Торайғыров университетінің хабаршысы
ҒЫЛЫМИ ЖУРНАЛЫ

НАУЧНЫЙ ЖУРНАЛ
Вестник Торайгыров университета

Торайғыров университетінің ХАБАРШЫСЫ

Энергетикалық сериясы

1997 жылдан бастап шығады

ВЕСТНИК Торайгыров университета

Энергетическая серия

Издается с 1997 года

ISSN 2710-3420

№ 4 (2024)

ПАВЛОДАР

НАУЧНЫЙ ЖУРНАЛ
Вестник Торайгыров университета

Энергетическая серия
выходит 4 раза в год

СВИДЕТЕЛЬСТВО
о постановке на переучет периодического печатного издания,
информационного агентства и сетевого издания
№ 14310-Ж
выдано
Министерство информации и общественного развития
Республики Казахстан

Тематическая направленность
публикация материалов в области электроэнергетики,
электротехнологии, автоматизации, автоматизированных и
информационных систем, электромеханики и теплоэнергетики

Подписной индекс – 76136

---

https://doi.org/10.48081/FYZZ1289

Бас редакторы – главный редактор
Талипов О. М.
доктор PhD, ассоц. профессор (доцент)

Заместитель главного редактора — Калтаев А.Г., *доктор PhD*
Ответственный секретарь — Сағындық Ә.Б., *доктор PhD*

Редакция алқасы – Редакционная коллегия

Клецель М. Я.,	*д.т.н., профессор*
Никифоров А. С.,	*д.т.н., профессор*
Новожилов А. Н.,	*д.т.н., профессор*
Никитин К. И.,	*д.т.н., профессор (Российская Федерация)*
Алиферов А. И.,	*д.т.н., профессор (Российская Федерация)*
Кошеков К. Т.,	*д.т.н., профессор*
Приходько Е. В.,	*к.т.н., профессор*
Кислов А. П.,	*к.т.н., доцент*
Нефтисов А. В.,	*доктор PhD*
Омарова А. Р.	*технический редактор*

За достоверность материалов и рекламы ответственность несут авторы и рекламодатели
Редакция оставляет за собой право на отклонение материалов
При использовании материалов журнала ссылка на «Вестник Торайгыров университета» обязательна

МРНТИ 28.23.25

https://doi.org/10.48081/SBNX6100

К. С. Беисова[1], Л. А. Авдеев[2], Ш. З. Телбаева[3], С. В. Войткевич[4], В. А. Иванов[5]

[1,2,3,4,5]Карагандинский технический университет имени Абылкаса Сагинова, Республика Казахстан, г. Караганда

[1]ORCID: https://orcid.org/0009-0002-5744-3868

[2]ORCID: https://orcid.org/0000-0003-4720-5823

[3]ORCID: https://orcid.org/0000-0001-6208-4113

[4]ORCID: https://orcid.org/0000-0003-4267-3468

[5]ORCID: https://orcid.org/0000-0003-2811-7908

*e-mail: k.beisova@mail.ru

ДООБУЧЕНИЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ ГЕНЕРАЦИИ И ОЗВУЧИВАНИЯ ОТВЕТОВ

В данной статье исследуются процессы, реализуемые при дообучении большой языковой модели (англ. large language model, далее – LLM) для обработки лекций в университетах и получения ответов на вопросы, заданные студентами, а также для озвучивания этих ответов. Таким образом, целью данного исследования является описание возможностей, возникающих при помощи использования LLM в образовании и, в частности, в автоматизации методов поддержки образовательного процесса. В этом исследовании также описаны этапы подготовки данных, дообучения по образовательным ресурсам во время установки и при необходимости настройки модели. Такая языковая модель, предназначенная для обработки и анализа текстов лекций, разработки алгоритмов генерации ответов на вопросы студентов и их озвучивания, является основным элементом данного исследования. В статье поэтапно описывается, как выбрать и установить необходимые программные компоненты, методика подготовки и обработки данных для последующего обучения модели, а также интеграция технологий преобразования текста в речь. Таким образом, полученные результаты показывают эффективность модели предварительного обучения в отношении понимания вопросов и ответов на них с использованием учебных материалов, что может значительно улучшить качество образовательного процесса. LLM

может быть использована при разработке интеллектуальных систем поддержки студентов и преподавателей за счет повышения интерактивности обучения посредством генерации и озвучивания ответов.

