319-56535-4_20>).

Plaza, L., Araujo, L., López-Ostenero, F., Martínez-Romo, J. ‘Automatic Recommendation of Forum Threads and Reinforcement Activities in a Data Structure and Programming Course’, Appl. Syst. Innov. 2023, 6, 83. (Available at: <https://doi.org/10.3390/asi6050083>).

Xiang, Y., Yu, H., Gong, Y., Huo, S., & Zhu, M. (2024). ‘Text Understanding and Generation Using Transformer Models for Intelligent E-commerce

Habib, J., Zhang, S., and Balog, K. ‘IAI MovieBot: A Conversational Movie Recommender System, 2020’. CIKM '20: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Pages 3405 – 340. [https://doi.org/10.1145/3340531.341743](https://doi.org/10.1145/3340531.3417433)

Neamah, A., and El-Ameer, A. ‘Design and Evaluation of a Course Recommender System Using Content-Based Approach (2018)’. 2018 International Conference on Advanced Science and Engineering (ICOASE). <https://doi.org/10.1109/ICOASE.2018.8548789>

Zhang, Y., Chen, X., Ai, Q., Yang , L., and Croft, B. “Towards Conversational Search and Recommendation: System Ask, User Respond (2018)”. CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Pages 177 – 186. <https://doi.org/10.1145/3269206.3271776>

Gao, C., Lei, W., He, X., Rijke, M., and Chua, T. “Advances and challenges in conversational recommender systems: A survey (2021)". AI Open Volume 2, 2021, Pages 100-126, ISSN 2666-6510. <https://doi.org/10.1016/j.aiopen.2021.06.002>