(19) Евразийское патентное ведомство

(11) 044634 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2023.09.18

(21) Номер заявки
202293409

(22) Дата подачи заявки
2022.12.21

(51) Int. Cl. ***G06F 40/00*** (2020.01)
G06F 40/20 (2020.01)
G06F 40/56 (2020.01)
G06N 20/00 (2019.01)

(54) СПОСОБ И СИСТЕМА ГЕНЕРАЦИИ ТЕКСТА ДЛЯ ЦИФРОВОГО АССИСТЕНТА

(31) **2022111787**
(32) **2022.04.29**
(33) **RU**
(43) **2023.09.13**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Тихонова Мария Ивановна (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) US-A1-20210303803
US-A1-20210174784
US-A1-20200356634
CN-A-109783603

(57) Изобретение в общем относится к области вычислительной техники, а в частности к способу и системе генерации текста для цифрового ассистента в диалоговых системах. Техническим результатом от реализации заявленного изобретения является повышение семантической точности генерации стилизованного текста из исходного текста. Указанный технический результат достигается благодаря осуществлению способа автоматической генерации текста для цифрового ассистента в диалоговых системах, содержащего этапы, на которых: получают входные данные, содержащие исходный текст на естественном языке и целевой стиль реплик цифрового ассистента; осуществляют кодирование исходного текста; выполняют векторизацию токенов; осуществляют обработку векторных представлений токенов исходного текста, с помощью модели машинного обучения на базе нейронной сети, обученной на стилизованных, в соответствии с заданным целевым стилем, текстовых репликах цифрового ассистента, в ходе которой осуществляется формирование массива векторизированных стилизованных текстов; осуществляют декодирование каждого векторизированного стилизованного текста из массива, причем в ходе декодирования выполняют, по меньшей мере, преобразование векторизированного стилизованного текста в токены и детокенизацию; выполняют фильтрацию массива стилизованных текстов; осуществляют ранжирование отфильтрованных стилизованных текстов и выбор лучшего стилизованного текста, причем выбор лучшего текста основан на попарном расстоянии между исходным текстом и каждым из возможных стилизованных текстов; отправляют стилизованный текст в диалоговую систему.

044634 B1

044634 B1

Область техники

Заявленное техническое решение в общем относится к области вычислительной техники, а в частности к способу и системе генерации текста для цифрового ассистента в диалоговых системах.

Уровень техники

В результате функционирования языка сложились его разновидности, принадлежащие к определенному стилю изложения текста, которому присущи определенные черты, языковые средства, жанры и т.д. Так, при публикации текста в научном журнале, такому тексту будут присущи черты научного стиля, в то время как при неформальном общении тексту будут присущи черты разговорного стиля, например, неформальные обращения на "ты", простота конструкции предложения, использование сленга и т.д.

В настоящее время, с развитием информационных технологий, активное развитие получили технологии переноса стиля речи в области обработки текстов на естественном языке (Natural Language Processing, NLP) и сегодня ее пытаются интегрировать в самые разные сферы. Автоматизация процесса или части процесса стилизации текста в определенном стиле может позволить существенно повысить эффективность в таких сферах, как журналистика, в издательских домах, например, редакторы текста, создание контента для медиа платформ и виртуальных ассистентов и т.д. Однако, не смотря на востребованность данной технологии, существуют ряд трудностей, не позволяющих, например, генерировать стилизованный текст с высокой точностью. Так, проблемами стилизации текста является обеспечение сохранности исходной информации, обеспечение отсутствия в сгенерированных текстах новых фактов, сохранение смысловой нагрузки исходного текста и т.д. Кроме того, одной из немаловажных проблем, также является возможность обеспечения универсальности технологии, позволяющей генерировать текст не только в одном стиле, но и обеспечивать возможность стилизации исходного текста в нескольких стилях, в зависимости от сферы применения. Поэтому создание эффективного и точного способа автоматической генерации текста в заданных стилях является существенной задачей.

Так, из уровня техники известен способ переноса стиля текста, раскрытый в источнике [1]. Указанный способ обеспечивает возможность генерирования стилизованного текста из исходного текста с помощью решения задачи машинного перевода. В качестве "языка", на который требуется перевести исходный текст, в данном случае выступает стиль текста.

К недостаткам указанного решения можно отнести высокую сложность реализации и узконаправленность данного решения, в связи с огромным набором требуемых обучаемых данных, и невозможность адаптации под разные стили ввиду особенностей технологии языкового перевода. Кроме того, указанное решение также не обеспечивает высокую точность, т.к. в процессе такого "перевода" может теряться смысл исходной фразы из-за изменения всех слов исходного текста.

Из уровня техники также известен способ предоставления логических ответов, которые подражают стилю речи пользователя, раскрытый в патенте РФ № RU 2693332 C1 (Общество с ограниченной ответственностью "Яндекс"), опубл. 02.07.2019. Указанный способ обеспечивает возможность выбора контекстного ответа на вопрос, в зависимости от контекста вопроса, за счет анализа векторного представления контекстного вопроса и поиска ближайшего ответа из набора ответов в базе данных. Недостатками данного решения являются невозможность генерирования стилизованного текста на основе исходного текста, высокие затраты вычислительной мощности и большой объем требуемой памяти на формирование базы данных (БД), ограниченность стилизации паттернами БД, низкая точность генерирования стилизованного текста ввиду подбора заранее созданных и сохраненных в БД стилистических ответов.

Общими недостатками существующих решений является отсутствие эффективного способа генерации стилизованного текста с высокой точностью, обеспечивающей сохранность исходной информации и отсутствие в сгенерированных текстах новых фактов. Также, указанный способ должен обеспечивать сохранение смысловой нагрузки исходного текста. Кроме того, указанный способ должен обеспечивать универсальность технологии стилизации текста, позволяющей генерировать текст не только в одном стиле, но и обеспечивать возможность стилизации исходного текста в нескольких стилях, в зависимости от сферы применения.

