|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** | | | | |
| БГТУ.СМК-Ф-4.2-К5-01 | | | | |
| Факультет | |  | И |  | Информационные и управляющие системы |
|  | |  | шифр |  | наименование |
| Кафедра | |  | И5 |  | Информационные системы и программная инженерия |
|  | |  | шифр |  | наименование |
| Дисциплина | |  | Технологии программирования | | |

КУРСОВАЯ РАБОТА

на тему

«Автоматизированная справочная система»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | | И5М42 |
| Гусейнов В.Г. | | | | | | |
| Фамилия И.О. | | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | | |
| Арсеньев Б.П. | |  |  | | | |
| Фамилия И.О. Подпись | | | | | | |
| «\_\_\_\_\_» |  | | | | 2018 г. | |

САНКТ-ПЕТЕРБУРГ

2018 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc532489812)

[1 БЛОК ВВОДА ИСХОДНЫХ ДАННЫХ 4](#_Toc532489813)

[2 БЛОК ОРГАНИЗАЦИИ ДАННЫХ 7](#_Toc532489814)

[3 БЛОК АНАЛИЗА ДАННЫХ 8](#_Toc532489815)

[4 БЛОК ВЫХОДНОЙ ИНФОРМАЦИИ 14](#_Toc532489816)

[ЗАКЛЮЧЕНИЕ 15](#_Toc532489817)

[СПИСОК ИСТОЧНИКОВ 16](#_Toc532489818)

ВВЕДЕНИЕ

Одной из сфер, в которой информационные технологии постепенно заменяют человека, является выдача справочной информации. Развитие технологий искусственного интеллекта и машинного обучения привели к появлению ботов, способных распознавать смысл сообщений на естественном языке и давать соответствующие ответы.

У приемной комиссии БГТУ «Военмех» есть группа в социальной сети ВКонтакте, через которую идет связь с абитуриентами. В ней поступающие задают вопросы, многие из которых не раз повторяются. Работники приемной комиссии тратят свое время на то, чтобы снова и снова отвечать на одни и те же вопросы.

Для решения этой проблемы предлагается разработать автоматизированную справочную систему, способную отвечать на распространенные вопросы самостоятельно.

1 БЛОК ВВОДА ИСХОДНЫХ ДАННЫХ

В качестве исходных данных в системы выступают вопросы абитуриентов на естественном языке.

Интерфейсом для ввода данных является раздел «Сообщения» сообщества приемной комиссии в социальной сети ВКонтакте. В этот раздел абитуриенты пишут свои вопросы. На данный момент на эти сообщения отвечают работники приемной комиссии, занимающиеся администрированием этого сообщества. ВКонтакте позволяет подключить к разделу «Сообщений» чат-бот.

Чат-бот – это программа, работающая на отдельном сервере, которая принимает уведомления о новых событиях из ВКонтакте и обрабатывает их.

Для реализации технологии чат-ботов ВКонтакте предоставляет программный интерфейс «API для чат-ботов» [1].

Для реагирования на события, происходящие в сообществе, необходимо о них узнать. Для этого в программной интерфейсе существуют два подхода – Callback API и Long Pull.

Callback API присылает уведомления на сервер, как только в сообществе происходит нужное событие. Событие может быть каким угодно: комментарий к фотографии, новая запись на стене, вступление в сообщество, отправка сообщения, и многое другое.

Для подключения Callback API в сообществе нужно указать адрес программы на сервере и выбрать события, которые нужно получать. Для того, чтобы бот распознавал текстовые команды, необходимо отметить событие «Входящее сообщение».

Уведомления от Callback API выглядят как JSON с основной информацией о событии:

*{   
   "type":"message\_new",   
   "object":{   
      "id":694,   
      "date":1499441696,   
      "out":0,   
      "user\_id":123456,   
      "read\_state":0,   
      "title":" ... ",   
      "body":"start"   
   },   
   "group\_id":1,   
   "secret":"sjr948dff3kjnfd3"   
}*

В этом примере пользователь с id 123456 отправил сообществу сообщение с текстом "start".

Для каждого события Callback API отправляет отдельный запрос на сервер. Программа на сервера должна подтверждать получение каждого запроса, отправляя в ответ строку "ok".

Второй способ получения обновлений — это подключение к Long Poll серверу. В отличие от Callback API, Long Poll сервер будет присылать только те обновления, которые связаны с сообщениями. Никаких других событий из сообщества в нем нет.

