(19) Евразийское патентное ведомство

(11) 038264 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2021.07.30

(21) Номер заявки
201990216

(22) Дата подачи заявки
2019.02.04

(51) Int. Cl. ***G06F 16/90*** (2006.01)
G10L 15/12 (2006.01)
G10L 17/18 (2006.01)

(54) СПОСОБ СОЗДАНИЯ МОДЕЛИ АНАЛИЗА ДИАЛОГОВ НА БАЗЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ОБРАБОТКИ ЗАПРОСОВ ПОЛЬЗОВАТЕЛЕЙ И СИСТЕМА, ИСПОЛЬЗУЮЩАЯ ТАКУЮ МОДЕЛЬ

(31) **2019102403**
(32) **2019.01.29**
(33) **RU**
(43) **2020.07.31**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК); ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)" (МФТИ) (RU)

(72) Изобретатель:
Антюхов Денис Олегович, Пугачёв Леонид Петрович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) US-A1-20090037398
WO-A1-2015049198
US-A1-20050005266
US-A1-20010049688
US-B2-9620145
US-B2-8117022

(57) Настоящее техническое решение в общем относится к области вычислительной обработки данных, а в частности к методам машинного обучения для построения моделей анализа диалогов на естественном языке. Предложен компьютерно-реализуемый способ создания модели анализа диалогов на базе искусственного интеллекта для обработки обращений пользователей, выполняемый с помощью по меньшей мере одного процессора и содержащий этапы, на которых получают набор первичных данных, причем набор включает в себя по меньшей мере текстовые данные диалогов между пользователями и операторами, содержащие обращения пользователей и ответы операторов; осуществляют обработку полученного набора данных, в ходе которой формируют обучающую выборку для искусственной нейронной сети, содержащую положительные и отрицательные примеры обращений пользователей на основании анализа контекста диалогов, причем положительные примеры содержат семантически связанный набор реплик оператора в ответ на обращение пользователя; выполняют выделение и кодирование векторного представления каждой реплики из упомянутых на предыдущем шаге положительных и отрицательных примеров обучающей выборки; применяют сформированную обучающую выборку для обучения модели определения релевантных реплик из контекста пользовательских обращений в диалогах.

038264 B1

038264 B1

Область техники

Настоящее техническое решение в общем относится к области вычислительной обработки данных, а в частности к методам машинного обучения для построения моделей анализа диалогов на естественном языке.

Уровень техники

В настоящее время системы автоматизированного распознавания естественного языка получили большое распространение в различных отраслях техники. Наиболее широкое применение данных технологий наблюдается в пользовательском секторе при использовании в различных программных приложениях, например поисковиках, навигаторах, приложениях по подбору товаров и т.п., например при использовании интеллектуальных ассистентов. Ключевой особенностью в работе таких интеллектуальных ассистентов является возможность точного распознавания речевых команд, формируемых пользователями.

Существующей сложностью является формирование моделей анализа речевых сообщений, которые с заданной точностью и скоростью позволяют быстро сформировать и предоставить ответ на запрос пользователя, особенно если речь идет о специализированной области их применения, что требует тщательной настройки и обучения такого рода моделей.

На сегодняшний момент из уровня техники известно достаточно много подходов в области создания и обучения моделей для обработки естественного языка (англ. "NLP" Natural Language Processing). Известен принцип создания моделей с помощью алгоритма машинного обучения, который заключается в применения способа фильтрации предложений с помощью рекуррентной нейронной сети и алгоритма "Мешок слов" (англ. "Bag of words") (патентная заявка US 20180268298, заявитель: Salesforce.com Inc., опубликовано 20.09.2018). Известный подход раскрывает принцип сентиментного анализа с помощью применения двух типов моделей - простой и сложной, которые классифицируют получаемое сообщение на естественном языке. Недостатками известного подхода является низкая точность и скорость работы, что обусловлено применением нескольких моделей, выбираемых в зависимости от типа и сложности получаемого обращения.

Сущность технического решения

Заявленное техническое решение предлагает новый подход в области применения искусственного интеллекта (ИИ) с помощью создания моделей машинного обучения для обработки обращений пользователя на естественном языке.

Решаемой технической проблемой или технический задачей является создание нового способа создания модели анализа обращений на естественном языке, обладающей высокой степенью точности распознавания контекста обращения и скоростью обработки входящих обращений.

Основным техническим результатом, достигающимся при решении вышеуказанной технической проблемы, является создание модели анализа обращений пользователя на естественном языке, обладающей высокой точностью распознавания контекста обращений, за счет обеспечения возможности ранжирования ответов на поступающие обращения пользователей.

Заявленный результат достигается за счет компьютерно-реализуемого способа создания модели анализа диалогов на базе искусственного интеллекта для обработки обращений пользователей, выполняемого с помощью по меньшей мере одного процессора и содержащего этапы, на которых

получают набор первичных данных, причем набор включает в себя, по меньшей мере, текстовые данные диалогов между пользователями и операторами, содержащие обращения пользователей и ответы операторов;

осуществляют обработку полученного набора данных, в ходе которой формируют обучающую выборку для искусственной нейронной сети, содержащую положительные и отрицательные примеры обращений пользователей на основании анализа контекста диалогов, причем положительные примеры содержат семантически связанный набор реплик оператора в ответ на обращение пользователя;

выполняют выделение и кодирование векторных представлений каждой реплики из упомянутых на предыдущем шаге положительных и отрицательных примеров обучающей выборки;

применяют сформированную обучающую выборку для обучения модели определения релевантных реплик из контекста пользовательских обращений в диалогах.

В одном из частных вариантов осуществления способа модель представляет собой по меньшей мере одну искусственную нейронную сеть.

В другом частном варианте осуществления способа положительные примеры формируются на основании законченных цепочек диалогов оператора с клиентом, причем такая цепочка содержит по меньшей мере одно вопросительное предложение.