Раскрытие изобретения

В заявленном техническом решении предлагается новый подход к генерации текста для цифрового ассистента в диалоговых системах. В данном решении используется алгоритм машинного обучения, который позволяет осуществлять генерацию стилизованного текста для цифрового ассистента из исходного текста с высокой точностью, обеспечивающей семантическую близость исходного и стилизованного текста и исключающий искажение стилизованного текста новыми фактами.

Таким образом, решается техническая проблема обеспечения возможности генерации стилизованного текста.

Техническим результатом, достигающимся при решении данной проблемы, является повышение семантической точности генерации стилизованного текста из исходного текста.

Дополнительным техническим результатом, проявляющимся при решении вышеуказанной проблемы, является обеспечение возможности генерации множества стилизованных текстов из одного исходного.

Указанные технические результаты достигаются благодаря осуществлению компьютерно-реализуемого способа автоматической генерации текста для цифрового ассистента в диалоговых систе-

мах, выполняемый по меньшей мере одним вычислительным устройством, и содержащий этапы, на которых:

a) получают входные данные, содержащие исходный текст на естественном языке и целевой стиль реплик цифрового ассистента;

b) осуществляют кодирование исходного текста, причем в ходе кодирования выполняют по меньшей мере токенизацию текстовых данных;

c) выполняют векторизацию токенов, полученных на этапе b);

d) осуществляют обработку векторных представлений токенов исходного текста, полученных на этапе c), с помощью модели машинного обучения на базе нейронной сети, обученной на стилизованных, в соответствии с заданным целевым стилем, текстовых репликах цифрового ассистента, в ходе которой осуществляется формирование массива векторизированных стилизованных текстов;

e) осуществляют декодирование каждого векторизированного стилизованного текста из массива, полученного на этапе d), причем в ходе декодирования выполняют по меньшей мере преобразование векторизированного стилизованного текста в токены и детокенизацию;

f) выполняют фильтрацию массива стилизованных текстов, полученных на этапе e);

g) осуществляют ранжирование отфильтрованных стилизованных текстов и выбор лучшего стилизованного текста, причем выбор лучшего текста основан на попарном расстоянии между исходным текстом и каждым из возможных стилизованных текстов;

h) отправляют стилизованный текст, полученный на этапе g), в диалоговую систему.

В одном из частных вариантов реализации способа фильтрация массива стилизованных текстов выполняется с помощью регулярных выражений и морфологического анализатора.

В другом частном варианте реализации способа фильтрация массива стилизованных текстов выполняется на основе совпадения имен собственных в исходном тексте и каждом стилизованном тексте из массива.

В другом частном варианте реализации способа совпадение имен собственных определяется с помощью количества неизменных именованных сущностей, содержащихся в исходном тексте и стилизованном тексте.

В другом частном варианте реализации способа количество неизменных именованных сущностей определяется на основе распознавания именованных сущностей в исходном тексте и каждом стилизованном тексте из массива.

В другом частном варианте реализации способ дополнительно содержит этап проверки наличия стилизации текста.

В другом частном варианте реализации способа проверка наличия стилизации текста осуществляется на основе определения произошла ли замена целевых индикаторов стиля.

Кроме того, заявленные технические результаты достигаются за счет системы автоматической генерации текста для цифрового ассистента в диалоговых системах, содержащей:

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 выполнение способа генерации текста для цифровых ассистентов в диалоговых системах.

Краткое описание чертежей

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

Фиг. 1 иллюстрирует блок-схему общего вида заявленной системы.

Фиг. 2 иллюстрирует блок-схему выполнения заявленного способа.

Фиг. 3 иллюстрирует пример общего вида вычислительного устройства, которое обеспечивает реализацию заявленного решения.

Осуществление изобретения

Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

Модель в машинном обучении (МО) - совокупность методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

Распознавание именованных сущностей (Named-entity recognition, NER) - это подзадача извлечения информации, которая направлена на поиск и классификацию упоминаний именованных сущностей в неструктурированном тексте по заранее определенным категориям, таким как имена собственные, имена персонажей, организации, местоположения, денежные значения, проценты и т.д.

Векторное представление слов (word embeddings, эмбеддинги) - общее название для различных подходов к моделированию языка и обучению представлений обработке естественного языка, направленных на сопоставление словам (и, возможно, фразам) из некоторого словаря векторов из n-мерного вещественного пространства R_n.

Токенизация - это процесс разбиения текста на текстовые единицы или токены (чаще всего в качестве

таких единиц выступают слова, но это могут быть также буквы, части предложения, сочетания слов и т д).

Языковая модель - это вероятностное распределение на множестве словарных последовательностей. В данном патенте термин "языковая модель" употребляется для описания нейросетевых языковых моделей, которые выполнены с возможностью моделирования языка посредством оценки вероятности той или иной последовательности символов.

Заявленное техническое решение предлагает новый подход, обеспечивающий повышение семантической точности генерации стилизованного текста, заключающейся в обеспечении сохранности смыслового содержания в стилизованном тексте, а также исключения дополнения стилизованного текста новыми фактами и логическими единицами языка. Одной из особенностей заявленного технического решения является возможность генерации множества стилизованных тексов из единственного исходного текста, что также обеспечивает автоматизацию процесса генерации стилизованных текстов и значительно снижает время стилизации текста по сравнению с ручной генерацией каждого отдельного стилизованного текста.

Заявленное техническое решение может быть реализовано на компьютере, в виде автоматизированной информационной системы (АИС) или машиночитаемого носителя, содержащего инструкции для выполнения вышеупомянутого способа.

Техническое решение также может быть реализовано в виде распределенной компьютерной системы или вычислительного устройства.