Ключевые слова: большие языковые модели, анализ данных, генерация ответов, установка моделей, дообучение моделей, автоматизация учебного процесса.

Введение

В настоящее время тематика искусственного интеллекта охватывает огромный перечень научных направлений, начиная с таких задач общего характера, как обучение и восприятие, и заканчивая такими специальными задачами, как игра в шахматы, доказательство математических теорем, сочинение поэтических произведений и диагностика заболеваний [1].

Современный термин «глубокое обучение» выходит за рамки нейробиологического взгляда на модели машинного обучения [2]. Наиболее успешные алгоритмы машинного обучения – это те, которые автоматизируют процессы принятия решений путем обобщения известных примеров [3].

Актуальность темы обусловлена растущим интересом к применению искусственного интеллекта в образовательной сфере.

Данная исследовательская работа посвящена рассмотрению процесса установки и дообучения LLM, направленного на улучшение качества образовательного процесса. В работе также представлены этапы настройки моделей, а также анализ их эффективности при обработке учебного материала, генерации и озвучивания ответов.

Практическая значимость работы заключается в создании инструментов, которые могут быть использованы для повышения интерактивности и доступности учебного контента.

Материалы и методы

Машинное обучение способно выполнять широкий спектр задач: изменение изображений, помощь в письме, обработка звука, генерация текста [4]. Однако в этой статье будет затронута последняя задача из данного списка.

Для генерации ответов использовались такие LLM, как упрощенный вариант модели GPT-2 (англ. Distilled GPT-2, далее – DistilGPT2) и упрощенный вариант модели BERT (англ. Distilled BERT, далее – DistilBERT).

Следующий шаг – это набор образовательных материалов с текстовыми данными, сведенными в таблицу в формате CSV (англ. Comma-Separated Values – значения, разделенные запятыми, далее – CSV) с использованием блоков таблиц и подразделов. Данный набор был использован в качестве обучающих данных для этой модели.

Среди нескольких пакетов или библиотек, использованных в этой работе, стоит упомянуть Python – основной язык программирования для анализа данных. Все вычислительные ресурсы и среды разработки были представлены с использованием среды Google Colab.

Ниже представлены основные библиотеки, использованные в исследовании:

Библиотека gtts (англ. Google Text-to-Speech – преобразование текста в речь, далее – gtts) для преобразования текста в речь и воспроизведения ответов модели в аудиоформате.

Библиотека transformers для работы с языковыми моделями, включающая такие функции, как токенизация текста, настройка и дообучение модели, генерация текстовых ответов.

Токенизация в сфере обработки естественного языка и машинного обучения – это процесс преобразования последовательности текста в более мелкие части, известные как токены. Токены могут быть размером с буквы или символы, или длиной со слова. [5].

Библиотека pandas для обработки данных.

Все эти процессы предполагают подготовку данных, создание модели и дополнительных элементов. Прежде всего, данные об обучении должны быть собраны, а затем переведены в необходимый формат CSV таким образом, чтобы каждая строка соответствовала отдельному блоку учебного материала.

Используемая далее модель DistilGPT2 дополнительно поддерживается библиотекой transformers от компании Hugging Face.

Платформа Hugging Face – это коллекция готовых современных предварительно обученных Deep Learning моделей. А библиотека Transformers предоставляет инструменты и интерфейсы для их простой загрузки и использования. Это позволит сэкономить время и ресурсы, необходимые для обучения моделей с нуля [6].

DistilGPT2 отличается высокой функциональностью для выполнения дальнейшего обучения на подготовленных обучающих данных. Такой подход позволит модели лучше выявлять закономерности в отношении последовательных и контекстуально значимых ответов с учетом предоставленной информации.

Весь процесс работы включал в себя несколько этапов, которые помогли достичь целей, поставленных перед исследованием. Первым этапом являлся сбор текстовых данных, который использовался для дообучения модели. После этого была произведена установка, настройка, а также дальнейшее дообучение модели DistilGPT2, чтобы она давала ответы, соответствующие входным данным. Наряду с этим, использовалась также модель DistilBERT для подтверждения соответствия полученных ответов заданному контексту.

И заключительный этап данного исследования – преобразование текстовых ответов с помощью модуля gTTS в аудио.