Ответ от Long Poll сервера выглядит как массив обновлений в JSON. Вот так выглядит отдельное событие:

*[   
   4,   
   2105994,   
   561,   
   123456,   
   1496404246,   
   "hello",   
   {   
      "attach1\_type":"photo",   
      "attach1":"123456\_417336473",   
      "attach2\_type":"audio",   
      "attach2":"123456\_456239018",   
      "title":" ... "   
   }   
]*

В этом примере пользователь с id 123456 прислал сообщение с текстом "hello" и двумя вложениями — аудио и фотографией.

Long Poll отправляет в ответе на запрос от программы сразу несколько событий. Чтобы сразу получать все обновления, подключение к Long Poll нужно поддерживать постоянно.

Для реализации блока входных данных предлагается использовать Long Poll, так как будет осуществляться обработка только сообщений, приходящих в сообщество.

2 БЛОК ОРГАНИЗАЦИИ ДАННЫХ

Для организации хранения данных требуется две сущности – вопрос и ответ.

Сущность вопроса включает в себя следующие атрибуты: идентификатор, текст вопроса, векторное представление (об этом в следующей главе).

Сущность ответа включает в себя следующие атрибуты: идентификатор, текст ответа.

Инфологическая модель базы данных предоставлена на рисунке 1.

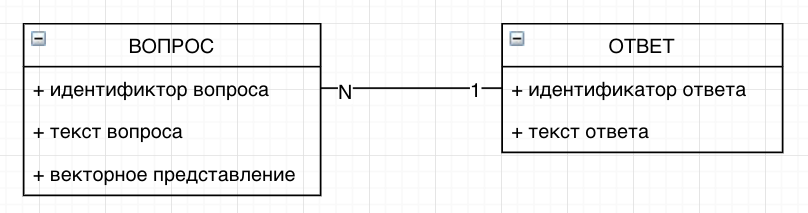


Рисунок 1 – Инфологическая модель базы данных

3 БЛОК АНАЛИЗА ДАННЫХ

Разрабатываемую систему можно отнести к семейству вопросно-ответных – это информационные системы, способные принимать вопросы и отвечать на них на естественном языке.

Такие системы можно разделить на два типа:

1. Узкоспециализированные – работают в конкретных областях (футбол, стоматология, цветы и так далее).
2. Общие – работают с информацией по всем областям знаний.

Второй тип очень сложный в реализации, до сих пор не создано подобных систем, способных отвечать на все поставленные вопросы. Реализацией таких технологий занимаются крупные компании, такие как Google, Yandex, Apple, Amazon.

Система, о которой идет речь в данной работе, относится к первому типу. Ее реализация, по сравнению с общими вопросно-ответными системами, на много проще.

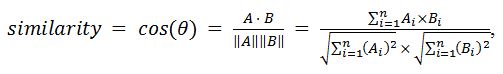
В узкоспециализированной системе заранее определяется конечное множество вопросов, на которые она может давать ответ. В случае, если она не находит у себя в базе вопрос, который с определенной вероятностью соответствует заданному, она отвечает общей фразой, например, «К сожалению, я не могу ответить на данный вопрос».

Таким образом, основная задача системы – поставить в соответствие вопросу, поступившему от абитуриента, вопрос из базы данных.

В основе разрабатываемой системы лежит практика часто задаваемых вопросов в расширенном виде. В структуру системы входит база вопросов по заданной тематике, которые могут задать пользователи. Каждому из них ставится в соответствие ответ, причем одному ответу могут соответствовать несколько вопросов. Таким образом, описанный подход сводит задачу нахождения ответа к поиску вопроса из базы, семантически близкого к заданному. Для того, чтобы решить эту задачу, формируются модели заданного вопроса и каждого вопроса из базы. В качестве моделей используются синтаксические деревья и геометрические вектора. В рамках рассматриваемой задачи сравнение подобных моделей является объективным показателем семантической близости вопросов.

Синтаксическое дерево — это построенный по определенному алгоритму граф, узлами которого являются отдельные части предложения. Ребрам, соединяющим узлы, соответствует их синтаксическая связь. В узле дерева могут располагаться таксономические единицы, отдельные слова предложения, или функциональные единицы, сочетания слов, которые при расщеплении перестают выполнять синтаксическую функцию. Существуют четыре основных алгоритма расположения узлов в графе: теория членов предложения, грамматика Теньера, грамматика зависимостей, грамматика непосредственно составляющих [2]. Теория членов предложения — это алгоритм, в котором в качестве вершины дерева выступает член предложения, не являющийся подчиненным по отношению ни к одной другой синтаксической единице. В соответствии с грамматикой Теньера вершиной синтаксического дерева является глагол-сказуемое. Кроме того, вводятся понятия актанты — функциональной единицы, обязательной по отношению к сказуемому и сирконстанты — необязательной (факультативной) функциональной единицы. Грамматика зависимостей — это алгоритм, при котором в узлах дерева располагаются таксономические единицы. Вершиной дерева является глагол-сказуемое или его аналитическая часть. В случае составного глагола, все связи в дереве подчинительные. Грамматика непосредственно составляющих — это алгоритм, в ходе выполнения которого каждая грамматическая единица делится на две более простых единицы. Подобное деление происходит вплоть до выделения в качестве узла отдельного слова, каждому узлу соответствует грамматический класс, среди которых все части речи, а также именная и глагольная составляющие.

