(19) Евразийское патентное ведомство

(11) 041027 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2022.08.30

(21) Номер заявки
202092862

(22) Дата подачи заявки
2020.12.23

(51) Int. Cl. ***G06F 40/295*** (2020.01)
G06N 20/00 (2019.01)

(54) СПОСОБ И СИСТЕМА ИЗВЛЕЧЕНИЯ ИМЕНОВАННЫХ СУЩНОСТЕЙ

(31) **2020128709**
(32) **2020.08.31**
(33) **RU**
(43) **2022.03.31**

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

(72) Изобретатель:
Емельянов Антон Александрович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) US-A1-20170060835
CN-A-108536679
WO-A1-2019228466
US-A1-20150286629

(57) Представленное изобретение относится, в общем, к области вычислительной техники, а в частности к способу и системе извлечения именованных сущностей. Техническим результатом является повышение точности предсказания именованных сущностей. Указанный технический результат достигается благодаря осуществлению способа извлечения именованных сущностей из текстовой информации, выполняемого по меньшей мере одним вычислительным устройством, содержащего этапы, на которых: получают текстовую информацию; выполняют разбиение текста на слова; выполняют токенизацию текста для получения последовательности токенов; формируют посредством нейронной сети для полученной последовательности токенов набор векторов; формируют на основе полученного набора векторов векторное представление последовательности токенов; посредством сравнения показателей полученного векторного представления последовательности токенов с заранее заданными показателями векторов, полученными в результате обучения нейронной сети, осуществляют предсказание именованных сущностей для векторного представления последовательности токенов; распознают полученные на предыдущем этапе именованные сущности посредством подбора метки слова.

041027 B1

041027 B1

Область техники

Представленное изобретение относится, в общем, к области вычислительной техники, а в частности к способу и системе извлечения именованных сущностей. Техническое решение находит применение во многих областях, связанных с обработкой текстов на естественных языках и извлечением информации из текста.

Уровень техники

В настоящий момент задача извлечения именованных сущностей из текстовых данных требуется в широком ряде направлений и прикладных задач. В частности, при построении диалоговых систем (чат-ботов, умных помощников) извлечение именованных сущностей необходимо для правильного понимая смысла текста, намерения пользователя и формирования ответа на запросы пользователя.

В промышленных областях, где требуется классификация документов, извлечение именованных сущностей может применяться для генерации новых признаков текста, чтобы улучшить качество работы классификации в условиях реального применения.

Задача извлечения именованных сущностей является подзадачей в ряде направлений, таких как анализ текстовых документов, классификация отзывов, кластеризация текстовых коллекций, в диалоговых системах и др. В настоящее время используется ряд подходов для решения задачи извлечения именованных сущностей из текстов на русском языке, каждый из которых обладает своими преимуществами и недостатками:

Named Entity Recognition (NER) от DeepPavlov, URL: http://docs.deeppavlov.ai/en/master/features/models/ner.html. Решение основано на дообучении языковой модели RuBERT под задачу извлечения именованных сущностей. Система имеет более 100 млн обучаемых параметров. Обучение с помощью этой модели требуют больших объемов GPU памяти (более 3 GB);

Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition. URL: https://arxiv.org/pdf/1709.09686.pdf. Решение основано на конкатенации Bi-LSTM char слоя, Token Embedding слоя для кодирования входной последовательности и Bi-LSTM-CRF слоя для предсказания сущностей. Данное решение показывает худшее качество, нежели современные модели, основанные на больших языковых моделях, однако в памяти при обучении занимает в разы меньше места;

Deep-NER: named entity recognizer based on deep neural networks and transfer learning. Модель имеет в основе ELMO (Matthew E. Peters, Mark Neumann, Mohit lyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. Deep contextualized word representations. 2018. URL: https://arxiv.org/abs/1802.05365) или BERT языковую модель. Далее идет BiLSTM-CRF слой для предсказаний. Система имеет более 100 млн обучаемых параметров и занимает в памяти при обучении более 3-4 GB GPU памяти. Работает хуже, чем модель от DeepPavlov.