Результаты и обсуждение

Для того чтобы понять, насколько эффективно LLM можно использовать в образовательном контексте, необходимо провести всесторонний анализ всех этапов: от настройки до дальнейшей интеграции с модулями, что включает в себя выбор моделей, настройку и дальнейшее обучение, проверку результатов, интеграцию в систему преобразования текста в речь.

Прежде всего, были проведены отбор и подготовка моделей. Для получения ответов в качестве основной модели была выбрана DistilGPT2, которая хоть и представляет собой упрощенный вариант модели GPT-2, но при этом обладает хорошим качеством генерации текстов без использования больших компьютерных мощностей. В связи с этим, с помощью библиотеки Transformers от Hugging Face была проведена настройка и дальнейшее дообучение модели, что позволило ей адаптироваться к входным данным.

На рисунке 1 показана установка библиотек и импорт модулей, которые будут использоваться при работе с большими языковыми моделями и данными:

!pip install datasets – устанавливает набор данных, предоставляющий инструменты для загрузки и предварительной обработки данных.

!pip install gtts – устанавливает библиотеку gtts для преобразования текста в аудиофайлы.

!pip install transformers – устанавливает библиотеку transformers, которая включает в себя инструменты для работы с такими моделями, как DistilGPT2 и DistilBERT.

from transformers import GPT2Tokenizer, GPT2LMHeadModel, Trainer, TrainingArguments, pipeline – импортирует классы и функции из библиотеки transformers. GPT2Tokenizer и GPT2LMHeadModel необходимы в работе с моделью DistilGPT2. Trainer и TrainingArguments настраивают и выполняют обучение. Библиотека pipeline предназначена для упрощения задач, связанных с обработкой естественного языка.

from datasets import load_dataset pipeline – импортирует функцию load_dataset из библиотеки datasets для загрузки наборов данных из библиотек. Таким образом, она подготавливает эти данные для обучения модели.

import pandas as pd pipeline – импортирует библиотеку pandas для обработки данных в табличной форме.

from gtts import gTTS pipeline – импортирует класс gTTS из библиотеки gtts. Класс gTTS расшифровывается также, как и библиотека gtts (Google Text-to-Speech – преобразование текста в речь). Он используется для реализации функции преобразования текста в речь.

from IPython.display import Audio, display pipeline – импортирует функцию Audio и display из модуля IPython.display для воспроизведения.

```
!pip install datasets
!pip install gtts
!pip install transformers

from transformers import GPT2Tokenizer, GPT2LMHeadModel, Trainer, TrainingArguments, pipeline
from datasets import load_dataset
import pandas as pd
from gtts import gTTS
from IPython.display import Audio, display
```

Рисунок 1 – Установка библиотек и импорт модулей

На рисунке 2 представлены шаги для загрузки как токенизатора, так и модели DistilGPT2.

```
tokenizer = GPT2Tokenizer.from_pretrained('distilgpt2')
model = GPT2LMHeadModel.from_pretrained('distilgpt2')
```

Рисунок 2 – Загрузка токенизатора и модели DistilGPT2

На рисунке 3 показаны команды, которые нужны, чтобы проверить, есть ли у токенизатора DistilGPT2 маркер заполнения, и, если нет, как его добавить.

Маркер заполнения – это специальный символ для выравнивания текстов разной длины, поэтому все они будут обрабатываться одинаково.

```
if tokenizer.pad_token is None:
    tokenizer.add_special_tokens({'pad_token': '<PAD>'})
    tokenizer.pad_token_id = tokenizer.convert_tokens_to_ids('<PAD>')
    model.resize_token_embeddings(len(tokenizer))
```

Рисунок 3 – Проверка наличия токена заполнения

На рисунке 4 изображены команды, которые упрощают процесс загрузки данных из CSV-файла в DataFrame, представляющий собой табличный формат данных в библиотеке pandas.

DataFrame – это проиндексированный многомерный массив значений [7].

