

ПРОЕКТИРОВАНИЕ ЧАТ-БОТОВ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ RASA

DESIGNING CHATBOTS USING THE RASA LIBRARY

**A. Pimanov
V. Samokhina**

Summary. Rasa is a popular open-source tool for developing intelligent chatbots. The text describes in detail the steps for installing Rasa on Windows and Ubuntu operating systems, including creating a virtual environment, configuring locales, and installing the necessary libraries. A detailed description of the creation and configuration of the Rasa project is given, including the definition of intents, sequences of interactions (stories) and areas of knowledge (domain) of the bot. The process of training a bot is also considered, including interactive training that allows you to adjust the model in real time based on real dialogues. The material highlights the possibilities of integrating Rasa with other applications through the REST API, which allows the use of chatbots in various platforms and services.

Keywords: Rasa, chatbot, machine learning, natural language analysis, intentions, entities.

Rasa — инструмент с открытым исходным кодом для разработки чат-ботов, основанных на машинном обучении. Rasa состоит из Rasa NLU и Rasa Core. Rasa NLU анализирует естественный язык, затем определяет намерения (intents) и сущности из пользовательского ввода. Rasa Core принимает решения в ходе диалогов, опираясь на подходы, основанные на машинном обучении, при этом учитывая контекст предыдущих сообщений.

Библиотека Rasa не требовательная к ресурсам системы. Библиотека отлично работает на центральном процессоре (CPU) и не требует графического процессора (GPU), что упрощает запуск на обычных машинах без необходимости в дорогом оборудовании.

Rasa можно установить на различных операционных системах и платформах. Основные поддерживаемые системы включают: Windows; Linux; macOS; Docker; об-

Пиманов Андрей Евгеньевич
Технический институт (филиал),
Федеральное Государственное Автономное
Образовательное Учреждение Высшего Образования
«Северо-Восточный федеральный университет
имени М.К. Аммосова», г. Нерюнгри
epimanov15@gmail.com

Самохина Виктория Михайловна
Кандидат педагогических наук, доцент, Технический
институт (филиал), Федеральное Государственное
Автономное Образовательное Учреждение
Высшего Образования «Северо-Восточный федеральный
университет имени М.К. Аммосова», г. Нерюнгри
vsamokhina@bk.ru

Аннотация. Rasa является популярным инструментом с открытым исходным кодом для разработки разговорных чат-ботов. В материале подробно описаны шаги по установке Rasa на операционных системах Windows и Debian, включая создание виртуального окружения, настройку локалей и установку необходимых библиотек.

В статье описаны шаги по созданию и настройке проекта Rasa, включая определение намерений (intents), последовательностей взаимодействий (stories) и областей знаний (domain) бота. Также рассматривается процесс обучения бота, включая интерактивное обучение, позволяющее корректировать модель в реальном времени на основе реальных диалогов. Материал освещает возможности интеграции Rasa с другими приложениями через REST API, что позволяет использовать чат-ботов в различных платформах и сервисах.

Ключевые слова: Rasa, чат-бот, машинное обучение, анализ естественного языка, намерения, сущности.

лачные платформы. Для запуска Rasa рекомендуется использовать версии Python 3.8–3.10.

Установка Rasa отличается от установки большинства Python библиотек. Вариант установки в операционной системе Windows для личного использования, является самым простым из приведённых вариантов. В данном случае рекомендуется создать виртуальное окружение, и выполнить установку Rasa командой — `pip3 install rasa`.

Для установки Rasa на сервере с системой, основанной на Debian, требуется выполнить следующие обязательные шаги. Во-первых, необходимо настроить локали системы. Этот шаг обязателен, если чат-бот должен обрабатывать кириллические символы. Локали определяют язык и региональные настройки, такие как формат даты и времени, валюты и сортировку текста. Для этого в консоли необходимо выполнить команду — `dpkg-reconfigure locales`.