В другом частном варианте осуществления способа при подборе релевантных реплик для ответа на фразу обращения клиента на стадии обучения модели для каждой ответной реплики рассчитывается скоринговый балл.

В другом частном варианте осуществления способа на этапе кодирования реплик в вектора реплики, представляющие предложения, кодируются как матрица семантических векторов.

Также указанный технический результат достигается за счет осуществления системы для обработки обращений пользователей в информационном канале с помощью искусственного интеллекта, которая

содержит по меньшей мере один процессор; по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают: получение пользовательского обращения с помощью информационного канала; обработку пользовательского обращения с помощью модели машинного обучения для автоматизированной обработки обращений пользователей, созданной с помощью способа по вышеописанному способу; формирование и передачу в информационном канале ответного сообщения на обращение пользователя.

В частном варианте реализации система представляет собой сервер, мейнфрейм или суперкомпьютер.

В другом частном варианте реализации системы информационный канал представляет собой чат-сессию, VoIP связь или канал телефонной связи.

В другом частном варианте реализации системы чат-сессия представляет собой чат с помощью мобильного приложения или чат на веб-сайте.

Описание чертежей

Признаки и преимущества настоящего изобретения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей.

Фиг. 1 иллюстрирует блок-схему выполнения заявленного способа.

Фиг. 2 иллюстрирует пример обработки данных для формирования обучающей выборки.

Фиг. 3 иллюстрирует архитектуру модели определения вопросительных предложений.

Фиг. 4 иллюстрирует способ обучения модели определения релевантных реплик.

Фиг. 5 иллюстрирует архитектуру модели определения релевантных реплик.

Фиг. 6 иллюстрирует пример применения обученной модели определения релевантных реплик.

Фиг. 7 иллюстрирует общий вид заявленной системы.

Осуществление изобретения

В данном техническом решении могут использоваться для ясности понимания работы такие термины как "оператор", "клиент", "сотрудник банка", которые в общем виде следует понимать, как "пользователь" системы.

Заявленный способ (100) создания модели анализа диалогов на базе искусственного интеллекта для обработки запросов пользователей, как представлено на фиг. 1, заключается в выполнении ряда последовательных этапов, осуществляемых процессором.

Начальным шагом (101) для формирования модели анализа диалогов является получение первичного ("сырого") набора данных, на которых будет строиться обучающая выборка для искусственной нейронной сети (ИНС). Набор первичных данных может представлять собой массив неразмеченных текстовых логов (записей) диалогов операторов с клиентами при обработке входящих обращений. Тематика текстовых логов может быть различной и меняться в зависимости от требований по итоговому формированию модели анализа для заданной отрасли ее итогового применения.

Под обращениями клиентов понимаются любые запросы, поступающие в информационные каналы взаимодействия с оператором контакт-центра или службы поддержки, например, финансово-кредитного учреждения. Как правило, первичные данные представляют собой преобразованные в текстовый вид записи разговоров клиентов с операторами. Информационные каналы для получения данных разговоров с операторами могут представлять собой, не ограничиваясь, телефонный канал, VoIP канал, чат-сессию на веб-сайте или в мобильном банковском приложении, чат-бот мессенджера и т.п. Любой тип канала, с помощью которого клиент может осуществлять диалог с оператором, может применяться для получения данных диалогов для последующего их перевода в текстовую форму для целей осуществления процесса обучения модели машинного обучения.

Как было указано выше, под оператором в настоящем решении может пониматься как человек, осуществляющий обработку целей обращения клиентов, так и программный алгоритм, например чат-бот или интеллектуальный автоответчик, способный также предоставлять сведения для обработки клиентских обращений.

На основании полученного массива первичных данных с логами на этапе (101) далее выполняется его обработка для формирования обучающей выборки (102) ИНС. Из полученного массива логов осуществляется формирование вопросно-ответных пар. Эта процедура включает в себя алгоритм сбора данных, использование модели определения вопросительного предложения и алгоритм формирования вопросно-ответных пар. С помощью модели определения вопросительных предложений происходит поиск вопросов клиента в текстовых логах. Следующая после найденного вопроса клиента реплика оператора (при условии, что она удовлетворяет ряду требований: не вопросительная, достаточно длинная, не содержит стоп-слов) считается ответом на этот запрос.

Обработка массива данных диалогов операторов с клиентами заключается в формировании на основании вопросно-ответных пар положительных и отрицательных примеров. Как правило, такие примеры формируются следующего вида: (контекст беседы (2-5 реплик), обращение или запрос клиента, ответ оператора на запрос) - положительный пример; (контекст беседы (2-5 реплик), обращение или запрос клиента, ответ оператора на какой-нибудь другой запрос) - отрицательный пример.

Ниже приведены примеры осуществления вопросно-ответных пар.

Пример 1 (положительный).

ctx: ['Здравствуйте, Иван Иванович!', 'Чем могу помочь?', 'Здравствуйте могу ли оформить кредитную карту']

rsp: 'Уточнить условия по кредитным картам и подать заявку вы можете по ссылке: http://www.sberbank.m/moscow/m/person/bank_cards/credit/'

Пример 2 (отрицательный).

ctx: ['Здравствуйте, Кирилл!', 'Чем могу вам помочь?', 'Здравствуйте, как подключить услугу мобильный банк?']

rsp:'Онлайн можно заказать только карты Visa Gold и MasterCard Gold'.

Формирование вопросно-ответных пар для создания обучающей выборки для ИНС осуществляется с помощью модели анализа вопросительных предложений, которая необходима для правильного разбиения контекста логов и формирования тренировочного набора данных. Под контекстом в данном случае понимается упорядоченный по времени набор реплик оператора и клиента. Последней репликой в контексте всегда является вопрос клиента.