В данном решении под системой подразумевается компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, четко определенную последовательность вычислительных операций (действий, инструкций). Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы)/ Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройства хранения данных, например, таких устройств, как оперативно запоминающие устройства (ОЗУ) и/или постоянные запоминающие устройства (ПЗУ). В качестве ПЗУ могут выступать, но, не ограничиваясь, жесткие диски (HDD), флэш-память, твердотельные накопители (SSD), оптические носители данных (CD, DVD, BD, MD и т.п.) и др.

Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

Термин "инструкции", используемый в этой заявке, может относиться, в общем, к программным инструкциям или программным командам, которые написаны на заданном языке программирования для осуществления конкретной функции, такой как, например, кодирование и декодирование текстов, фильтрация, ранжирование, трансляция текстов в диалоговую систему и т.п. Инструкции могут быть осуществлены множеством способов, включающих в себя, например, объектно-ориентированные методы. Например, инструкции могут быть реализованы, посредством языка программирования Python, C++, Java, Python, различных библиотек (например, MFC; Microsoft Foundation Classes) и т.д. Инструкции, осуществляющие процессы, описанные в этом решении, могут передаваться как по проводным, так и по беспроводным каналам передачи данных, например Wi-Fi, Bluetooth, USB, WLAN, LAN и т.п.

На фиг. 1 приведен общий вид системы 100 генерации текста для цифрового ассистента в диалоговых системах. Система 100 включает в себя основные функциональные элементы, такие как: модуль кодирования/декодирования 101, модуль переноса стиля 102, модуль фильтрации стилизованных текстов 103, модуль ранжирования 104 стилизованных текстов. Более подробно элементы системы 100 раскрыты на фиг. 3.

В качестве диалоговой системы может выступать система 120 и представлять собой различные решения, например, голосовые помощники, чат-боты, роботизированные колл-центры, и иные технологии, воплощающие автоматизированный процесс общения с пользователем, таким как пользователь 110. Стоит отметить, что под диалоговыми системами в данном решении следует понимать любую автоматизированную человеко-машинную систему, работающую в режиме диалога, при котором она отвечает на каждую команду пользователя и по мере надобности обращается к нему за информацией. В качестве цифровых ассистентов (виртуальные цифровые помощники) могут выступать системы автоматизации взаимодействия с пользователем, реализованные на основе искусственного интеллекта в диалоговом формате (чат-бот, навыки для голосового помощника и т.д.). Так, в одном частном варианте осуществления цифровой ассистент может представлять собой компьютерную систему, которая имитирует в диалоговом формате разговор с пользователями.

Под стилизацией текста в данном решении понимается генерирование текста путем преобразования принятого исходного текста в текст, которому присущи стилистические речевые черты. Так, стилистическими чертами могут являться эмоциональный окрас текста (веселый, грустный и т.д.). В другом частном варианте осуществления стилистическими речевыми чертами может являться условие и цели общения в какой-то сфере общественной деятельности, например, официально-деловой деятельности, публицистической деятельности, разговорной, художественной и т.д. Стоит отметить, что стилизацией текста также

может являться придание характерных черт тексту, присущих особенностям общения отдельно взятых личностей, персонажей, литературных героев и т.д., не ограничиваясь. Так, в еще одном частном варианте осуществления стилизацией текста может являться преобразование исходного текста в стилизованный, в соответствии с заданным стилем общения конкретного цифрового помощника, которому присущи разговорные черты стиля, например, использование определенных местоимений, подчеркивающих неформальный/формальный стиль общения ("Ты" и "Вы), род, число и т.д.

Модуль кодирования/декодирования 101 может быть реализован на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением, и включать набор моделей для токенизации и детокенизации текста, векторизации токенизированного текста и преобразования токенов в текст, например, одну или несколько моделей машинного обучения для преобразования текстовой информации в векторную форму, например, BERT, ELMo, ULMFit, XLNet, RoBerta, RuGPT3 и другие. В одном частном варианте осуществления модуль 101 может быть

реализован на базе системы 300, которая более подробно раскрыта на фиг. 3. Стоит отметить, что определенный метод токенизации и векторизации зависит от выбранной языковой модели, на базе которой реализован модуль 102. Например, при использовании модели RuGPT3, токенизация осуществляется методом BPE (Byte Pair Encoding), а последующая векторизация - путем замены каждого токена на его индекс в словаре языковой модели, составленном на этапе изначального обучения модели. Кроме того, в еще одном частном варианте осуществления, в качестве метода токенизации может использоваться токенизация по словам. Пример токенизации по словам и кодирование слов индексами в словаре:

'мама мыла раму' → *['мама', 'мыла', 'раму']* → *[235, 376, 1056]*

Модуль 102 может быть реализован на базе по меньшей мере одной нейронной сети, заранее обученной на конкретных наборах стилизованных, в соответствии с заданными стилями, текстов. В качестве модели машинного обучения, реализующей функцию генерации стилизованных, в соответствии с заданным стилем, текстов, может быть использована, например, генеративная языковая модель, такая как RuGPT3, XLNet и т.д. В одном частном варианте осуществления, при реализации заявленного решения, МО являлась русскоязычная генеративная языковая модель RuGPT3-Large. Модель обучена на источниках из разных доменов: Википедия, книги, новости, русский Common Crawl и т.д. Модель обучали 14 дней на 128 GPU с контекстным окном 1024 и дополнительно несколько дней на 16 GPU с контекстом 2048. Финальная модель имеет перплексию 13.8 на тестовом наборе данных. На данном этапе обучения, результатом обучения языковой модели являлась возможность предсказывания вероятности следующего токена на основе предыдущего начального фрагмента текста. Так, если в процессе обучения модель часто встречала в обучающих данных определенное словосочетание, то при предсказании следующего после известного из словосочетания токена, модель с высокой вероятностью будет предсказывать именно токен из словосочетания в обучающем наборе данных. Далее, для выполнения непосредственно самого процесса генерации стилизованного текста из исходного текста, выполнялось дообучение обученной модели. Для дообучения модели использовалась процедура fine-tune. На указанном этапе выполнялось настраивание весов обученной модели в соответствии с решаемой задачей. Так, при появлении в модели исходного предложения, за счет измененных весовых коэффициентов, наиболее вероятным предложением для модели будет перефраз данного предложения в определенном стиле, т.е. осуществляется повышение вероятности продолжения текстового фрагмента в том формате, в котором должен быть сгенерирован стилизованный текст. При дообучении модели использовались датасеты с различными стилизованными репликами цифровых ассистентов. Так, в одном частном варианте осуществления решалась задача стилизации исходного текста под три стиля речи цифровых ассистентов. Изначальные данные для дообучения содержали 2174, 2436 и 32242 реплик в стиле каждого ассистента соответственно. При этом исходные данные содержали лишь реплики в стиле конкретных ассистентов, но не содержали исходного текста, из которого осуществляется генерация таких стилизованных реплик.

