

Содержание

Введение	3
Основные результаты практики	4
Настройка инструментов	4
Rasa	5
Заключение	6
Список источников	6

Введение

Темой работы является «Исследование и разработка системы искусственного интеллекта в виде виртуального собеседника». Цель работы - разработать чат-бота, умеющего распознавать 5 сценариев диалога.

Данная задача находится в области NLP. Обработка естественного языка (NLP) – это общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез — генерацию грамотного текста. Решение этих проблем будет означать создание более удобной формы взаимодействия компьютера и человека.

Формой отчетности являлась ежедневная аудио-конференция команды и руководителя длительностью от 15 минут до часа, в ходе которой каждый практикант информировал остальных о том, каких результатов достиг за прошедшие сутки, какой задачей собирается заниматься и существуют ли какие-либо факторы, которые мешают ему выполнять намеченный план.

При разработке нашей командой использовался один из методов гибкой системы управления проектами Agile под названием Scrum. Суть заключается в том, что в течение определенного времени, называемого спринтом, нужно выполнить определенный объем работ, который определяется перед тем, как приступить к задаче. После того, как спринт завершается происходит демонстрация результатов, на которой обсуждаются хорошо реализованные решения и те, которые можно было бы сделать по-другому.

Проект включал в себя этапы подготовки окружения на стороне сервера и построение бота на основе инструментов Rasa [1]. Каждый из членов команды занимался проработкой своего сценария для бота, который по легенде является вымышленным персонажем из игровой вселенной S.T.A.L.K.E.R.

Создание подобных систем представляет собой актуальную проблему для бизнеса, так как зачастую количество сообщений от пользователей во много раз превосходит количество имеющихся сотрудников в штате. Другим важным аспектом актуальности проблемы является стремление интеграции естественного способа общения человека с компьютером. Чат-боты, умеющие распознавать человеческую речь, получили широкое распространение в банковской сфере, в частности они используются при запросе пользователем консультанта. В таком случае бот отвечает на вопросы человека и в крайнем случае передает сообщение эксперту.

Основные результаты практики

1. Настройка инструментов

1.1. Запуск машины в облаке

В качестве платформы для размещения сервера были выбраны облачные решения Amazon Elastic Compute Cloud [2], что позволило в процессе разработки увеличить мощности машины для работы всего необходимого ПО. Наиболее оптимальным выбором для установки операционной системы оказался дистрибутив Linux под названием CentOS.

1.2. Установка веб-сервера

Для распределения нагрузки на сервер и настройки прокси-соединения использовался веб-сервер Nginx [3]. Было настроено HTTPS-соединение с помощью клиента Certbot для Let's Encrypt [4] открытого центра центра сертификации. Данный шаг был необходим для дальнейшей настройки соединения с Telegram API с использованием технологии веб-хуков.

1.3. Настройка Elasticsearch

В двух сценариях диалогов из пяти необходимо отвечать на вопрос пользователя по конкретной сущности. В таком случае необходимо применять поиск по существующим документам.

Распространенным выбором для поисковой машины с нечетким и полнотекстовым поиском по локальным документам является Elasticsearch [5]. В проекте данный инструмент использовался также в качестве хранилища документов, т.к. он умеет автоматически распределять данные на несколько узлов.

1.4. Непрерывная доставка

Для реализации непрерывной доставки применяется программное обеспечение Jenkins [6]. Сама система настроена на связь с удаленным приватным репозиторием с помощью асинхронных ключей шифрования ssh-rsa и технологии веб-хуков, когда при любом изменении в репозитории происходит HTTP POST запрос на сервер с проектом. Данный запрос является сигналом к действию для сборщика.

2. Rasa

2.1. Описание

Rasa представляет собой инструмент для быстрого и удобного построения чат-бота, в основе которого лежат методы машинного обучения с использованием нейронных сетей.

Диалог с пользователем состоит из определения его намерений и заполнения слотов, которые будут использоваться дальше в логике программы. Слотами могут быть имя пользователя, его имя и так далее.

Классификация намерений пользователя осуществляется при помощи машины опорных векторов, которая обучается на выборках, сгенерированных в специальном формате.