Для обработки кириллицы подойдет локаль — ru_RU.UTF-8. Можно использовать ее на сервере как локаль по умолчанию.

Во-вторых, необходимо установить Miniconda или любой его аналог. Miniconda включает в себя основные компоненты, необходимые для управления пакетами и средами, предоставляя легкий и гибкий способ установки Python библиотек. Установить Miniconda можно последовательно выполнив следующие команды:

Листинг 1

```
wget https://repo.anaconda.com/miniconda/
Miniconda3-latest-Linux-x86_64.sh
sh Miniconda3-latest-Linux-x86_64.sh
```

Третьим шагом является установка менеджера пакетов Pip на сервере. Для этого используется команда — sudo apt-get install python3-pip. Pip облегчает процесс установки, обновления и удаления пакетов Python. Установку Rasa с помощью Pip можно завершить аналогично процессу установки на Windows.

Перед началом обучения необходимо создать проект Rasa, что можно сделать с помощью команды rasa init. После ее ввода потребуется выбрать директорию для размещения начальных файлов проекта. Далее можно запустить процесс обучения начальной модели. После обучения будет предложено протестировать модель, пообщавшись с ней через консольный интерфейс.

На текущем этапе была создана и обучена модель по умолчанию. Для того чтобы научить бота понимать новые намерения, необходимо определить их в файле data/nlu.yml. Намерения состоят из имени и примеров сообщений, которые определяют данное намерение. Ниже приведен файл data/nlu.yml в котором определены намерения приветствия и прощания.

Листинг 2

```
version: «3.1»
nlu:
— intent: greet
examples: |
— hey
— hello
— hi
— intent: goodbye
examples: |
— cu
— good by
— see you later
```

Для каждого намерения необходимо задать минимум 2 примера, описывающих его.

В файле data/stories.yml можно описать последовательности взаимодействий, которые демонстрируют, как бот должен реагировать на пользовательские запросы. Файл состоит из нескольких историй. Каждая история описывает последовательность действий и ответов бота на основе определенных пользовательских намерений. Компонента story имя истории, которое помогает идентифицировать её цель или контекст. Компонента steps внутри каждой истории описываются шаги, которые включают намерения пользователя и действия бота. Компонента intent намерение пользователя, которое распознается моделью NLU. Компонента action действие, которое выполняет бот в ответ на намерение пользователя. Это может быть отправка заранее определенного сообщения или выполнение пользовательского действия (например, запрос к базе данных или вызов API). Пример файла data/stories.yml:

Листинг 3

```
version: «3.1»
stories:
— story: happy path
steps:
— intent: greet
— action: utter_greet
— intent: mood_great
— action: utter_happy
```

Файл domain.yml в Rasa определяет область знаний чат-бота. Он содержит информацию о намерениях, сущностях, слотах, действиях и ответах, которые использует бот. Этот файл является важной частью настройки бота, так как он связывает пользовательские запросы с ответами и действиями бота. Компонента intents определяет намерения, которые бот может распознавать. Компонента entities определяет сущности, которые бот может извлекать из пользовательских запросов. Компонента slots определяет слоты, которые используются для хранения информации в течение диалога. Компонента responses определяет стандартные ответы, которые бот может использовать. Компонента actions определяет пользовательские действия, которые бот может выполнять. Пример файла:

Листинг 4

```
version: «3.1»
intents:
— greet
— goodbye
responses:
utter_greet:
— text: «Hey! How are you?»
```

В файле config.yml параметр language определяет язык, на котором будет обучена модель Rasa, а также язык, на котором она будет обрабатывать пользователь-

ские запросы. Этот параметр указывает, какому языковому окружению соответствует проект, и влияет на выбор токенизаторов и других компонентов, зависящих от языка. Для модели обучаемой ниже, данный параметр был указан, как ru.

Файл domain.yml представленный ниже задает намерения приветствия и прощания.