Для формирования датасета (обучающего набора данных) в формате, подходящем для дообучения модели, состоящего из пар "исходная реплика" - "реплика в стиле ассистента", к стилизованным репликам был применен парафразер, например, парафразер на основе генеративной модели RuT5. Модель парафразера известна из уровня техники и раскрыта, например, в источнике, доступном по ссылке в Интернет: https://huggingface.co/cointegrated/rut5-base-paraphraser. Для каждой стилизованной реплики было сгенерировано 10 исходных вариантов парафразера, из которых затем было выбрано 2 наиболее близких исходных реплики на основе семантической метрики, например, метрики LabSe. Данная метрика оценивает косинусное сходство между векторными представлениями предложений, полученными с помощью модели, которое соответствует семантической близости. Таким образом итоговые датасеты для дообучения RuGPT3 под задачу переноса стиля содержали 4348, 4872, 64484 пар предложений ("исходная реплика" - "реплика в стиле ассистента"). Помимо этого, к данным был также добавлен тэг, характеризующий конкретного цифрового ассистента с соответствующим ему стилем речи для обеспечения возможности определения в каком именно стиле сгенерируется стилизованный текст. На полученных данных модель дообучалась 5 эпох. Оценка полученной модели производилась на тестовом сете отдельно по каждому ассистенту. В качестве тестового набора данных использовалось 1097 реплик для каждого ассистента из

наборов ассистентов. Для оценки качества были использованы следующие метрики: 1) BLEU (Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation (PDF). ACL-2002: 40th Annual meeting of the Association for Computational Linguistics, pp. 311-318. CiteSeerX 10.1.1.19.9416.). Алгоритм BLEU сравнивает фразы двойного перевода с фразами, которые он находит в эталонном варианте, и взвешенно подсчитывает количество совпадений. Эти совпадения не зависят от позиции. Высшая степень совпадения указывает на более высокую степень сходства с эталонным переводом и более высокий балл. Внятность и грамматика не учитываются. 2) Число общих N-Gram, где в качестве N-Gram брались последовательности слов длины от 3 до 8. 3) Levenshtein - расстояние Левенштейна (В. И. Левенштейн. Двоичные коды с исправлением выпадений, вставок и замещений символов. Доклады Академий Наук СССР, 1965. 163.4:845-848.) (редакционное расстояние, дистанция редактирования). Указанная метрика, измеряющая по модулю разность между двумя последовательностями символов. Она определяется как минимальное количество односимвольных операций (а именно вставки, удаления, замены), необходимых для превращения одной последовательности символов в другую. 4) Jaccard index - индекс Жаккара (Jaccard P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines //Bull. Soc. Vaudoise sci. Natur. 1901. V. 37. Bd. 140. S. 241-272.), который вычислялся как число общих токенов в стилизованной реплике и исходном предложении, деленное на объединение токенов в этих двух фрагментах текста.

Результаты оценки модели для трех цифровых ассистентов, каждому из которых присущ свой стиль общения, приведены в табл. 1.

Таблица 1

Исходный vs стилизованный текст	BLEU	Common n-grams	Jaccard index	Levenshtein
Цифровой ассистент 1	0.966	0.986	0.983	1.764
Цифровой ассистент 2	0.967	0.989	0.984	1.808
Цифровой ассистент 3	0.995	0.992	0.989	0.432

Из табл. 1 видно, что BLEU score, число общих n-gram и индекс Жаккара имеют уровень более 95%. Расстояние Левенштейна для всех ассистентов также в среднем меньше 2, что указывает на то, что сгенерированные предложения в среднем отличаются от ответов-образцов не более чем на 2 символа.

Модуль 103 может быть реализован на базе по меньшей мере одного вычислительного устройства и включать морфологический анализатор, например, Unigram анализатор, N-gram анализатор, анализатор на основе регулярных выражений и т.д. Также, модуль 103 может содержать нейронную сеть, обученную на решение задачи NER (распознавание именованных сущностей), например, Slovnet BERT NER, DeepPavlov BERT NER и т.д., не ограничиваясь. В одном частном варианте осуществления модуль 103, может содержать тяжелую модель с BERT-архитектурой, и быть обучен на небольшом вручную аннотированном датасете.

Модуль ранжирования 104 стилизованных текстов может быть реализован на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением для вычисления расстояния Левенштейна. Так, указанный модуль 104 выполнен с возможностью осуществления алгоритма вычисления попарных расстояний между исходным текстом и стилизованными текстами.

Для специалиста в данной области техники очевидно, что, хотя и описанные выше модули представлены как отдельные устройства, указанные модули также могут быть объединены в составе одного устройства, например, системы 300.