Следующие команды преобразуют эти данные и токенизируют. Токенизация здесь также выполняет роль заполнения и обрезки последовательности текста до определенной длины для последующего обучения модели.

```
if tokenizer.pad_token is None:
    tokenizer.add_special_tokens({'pad_token': '<PAD>'})
    tokenizer.pad_token_id = tokenizer.convert_tokens_to_ids('<PAD>')
    model.resize_token_embeddings(len(tokenizer))

df = pd.read_csv('training_data.csv', delimiter=",", encoding='utf-8')

texts = df['content'].tolist()

data = {"text": texts}
df = pd.DataFrame(data)
df.to_csv('text_data.csv', index=False)

dataset = load_dataset("csv", data_files={'train': 'text_data.csv'})

def tokenize_function(examples):
    encodings = tokenizer(examples["text"], truncation=True, padding='max_length', max_length=512)
    encodings['labels'] = encodings['input_ids']  # Используем те же идентификаторы токенов как метки
    return encodings

tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

Рисунок 4 – Загрузка и подготовка данных для обучения

Существует два подхода к задаче построения нейронной сети-классификатора. Первый подход заключается в построении сети, варьируя архитектуру. Данный метод основан на точной классификации прецедентов. Второй подход состоит в подборке параметров (весов и порогов) для сети с заданной архитектурой [8].

Ниже на рисунке 5 рассмотрен процесс применения второго подхода, а именно: как устанавливаются некоторые из параметров обучения и как выполняется сам процесс. В этом фрагменте кода задаются следующие параметры: директория для сохранения результатов, стратегия оценки, скорость обучения, которая в основном определяет, насколько сильно обновляются веса модели во время обучения, или, другими словами, параметры, которые модель обучается настраивать во время обучения, размер батча означает, сколько примеров обрабатывается одновременно при одном обновлении параметров модели. Количество эпох представляет собой количество периодов, которые являются скалярным числом, указывающим, как часто алгоритм обучения будет проходить через весь набор обучающих данных, и, наконец, коэффициент веса для регуляризации, который поможет избежать переобучения, добавляя штраф за большие значения весов модели. После настройки всего этого был инициализирован объект класса Trainer с заданными параметрами вместе с выбранным набором данных, и запущено фактическое обучение модели.

Описание объекта – это определение его свойств и методов, которые этот объект может принимать [9].

```
training_args = TrainingArguments(
    output_dir='./results',          # Директория для сохранения результатов обучения
    evaluation_strategy="no",        # Оценка модели во время тренировки не проводится
    learning_rate=2e-5,              # Скорость обучения модели
    per_device_train_batch_size=2,   # Размер батча для обучения модели на каждом устройстве
    per_device_eval_batch_size=2,    # Размер батча для оценки модели на каждом устройстве
    num_train_epochs=3,              # Количество эпох для обучения модели
    weight_decay=0.01,               # Коэффициент веса для регуляризации
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets['train'],
    eval_dataset=tokenized_datasets['train']  # Используем тот же набор данных для оценки
)
trainer.train()
```

Рисунок 5 – Настройка и запуск тренировки модели

На рисунке 6 приведены команды для сохранения и загрузки модели и токенизатора, а также настройка параметров генерации ответа.

```
model.save_pretrained('./results')
tokenizer.save_pretrained('./results')
model = GPT2LMHeadModel.from_pretrained('./results')  # Путь к сохраненной модели
tokenizer = GPT2Tokenizer.from_pretrained('./results')  # Путь к сохраненному токенизатору

question = "Based on the context, what is a robot?"

input_ids = tokenizer.encode(question, return_tensors='pt')

output = model.generate(
    input_ids,                          # Входные данные для генерации текста
    max_length=150,                     # Максимальная длина ответа
    num_return_sequences=1,             # Количество генерируемых последовательностей
    temperature=0.7,                    # Контроль случайности
    top_k=50,                           # Ограничение на количество наиболее вероятных токенов
    top_p=0.9,                          # Условие для nucleus sampling (суммарная вероятность)
    pad_token_id=tokenizer.eos_token_id  # Идентификатор токена паддинга
)

generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
```

Рисунок 6 – Настройка параметров генерации ответа

Следующим шагом является загрузка модели DistilBERT для проверки сгенерированных ответов, а также задаваемый модели вопрос согласно входным данным, как показано на рисунке 7.

```
nlp = pipeline("question-answering", model="distilbert-base-uncased-distilled-squad")

context = '\n'.join(texts)
question = "Based on the context, what is a robot?"
```

Рисунок 7 – Проверка ответа с использованием модели BERT

На рисунке 8 более подробно описывается, как генерируется и отображается ответ на экране, что является конечным результатом всего этого процесса.

