

Возможности Google Colab и Jupyter Notebook для решения задач искусственного интеллекта

Е. В. Абрамова¹, Л. А. Максименко^{1},*

¹ Сибирский государственный университет геосистем и технологий, г. Новосибирск,
Российская Федерация

* e-mail: maksimenko_la@mail.ru

Аннотация. В данной статье рассмотрены возможности программного обеспечения Google Colaboratory и Jupyter Notebook для решения задач искусственного интеллекта. Были изучены основные функции и методы использования интерактивного блокнота Jupyter Notebook и облачного сервиса Google Colaboratory. Примером реализации полученных знаний явилось создание чат-бота «Организатор конференций и семинаров» в Colab, а также решение простых арифметических задач в блокнотах Jupyter Notebook. В ходе работы были рассмотрены основные принципы работы с библиотеками, написание кодов, а также создание чат-бота без использования машинного обучения.

Ключевые слова: Google Colaboratory, Jupyter Notebook, чат-бот, облачный сервис, машинное обучение, пользователь, Python

The capabilities of Google Colab and Jupyter Notebook for solving artificial intelligence problems

E. V. Abramova¹, L.A. Maksimenko¹

¹ Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation

* e-mail: maksimenko_la@mail.ru

Abstract. This article discusses the capabilities of the Google Collaboratory and Jupyter Notebook software for solving artificial intelligence problems. The main functions and methods of using the Jupyter Notebook interactive notebook and the Google Collaboratory cloud service were studied. An example of the implementation of the acquired knowledge was the creation of a chatbot "Organizer of conferences and seminars" in Colab, as well as the solution of simple arithmetic problems in Jupyter Notebook notebooks. In the course of the work, the basic principles of working with libraries, writing codes, as well as creating a chatbot without using machine learning were considered

Keywords: Google Colaboratory, Jupyter Notebook, Chatbot, Cloud service, Machine Learning, User, Python

Введение

В настоящее время разработано большое количество программ и приложений для решения задач искусственного интеллекта [1-4]. В данной статье рассмотрены возможности программного обеспечения Google Colaboratory и Jupyter Notebook для решения задач искусственного интеллекта.

Главная особенность Google Colab – бесплатные мощные графические процессоры и тензорный процессор, благодаря которым можно заниматься не только базовой аналитикой данных, но и более сложными исследованиями в об-

ласти машинного обучения. Google Colab реализует максимально простые процессы: в нем есть как базовые библиотеки, так и более сложные, не нужно ставить программы и среды самостоятельно, можно просто сразу написать код. Если же базовых библиотек недостаточно, всегда можно добавить необходимые с помощью установщика PIP.

Jupyter Notebook - это оригинальное веб-приложение для создания вычислительных документов и обмена ими. Он предлагает простой, оптимизированный интерфейс, ориентированный на работу с документами. Работу можно проводить как в браузере, так и в приложении, скаченном на ваше устройство. Основные возможности:

- Jupyter поддерживает более 40 языков программирования;
- записными книжками можно делиться с другими пользователями с помощью электронной почты и программы просмотра записных книжек;
- разработанный код может создавать насыщенный интерактивный вывод: изображения, видео и пользовательские типы;
- возможно использование инструментов для обработки больших данных.

В Jupyter Notebook существует раздел Voilà (Вуаля), что помогает обмениваться информацией, превращая записные книжки в безопасные автономные веб-приложения, которые можно настраивать и использовать совместно. Project Jupyter продвигает открытые стандарты, которые сторонние разработчики могут использовать для создания пользовательских приложений.

Методы и материалы

Anaconda — дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения. Основная особенность дистрибутива — оригинальный менеджер разрешения зависимостей conda с графическим интерфейсом Anaconda Navigator, что позволяет отказаться от стандартных менеджеров пакетов. Благодаря чему работа может проходить в режиме офлайн.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Python используется в анализе данных, машинном обучении и веб-разработке, а также в других сферах, включая разработку игр. За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм. Python является наиболее распространённым ПО для написания собственного чат-бота, способного поддерживать беседу или ответить на ваши вопросы в различных сферах деятельности.