На фиг. 2 представлена блок схема способа 200 автоматической генерации текста для цифрового ассистента в диалоговых системах, который раскрыт поэтапно более подробно ниже. Указанный способ 200 заключается в выполнении этапов, направленных на обработку различных цифровых данных. Обработка, как правило, выполняется с помощью системы, например, системы 100, которая также может представлять, например, сервер, компьютер, мобильное устройство, вычислительное устройство и т.д. На этапе 210 система 100 получает входные данные, содержащие исходный текст на естественном языке и целевой стиль реплик цифрового ассистента. Так, входные данные могут быть получены от диалоговой системы 120 по каналам передачи данных, таких как Интернет. Исходный текст, полученный от диалоговой системы 120, может представлять, например, диалоговую реплику, такую как ответ на вопрос пользователя, вопрос пользователю и т.д. Целевой стиль реплик цифрового ассистента может представлять определенные характерные черты речевого стиля, например, разговорный стиль общения, которому присущи, например, наличие неформальных обращений к пользователю, деловой стиль общения, которому присущи официальные обращения и т.д. Также, в одном частном варианте осуществления целевой стиль реплик может указывать на эмоциональный окрас текста, например, грустный, веселый, нейтральный и т.д. Целевой стиль реплик цифрового ассистента может быть определен настройками диалоговой системы 120. Так, диалоговая система, такая как система 120, может содержать набор цифровых ассистентов каждому из которых присущ свой стиль общения. При начале диалога пользователь в настройках системы может выбрать определенного ассистента. Указанные данные о выбранном ассистенте также переда-

ются в систему 100. На этапе 220 осуществляют кодирование исходного текста, причем в ходе кодирования выполняют по меньшей мере токенизацию текстовых данных. Указанный этап 220 может выполняться модулем 101. Входной текст может быть разделен на токены. Под токеном в данном решении следует понимать последовательность символов в тексте, которая имеет значение для анализа. В еще одном частном варианте осуществления токенизация текста может быть выполнена с помощью алгоритма BPE (Byte Pair encoding). В еще одном частном варианте осуществления токенизация может представлять собой разбиение текста на слова по пробелу между словами. Далее составляется словарь токенов фиксированного размера (например, 30000 токенов), где каждому токену сопоставляется его индекс в словаре.

Пример токенизации на слова:

['Вот что я нашел по вашей заявке' → '<Вот> <что> <я> <нашел> <по> <твоей> <заявке> ']

На указанном этапе 230 выполняется векторизация токенизированных текстов. Как упоминалось выше, метод токенизации зависит от языковой модели, которая используется в модуле 102 на этапе 240. Так, например, при использовании языковой модели RuGPT3, каждому токену сопоставляется его индекс в словаре. Таким образом, токенизированный фрагмент текста (список токенов) после векторизации отображается в вектор индексов данных токенов в словаре. Приме векторизации при токенизации по словам:

'мама мыла раму' → ['мама', 'мыла', 'раму'] → [235, 376, 1056]

Далее способ 100 переходит к этапу 240.

На этапе 240 осуществляют обработку векторных представлений токенов исходного текста, с помощью модуля 103, в ходе которой осуществляется формирование массива векторизированных стилизованных текстов. Как упоминалось выше, модель машинного обучения на базе нейронной сети была дообучена на стилизованных, в соответствии с заданным целевым стилем, текстовых репликах цифровых ассистентов. На этапе 240 на вход модели поступает векторное представление исходного текста. На выходе модель генерирует несколько вариантов стилизованных текстов (кандидатов) в форме векторных представлений. Указанные векторные представления сохраняются в массив стилизованных текстов. Количество кандидатов, генерируемых моделью зависит от применяемых методов выборки. В одном частном варианте осуществления для модели использовались следующие критерии выборок: выборка top P (top_p=0.92), выборка top K (top_k=50), температурная выборка (temperature=0.85). Более подробно указанные методы выборок раскрыты в источнике, найдено в Интернет по ссылке: https://towardsdatascience.com/how-to-sample-from-language-models-682bceb97277. Общий принцип работы модели заключается в возможности предсказывать вероятность следующего токенав определенном контексте. Так, на первом этапе обучения осуществляется возможность языковой модели предсказывать вероятность следующего токена на основе предыдущего начального фрагмента текста. Для реализации возможности генерирования указанной моделью стилизованных текстов, далее выполняется изменение ее весов таким образом, чтобы те вероятности для следующего токена, которые предсказывает модель, отвечали текущей задачи стилизации текста. Указанная изменение весов осуществляется на основе обучающего набора данных. Дообученная модель далее способна генерировать из исходного текста множество стилизованных текстов с разной степенью вероятности токенов в этом тексте. Поскольку сгенерированное множество стилизованных текстов может быть очень велико, а распределение вероятности в части из них будет слишком малым, то такие кандидаты (сгенерированные стилизованные тексты) могут быть отсечены на основе критериев выборок, указанных выше. В результате, на выходе модели формируется массив стилизованных текстов.

Стоит отметить, что обеспечение возможности генерации масссива стилизованных текстов, а не одного стилизованного текста, повышает точность стилизации текста, т.к. обеспечивается возможность дальнейшей проверки и обработки всех вариантов стилизованного текста для выбора наиболее семантически близкого, к исходному, текста из массива. Кроме того, генерация именно массива стилизованных текстов позволяет в дальнейшем исключить добавление новых фактов и/или некорректное изменение определенных слов на основе дальнейшей обработки таких текстов.

На этапе 250 сформированный массив векторизированных стилизованных текстов поступает в модуль 101. На указанном этапе 250 осуществляют декодирование каждого векторизированного стилизованного текста из массива, причем в ходе декодирования выполняют по меньшей мере преобразование векторизированного стилизованного текста в токены и детокенизацию. Так, например, в ходе указанного процесса каждый вектор фиксированной длины на основе его размерности сопоставляется токен по индексу словаря, что позволяет представить каждый вектор в виде токена. Процесс детокенизации является обратным процессом к токенизации и заключается в объединении токенов в текст. В результате выполнения данного этапа 250, массив векторизированных стилизованных текстов преобразуется в массив стилизованных текстов на естественном языке.