Томита-парсер. URL: https://yandex.ru/dev/tomita/. Томита-парсер создан для извлечения структурированных данных из текста на естественном языке. Вычленение фактов происходит при помощи контекстно-свободных грамматик и словарей ключевых слов. Парсер позволяет писать свои грамматики и добавлять словари для нужного языка. С его помощью можно извлекать именованные сущности. Основной недостаток состоит в сложности написания правил: для каждой новой сущности требуется новая группа правил, что требует затраты времени специалиста. Однако данный инструмент не требует обучающих данных. Основные недостатки приведенных нейросетевых моделей состоят в том, что они требуют большие объемы памяти при обучении и предсказании и наличия соответствующего оборудования. Использование томита-парсера требует высококвалифицированного специалиста для написания грамматик.

Раскрытие изобретения

Технической проблемой или задачей, поставленной в данном техническом решении, является создание нового эффективного, простого и надежного метода извлечения именованных сущностей из текста на русском языке. При этом решение должно обеспечить высокое качество извлечения именованных сущностей и требовать как можно меньше GPU памяти (менее 2 GB). Также решение должно уметь решать задачу классификации именованных сущностей (одновременно - Joint learning (Bing Liu, Lane Ian. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv:1609.01454) или по отдельности).

Техническим результатом является повышение точности предсказания именованных сущностей. Указанный технический результат достигается благодаря осуществлению способа извлечения именованных сущностей из текстовой информации, выполняемого по меньшей мере одним вычислительным устройством, содержащего этапы, на которых:

получают текстовую информацию;

выполняют разбиение текста на слова;

выполняют токенизацию текста для получения последовательности токенов;

формируют посредством нейронной сети для полученной последовательности токенов набор векторов;

формируют на основе полученного набора векторов векторное представление последовательности токенов;

посредством сравнения показателей полученного векторного представления последовательности

токенов с заранее заданными показателями векторов, полученными в результате обучения нейронной сети, осуществляют предсказание именованных сущностей для векторного представления последовательности токенов;

распознают полученные на предыдущем этапе именованные сущности посредством подбора метки слова.

В одном из частных примеров осуществления способа векторное представление последовательности токенов на основе набора векторов формируют посредством расчета взвешенной суммы или средних значений показателей векторов, содержащихся в наборе векторов.

В другом частном примере осуществления способа дополнительно выполняют этап, на котором корректируют размерность векторов, содержащихся в полученном векторном представлении последовательности токенов, для получения векторного представления последовательности токенов с заданной размерностью.

В другом частном примере осуществления способа дополнительно выполняют этапы, на которых:

на основе показателей векторов, содержащихся в векторном представлении последовательности токенов, и показателей предыдущих векторов в упомянутой последовательности посредством рекуррентного слоя нейронной сети определяют зависимости показателей векторов токенов;

добавляют информацию о зависимостях показателей векторов токенов к представлению последовательности токенов посредством формирования нового векторного представления последовательности токенов.

В другом частном примере осуществления способа дополнительно выполняют этапы, на которых:

определяют для каждого вектора в векторном представлений последовательности токенов зависимости его показателей от показателей других векторов;

генерируют на основе показателей векторного представления последовательности токенов и данных о зависимостях показателей векторов, определенных на предыдущем этапе, новое векторное представление последовательности токенов.

В другом частном примере осуществления способа финальная метка слова может быть определена посредством нейронных сетей методом голосования.

В другом частном примере осуществления способа дополнительно выполняют этапы, на которых:

на основе показателей векторного представления последовательности токенов определяют максимальные значения этих показателей по всем векторам и на их основе формируют вектор, содержащий максимальные значения показателей полученного векторного представления последовательности токенов;

на основе показателей векторного представления последовательности токенов определяют средние значения этих показателей по всем векторам и на их основе формируют вектор, содержащий средние значения показателей полученного векторного представления последовательности токенов;

на основе показателей вектора, содержащего максимальные значения показателей векторного представления последовательности токенов, вектора, содержащего средние значения показателей полученного векторного представления последовательности токенов, и показателей последнего вектора в векторном представлении последовательности токенов формируют результирующую вектор;

осуществляют классификацию результирующего вектора для определения класса векторного представления последовательности токенов. В другом предпочтительном варианте осуществления заявленного решения представлена система извлечения именованных сущностей, содержащая по меньшей мере одно вычислительное устройство и по меньшей мере одно устройство памяти, содержащее машиночитаемые инструкции, которые при их исполнении по меньшей мере одним вычислительным устройством выполняют вышеуказанный способ.