Чат-бот — это программа, которая выясняет потребности пользователей, а затем помогает удовлетворить их. Наиболее популярные приложения для чат-ботов:

- чат-боты для обслуживания клиентов: многие компании используют чат-ботов в качестве первого контакта, когда клиентам нужна помощь;

- чат-боты в электронной коммерции: розничные компании и телекоммуникационные провайдеры используют чат-ботов в качестве дополнительного канала взаимодействия со своими клиентами. Бот необходим для ведения клиентов через линейный поток процессов, чтобы выполнить запросы или транзакции.

- виртуальные помощники в чате: личные помощники, такие как Siri, Алиса, стали популярными, поскольку их преимущества легко доступны, и они без труда внедрились в повседневную жизнь потребителей.

Работа чат-бота происходит согласно заложенной в него программе. Боты бывают разных видов, в зависимости от принципа работы. Основных вида три:

- чат-бот на основе правил. Это базовый чат-бот. Пользователь взаимодействует с ботом, используя predefined параметры, чтобы получить ответ от бота, пользователю необходимо выбрать нужный параметр. Такие боты получают запрос пользователя, анализируют его, а затем предлагают результаты в виде кнопок.

- независимые чат-боты с ключевыми словами – боты, использующие машинное обучение. В отличие от чат-ботов на основе правил, они анализируют то, что хочет пользователь, и реагируют соответствующим образом. Эти боты используют настраиваемые ключевые слова и машинное обучение, чтобы определить, как эффективнее и результативнее реагировать на запросы пользователей.

- NLP (Natural Language Processing - нейролингвистическое программирование) чат-боты. На данный момент они самые продвинутые. Эти чат-боты представляют собой комбинацию лучших чат-ботов на основе правил и ключевых слов. Они используют обработку естественного языка, чтобы понять контекст запросов и намерения пользователей и действовать соответственно. Такие чат-боты могут легко обрабатывать несколько запросов от одного и того же пользователя.

Примером активного использования может послужить чат-бот «Самострой.net», созданный Государственной инспекцией по контролю за использованием объектов недвижимости Москвы. Благодаря чему жители города Москвы имеют возможность получать данные о законности использования интересующего объекта капитального строительства или земельного участка, без визитов и письменных обращений в Госинспекцию по недвижимости. Для того чтобы воспользоваться ботом всего лишь необходимо отправить в чат боту кадастровый номер. Интересующие сведения можно получить в публичной кадастровой карте при вводе адреса. Также бот имеет возможность проверить объект недвижимости при вводе кадастрового номера.

Результаты

Целью работы стало создание чат-бота на актуальную тему: «Организатор конференций и семинаров», который сможет работать без машинного обучения. Создание чат-бота «Организатор конференции семинаров» проводилось в облачном сервисе Google Colaboratory (Google Colab).

Блок приветствия разработанного программного кода приведен на рис.1.

Блок приветствия. Фрагменты кода и диалога <pre>name = input ("Давай познакомимся \n, как тебя зовут? ") print ("Привет,", name) name1 = input ("Твоя фамилия?") print ("Привет, я БОБ, и теперь я тебя запомнил.", name, name1) name2 = int(input ("Сколько тебе лет?")) if name2 > 18: print ("Ок! Лучшее время для участия в Конференции") if name2 < 18: print ("школьники в этой Конференции не участвуют") hello1 = ("Боб очень рад, что дела ") name3 = input ("Как у тебя дела?") greeting2 = hello1 + name3 print (greeting2)</pre>	
Диалог 1: Давай познакомимся Как тебя зовут? <i>Матвей</i> Привет, Матвей Твоя фамилия? <i>Петровский</i> Привет, я БОБ, и теперь я тебя запомнил: Матвей Петровский Сколько тебе лет? <i>21</i> Ок! Лучшее время для участия в Конференции Как у тебя дела? <i>Отлично</i> Боб очень рад, что у тебя дела идут Отлично	Диалог 2: Давай познакомимся Как тебя зовут? <i>Сергея</i> Привет, Сергей Твоя фамилия? <i>Петров</i> Привет, я БОБ, и теперь я тебя запомнил: Сергея Петров Сколько тебе лет? <i>12</i> Боб тебя огорчит - Школьники в этой Конференции не участвуют Как у тебя дела? <i>хорошо</i> Боб очень рад, что у тебя дела идут хорошо