На фиг. 2 представлен пример применения модели определения вопросительных предложений для формирования на этапе (102) обучающей выборки для ИНС. Модель определения вопросительных предложений (220) представляет собой модель машинного обучения, например искусственную нейронную сеть. Для обучения модели (220) может быть использован набор данных (иногда называют - "датасет") OpenSubtitles (OPUS) (http://opus.nlpl.eu/) (221), а также данные чатов с оператором (222). Датасет OPUS (221) представляет собой открытый набор данных, состоящий из субтитров к фильмам на различных языках, который используется как источник разговорной лексики, обычно встречающейся в художественном кино.

В качестве положительных примеров из наборов данных (221)-(222) были выбраны все предложения, содержащие знак вопроса, в качестве отрицательных - все остальные предложения. В OPUS (221) все предложения заканчивающиеся знаком вопроса - вопросительные, так как пунктуация в субтитрах всегда правильная. Это относится и к вопросительным предложениям из данных чатов (222). Извлеченные таким образом вопросительные предложения также проходят дополнительную фильтрацию: отбрасываются короткие или содержащие стоп-слова предложения. Аналогично положительным примерам, большая часть предложений из OPUS (221), не содержащих знак вопроса - не вопросительные и выбираются в качестве отрицательных примеров.

Дополнительно отбрасываются предложения, содержащие вопросительные слова или слишком короткие, где заранее задан размер слова.

Отрицательные примеры могут быть сгенерированы в любом количестве, что позволяет добиться любого соотношения положительных и отрицательных примеров в обучающей выборке для ИНС. Эксперименты показали, что наилучшего качества удается добиться, когда соотношение положительных и отрицательных примеров 1:1.

В примерном варианте осуществления обучающая выборка балансируется, вся пунктуация вырезается, чтобы модель (220) строила свои предсказания основываясь исключительно на семантике слов в предложении. Использовались все полученные сырые данные, но количество положительных и отрицательных примеров было одинаковым в каждом батче (фрагмент данных) при обучении модели (220). Для семантического анализа применяется семантическая модель слов fasttext и рекуррентная нейронная сеть на основе LSTM (англ. "Long short-term memory" - долгая краткосрочная память) для моделирования семантики предложения. FastText - это библиотека для изучения встраивания слов и классификации текста, созданная исследовательской лабораторией AI в Facebook™. Точность такой процедуры обработки набора данных составляет около 95%.

LSTM - тип рекуррентных нейронных сетей с обратной связью, который широко применяется в индустрии для моделирования временных рядов и других последовательностей. Наиболее широкое применение данная архитектура нашла в компьютерной лингвистике, где она применяется для моделирования семантики предложений или целых абзацев текста.

На фиг. 3 представлена архитектура модели определения вопросительных предложений (220). Архитектура модели определения вопросительных предложений (220) представлена на примере нейросетевой модели в виде ациклического вычислительного графа.

На архитектуре модели (220) указаны примеры размерностей входного и выходного тензора для каждого блока.

Пример записи.

Вход: (Нет,, 20) Выход: (Нет, 20, 300).

Данный пример означает, что блок на вход принимает тензор размерностью (batch_size, 20) и отдает тензор размерностью (batch_size, 20, 300). Размер батча (пакета данных) для обученной модели может быть любым (это влияет только на быстродействие и зависит от среды исполнения), для этого в нотации указывается (Нет).

Модель (220) содержит входной узел для текстовых данных (inp_ctx_0) (2201) и один выходной узел предсказания модели (relevance) (2211). В качестве пре-тренированных эмбеддингов (векторных

представлений слов) используется модель FastText, содержащая тексты на русском языке. В качестве энкодера используется двунаправленный LSTM-модуль.

Модуль векторизации слов (2202) содержит предобученную модель (англ. "word embedding") для векторизации на уровне слов. Каждое из предложений, подаваемых на вход модели, уже разбито на токены - представлено в виде списка слов. При этом все предложения представляются в виде последовательностей равной длины (это нужно для эффективной обработки батчей). Короткие предложения дополняются до этой фиксированной длины нулевым токеном, слишком длинные - обрезаются. Здесь и далее величина длины последовательностей будет обозначаться как MAX_LEN. В экспериментах использовалось значение MAX_LEN = 24, однако не ограничиваясь.

Word embedding - векторное представление слова, полученное с помощью дистрибутивной модели языка (обычно программных инструментов анализа семантики естественных языков word2vec, fasttext или glove). Это вектор размерности порядка нескольких сотен (100-1000). Характерной особенностью является то, что похожие по смыслу слова представляются близкими (по Евклидовой метрике L2) векторами.

Каждому слову в соответствие ставится семантический вектор - т.н. word embedding (см. источник информации https://ru.wikipedia.org/wiki/Word2vec). Для этого используется обученная на тематическом, например банковском, наборе данных модель fasttext (см. https://arxiv.org/abs/1607.04606). Преимуществом модели fasttext над базовой word2vec является возможность обработки (векторизации) слов, отсутствовавших в обучающей выборке. Семантические векторы fasttext имеют размерность порядка нескольких сотен, эту размерность можно обозначить как EMB_DIM. В проведенных экспериментах с архитектурой представленной модели (220) использовался EMB_DIM=300.

В результате этих процедур каждому предложению на входе модели (220) ставится в соответствие матрица размерностью (MAX_LEN, EMB_DIM). Эта функциональность инкапсулирована в модуль word_embedding_model (2202). Для векторизации предложений контекста и ответа используется модуль (2202). При этом модуль векторизации слов (2202) не обучается в процессе настройки модели (220), т.к. векторы слов в нем зафиксированы и более не изменяются.

Модуль векторизации предложения (2203) содержит модель векторизации всего предложения. Каждое из предложений представляется в виде матрицы из (MAX_LEN, EMB_DIM) модуля (2202) и кодируется в вектор фиксированной размерности. Для этого применяется рекуррентная нейронная сеть типа LSTM. Матрица, полученная с помощью модуля (2202) обрабатывается слева-направо модулем LSTM, в качестве векторного представления предложения соответствует последнее внутреннее состояние LSTM (то есть, соответствующее последнему слову в предложении).