Листинг 5

```
version: «3.1» intents: — greet — goodbye
```

Файл data/nlu.yml описывает примеры сообщений для ранее определенных намерений на русском языке.

Листинг 6

```
version: «3.1» nlu: — intent: greet examples: | — привет — здравствуй — intent: goodbye examples: | — пока — прощай.
```

Начать обучение с подготовленным набором данных можно командой — `rasa train`, а запустить обученную модель можно с помощью команды — `rasa shell`.

При вводе различных сообщений Rasa вернёт JSON ответ, где будет определено имя намерения (`name`) и точность прогноза (`confidence`). Примеры пользовательского ввода, с намерением и точностью прогноза для выше представленных файлов `domain.yml` и `data/nlu.yml` приведены в таблице.

Таблица 1.

Соответствие пользовательского ввода с определенным Rasa намерением

Текст сообщения	Намерение	Точность
привет	greet	1.0
приветствую	greet	1.0
приве	greet	0.9999997615814209
прив	greet	0.9999837875366211
при	greet	0.9402692914009094

Для всех вводимых сообщений, намерение определено корректно — `greet`. Существует два способа получения ответов со стороны Rasa. В виде заготовленного текста и в формате JSON с выходными параметрами. Итоговый вывод зависит от наполнения файлов `domain.yml` и `data/nlu.yml`.

Rasa также позволяет обучать ботов в интерактивном режиме, непосредственно во время его использования. Интерактивное обучение помогает исправлять ошибки

и улучшать модель на основе реальных диалогов. Для запуска интерактивного обучения используется команда — `rasa interactive`.

Интерактивный сеанс обучения происходит в формате диалога с дополнительными этапами. Пользователь вводит сообщение, после чего Rasa пытается распознать намерение. Далее система запрашивает у пользователя подтверждение, чтобы убедиться в правильности распознанного намерения. На этом этапе возможно два сценария: если намерение распознано корректно, диалог продолжается. Пример такого взаимодействия представлен ниже:

Листинг 7

```
? Your input -> Hello
? Your NLU model classified 'Hello' with intent 'greet' and there are no entities, is this correct? Yes
? The bot wants to run 'utter_greet', correct? Yes
? The bot wants to run 'action_listen', correct? Yes
```

Если намерение было распознано не верно, Rasa предложит список намерений, к которому может отно-

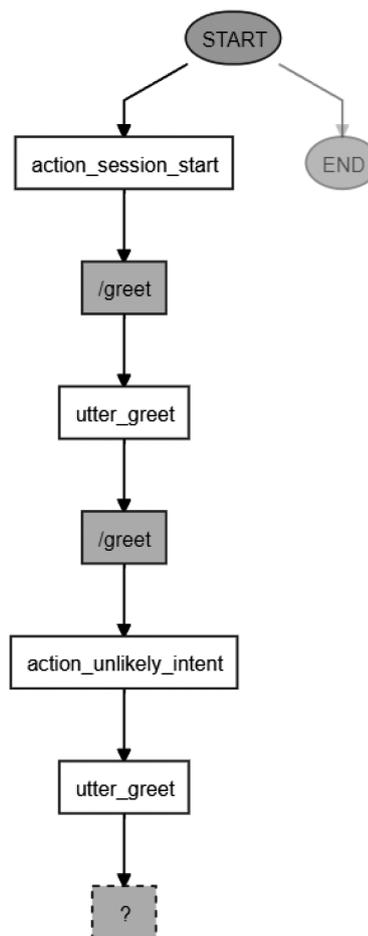


Рис. 1. Блок схема процесса обучения из веб-интерфейса Rasa

ситься сообщение пользователя. Из них можно выбрать существующее или создать новое.