```
result = nlp(question=question, context=context)
print("BERT Validation Answer:", result['answer'])
```

```
BERT Validation Answer: an automatic machine that combines the properties of working and information machine
```

Рисунок 8 – Сгенерированный ответ

На рисунке 9 представлен процесс интеграции системы преобразования текста в речь с использованием класса gTTS: создание аудиофайла с сгенерированным ответом, сохранение и воспроизведение его для прослушивания.

Однако это не единственный способ применить данную библиотеку. Она предоставляет множество языков и настроек, таких как скорость речи, голос и многие другие [10].

```
# Шаг 1: Создание объекта gTTS с ответом от BERT
bert_answer = result['answer']  # Извлекаем ответ из результата BER
tts = gTTS(text=bert_answer, lang='en')
```

```
# Шаг 2: Сохранение аудио файла
tts.save("output.mp3")
```

```
# Шаг 3: Озвучивание ответа
audio = Audio("output.mp3", autoplay=True)
display(audio)
```

Рисунок 9 – Создание и воспроизведение аудиофайла с сгенерированным ответом

Выводы

Таким образом, на текущем этапе работы достигнуты обзорные результаты в разработке и интеграции системы, объединяющей генерацию текстовых ответов и преобразование их в аудиоформат. Выбор модели DistilGPT2 для генерации текстов обеспечил эффективное использование

ограниченного количества вычислительных ресурсов при сохранении качества результатов.

Интеграция DistilBERT для проверки качества сгенерированных ответов позволила повысить точность и релевантность выходных данных. Также была реализована интеграция текстово-речевой системы с помощью библиотеки gTTS, что позволило преобразовать текстовые ответы в аудиоформат и воспроизвести их.

Результаты показывают, что данный подход корректно справляется со сложностями, возникающими при генерации и проверке ответов и выполняет свою задачу преобразования таких ответов в аудио. Однако, в перспективе есть возможность рассмотреть способы улучшения модели. Дополнение может быть реализовано с использованием более сложных алгоритмов. Таким образом, система станет более точной, что позволит расширить её возможности и адаптировать её к более сложным задачам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 **Рассел, С., Норвиг, П.** Искусственный интеллект : современный подход, 2-е изд. [Текст] // Издательский дом «Вильямс» – 2007. – 34 с.

2 **Гудфеллоу, Я., Бенджио, И., Курвилль, А.** Глубокое обучение [Текст] // пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М. : ДМК Пресс – 2018. – 32 с.

3 **Мюллер, А., Гвидо, С.** Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными [Текст] // ООО «Альфа-книга» – 2017. – 14 с.

4 **Капаца, Е.** Машинное обучение доступным языком [Текст] // «Автор» – 2023. – 8 с.

5 Что такое токенизация? [Электронный ресурс]. – https://nft.ru/article/chto-takoe-tokenizatsiia-1011 (Дата обращения: 09.09.2024).

6 Введение в библиотеку Transformers и платформу Hugging Face [Электронный ресурс]. – https://habr.com/ru/articles/704592/ (Дата обращения: 09.09.2024).

7 Введение в анализ данных с помощью Pandas [Электронный ресурс]. – https://habr.com/ru/articles/196980/ (Дата обращения: 09.09.2024).

8 **Местецкий, Л. М.** Математические методы распознавания образов [Текст] // МГУ, ВМиК, кафедра «Математические методы прогнозирования» – 2002. – 32 с.

9 Понятие класса, экземпляра класса и объекта в ООП. [Электронный ресурс]. – https://webkyrs.info/page/poniatie-klassa-ekzempliara-klassa-i-obekta-v-oop (Дата обращения: 09.09.2024).

10 Использование библиотеки gTTS [Электронный ресурс]. – https://wdinfo.ru/docs/python/examples-python/sintez-rechi-ili-kak-preobrazovat-tekst-v-golos/#:~:text=gTTS%20%2D%20%D1%8D%D1%82%D0%BE%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0%2C%20%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82,%D0%B2%20%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D1%87%D0%BD%D1%8B%D1%85%20%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B0%D1%85%2C%20%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D0%B0%D1%8F%20MP3. (Дата обращения: 09.09.2024).