В результате работы модуля (2203) каждое предложение будет представлено в виде вектора фиксированной размерности LSTM_DIM. В качестве примера работы была использована размерность ячейки LSTM_DIM=340. Таким образом, контексту запроса, состоящему из CTX_LEN реплик максимум по SEQ_LEN слов (вход inp_ctx) в соответствие ставится матрица (CTX_LEN, LSTM_DIM). Кандидату, состоящему из одного предложения, в соответствие ставится вектор LSTM_DIM. Кандидат представляет собой один из возможных вариантов ответа для данного контекста. Эта функциональность инкапсулирована в модуль векторизации предложений (2203). Модуль (2203) содержит большую часть обучаемых параметров модели (2-5 млн в зависимости от конфигурации) и является наиболее вычислительно "тяжелым".

Модули субдискретизации (пулинга) (2204, 2205) получают на вход вектор фиксированной размерности из внутреннего состояния RNN модуля векторизации предложений (2203). В частном варианте осуществления модули (2204, 2205) могут являться частью модуля векторизации предложений (2203).

Модуль конкатенации (2206) предназначен для конкатенации векторов, получаемых от модулей (2204, 2205) в один, для их последующей передачи в многослойный перцептрон (2207)в виде единого вектора.

Многослойный перцептрон (англ. "MLP/Multilayer perceptron") (2207), в частности двуслойный, в котором полносвязные (англ. "Dense") слои перемежаются с регуляризационными (англ. "Dropout"). Значение Dropout для данного примера MLP=0.3. Dropout - способ регуляризации нейронных сетей, который служит для борьбы с переобучением (см. например, http://imlr.org/papers/volume15/srivastaval4a/srivastaval4a.pdf).

Модуль (2208) является выходным нейроном с сигмоидальной функцией модели определения вопросительных предложений активации (220) и содержит предсказание модели (220), выполненное на основании обработки входных текстовых данных. В представленном примере архитектура представленной модели (220) достигла 0.945 AUC (Area under the ROC Curve), 0.875 ACC (Accuracy/точность) на валидационной выборке. Площадь под ROC-кривой AUC (Area under the ROC Curve) является агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок. Чем больше значение AUC, тем "лучше" модель классификации. Данный показатель часто используется для сравнительного анализа нескольких моделей классификации.

Далее рассмотрим этап генерирования обучающей выборки (102) для модели анализа обращений. Генерация обучающей выборки (102) выполняется с помощью модели (220) выделения вопросительных предложений из входного набора данных (210), который представляет собой неразмеченные диалоги чатов между клиентами и операторами (210). Каждая реплика клиента в каждом чате обрабатывается с

помощью упомянутой модели (220).

На этапе (ЮЗ) реплики токенизируются и представляются как последовательность векторов слов, после чего подаются на вход модели (220) выделения вопросительных предложений. В ходе обработки реплик модель (220) оценивает вероятность того, что реплика является вопросительным предложением. В случае если после реплики-запроса следуют несколько сообщений оператора, формируется положительный обучающий пример (231). В случае если подряд идет несколько реплик клиента, в качестве вопросительной выбирается та, для которой предсказанная вероятность оказалась выше всего. Если же подряд идет несколько реплик-ответов оператора в ответ на запрос клиента, то обучающий пример формируется с каждой из них.

В положительный обучающий пример (231) в качестве контекста включаются все реплики вплоть до запроса клиента (включительно). В качестве ответа используется следующая реплика оператора. В контекст включаются последние n реплик (как клиента, так и оператора) предшествующие запросу клиента, где n - параметр модели, например, от 1 до 6. В примерном варианте реализации в ходе обработки набора данных (210) моделью (220) был сформирован обучающий набор, который содержал порядка 1000000 положительных примеров (231).

Отрицательные примеры (232) были сформированы путем замены в положительном примере правильного ответа на произвольный из множества всех возможных ответов оператора (которых на момент обучения было около 1000000).

На основании сформированной обучающей выборки на этапе (104) осуществляют обучение модели определения релевантных реплик (240). На фиг. 4 представлен пример обучения модели определения релевантных реплик (240). На вход модели (240) поступают данные обучающей выборки (230), сформированной на основании полученных положительных (231) и отрицательных (232) примеров обработки обращений клиентов.

Из тренировочной части обучающей выборки (230) формируются обучающие батчи. Соотношение положительных и отрицательных примеров в батче выбирается приблизительно равным. Типичный размер батча - 256, 512. Модель (240) обучается в течении 32 эпох, в конце каждой валидируясь на отложенной выборке. Отложенная выборка представляет собой часть датасета, не используемую при обучении модели, но которая применяется для ее валидации (расчета метрик). Как пример, отложенная выборка может составлять 10% от исходной сгенерированной обучающей выборки.

Опционально при процессе валидации, в дополнение к обучающей выборке (230), полученной в автоматическом режиме из "сырых" (другими словами неразмеченных) данных, может быть использован размеченный вручную набор данных из вопросов и ответов. Если вручную размеченный набор данных достаточно большой (тысячи пар вопрос-ответ), то таким образом можно дополнительно дообучить модель (240) на этих парах. В этом случае выполняется замещение обучающей выборки (230) на размеченный вручную набор данных, с помощью которого продолжается дальнейшее обучение модели (240). Это приводит к существенному росту метрик качества на вопросах из дополнительного набора данных.

Если данных немного, то выполняется валидация модели (240) на них с помощью вычисления соответствующих метрик качества. Как правило, для вопросно-ответной системы, включающей модель (240), рассчитываются метрики recall@k и precision@k - модель с максимальными значениями этих метрик можно выбрать для последующей сериализации в pickle (модуль pickle реализует алгоритм сериализации и десериализации объектов Python). Значение данной метрики определяется частотой попадания верного ответа на вопрос в топ-K ранжированных по релевантности ответов модели. Это значение вычисляется по формуле: (количество релевантных ответов вплоть до k-той позиции в ранжированном списке ответов)/(общее количество релевантных ответов).