Для распознавания сущностей в Rasa применяются нейронная сеть с архитектурой долгой краткосрочной памяти и условные случайные поля. Сущности используются для конкретизации запросов пользователя, определения необходимых заказов и заполнения слотов.

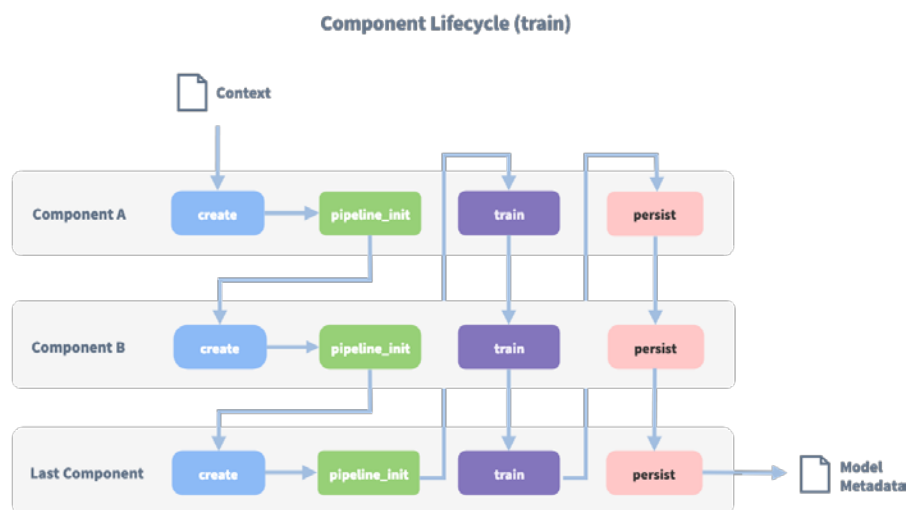


Рис. 1. Схема модели обработки естественного языка в Rasa

2.2. Проработка сценария

Моим сценарием общения бота с человеком является продажа различных предметов виртуальному собеседнику.

При обработке фразы программой первоначально определяется намерение пользователя, далее проверяется наличие сущности в виде описания предмета, и если она присутствует, то заполняется соответствующий слот.

Дальнейшая обработка сценария происходит внутри скрипта. При наличии пустого слота программа выдает сообщение с требованием уточнить

описание предмета. В противном случае отправляется HTTP GET запрос в поисковую машину Elasticsearch, куда заранее были добавлены тексты, содержащие названия, цены и описания предметов. При нахождении подходящего документа происходит его выдача, при пустом ответе выдается сообщение о том, что такого предмета не существует.

Заключение

Поставленные задачи были выполнены в полном объеме и закончены в установленные сроки. Разработанный бот умеет распознавать пять сценариев и работать с каждым из них по отдельности, т.к. полностью выполняет поставленные в начале работы задачи.

Для дальнейшего усовершенствования продукта необходимо усложнить текущие сценарии путем разветвления структуры диалога и увеличением обучающей выборки. Как было упомянуто ранее, проблема разработки подобных систем является актуальной на данный момент. Поэтому все полученные наработки при незначительном изменении можно применять в реальных бизнес-приложениях.

Список источников

1. Rasa – open source machine learning toolkit. [Электронный ресурс]: **URL:** <https://rasa.com/docs/> (дата обращения: 19.04.2019)
2. Amazon EC2 – это веб-сервис, предоставляющий безопасные масштабируемые вычислительные ресурсы в облаке. [Электронный ресурс]: **URL:** <https://aws.amazon.com/ru/ec2/> (дата обращения: 19.04.2019)
3. Nginx – это HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения. [Электронный ресурс]: **URL:** <https://nginx.org/> (дата обращения: 19.04.2019)
4. Let's Encrypt – free, automated, and open Certificate Authority. [Электронный ресурс]: **URL:** <https://letsencrypt.org/> (дата обращения: 19.04.2019)
5. Elasticsearch – distributed, RESTful search and analytics engine. [Электронный ресурс]: **URL:** <https://www.elastic.co/> (дата обращения: 19.04.2019)
6. Jenkins – open source automation server written in Java. [Электронный ресурс]: **URL:** <https://jenkins.io/> (дата обращения: 19.04.2019)