11 **Мэтиз, Э.** Изучаем Python : программирование игр, визуализация данных, веб-приложения. 3-е изд. [Текст] // СПб.: Питер – 2020. – С. 318–391.

REFERENCES

1 **Russell, S., Norvig, P.** Iskusstvennyy intellekt: sovremennyy podkhod, 2-e izd. [Artificial Intelligence: A Modern Approach, 2nd ed.] [Text]. – Izdatel'skiy dom «Viliams» – 2007. – P. 34.

2 **Goodfellow, I., Bengio, Y., Courville, A.** Glubokoye obucheniye [Deep Learning] [Text]. – Translated by A. A. Slinkin. – 2-e izd., ispr. – M. : DMK Press – 2018. – P. 32.

3 **Muller, A., Guido, S.** Vvedeniye v mashinnoe obucheniye s pomoshch'yu Python. Rukovodstvo dlya spetsialistov po rabote s dannymi [Introduction to Machine Learning with Python: A Guide for Data Scientists] [Text]. – OOO «Al'fa-kniga» – 2017. – P. 14.

4 **Kapatsa, E.** Mashinnoe obucheniye dostupnym yazykom [Machine Learning in Simple Language] [Text]. – «Avtor» – 2023. – P. 8.

5 Chto takoye tokenizatsiya? [What is Tokenization?] [Electronic Resource]. – https://nft.ru/article/chto-takoe-tokenizatsiia-1011 (Accessed: 09.09.2024).

6 Vvedeniye v bibliotéku Transformers i platformu Hugging Face [Introduction to the Transformers Library and Hugging Face Platform] [Electronic Resource]. – https://habr.com/ru/articles/704592/ (Accessed: 09.09.2024).

7 Vvedeniye v analiz dannykh s pomoshch'yu Pandas [Introduction to Data Analysis with Pandas] [Electronic Resource]. – https://habr.com/ru/articles/196980/ (Accessed: 09.09.2024).

8 **Mestetskiy, L. M.** Matematicheskie metody raspoznavaniya obrazov [Mathematical Methods of Pattern Recognition] [Text]. – MGU, VMiK, kafedra «Matematicheskie metody prognozirovaniya» – 2002. – P. 32.

9 Ponyatiye klassa, ekzemplyara klassa i obyekta v OOP [Concept of Class, Class Instance, and Object in OOP] [Electronic Resource]. – https://webkyrs.info/page/poniatie-klassa-ekzempliara-klassa-i-obekta-v-oop (Accessed: 09.09.2024).

10 Ispol'zovaniye bibliotéki gTTS [Using the gTTS Library] [Electronic Resource]. – https://wdinfo.ru/docs/python/examples-python/sintez-rechi-ili-kak-preobrazovat-tekst-v-golos/#:~:text=gTTS%20%2D%20%D1%8D%D1%82%D0%BE%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0%2C%20%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82,%D0%B2%20%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D1%87%D0%BD%D1%8B%D1%85%20%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B0%D1%85%2C%20%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D0%B0%D1%8F%20MP3. (Accessed: 09.09.2024).

11 **Metiz, E.** Izuchayem Python: programmiruyem igry, vizualizatsiya dannykh, veb-prilozheniya. 3-e izd. [Learning Python: Game Programming, Data Visualization, Web Applications, 3rd ed.] [Text]. – SPb.: Piter – 2020. – P. 318–391.

Поступило в редакцию 12.09.24
Поступило с исправлениями 14.10.24
Принято в печать 04.12.24

**К. С. Беисова[1], Л. А. Авдеев[2], Ш. З. Телбаева[3],*
С. В. Войткевич[4], В. А. Иванов[5]
[1,2,3,4,5]Әбілқас Сағынов атындағы Қарағанды техникалық университеті, Қарағанды қ.
12.09.24 ж. баспаға түсті.
14.10.24 ж. түзетулерімен түсті.
04.12.24 ж. басып шығаруға қабылданды.

ЖАУАПТАРДЫ ҚҰРУ ЖӘНЕ ДАУЫСТАУ ҮШІН ҮЛКЕН ТІЛДІК МОДЕЛЬДЕРДІ ЖЕТІЛДІРУ

Бұл мақалада үлкен тілдік модельді (ағылш. large language model, бұдан әрі – LLM) университеттердегі дәрістерді өңдеуге және студенттер қойған сұрақтарға жауап алуға, сондай-ақ осы жауаптарды айтуға арналған. Осылайша, бұл зерттеудің мақсаты білім беруде LLM қолдану арқылы, атап айтқанда, білім беру процесін қолдау әдістерін автоматтандыруда туындайтын мүмкіндіктерді сипаттау болып табылады. Бұл зерттеу сонымен қатар деректерді