Например: модели задали 10 вопросов, 5 раз верный ответ был первым в списке сортированных ответов, и 8 раз верный ответ вошел в топ-3 сортированных по релевантности ответов. В таком случае для такого теста recall@1=5/10=0.5, recall@3=8/10 (считая, что существует единственный релевантный ответ на каждый вопрос).

Обучение модели (240) в среднем занимает 2-3 ч в зависимости при использовании графического ускорителя GPU NVIDIA 1080TL Модель (240) с максимальным значением точности (accuracy) на отложенной выборке сериализуется в бинарный формат (pickle) для дальнейшего использования.

На фиг. 5 представлена архитектура модели (240) определения релевантных реплик. Модель определения релевантности (240) предназначена для оценки релевантности данной пары контекст-ответ. Модель имеет два входных узла - для контекста диалога (2401) и для реплики - кандидата (2402). Модель имеет один выходной узел (2407), который представляет собой модуль для определения скорингового балла релевантности, который может принимать значения от 0 до 1.

Модуль (2403) векторизации слов аналогичен по своему функционалу модулю (2202), который также осуществляет векторизацию на уровне слов. Обозначение "Итеративный" означает, что модуль (2404) может выполнять предписанную обработку данных последовательно несколько раз. В данном случае параметр "Итеративный" для модуля векторизации слов (2404) указывает, что модуль векторизации слов (240) применяется по очереди для каждой реплики контекста (которых 3 штуки в данном примере). Для реплики кандидата это не требуется, так как она состоит из одного предложения (соответст-

венно векторизация отрабатывает один раз).

Модуль (2405) предназначен для векторизации предложений и по своему функционалу повторяет функционал модуля (2203). На представленной на фиг. 5 схеме узлы субдискретизации и конкатенации инкапсулированны внутрь модуля (2405) и не показаны на схеме явным образом. В данном случае параметр "Итеративный" указывает, что модуль векторизации предложений (2406) применяется по очереди для каждой реплики контекста (которых 3 штуки в данном примере).

Модуль (2407) представляет собой модуль вычисления релевантности. Данный модуль (2407) принимает на вход векторные представления контекста и кандидата и возвращает единственное число [0,1] - скоринговый балл релевантности. Скоринговый балл вычисляется на основании расчета ряда факторов, включающих в себя скалярное произведение между вектором-кандидатом и каждым из векторов в контексте, скалярное произведение между вектором-кандидатом и суммой векторов контекста; конкатенации векторов контекста и вектора кандидата; вычисления скалярного произведения с вектором-кандидатом.

Результат конкатенации векторов контекста и вектора кандидата подается на вход в двуслойный перцептрон. Размерность выходного слоя равна LSTM_DIM. На выходе формируется матрица (CTX_LEN, LSTM_DIM). Вычисляется скалярное произведение с вектором-кандидатом, и в результате на выходе определяется CTX_LEN факторов для полученного контекста. Длина контекста обозначается как CTX_LEN и может быть от 1 (контекст - только вопрос) и до бесконечности (контекст - весь диалог). Типичные значения: [1:5]. В качестве примера реализации, при CTX_LEN=3 получается 7 факторов для вычисления релевантности.

Эти факторы подаются на вход еще одному двухслойному перцептрону с сигмоидальной функцией активации на последнем слое. Результатом работы модуля (2406) является одно число - скоринговый балл релевантности, являющийся финальным выходом всей модели (240). Модель (240) обучается как бинарный классификатор, предсказывая, релевантен ли данный кандидат данному контексту или нет, то есть модель (240) позволяет определить ответ на получаемый вопрос в обращении.

Обученная модель (240) может быть использована для построения вопросно-ответной системы следующим образом. Из всех возможных ответов оператора выделяется некоторое ограниченное множество кандидатов. В процессе выделения из кандидатов исключаются слишком короткие, слишком длинные, несодержательные и дубликатные ответы. Это процесс может быть как полностью автоматизированным, так и полуавтоматизированным, при котором итоговый список кандидатов дополнительно проверяется вручную специалистом, что позволяет получить дополнительное качество работы всей системы. В результате получается множество кандидатов (обычно от сотен до тысяч), каждым из которых модель (240) сможет ответить на запрос.

Для ответа на запрос моделью (240) оценивается релевантность контекста запроса каждому из загруженных кандидатов. Топ-K кандидатов (типичное значение k=3) возвращается в качестве наиболее вероятных вариантов ответа на запрос в обращении клиента.

После получения обученной модели (240) определения релевантности реплик данная модель (240) может использоваться в дальнейшем в автоматизированных системах анализа диалогов, поступающих со стороны клиента. Например, такими системами могут выступать чат-боты, интеллектуальные ассистенты, размещаемые на веб-сайтах, виджеты, телефонные роботы и т.п.

На фиг. 6 представлен пример применения обученной модели (240) определения релевантных реплик для обработки обращений клиента (10), которые могут поступать на ресурс (20) при обращении. Под ресурсом (20) может использоваться веб-ресурс (веб-сайт, портал и т.п.), кол-центр, мобильное приложение и т.п. Клиент (10) может сформировать свое обращение в виде телефонного звонка, посредством чат-сессии, VoIP звонке, использованию специализированного виджета или программного обеспечения и т.п. Ресурс (20) при получении информации обращения от клиента (10) передают контекст обращения в обученную модель (240), которая определяет вопросно-ответную пару и передает в модуль формирования ответа на обращение клиента (250) данные для генерирования ответа на вопрос клиента (10). Ответ на обращение клиента (10) передается от модуля (250), как правило, в том же информационном канале, из которого поступило обращение. Ответ может представлять собой ответ с помощью чат-бота, телефонного робота, интерактивная информация, гиперссылка или комбинация вариантов ответа и т.д.