Пример:

Исходный текст [*'Вот что я нашел по вашей заявке*]

Массив стилизованных текстов: [*'Вот что я нашла по заявке', 'Вот что я нашел по твоей заявке', 'Вот что я нашла по вашей заявке', 'Что я нашла по твоей заявке'...*]

На этапе 260 выполняют фильтрацию массива стилизованных текстов. На указанном этапе 260 с помощью модуля 103 осуществляется фильтрация указанного массива. Так, на данном этапе исключаются те кандидаты, которые не удовлетворяют заданным характерным стилевым чертам. Так, например, при генерации стилизованной реплики для цифрового ассистента, имитирующего разговорное неформальное общение в качестве женского пола, будут по меньшей мере следующие критерии: по роду (женский род) и обращению (на Ты). Так, в результате генерации стилизованного текста для женского персонажа, который придерживается делового стиля общения (присущи характерные стилистические черты делового стиля общения), исключается кандидат 'Вот что я нашел по твоей заявке'. Указанная фильтрация также повышает точность стилизации текста. Для проверки корректности используются регулярные выражения, написанные на основе библиотеки re и морфологический анализатор MorphAnalyzer из библиотеки pymorphy2. На выходе данного модуля имеем отфильтрованный список возможных кандидатов, которые прошли проверку корректности.

Пример:

[*'Вот что я нашла по заявке', 'Что я нашла по твоей заявке'...*].

Кроме того, в одном частном варианте осуществления фильтрация массива стилизованных текстов выполняется на основе совпадения именованных сущностей (например, имен собственных) в исходном тексте и каждом стилизованном тексте из массива. Для этого в исходном и каждом из стилизованных текстов выполняется алгоритм распознавания именованных сущностей, например, с помощью модуля 103, и происходит сравнение количества неизменных именованных сущностей, содержащихся в исходном тексте и стилизованном тексте. Как указывалось выше, на этапе 240, генерируется массив стилизованных текстов. Для повышения семантической точности между стилизованным и исходным текстом, стилизованный текст не должен искажать факты и не быть дополнен новыми логическими связями. В свою очередь, указанный массив стилизованных текстов может содержать релевантные стилизованные текста, которые удовлетворяют характерным стилистическим чертам, однако такие тексты будут некорректными, например, ввиду изменения фактов такого текста.

Пример:

Исходный текст: *Думаю тебе стоит прочитать произведение Александра Сергеевича Пушкина «Евгений Онегин»*

Некорректный стилизованный текст: *Думаю Вам стоит прочитать произведение Александра Васильевича Пушкина «Евгений Онегин»*

Как видно из примера, сгенерированный текст удовлетворяет стилистическим чертам официального стиля (обращение на "Вы"), т.е. является релевантным, однако, изменяет факты текста (имя автора произведения).

Для решения данной проблемы был предложен подход, заключающийся в сравнении именованных сущностей в исходном тексте и каждом стилизованном тексте из набора. Для этого, на первом шаге в исходном тексте выполняют распознавание именованных сущностей (например, имена, названия топонимов и организаций) и сохраняют их в памяти, например, в памяти системы 100, в виде файла данных. На следующем шаге выполняют распознавание именованных сущностей в каждом стилизованном тексте из массива. После этого выполняют сравнение неизменных сущностей между исходными и стилизованными текстами. Соответственно, кандидаты, отличающиеся от исходного текста по именованным сущностям, отбрасываются, что дополнительно повышает семантическую точность стилизации текста за счет исключения изменений/дополнений фактами стилизованного текста.

Таким образом, на этапе 260 осуществляется фильтрация массива стилизованных текстов.

На этапе 270 осуществляют ранжирование отфильтрованных стилизованных текстов и выбор лучшего стилизованного текста, причем выбор лучшего текста основан на попарном расстоянии между исходным текстом и каждым из возможных стилизованных текстов. На этапе 270 набор отфильтрованных стилизованных текстов ранжируется по посимвольной близости с исходным текстом. Указанное ранжирование осуществляется на основе расстояния Левенштейна. Для этого вычисляются попарные расстояние между исходной нейтральной фразой и каждым из возможных кандидатов. В качестве лучшего выбирается кандидат, расстояние Левенштейна для которого минимально. Расстояние Левенштейна вычисляется с помощью функции distance из библиотеки Levenshtein. В результате получаем стилизованную реплику цифрового ассистента, с наименьшим изменением исходного текстового фрагмента, что соответственно повышает точность всего алгоритма стилизации и уменьшает вероятность добавления новых

фактов. Кроме того, в одном частном варианте осуществления заявленного решения полученный в результате ранжирования текст может быть дополнительно проверен на наличие стилизации. Проверка наличия стилизации текста осуществляется на основе определения произошла ли замена целевых индикаторов стиля. Так, например, система 100 может дополнительно содержать модуль проверки стилизации, реализованный на базе вычислительного устройства. Указанный модуль выполнен с возможностью проверки для пары (исходная реплика, стилизованная реплика,) происходит ли при стилизации замена целевых индикаторов стиля, например, рода/числа или числа в обращении Ты/Вы по сравнению с оригинальной фразой (есть много нейтральных фраз, для которых замена не требуется). Данный модуль выполнен с возможностью замены только в том случае, если происходит хотя бы одно изменение. Таким образом, в случае если происходит замена, то на выходе мы получаем стилизованную реплику, а если замены не произошло - то исходную, которая является нейтральной и подходит для любого стиля. На этапе 280 выполняют передачу стилизованного текста в диалоговую систему. На указанном этапе 280 стилизованная реплика цифрового ассистента может быть сохранена в памяти системы в виде файла и отправлена в диалоговую систему посредством, например, канала передачи данных для последующего отображения пользователю, например, с помощью интерфейса ввода-вывода диалоговой системы 120.

Таким образом, в вышеприведенных материалах были описаны система и способ генерации текста для цифровых ассистентов в диалоговых системах, обеспечивающий высокую семантическую точность генерации стилизованного текста.