Краткое описание чертежей

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей, на которых:

на фиг. 1 представлена общая схема взаимодействия элементов системы извлечения именованных сущностей;

на фиг. 2 представлен пример общего вида системы извлечения именованных сущностей.

Осуществление изобретения

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

В данном техническом решении под системой подразумевается, в том числе, 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 операций (действий, инструкций).

Под устройством обработки команд подразумевается электронный блок, вычислительное устройство, либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного

или более устройств хранения данных. В роли устройства хранения данных могут выступать, но не ограничиваясь, жесткие диски (HDD), флеш-память, ПЗУ (постоянное запоминающее устройство), твердотельные накопители (SSD), оптические приводы.

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

База данных (БД) - совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения (ISO/IEC 2382:2015, 2121423 "database").

Сущность - любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Экземпляр сущности - это конкретный представитель данной сущности. Например, представителем сущности "Сотрудник" может быть "Сотрудник Иванов". Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности. Именованная сущность - объект определенного типа, имеющий имя, название или идентификатор. Какие типы выделяет система, определяется в рамках конкретной задачи. Для новостного домена обычно это люди (PER), места (LOC), организации (ORG) и разное, объекты широкого спектра: события, слоганы и т.д.

В соответствии со схемой, приведенной на фиг. 1, система 100 извлечения именованных сущностей содержит соединенные между собой: модуль 10 предобработки и токенизации текста; модуль 20 формирования векторов; модуль 30 формирования векторного представления последовательности токенов; модуль 40 определения зависимости показателей векторов токенов; модуль 50 определения типов зависимостей между токенами в последовательности векторного представления токенов; модуль 60 корректировки размерности векторов; модуль 70 предсказания именованных сущностей для токенов; модуль 80 предсказания класса последовательности векторного представления токенов; и модуль 90 распознавания именованных сущностей.

Модуль 10 предобработки и токенизации текста может быть реализован на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением, и включать набор моделей для токенизации текста, например, набор моделей WordPiece (см. Yonghui Wu, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Mike Schuster, Zhifeng Chen. 2016. Googles neural machine translation system: Bridging the gap between human and machine translation, volume arXiv:1609.08144).

Модуль 20 формирования векторов может быть реализован на базе языковой модель BERT. BERT - это нейронная сеть от Google, показавшая с большим отрывом state-of-the-art результаты на целом ряде задач. С помощью BERT можно создавать программы с ИИ для обработки естественного языка, отвечать на вопросы, заданные в произвольной форме, создавать чат-боты, автоматические переводчики, анализировать текст и так далее. Модуль 30 формирования векторного представления последовательности токенов может быть реализован на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением для определения взвешенной сумма или среднего (в зависимости от конфигурации) значения показателей векторов для формирования векторного представления последовательности токенов.

Модуль 40 определения зависимости показателей векторов токенов может быть реализован на базе рекуррентной нейронной сети (RNN), выполненной с возможностью на основе значений показателей векторов, содержащихся в векторном представлении последовательности токенов, определять рекуррентные зависимости между показателями векторов. Граф вычислений рекуррентных нейронных сетей также может содержать циклы, которые отражают зависимости значения переменной в следующий момент времени от ее текущего значения.