Рис. 1. Блок приветствия

На первом этапе был создан чат-бот, запрограммированный как «1 вопрос – 1 ответ». Для этого был составлен программный код, где был указан вопрос и определенный ответ к нему. Установив значения переменной question, как «Какая у вас тематика секции?», «Какого числа проходит ваша секция?», «В какое время проходит ваша секция?» запустили программу. Результатом стало мгновенное получение ответа, таким образом мы научили бота отвечать на 1 вопрос заранее установленным ответом. На рис. 2 представлен фрагмент протокола чат-бота «Организатор конференций и семинаров».

На втором этапе происходит усовершенствование бота. Были выделены следующие требования: бот должен давать ответ при "неточном" совпадении с шаблоном вопроса; нужно создать базу вопросов и ответов и предусмотреть разные ответы бота на один вопрос. Задачей было запрограммировать бот так, чтобы он смог работать с «неточным» совпадением вопроса. В начале мы заменили ко-

манду на `question = input ()`, чтобы при запуске программа каждый раз запрашивала ваш вопрос. После чего мы задали вопрос и запустили программу. При этом вопрос был составлен иначе, но несмотря на это без проблем получаем ответ. Далее, импортируем библиотеку регулярных выражений (`re`) в блоке подключения библиотек, добавив команду `import re`. Затем в строки импорта библиотеки добавили новую функцию `normalize ()` для удаления знаков препинания и приведения текста к нижнему регистру, а также функцию `get_rank () (text1, text2)` и команды: `text1 = normalize(text1); text2 = normalize(text2)`. Они нужны, чтобы в функции `get_rank` использовать уже отфильтрованный текст (без пробелов, знаков препинания и в нижнем регистре). После чего получили правильный ответ на поставленный вопрос. Протокол работы с «неточным» вопросом представлен на рис.3.

```
# 1 Вопрос = 1 Ответ // первая версия бота
question = "Какая у вас тематика секции?" # Вопрос, который мы задаем боту
# if - оператор условия
if question == "Какая у вас тематика секции?": # если (if) в переменной question содержится
    # текст "Какая у вас тематика секции?", то
    print("Кадастровая деятельность, кадастровый учет и регистрация права") # вывести текст "Кадастровая деятельность, кадастровый учет и регистрация права"
if question == "Какого числа проходит ваша секция?":
    print("24 ноября 2022 года")
if question == "В какое время проходит ваша секция?":
    print("С 14 - 17 ")
```

Рис. 2. Фрагмент протокола написания чат-бота

На втором этапе происходит усовершенствование бота. Были выделены следующие требования: бот должен давать ответ при "неточном" совпадении с шаблоном вопроса; нужно создать базу вопросов и ответов и предусмотреть разные ответы бота на один вопрос. Задачей было запрограммировать бот так, чтобы он смог работать с «неточным» совпадением вопроса. В начале мы заменили команду на `question = input ()`, чтобы при запуске программа каждый раз запрашивала ваш вопрос. После чего мы задали вопрос и запустили программу. При этом вопрос был составлен иначе, но несмотря на это без проблем получаем ответ. Далее, импортируем библиотеку регулярных выражений (`re`) в блоке подключения библиотек, добавив команду `import re`. Затем в строки импорта библиотеки добавили новую функцию `normalize ()` для удаления знаков препинания и приведения текста к нижнему регистру, а также функцию `get_rank () (text1, text2)` и команды: `text1 = normalize(text1); text2 = normalize(text2)`. Они нужны, чтобы в функции `get_rank` использовать уже отфильтрованный текст (без пробелов, знаков препинания и в нижнем регистре). После чего получили правильный ответ на поставленный вопрос. Протокол работы с «неточным» вопросом представлен на рис.3.

```

question = input()

# В каждом условии if используем созданную выше функцию и установим па-
# раметры.
# Если значение функции get_rank < 0.4, то выводится соответствующий от-
# вет
if get_rank(question, "Какая у вас тематика секции?") < 0.4:
    print("Кадастровая деятельность, кадастровый учет и регистрация права")

if get_rank(question, "Какого числа проходит ваша секция?") < 0.4:
    print("24 ноября 2022 года")

if get_rank(question, "В какое время проходит ваша секция?") < 0.4:
    print("С 14 - 17")

какого числа проходит секция
24 ноября 2022 года