Кроме того, стоит отметить, что благодаря реализации заявленного решения также обеспечивается универсальность стилизации текста, позволяющая генерировать текст не только в одном стиле, но и обеспечивать возможность стилизации исходного текста в нескольких стилях, в зависимости от сферы применения. Указанная особенность исключает необходимость в отдельном формировании для каждого стиля уникальной стилизованной реплики. Благодаря применению заявленного решения выполняется возможность подачи одного исходного текста и генерирование на его основе разных стилизованных текстов.

Теперь рассмотрим один из примеров реализации заявленного технического решения.

Одно из возможных применений системы заключается в стилизации изначально нейтрального ответа под стиль заданного ассистента. Так, как упоминалось выше, диалоговые системы могут содержать набор цифровых ассистентов, каждому из которых присущ свой стиль общения. При запросе пользователя, диалоговая система генерирует изначально нейтральную реплику (исходный текст). Указанный способ 200 обеспечивает возможность генерации из указанной нейтральной реплики, например, ответа цифрового ассистента, стилизованную реплику (стилизованный текст) в соответствии с заданным стилем цифрового ассистента. Это исключает необходимость в генерации множества стилизованных реплик под каждого цифрового ассистента. Также, еще одним преимуществом такого подхода является гибкость в выборе стиля, генерация текстового ответа - вычислительно сложная операция, под нее необходимо обучать большую модель.

Заявленное решение в свою очередь позволяет стилизовать сгенерированный исходный ответ с помощью системы 100 под разные стили.

На фиг. 3 представлен пример общего вида вычислительной системы (300), которая обеспечивает реализацию заявленного способа или является частью компьютерной системы, например, модулями 101-103, сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

В общем случае система (300) содержит такие компоненты, как: один или более процессоров (301), по меньшей мере одну память (302), средство хранения данных (303), интерфейсы ввода/вывода (304), средство В/В (305), средство сетевого взаимодействия (306), которые объединяются посредством универсальной шины.

Процессор (301) выполняет основные вычислительные операции, необходимые для обработки данных при выполнении способа (200). Процессор (301) исполняет необходимые машиночитаемые команды, содержащиеся в оперативной памяти (302). Память (302), как правило, выполнена в виде ОЗУ и содержит необходимую программную логику, обеспечивающую требуемый функционал.

Средство хранения данных (303) может выполняться в виде HDD, SSD дисков, рейд массива, флэш-памяти, оптических накопителей информации (CD, DVD, MD, Blue-Ray дисков) и т.п. Средства (303) позволяют выполнять долгосрочное хранение различного вида информации, например сгенерированных стилизованных реплик, идентификаторов пользователей, идентификаторов цифровых ассистентов и т.п.

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. SAT A. 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. Выбор интерфейсов (304) зависит от конкретного исполнения системы (300). которая может быть реализована на базе широко класса устройств, например, персональный компьютер, мейнфрейм. ноутбук, серверный кластер, тонкий клиент, смартфон, сервер и т.п.

В качестве средств В/В данных (305) может использоваться: клавиатура, джойстик, дисплей (сен-

сорный дисплей), монитор, сенсорный дисплей, тачпад. 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. Средства сетевого взаимодействия (306) выбираются из устройств, обеспечивающий сетевой прием и передачу данных, например. Ethernet карту. WLAN/Wi-Fi модуль. Bluetooth модуль. BLE модуль. NFC модуль. IrDa. RFID модуль. GSM модем и т.п. С помощью средств (305) обеспечивается организация обмена данными между, например, системой (300). представленной в виде сервера и вычислительным устройством пользователя, на котором могут отображаться полученные данные (сгенерированная стилизованная реплика цифрового ассистента) по проводному или беспроводному каналу передачи данных, например, WAN, PAN, ЛВС (LAN), Интранет, Интернет, WLAN, WMAN или GSM.

Конкретный выбор элементов системы (300) для реализации различных программно-аппаратных архитектурных решений может варьироваться с сохранением обеспечиваемого требуемого функционала.

Представленные материалы заявки 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. Таким образом, объем настоящего технического решения ограничен только объемом прилагаемой формулы.

Источники информации.

1. An Empirical Study on Multi-Task Learning for Text Style Transfer and Paraphrase Generation, Paweł Bujnowskia, Kseniia Ryzhovac, Hyungtak Choib, Katarzyna Witkowskad, Jarosław Piersaa, Tymoteusz Krumholca and Katarzyna Beksa. Найдено в Интернет по ссылке: https://aclanthology.org/2020.coling-industry.6.pdf , 20.04.2022.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ автоматической генерации текста для цифрового ассистента в диалоговых системах, выполняемый по меньшей мере одним вычислительным устройством и содержащий этапы, на которых:

a) получают входные данные, содержащие исходный текст на естественном языке и целевой стиль реплик цифрового ассистента;

b) осуществляют кодирование исходного текста, причем в ходе кодирования выполняют, по меньшей мере, токенизацию текстовых данных;

c) выполняют векторизацию токенов, полученных на этапе b);

d) осуществляют обработку векторных представлений токенов исходного текста, полученных на этапе c), с помощью модели машинного обучения на базе нейронной сети, обученной на стилизованных, в соответствии с заданным целевым стилем, текстовых репликах цифрового ассистента, в ходе которой осуществляется формирование массива векторизированных стилизованных текстов;

e) осуществляют декодирование каждого векторизированного стилизованного текста из массива, полученного на этапе d), причем в ходе декодирования выполняют, по меньшей мере, преобразование векторизированного стилизованного текста в токены и детокенизацию;

f) выполняют фильтрацию массива стилизованных текстов, полученных на этапе e);

g) осуществляют ранжирование отфильтрованных стилизованных текстов и выбор лучшего стилизованного текста, причем выбор лучшего текста основан на попарном расстоянии между исходным текстом и каждым из возможных стилизованных текстов;

h) отправляют стилизованный текст, полученный на этапе g), в диалоговую систему.