На фиг. 7 представлен пример общего вида вычислительной системы (300), которая обеспечивает реализацию заявленного способа (100) или является частью компьютерной системы, например сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

В общем случае система (300) содержит объединенные общей шиной информационного обмена один или несколько процессоров (301), средства памяти, такие как ОЗУ (302) и ПЗУ (303), интерфейсы ввода/вывода (304), устройства ввода/вывода (305) и устройство для сетевого взаимодействия (306).

Процессор (301) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей как Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (300) также необходимо учитывать графический

процессор, например, GPU NVIDIA или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа (100), а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (302) представляет собой оперативную память и предназначено для хранения исполняемых процессором (301) машиночитаемых инструкций для выполнения необходимых операций по логической обработке данных. ОЗУ (302), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом в качестве ОЗУ (302) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (303) представляет собой одно или более устройств постоянного хранения данных, например жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAKD и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов системы (300) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (304). Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь, PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п. Для обеспечения взаимодействия пользователя с вычислительной системой (300) применяются различные средства (305) В/В информации, например клавиатура, дисплей (монитор), сенсорный дисплей, тачпад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

Средство сетевого взаимодействия (306) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (306) может использоваться, но не ограничиваться, Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др. Дополнительно могут применяться также средства спутниковой навигации в составе системы (300), например GPS, ГЛОНАСС, BeiDou, Galileo.

Как было представлено на фиг. 6, ресурс (20), к которому осуществляет обращение клиент (10), может быть организован с помощью системы (300), которая может представлять собой сервер для обеспечения требуемого функционала по обработке поступающих обращений, распознаванию ответных реплик с помощью обученной модели (240) и генерирования ответных сообщений с помощью модуля (250), которые передаются по различным информационным каналам проводного и/или беспроводного типа. Обращения клиентов (10) также могут формироваться с помощью устройства, которое содержит частичный функционал системы (300), в частности устройство клиента (10) может представлять собой смартфон, компьютер, планшет, терминал и любое другой устройство, которое обеспечивает коммуникационный канал с ресурсом (20) для формирования и передачи обращения и получения требуемого ответа, который также может включать различный тип цифровой информации.

Таким образом, при применении модели определения релевантных ответов (240), созданной с помощью заявленного способа (100), достигается более точный подбор в автоматизированном режиме ответных пар по поступающему контексту в пользовательских обращениях, что позволяет создать новый, более усовершенствованный способ обучения и применения моделей машинного обучения в системах, основанных на использовании ИИ.

Представленные материалы описания раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Компьютерно-реализуемый способ создания модели анализа диалогов на базе искусственного интеллекта для обработки обращений пользователей, выполняемый с помощью по меньшей мере одного процессора и содержащий этапы, на которых

получают набор первичных данных, причем набор включает в себя, по меньшей мере, текстовые данные диалогов между пользователями и операторами, содержащие обращения пользователей и ответы операторов;

осуществляют обработку полученного набора данных, в ходе которой формируют обучающую выборку для искусственной нейронной сети, содержащую положительные и отрицательные примеры обращений пользователей на основании анализа контекста диалогов, причем положительные примеры содержат семантически связанный набор реплик оператора в ответ на обращение пользователя;

выполняют выделение и кодирование векторных представлений каждой реплики из упомянутых на предыдущем шаге положительных и отрицательных примеров обучающей выборки;

применяют сформированную обучающую выборку для обучения модели определения релевантных реплик из контекста пользовательских обращений в диалогах.

2. Способ по п.1, характеризующийся тем, что модель представляет собой по меньшей мере одну искусственную нейронную сеть.

3. Способ по п.1, характеризующийся тем, что положительные примеры формируются на основании законченных цепочек диалогов оператора с пользователем, причем такая цепочка содержит по меньшей мере одно вопросительное предложение.

4. Способ по п.1, характеризующийся тем, что при подборе релевантных реплик для ответа на фразу обращения пользователя на стадии обучения модели для каждой ответной реплики рассчитывается скоринговый балл.

5. Способ по п.1, характеризующийся тем, что на этапе кодирования реплик в векторное представление реплики, причем представляющие предложения, кодируются как матрица семантических векторов.

6. Система для обработки обращений пользователей в информационном канале с помощью искусственного интеллекта, содержащая

по меньшей мере один процессор;

по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают

получение пользовательского обращения с помощью информационного канала;

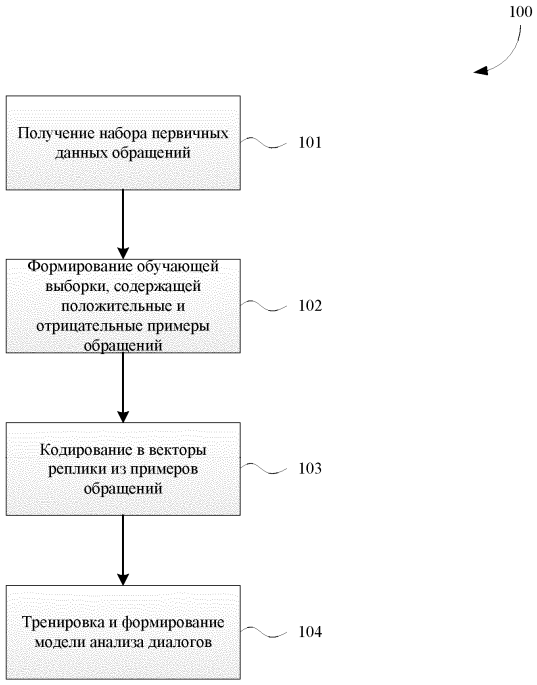
обработку пользовательского обращения с помощью модели машинного обучения для автоматизированной обработки обращений пользователей, созданной с помощью способа по любому из пп.1-5;

формирование и передачу в информационном канале ответного сообщения на обращение пользователя.

