

А.В. Матыкова, Д.Е. Сальников

Тюменский государственный университет, г. Тюмень

УДК 004.912

РАЗРАБОТКА СИСТЕМЫ КЛАССИФИКАЦИИ ОБРАЩЕНИЙ В ОТДЕЛ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

Аннотация. Техническая поддержка играет важную роль в ИТ-компаниях, так как основная ее цель – решение проблем, возникающих при эксплуатации внедренного продукта. В данной статье описана разработка системы, выполняющей часть функций первой линии технической поддержки и решающей проблему круглосуточной доступности технической поддержки для клиента.

Ключевые слова: техническая поддержка, обработка естественного языка (Natural Language Processing, NLP), метод k ближайших взвешенных соседей, наивный байесовский классификатор, чат-бот Telegram, система управления проектами Trello.

Введение

Отдел технической поддержки сопровождает клиентов после внедрения программного продукта, решает возникшие во время его эксплуатации проблемы, проводит консультации, принимает заявки на разработку нового функционала или обучение персонала компании-заказчика. От скорости обработки поступающих обращений во многом зависит лояльность клиента и то, останется ли заказчик с этой компанией или же уйдет к конкуренту, предложившему лучший клиентский сервис. В идеале обращения клиентов должны приниматься круглосуточно, семь дней в неделю. На регистрацию и классификацию обращений должно тратиться минимальное время, необходимо как можно быстрее отправить

клиенту ответ с данными о регистрации и планируемом времени решения его проблемы. Решать такую задачу традиционным способом, предполагающим круглосуточное дежурство группы специалистов технической поддержки, неэффективно, так как существенно возрастают затраты на организацию технической поддержки, а значит и стоимость поддержки для клиента. В свете вышеизложенного, актуальной является задача создания программного обеспечения, выполняющего такие функции первой линии технической поддержки, как: прием и классификация обращений; распределение обращений в очереди задач профильных специалистов; отправка клиентам сообщений о регистрации обращений.

Разработана система для первой линии технической поддержки, состоящая из трех частей:

- чат-бот, который ведет диалог с пользователем при помощи команд меню и позволяет ему начать новое обращение, дополнить или получить статус зарегистрированного ранее;
- классификатор текста, разделяющий поступающие обращения на классы «В работу аналитику» и «В работу разработчику»;
- интеграционный модуль, добавляющий карточки с информацией об обращении в соответствующую очередь задач в системе управления проектами организации Trello.

Чат-бот

Чат-бот, получающий обращения, разработан для мессенджера Telegram. При разработке использовалась библиотека telebot [1], являющаяся оберткой над HTTP-интерфейсом под названием Bot API [2], предоставляемым компанией Telegram. Перед тем, как начать программирование чат-бота, необходимо создать его, обратившись к чат-

боту BotFather. После создания он предоставляет токен для получения доступа к HTTP-интерфейсу. По правилам Telegram, пользовательское имя создаваемого бота должно оканчиваться на «bot».

Разработанный чат-бот может:

- принимать новое обращение в текстовом формате и с вложением;
- предоставлять информацию о статусе обращения;
- предоставлять возможность дополнения обращения новыми сведениями, которые сохранятся в комментариях к карточке;
- предоставлять описание работы с чат-ботом и ссылку на сайт компании с контактными данными.

При первом запуске чата необходимо авторизоваться с помощью токена – строки, предоставляемой клиенту при заключении договора об оказании услуг технической поддержки. Это позволит защитить чат-бот и связанную с ним доску Trello от пользователей, не являющихся клиентами компании, а также привязать ID чата к информации о клиенте, чтобы в дальнейшем отражать ее в создаваемых карточках (организация, контактное лицо, используемая система – Directum или DirectumRX).

Для удобства редактирования контактная информация по каждому клиенту хранится в базе данных системы DirectumRX, используемой в компании. При необходимости изменить что-либо из сведений о клиенте пользователь может зайти в справочник «Контактная информация» и в графическом интерфейсе DirectumRX добавить сведения о новом клиенте, удалить сведения о клиенте, с которым больше не ведется сотрудничество, или отредактировать какой-либо из пунктов информации. Все изменения, совершенные таким образом, отразятся в таблице, с которой ведет работу представленная система.

Извлечение и предобработка данных, обучение классификатора

На момент извлечения данных в организации имелись 2 справочника, содержащих в себе обращения с января 2015 года по май 2020. После анализа извлеченных данных было принято решение классифицировать их на 2 класса: «В работу аналитику» и «В работу разработчику».