Модуль 50 определения типов зависимостей между токенами в последовательности векторного представления токенов может быть реализован на базе нейронной сети, содержащей Multi-Head attention нейросетевой слой. Multi-head attention - это специальный слой, который дает возможность каждому входному вектору взаимодействовать с другими словами через механизм внимания (attention mechanism), вместо передачи параметров "hidden state" как в RNN или соседних слов как в CNN. Ему на вход даются вектора Query, и несколько пар Key и Value (на практике, Key и Value это всегда один и тот же вектор). Каждый из них преобразуется обучаемым линейным преобразованием, а потом вычисляется скалярное произведение Q со всеми K по очереди, прогоняется результат этих скалярных произведений через softmax, и с полученными весами все вектора суммируются в единый вектор. Модуль 60 корректировки размерности векторного представления последовательности токенов может быть реализован на базе линейных слоев нейронной сети, предназначенных для обработки показателей векторов для корректировки их размерностей. Весовые коэффициенты для линейных слоев могут быть заданы разработчиком модуля 60 при его проектировании. Модуль 70 предсказания именованных сущностей для токенов может быть реализован на базе по меньшей мере одной нейронной сети, заранее обученной на конкретных наборах именованных сущностей и соответствующих им наборах векторов. В частности, для предсказания именованных сущностей может быть использован слой softmax, широко известный из уровня техники (см.,

например, https://ru.wikipedia.org/wiki/Softmax).

Модуль 80 предсказания класса векторного представления токенов может быть реализован на базе по меньшей мере одной нейронной сети, выполненной с возможностью осуществления операций подвыборки по максимальному (Max Pooling) и среднему (Average Pooling) значению, содержащей по меньшей мере один линейный слой для корректировки размерности векторов и слой softmax. Модуль 90 распознавания именованных сущностей может быть реализован на базе по меньшей мере одной вычислительного устройства, выполненного с возможностью преобразования данных о предсказанных именованных сущностях в слова. Упомянутый алгоритм преобразования может быть заранее задан разработчиком упомянутого модуля 90 при его проектировании.

На первом этапе работы системы 100 извлечения именованных сущностей в модуль 10 предобработки и токенизации текста поступает текст, например, от источника данных 1. Источником 1 данных может быть по меньшей мере одна база данных (БД), в которой содержится текстовая информация, либо источником 1 данных может быть устройство пользователя, например, портативный или стационарный компьютер, мобильный телефон или смартфон, планшет и пр. Модуль 10 предобработки и токенизации текста выполняет разбиение текста на слова по пробелу между словами и его токенизацию с помощью модели WordPiece для получения последовательности токенов. Например, для текста "Хочу оформить дебетовую карту" последовательность токенов будет представлять: 'Х', '##очу', 'о', '##фор', '##ми', '##ть', 'де', '##бет', '##овую', 'карт', '##у'. Также упомянутый модуль 10 добавляет специальный токен "[CLS]" в начало последовательности токенов.

Соответственно, сформированная последовательность токенов далее упомянутым модулем 10 передается в модуль 20 формирования векторов, который известными из уровня техники методами (см., например, статью Anton Emelyanov, Ekaterina Artemova. Gapping parsing using pretrained embeddings, attention mechanism and NCRF. 2019. Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" (2019). Issue 18. Supplementary volume. Pages 21-30. URL: http://www.dialog-21.ru/media/4870/-dialog2019scopusvolplus.pdf) формирует для последовательности токенов набор векторов, например, 12 векторных представлений последовательности токенов, после чего полученный набор векторов направляются упомянутым модулем 20 в модуль 30 формирования векторного представления последовательности токенов. На основе полученного набора векторов модуль 30 формирует векторное представление последовательности токенов, например, посредством расчета взвешенной суммы или средних значений показателей 12 упомянутых векторных представлений последовательности токенов.

Далее полученное модулем 30 векторное представление последовательности токенов направляется в модуль 60 корректировки размерности, который корректирует размерность векторов, содержащихся в полученном векторном представлении последовательности токенов, например, посредством умножения показателей векторов на сгенерированное случайное нецелое число, для получения векторного представления последовательности токенов с заданной размерностью.