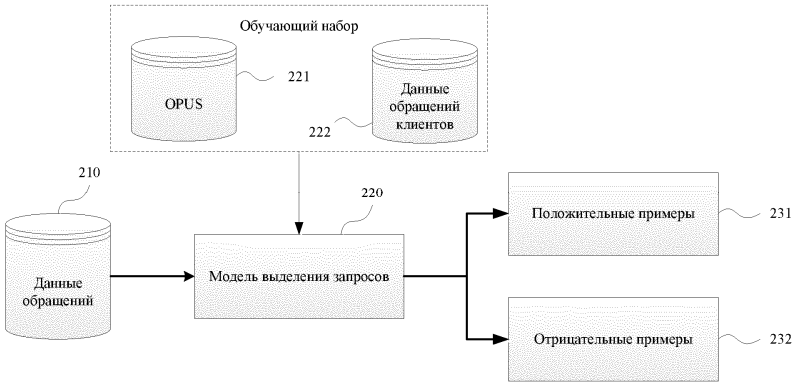
7. Система по п.6, характеризующаяся тем, что представляет собой сервер, мейнфрейм или суперкомпьютер.

8. Система по п.6, характеризующаяся тем, что информационный канал представляет собой чат-сессию, VoIP связь или канал телефонной связи.

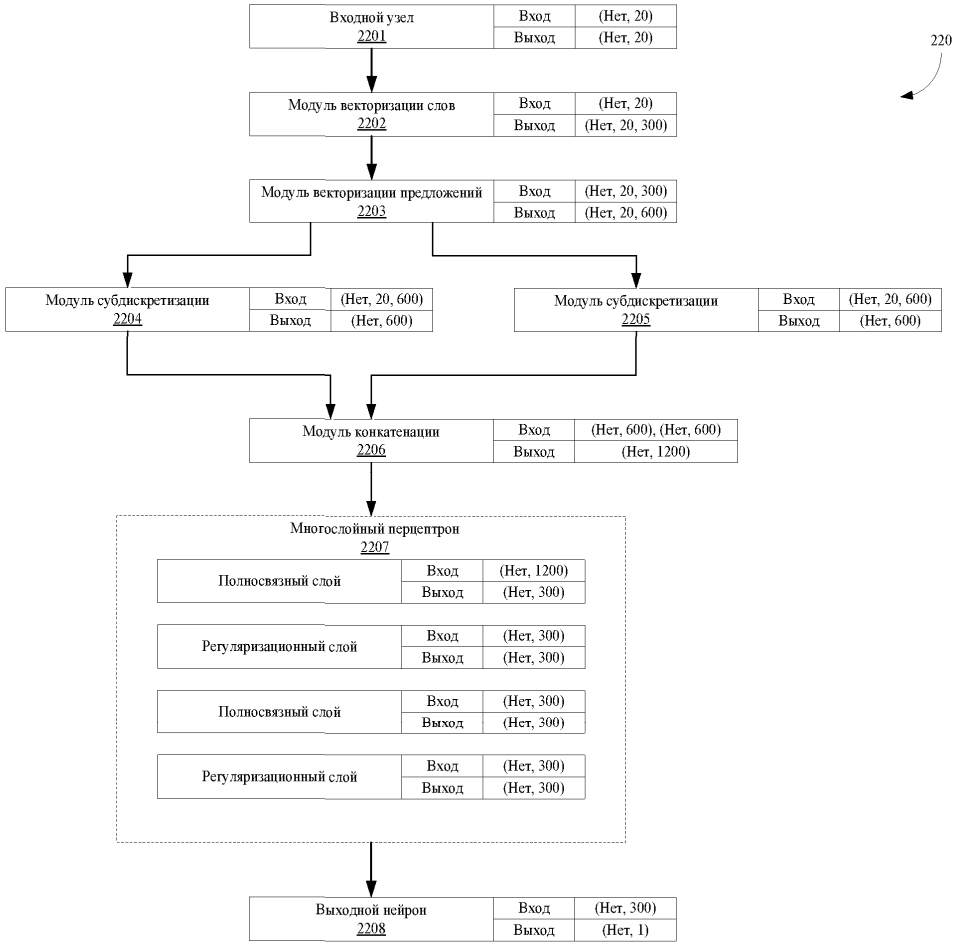
9. Система по п.6, характеризующаяся тем, что чат-сессия представляет собой чат с помощью мобильного приложения или чат на веб-сайте.



Фиг. 1


Обучающий набор
OPUS
221
Данные обращений клиентов
222
210
220
Данные обращений
Модель выделения запросов
Положительные примеры
231
Отрицательные примеры
232