2. Способ по п.1, характеризующийся тем, что фильтрация массива стилизованных текстов выполняется с помощью регулярных выражений и морфологического анализатора.

3. Способ по п.1, характеризующийся тем, что фильтрация массива стилизованных текстов выполняется на основе совпадения имен собственных в исходном тексте и каждом стилизованном тексте из массива.

4. Способ по п.3, характеризующийся тем, что совпадение имен собственных определяется с помощью количества неизменных именованных сущностей, содержащихся в исходном тексте и стилизованном тексте.

5. Способ по п.4, характеризующийся тем, что количество неизменных именованных сущностей определяется на основе распознавания именованных сущностей в исходном тексте и каждом стилизованном тексте из массива.

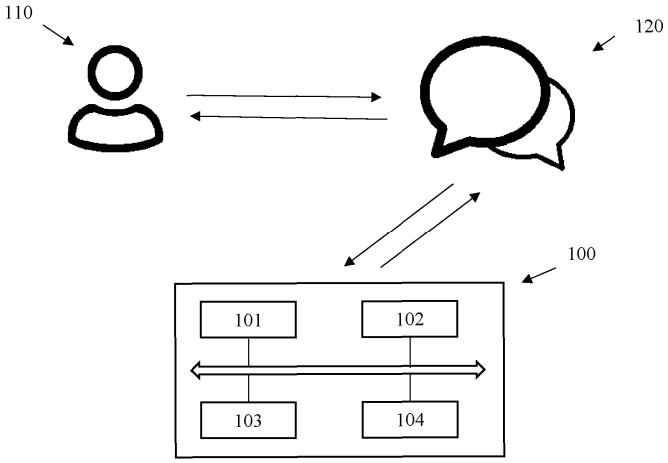
6. Способ по п.1, характеризующийся тем, что дополнительно содержит этап проверки наличия стилизации текста.

7. Способ по п.6, характеризующийся тем, что проверка наличия стилизации текста осуществляется на основе определения, произошла ли замена целевых индикаторов стиля.

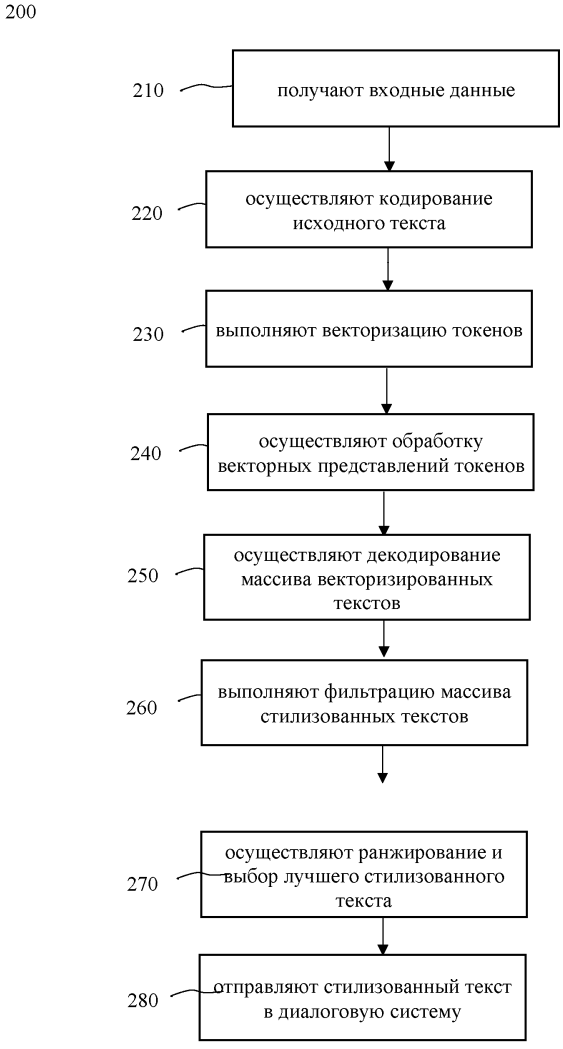
8. Система автоматической генерации текста для цифрового ассистента в диалоговых системах, содержащая:

по меньшей мере один процессор;

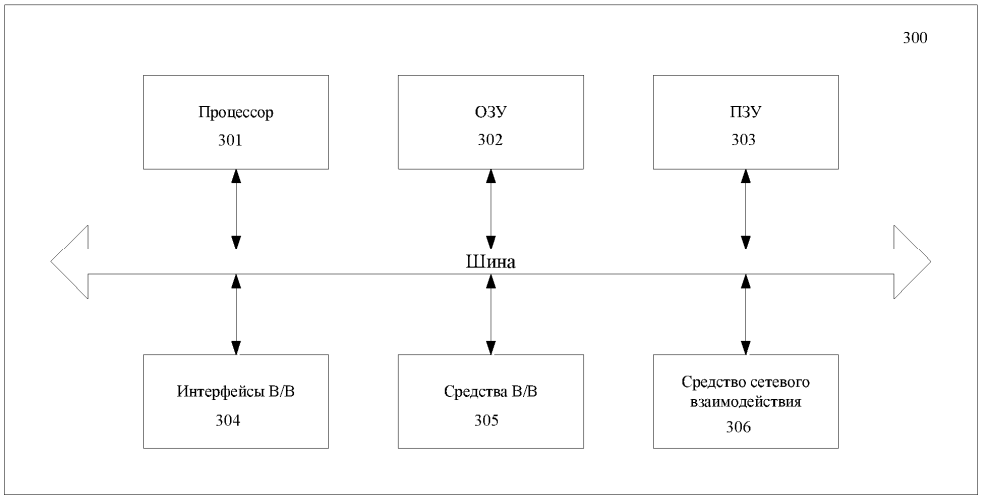
по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа по любому из пп.1-7.



Фиг. 1



Фиг. 2



Фиг. 3

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