```

Рис. 3. Итоговая работа с «неточным» выполнением вопроса

Для усложнения работы чат-бота создаем еще более развитую базу вопросов и ответов. Для этого были произведены некоторые изменения в программном коде и составлены дополнительные вопросы и ответы на них. После запуска бота проверяем его работу на исправность. Фрагмент программного кода чат-бота «Организатор конференций и семинаров» представлен на рис.4.

```

# Добавим боту способность давать разные ответы на один вопрос
for pair in database:
    if get_rank(question, pair["question"]) < 0.4:
        answer = random.choice(pair["answer"])
        print(answer)

какая у вас тематика секции
Применение систем искусственного интеллекта и «больших данных»
в геоинженерии

```

Рис. 4. Программный код чат-бота «Организатор конференций и семинаров» (фрагмент 2)

Принцип работы Jupyter notebook был рассмотрен на примере решения простой арифметической задачи. Для начала работы создаем папку с примерами, присваивая ей имя “Untitled folder”, переименовываем ее в “notebooks”. В данной папке создаем ноутбук. Решаем простую арифметическую задачу: выставим

свойство “Code”, введем в ячейке “2 + 3” без кавычек и нажимаем Ctrl+Enter или Shift+Enter, в первом случае код будет выполнен интерпретатором Python, во втором – будет выполнен код и создана новая ячейка, которая расположится уровнем ниже. В результате мы получаем решение задачи.

Обсуждение

Ответа на вопрос чем занимается искусственный интеллект, не может быть однозначным. Повсеместное использование ИИ создаёт предпосылки для перехода на качественно новую ступень развития, для студентов, специалистов по обработке данных и исследователей в различных областях науки и техники, в том числе и кадастровой деятельности. Рассмотрев подробно Google Colab и Jupyter Notebook для решения задач искусственного интеллекта, мы выделили много различных возможностей. Colaboratory позволяет писать и выполнять код Python в браузере. При этом: не требуется никакой настройки; бесплатный доступ к графическим процессорам; предоставлять доступ к документам другим людям очень просто. В Colab возможно импортировать набор данных изображения, сориентировать на него классификатор изображений и оценить модель с помощью нескольких строк кода. Код в блокнотах Colab исполняется на облачных серверах Google, то есть можно использовать аппаратное обеспечение Google, независимо от мощности вашей машины, нужен только браузер.

Заключение

В проведенной работе были изучены основные функции и методы использования интерактивного блокнота Jupyter Notebook и облачного сервиса Google Colaboratory. Примером реализации стало создание чат-бота «Организатор конференций и семинаров» в Colab, а также решение простой арифметической задачи в блокнотах Jupyter Notebook. В ходе работы были получены основные навыки работы с библиотеками, написания кодов, работы с блокнотами, а также создание чат-бота без использования машинного обучения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р 59276-2020 Национальный Стандарт Российской Федерации Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения Системы искусственного интеллекта. – Текст: электронный: <https://docs.cntd.ru/document/1200177291>
2. Михайлова, И.С. Перспективы использования искусственного интеллекта в сфере образования / И. С. Михайлова, В. В. Шевцов // Современные научные исследования и разработки. - 2018. - № 6 (23). - С. 475-476. – <https://eLIBRARY.ru/item.asp?id=35690383>
3. Агентство «Синьхуа» представило англоязычного ведущего новостей, созданного с помощью ИИ // RusBase: портал. – 2018. – 8 ноября. – <https://rb.ru/story/sinhua-anchor-ai/>
4. Зайцев, А. Тенденции в области искусственного интеллекта. Современные методы машинного обучения / А. Зайцев // Видео наука: сетевой журн. - 2018. - №1(9). – URL: <https://videonauka.ru/stati/32-vystavkikonferentsii-seminary/182-tendentsii-v-oblastiiskusstvennogo-intellekta-sovremennye-metodymashinnogo-obucheniya>

© Е. В. Абрамова, Л. А. Максименко, 2023