Представление слова в векторном виде — сопоставление слова из словаря геометрическому вектору в пространстве Rn, где под словарём понимается пространство конечной размерности N, равной количеству всех представляемых в векторном виде слов. Задачей определения семантической близости между словами занимается дистрибутивная семантика. Увеличение размерности словесного векторного пространства способствует повышению точности определения смысловой близости, однако существует некоторая критическая размерность, превышая которую, модель не приносит заметного увеличения точности. Обычно размерность вектора устанавливается в диапазоне от 100 до 1000. Любой алгоритм построения векторного пространства стремится к максимизации косинусного сходства между векторами семантически близких слов. Косинусное сходство определяется формулой:

,

где A и B — вектора, расстояние между которыми вычисляется, θ — угол между ними. Одним из наиболее эффективных семантических анализаторов на сегодняшний день является word2vec — программное средство для построения словесных векторных пространств, разработанное компанией Google в 2013 году [3].

Word2vec основан на двухслойной нейронной сети прямого распространения, поэтому у пользователей существует возможность обучить сеть на собственных текстовых корпусах и, таким образом, получить наиболее подходящую для решения текущей задачи векторную модель. Результаты обучения модели зависят от выбранной пользователем модельной архитектуры. Всего в word2vec реализованы два алгоритма обучения: Continuous Bag of Words (CBOW) и Skip-gram.

При использовании архитектуры CBOW алгоритм предсказывает слово, исходя из его контекста, то есть анализируя наборы слов, находящиеся по правую и левую стороны от данного. При этом результат работы алгоритма не зависит от порядка контекстных слов. Входным элементом в нейронную сеть выступает набор контекстных векторов w(t-k),.., w(t-1), w(t+1), ..., w(t+k) , а выходным вектором — w(t), где w(t) — вектор предсказанного на основе контекста слова. Архитектура Skip-gram отличается от CBOW тем, что предсказывает набор слов вокруг, основываясь на данном слове. Входным вектором выступает w(t), а выходным элементом — множество M = {w(t-k),.., w(t-1), w(t+1), ..., w(t+k)}, где M — множество векторов. Каждое слово, соответствующее векторам из множества М, характеризует слово, соответствующее входному вектору. Схема работы алгоритмов CBOW и Skip-gram показана на рисунке 2.

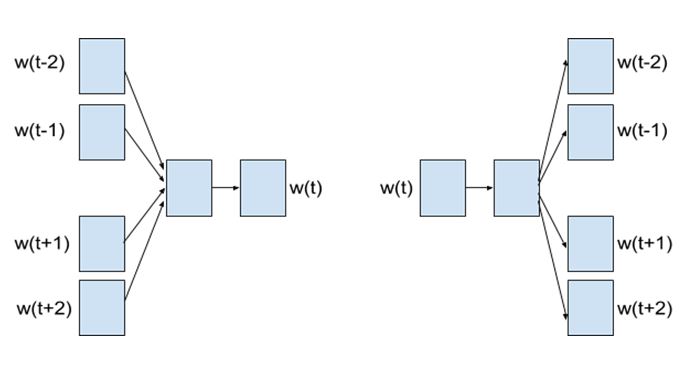


Рисунок 2 – Схема работы алгоритмов CBOW и Skip-gram

Работу word2vec можно разделить на пять этапов. На первом этапе происходит статистическая обработка входного текстового корпуса, то есть для каждого слова рассчитывается количество вхождений его в исходный корпус.

На втором этапе происходит сортировка слов по частоте вхождения, а также, в целях оптимизации работы с памятью, удаляются так называемые гапаксы — слова, встречающиеся редко в сравнении с другими словами текста. Результаты работы данного этапа сохраняются в хеш-таблице.