Текст обращений содержал в себе цифры, пунктуацию, пробельные символы (переносы строк, табуляцию), поэтому их предобработка заключалась в удалении этих символов, переводе всех букв в нижний регистр, лемматизации слов и векторизации текста обращений. Также перед лемматизацией были удалены стоп-слова путем сравнения всех слов с набором стоп-слов, содержащимся в библиотеке nltk. Лемматизация проводилась с помощью средств библиотеки MyStem [3], векторизация – подходом bag-of-words с использованием меры TF-IDF [4] (данный процесс был выполнен с помощью библиотеки sklearn).

В итоге извлечения и предобработки данных был получен датасет, состоящий из 1341 объекта и 3305 признаков.

Для классификации обращений сравнивались два метода с различными значениями параметров: метод ближайших соседей и наивный байесовский классификатор, реализованные в библиотеке scikit-learn [5]. Метод ближайших соседей показал наилучшие результаты при значении параметров $weight = \text{«distance»}$ (то есть, вариация метода ближайших соседей – метод k ближайших взвешенных соседей) и $k = 9$, наивный байесовский классификатор – при значении параметра $alpha = 0.9$ (параметр аддитивного сглаживания – в данном случае используется сглаживание Лидстоуна).

В результате проведения кросс-валидации с разбиением на 5 блоков наиболее подходящим методов классификации является наивный байесовский классификатор – он показал среднюю точность 74%, в то время как метод k ближайших взвешенных соседей показал более низкую точность, равную 60%. Таким образом, для классификации обращений на 2 вида – «В работу аналитику» и «В работу разработчику» - использовался наивный байесовский классификатор, реализованный в библиотеке `scikit-learn`, с параметром `alpha`, равным 0.9.

Взаимодействие с Trello

После получения информации об обращении из чат-бота создается карточка в Trello. Для работы с Trello используется библиотека `py-trello` [6], также являющаяся оберткой над Trello API [7], предоставляемой компанией. Аналогично разработке чат-бота для Telegram, перед тем, как начать описание взаимодействия системы с Trello, необходимо получить токен доступа к API.

При создании карточки генерируется шестизначный ID, по которому клиент впоследствии может узнать статус обращения или дополнить его, обратившись к чат-боту. С помощью классификации определяется, в какой из списков должна быть помещена эта карточка, и затем происходит непосредственно создание карточки с шаблонным названием и описанием обращения внутри.

Внутри карточки содержится информация об обращении, а именно:

- текст обращения (то, как клиент описал свою проблему);
- контактное лицо (с кем можно связаться для уточнения информации по обращению – данная информация подставляется чат-ботом, исходя из токена и привязанной к нему информации);
- вложение (в виде ссылки на скачивание вложенного файла).

Также на каждую карточку ставится цветная метка, указывающая на используемую в организации заказчика систему – оранжевым обозначается Directum, синим – DirectumRX.

Заключение

Разработанная система способна получать обращения с помощью чат-бота в мессенджере Telegram, классифицировать эти обращения в соответствии с правилами, принятыми в компании, размещать обращения и дополнительную информацию в системе Trello. Система принимает обращения круглосуточно, семь дней в неделю, что позволяет клиенту не привязываться к графику работы техподдержки, получать информацию о состоянии работ по обращению и дополнять его в любое время. Также система позволила снять часть обязанностей со специалиста технической поддержки, дав ему больше времени и высвободив силы на выполнение других задач.

СПИСОК ЛИТЕРАТУРЫ

1. Github – mullwar/telebot. URL: <https://github.com/mullwar/telebot> (Дата обращения: 01.06.2020).
2. Справочник по Telegram Bot API. URL: <https://tlgrm.ru/docs/bots/api> (Дата обращения: 01.06.2020).
3. Segalovich I. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine //MLMTA. – 2003. – С. 273-280.
4. Trstenjak B., Mikac S., Donko D. KNN with TF-IDF based framework for text categorization //Procedia Engineering. – 2014. – Т. 69. – С. 1356-1364.
5. scikit-learn: machine learning in Python. URL: <https://scikit-learn.org/stable/index.html> (Дата обращения: 03.05.2020).

6. Adrien Lemaire, Kyle Valade, Rick van Hattem. py-trello Documentation. URL: <https://readthedocs.org/projects/py-trello-dev/downloads/pdf/latest/> (Дата обращения: 01.06.2020).
7. Trello API documentation. URL: <https://help.trello.com/article/756-trello-api-documentation> (Дата обращения: 01.06.2020).