В альтернативном варианте реализации заявленного решения полученное векторное представление последовательности токенов упомянутым модулем 30 может быть направлено в модуль 40 определения зависимости показателей векторов токенов, который на основе показателей векторов, содержащихся в векторном представлении последовательности токенов, и показателей предыдущих векторов в упомянутой последовательности, посредством рекуррентного слоя определяет зависимости показателей векторов токенов, которые могут быть выражены в виде векторов. Информация о зависимостях показателей векторов токенов добавляется упомянутым модулем 40 к векторному представлению последовательности токенов посредством формирования нового векторного представления последовательности токенов, после чего сформированное модулем 40 векторное представление последовательности токенов с информацией о зависимостях показателей векторов токенов направляется в модуль 50 определения типов зависимостей между токенами, либо в модуль 60 корректировки размерности векторного представления последовательности токенов для его обработки описанным ранее способом. Модуль 50 определения типов зависимостей между токенами при получении векторного представления последовательности токенов для каждого вектора известными из уровня техники методами определяет зависимости его показателей от показателей других векторов, входящих векторное представление последовательности токенов, которые также могут выражены в виде векторов, после чего упомянутый модуль 50 корректирует векторное представление последовательности токенов путем генерирования на основе показателей векторного представления последовательности токенов и данных о зависимостях показателей векторов нового векторного представления последовательности токенов. Данные о зависимостях показателей векторов характеризуют типы зависимостей между токенами, содержащимися в векторном представлении последовательности токенов, и могут указывать на наличие семантической или морфологической зависимости между токенами. Соответственно, сформированное векторное представление последовательности токенов модулем 50 направляется в модуль 60 корректировки размерности векторного представления последовательности токенов для его обработки описанным ранее способом.

После того, как векторное представление последовательности токенов с заданной размерностью было сформировано упомянутым модулем 60, оно направляется в модуль 70 предсказания именованных

сущностей для токенов. Модуль 70 предсказания именованных сущностей для токенов на основе сравнения показателей полученного векторного представления последовательности токенов с заранее заданными показателями векторов, полученными в результате обучения нейронной сети, осуществляет предсказание именованных сущностей для векторного представления последовательности токенов. Например, для слова "дебетовую" может быть определена следующая последовательность меток именованных сущностей в IO разметке: 'I_DEB_CARD', 'I_DEB_CARD', 'I_O'. Метка "O" означает, что сущности нет (в данном случае модель ошиблась на одном токене). Финальное предсказание именованной сущности для слова "дебетовую" будет "DEB_CARD".

Далее данные об именованных сущностях упомянутым модулем 70 направляются в модуль 90 распознавания именованных сущностей, который для каждой именованной сущности известными из уровня техники методами подбирает метку слова, причем финальная метка слова может быть определена посредством нейронных сетей методом голосования, а в качестве финальной метки выбирается та, у которой наибольшее их число. Например, для метки "I_DEB_CARD" может быть определено 2 голоса, а для метки "I_O" - 1 голос. Соответственно, финальное предсказание метки именованной сущности для слова также будет "DEB_CARD".

Также дополнительно векторное представление последовательности токенов с заданной размерностью может быть обработано модулем 80 предсказания класса последовательности векторного представления токенов. Упомянутый модуль 80 на основе показателей векторного представления последовательности токенов с заданной размерностью, полученного от упомянутого ранее модуля 60, при помощи операции Max Pooling определяет максимальные значения этих показателей по всем векторам и на их основе формирует вектор, содержащий максимальные значения показателей полученного векторного представления последовательности токенов. Также упомянутый модуль 80 на основе показателей векторного представления последовательности токенов с заданной размерностью при помощи операции Average Pooling определяет средние значения этих показателей по всем векторам и на их основе формирует вектор, содержащий средние значения показателей полученного векторного представления последовательности токенов, после чего на основе показателей вектора, содержащего максимальные значения показателей векторного представления последовательности токенов, вектора, содержащего средние значения показателей полученного векторного представления последовательности токенов, и показателей последнего вектора в векторном представлении последовательности токенов упомянутый модуль 80 формирует результирующую вектор, например, посредством операции конкатенации. Далее результирующий вектор обрабатывается линейным слоем, входящим в состав модуля 80, для корректировки его размерности описанным ранее способом, и слоем softmax для классификации результирующего вектора и определения на основе результата классификации класса для векторного представления последовательности токенов, для которого был сформирован результирующий вектор.