На третьем этапе для сжатия данных к полученной хеш-таблице применяется код Хаффмана — алгоритм оптимального префиксного кодирования. В результате применения данного алгоритма чаще встречающиеся слова кодируются меньшим количеством бит, а реже встречающиеся — большим.

Четвертый этап заключается в суб-сэмплировании самодостаточной выборки из текстового корпуса (например, предложения или абзаца). В ходе данного процесса из выборки удаляются наиболее часто встречающиеся слова, так как они обычно не несут значимого смысла. Операция суб-сэмплирования применяется для уменьшения времени обучения модели.

Hа пятом этапе к получившейся выборке применяется один из алгоритмов обучения, рассмотренных выше: CBOW или Skip-gram.

Разработанная вопросно-ответная система включает в себя следующие элементы: управляющие скрипты; обученная на Национальном корпусе русского языка векторная Skip-gram-модель в 300-мерном пространстве [4, 5], содержащая более 184000 лемм; база ответов на вопросы, сами вопросы, а также их векторное представление.

Так как технология Word2vec применима как к отдельным словам, так и к текстам, в данном случае являющимся вопросительными предложениями, то с её помощью на основе любой текстовой выборки можно построить соответствующее векторное представление путём суммирования векторов слов, входящих в выборку. Каждое слово из предложения должно быть лемматизировано, так как в векторную модель входят только леммы, то есть слова в словарной форме, а сама форма не имеет значения в определении смысла слова. Некоторые части речи, такие как существительные, глаголы и прилагательные, в большинстве случаев являются семантически значимыми, тогда как другие части речи, например, предлоги, союзы и местоимения не несут смысловой нагрузки. Поэтому при анализе предложения к ним применяется фильтрация, в ходе которой из него исключаются так называемые стоп-слова [6] и знаки препинания. Пример подготовки предложения к построению векторной модели изображен на рисунке 3.



Рисунок 3 – Пример анализа предложения

Процесс реализации вопросно-ответной системы разделён на несколько этапов. Так как система базируется на практике часто задаваемых вопросов, то на первом этапе осуществляется проектирование базы вопросов и ответов.

На втором этапе в базу вносятся векторные представления каждого вопроса, построенные по одному из ранее описанных алгоритмов. Затем этот же алгоритм применяется к вопросу, заданному пользователем системы. Таким образом, система получает необходимую для определения семантической близости информацию, на основе которой может делать предположения о смысловом сходстве или различии заданного вопроса и вопросов из базы.

На заключительном этапе для каждого вопроса из базы и заданного пользователем вопроса вычисляются значения косинусного сходства. Результатом работы системы является ответ на тот вопрос из базы, косинусное сходство с которым максимально.

4 БЛОК ВЫХОДНОЙ ИНФОРМАЦИИ

После того, как система получила ответ на вопрос, заданный пользователем, этот ответ отправляется в чат с ним. Для этого опять используется программный интерфейс ВКонтакте.

Для отправки сообщения используется метод messages.send [1]. В ответе метод вернёт серверу id отправленного сообщения или ошибку. При использовании Callback API, в случае удачной отправки будет сгенерировано событие message\_reply.

Таким образом, сообщение будет доставлено в диалог с пользователем.

ЗАКЛЮЧЕНИЕ

В результате работы были описаны блоки, описывающие входные данные, организацию хранения данных, методы обработки и выходные значения системы для автоматического ответа на вопросы.

Эта разработка может использоваться приемной комиссией БГТУ «Военмех» для автоматизации ответов на вопросы в социальной сети ВКонтакте, которые часто задаются абитуриентами входе приемной комиссии.

СПИСОК ИСТОЧНИКОВ

1. API для чат-ботов [Электронный ресурс] https://vk.com/dev/bots\_docs?f=2.1.%2BCallback%2BAPI (дата обращения 12.12.2018)
2. Касевич В.Б. Структура предложения. Элементы общей лингвистики. — М.: Наука, 1977. —183 с.
3. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed Representations of Words and Phrases and their Compositionality // Proceedings of the 26th International Conference on Neural Information Processing Systems, USA, 2013 – С3111–3119.
4. Kutuzov A., Andreev I. Texts in, Meaning Out: Neural Language Models in Semantic Similarity Task for Russian // Proceedings of the Dialog 2015 Conference. Moscow, Russia, 2015 – С143–154.
5. Официальный сайт проекта RusVectōrēs [Электронный ресурс] http://ling.go.mail.ru (дата обращения 12.12.2018).
6. Manning C.D., Raghavan P., Schütze H. An Introduction to Information Retrieval, Cambridge University Press, Cambridge, England, 2009. —547 с.