Листинг 8

```
? Your input -> Visit you
? Your NLU model classified 'Visit you' with intent
'goodbye' and there are no entities, is this correct? No
? What intent is it? 0.00 greet
? Please mark the entities using [value](type) notation
Visit you
? The bot wants to run 'action_unlikely_intent' to indicate
that the last user message was unexpected at this point in
the conversation. Check out UnexpectEDIntentPolicy to
learn more. Is that correct? Yes
? The bot wants to run 'action_listen', correct? Yes
```

Завершение каждой развилки сопровождается выводом истории чата. С помощью веб-интерфейса можно ознакомиться с блок схемой процесса обучения. Блок схема текущего обучения приведена на рис. 1.

После завершения обучения Rasa обновит файлы stories.yml, nlu.yml и domain.yml.

Для интеграции с внешними приложениями Rasa предоставляет API. Запуск сервера API осуществляется командой — `rasa run --enable-api`. После ее выполнения Rasa начнет работать в режиме веб-сервера и будет доступна для взаимодействия через REST API. При использовании Rasa API доступны следующие эндпоинты: `POST /webhooks/rest/webhook`, `GET /status`, `POST /model/parse`, `POST /conversations/<conversation_id>/execute`, `POST /model/train`.

Rasa — мощная и гибкая платформа для создания интеллектуальных чат-ботов. Она предоставляет полный набор инструментов для разработки, обучения и развертывания моделей, которые способны понимать и обрабатывать естественный язык. Rasa позволяет создавать сложные диалоги и настраивать логику работы бота, учитывая специфические бизнес-потребности. Используя Rasa, можно создавать ботов, которые поддерживают многошаговые диалоги, учитывают контекст и управляют состоянием сессий.

ЛИТЕРАТУРА

1. Астапов Р.Л. Автоматизация подбора параметров машинного обучения и обучение модели машинного обучения / Р.Л. Астапов, Р.М. Мухаммадеева // Актуальные научные исследования в современном мире. — 2021. — № 5–2(73). — С. 34–37. — EDN GJEUNW.
2. Сивцев Д.А. Использование машинного обучения для совершенствования алгоритма исполнителя «Змейка» / Д.А. Сивцев, А.Н. Гордиенко // Экономика и право: современные интеграционные процессы: сборник научных трудов V Международной молодежной научно-практической конференции, Ставрополь, 18 апреля 2023 года. — Ставрополь: Общество с ограниченной ответственностью «СЕКВОЙЯ», 2023. — С. 117–122. — EDN OXIVJ.
3. Свидетельство о государственной регистрации программы для ЭВМ № 2022668932 Российская Федерация. ТЕРН Интеллектуальный анализ текстов: № 2022667562: заявл. 27.09.2022: опубли. 13.10.2022 / Е.А. Лозовая; заявитель АКЦИОНЕРНОЕ ОБЩЕСТВО «ТЕРН». — EDN DLFYIV.
4. Хорзов Д.С. Библиотеки machine-learning как инструмент начинающего разработчика / Д.С. Хорзов // Культура информационной безопасности: вызовы времени: Материалы XIV Международной научно-практической конференции студентов, аспирантов и молодых учёных, Москва, 23–24 ноября 2023 года. — Москва: Московский государственный лингвистический университет, 2024. — С. 210–216. — EDN JQIIY.
5. Корбут К.Э. Электронное обучение или машинное обучение: четвёртая научно-техническая революция — прогресс или вызов человечеству? / К.Э. Корбут // Интеллектуальный потенциал образовательной организации и социально-экономическое развитие региона: Сборник материалов международной научно-практической конференции Академии МУБиНТ, Ярославль, 08–10 апреля 2019 года / Образовательная организация высшего образования (частное учреждение) «Международная академия бизнеса и новых технологий (МУБиНТ)». — Ярославль: Образовательная Организация Высшего Образования (Частное Учреждение) «Международная Академия Бизнеса И Новых Технологий (МУБиНТ)», 2019. — С. 176–178. — EDN IAJVKL.

© Пиманов Андрей Евгеньевич (epimanov15@gmail.com); Самохина Виктория Михайловна (vsamokhina@bk.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»