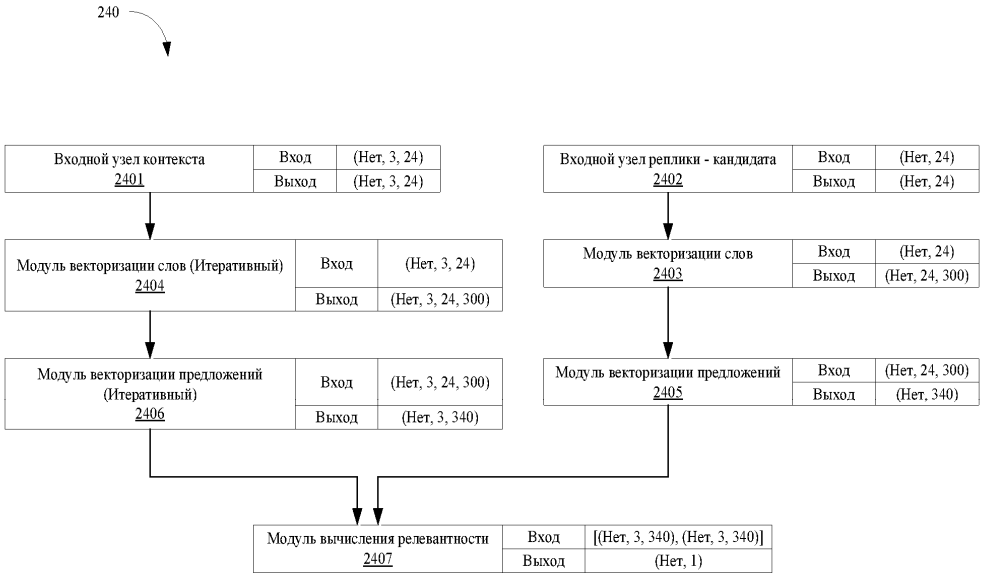
Фиг. 2


Входной узел 2201
Вход (Нет, 20)
Выход (Нет, 20)
220
Модуль векторизации слов 2202
Вход (Нет, 20)
Выход (Нет, 20, 300)
Модуль векторизации предложений 2203
Вход (Нет, 20, 300)
Выход (Нет, 20, 600)
Модуль субдискретизации 2204
Вход (Нет, 20, 600)
Выход (Нет, 600)
Модуль субдискретизации 2205
Вход (Нет, 20, 600)
Выход (Нет, 600)
Модуль конкатенации 2206
Вход (Нет, 600), (Нет, 600)
Выход (Нет, 1200)
Многослойный перцептрон 2207
Полносвязный слой
Вход (Нет, 1200)
Выход (Нет, 300)
Регуляризационный слой
Вход (Нет, 300)
Выход (Нет, 300)
Полносвязный слой
Вход (Нет, 300)
Выход (Нет, 300)
Регуляризационный слой
Вход (Нет, 300)
Выход (Нет, 300)
Выходной нейрон 2208
Вход (Нет, 300)
Выход (Нет, 1)

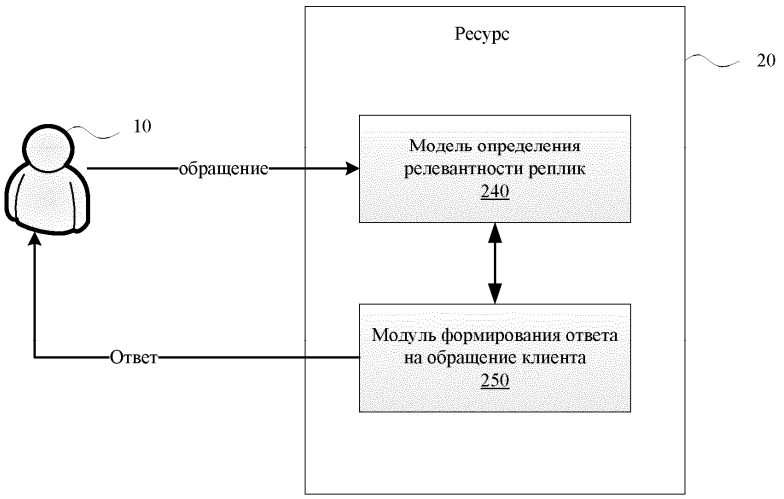
Фиг. 3


Обучающая выборка
Положительные примеры
231
Отрицательные примеры
232
230
Модель определения
релевантности реплик
240

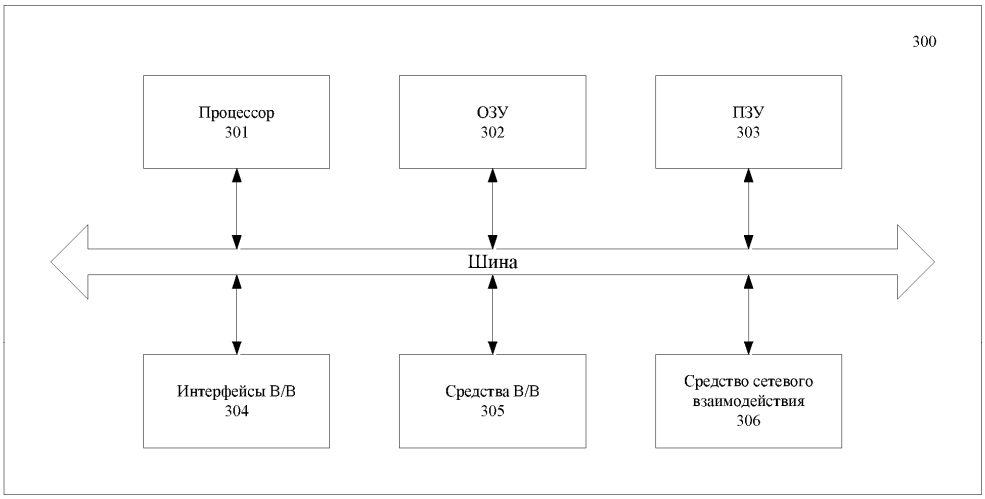
Фиг. 4


240
Входной узел контекста
2401
Вход (Нет, 3, 24)
Выход (Нет, 3, 24)
Входной узел реплики - кандидата
2402
Вход (Нет, 24)
Выход (Нет, 24)
Модуль векторизации слов (Итеративный)
2404
Вход (Нет, 3, 24)
Выход (Нет, 3, 24, 300)
Модуль векторизации слов
2403
Вход (Нет, 24)
Выход (Нет, 24, 300)
Модуль векторизации предложений
(Итеративный)
2406
Вход (Нет, 3, 24, 300)
Выход (Нет, 3, 340)
Модуль векторизации предложений
2405
Вход (Нет, 24, 300)
Выход (Нет, 340)
Модуль вычисления релевантности
2407
Вход [(Нет, 3, 340), (Нет, 3, 340)]
Выход (Нет, 1)

Фиг. 5



Фиг. 6



Фиг. 7

Евразийская патентная организация, ЕАПВ
Россия, 109012, Москва, Малый Черкасский пер., 2