дайындау кезеңдерін, орнату кезінде білім беру ресурстарын оқытуды және қажет болған жағдайда модельді теңшеуді сипаттайды. Дәріс мәтіндерін өңдеуге және талдауға, студенттердің сұрақтарына жауап беру алгоритмдерін жасауға және оларды айтуға арналған мұндай тілдік модель осы зерттеудің негізгі элементі болып табылады. Мақалада қажетті бағдарламалық жасақтама компоненттерін қалай таңдауға және орнатуға болатындығы, модельді кейіннен оқыту үшін деректерді дайындау және өңдеу әдістемесі, сондай-ақ мәтінді сөйлеуге арналған технологияларды интеграциялау кезең-кезеңімен сипатталған. Осылайша, алынған нәтижелер білім беру процесінің сапасын айтарлықтай жақсарта алатын оқу материалдарын пайдалана отырып, сұрақтар мен жауаптарды түсінуге қатысты алдын ала оқыту моделінің тиімділігін көрсетеді. LLM жауаптарды генерациялау және дауыстау арқылы оқытудың интерактивтілігін арттыру арқылы студенттер мен оқытушыларды қолдаудың интеллектуалды жүйелерін әзірлеуде пайдаланылуы мүмкін.

Кілтті сөздер: үлкен тілдік модельдер, деректерді талдау, жауаптар генерациясы, модельдерді орнату, модельдерді оқыту, оқу процесін автоматтандыру.

**K. S. Beisova[1], L. A. Avdeev[2], Sh. Z. Telbaeva[3], S. V. Voitkevich[4], V. A. Ivanov[5]*
[1,2,3,4,5]Abylkas Saginov Karaganda Technical University, Karaganda
Received 12.09.24
Received in revised form 14.10.24
Accepted for publication 04.12.24

FINE-TUNING OF LARGE LANGUAGE MODELS FOR GENERATING AND VOICING ANSWERS

This article examines the processes implemented during the completion of the large language model (English: large language model, hereinafter referred to as LLM) for processing lectures at universities and obtaining answers to questions posed by students, as well as for voicing these answers. Thus, the purpose of this study is to describe the opportunities that arise through the use of LLM in education and, in particular, in automating methods to support the educational process. This study also describes the stages of data preparation, additional training on educational resources during installation and, if necessary, model configuration. Such a language

model, designed for processing and analyzing lecture texts, developing algorithms for generating answers to students' questions and voicing them, is the main element of this study. The article describes step by step how to select and install the necessary software components, the methodology of data preparation and processing for subsequent training of the model, as well as the integration of text-to-speech technologies. Thus, the results obtained show the effectiveness of the pre-learning model in terms of understanding questions and answering them using educational materials, which can significantly improve the quality of the educational process. LLM can be used in the development of intelligent support systems for students and teachers by increasing the interactivity of learning through the generation and voicing of responses.

Keywords: large language models, data analysis, response generation, model installation, model completion, automation of the educational process.

Теруге 04.12.2024 ж. жіберілді. Басуға 30.12.2024 ж. қол қойылды.
Электронды баспа
29.9 Mb RAM
Шартты баспа табағы 22,2. Таралымы 300 дана. Бағасы келісім бойынша.
Компьютерде беттеген: А. К. Мыржикова
Корректор: А. Р. Омарова, Д. А.Кожас
Тапсырыс №4317

Сдано в набор 04.12.2024 г. Подписано в печать 30.12.2024 г.
Электронное издание
29.9 Mb RAM
Усл. печ. л. 22,2. Тираж 300 экз. Цена договорная.
Компьютерная верстка: А. К. Мыржикова
Корректор: А. Р. Омарова, Д. А. Кожас
Заказ № 4317

«Toraighyrov University» баспасынан басылып шығарылған
Торайғыров университеті
140008, Павлодар қ., Ломов к., 64, 137 каб.

«Toraighyrov University» баспасы
Торайғыров университеті
140008, Павлодар қ., Ломов к., 64, 137 каб.
67-36-69
E-mail: kereku@tou.edu.kz
www.vestnik-energy.tou.edu.kz