Информация о классе векторного представления последовательности токенов может указывать, например, на наличие или отсутствие именованных сущностей в текстовой информации или на вид запроса (намерения) пользователя, содержащемся в текстовой информации, для которой было сформировано упомянутое векторное представление последовательности токенов. Например, если текстовая информация представляет собой текст "на моей дебетовой карте с номером xxxx xxxx xxxx xxxx были операции, которых я не делал. Что делать?", в размеченном виде который будет представлен как "O O B_DEB_CARD I_DEB_CARD O B_NUM_CARD I_NUM_CARD O O O O O O O O O O O", то векторному представлению последовательности токенов, сформированному для данной текстовой информации, модулем 80 будет определен класс "позвать_оператора", указывающий на вид запроса пользователя - вызов оператора. Если текстовая информация будет представлять текст "Я хочу дебетовую карту", то класс будет определен как "хочу_карту", указывающий на вид запроса - оформление карты. Также вид запроса пользователя может указывать на желание получить кредит или прочую услугу. Данные о метках слов модулем 90 распознавания именованных сущностей и данные классификации векторного представления последовательности токенов модулем 80 предсказания класса последовательности векторного представления токенов далее могут быть направлены в устройство 2 хранения данных, для их последующего отображения пользователю по соответствующему запросу. Таким образом, за счет того, что векторное представление последовательности токенов формируют на основе набора векторов, полученных посредством нейронной сети в результате обработки данных последовательности токенов, а предсказание именованных сущностей для векторного представления последовательности токенов выполняют посредством сравнения показателей полученного векторного представления последовательности токенов с заранее заданными показателями векторов, полученными в результате обучения нейронной сети, повышается точность предсказания именованных сущностей. Также дополнительно точность предсказания именованных сущностей обеспечивается за счет того, что при предсказании учитывается информацию о зависимостях показателей векторов токенов от других векторов, содержащихся в векторном представлении последовательности токенов.

В общем виде (см. фиг. 2) система (200) извлечения именованных сущностей содержит объединенные общей шиной информационного обмена один или несколько процессоров (201), средства памяти, такие как ОЗУ (202) и ПЗУ (203), интерфейсы ввода/вывода (204), устройства ввода/вывода (205), и уст-

ройство для сетевого взаимодействия (206).

Процессор (201) (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 (200) также необходимо учитывать графический процессор, например, GPU NVIDIA с программной моделью, совместимой с CUDA, или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (202) представляет собой оперативную память и предназначено для хранения исполняемых процессором (201) машиночитаемых инструкций для выполнения необходимых операций по логической обработке данных. ОЗУ (202), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом, в качестве ОЗУ (202) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (203) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов системы (200) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (204). 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 (200) применяются различные средства (205) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, 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 (206) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (206) 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 (200), например, GPS, ГЛОНАСС, BeiDou, Galileo. Конкретный выбор элементов системы (200) для реализации различных программно-аппаратных архитектурных решений может варьироваться с сохранением обеспечиваемого требуемого функционала. Модификации и улучшения вышеописанных вариантов осуществления настоящего технического решения будут ясны специалистам в данной области техники. Предшествующее описание представлено только в качестве примера и не несет никаких ограничений. Таким образом, объем настоящего технического решения ограничен только объемом прилагаемой формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ извлечения именованных сущностей из текстовой информации, выполняемый по меньшей мере одним вычислительным устройством, содержащий этапы, на которых:

получают текстовую информацию;

выполняют разбиение текста на слова;

выполняют токенизацию текста для получения последовательности токенов;

формируют посредством нейронной сети для полученной последовательности токенов набор векторов;

формируют на основе полученного набора векторов векторное представление последовательности токенов;

посредством сравнения показателей полученного векторного представления последовательности токенов с заранее заданными показателями векторов, полученными в результате обучения нейронной сети, осуществляют предсказание именованных сущностей для векторного представления последовательности токенов;

распознают полученные на предыдущем этапе именованные сущности посредством подбора метки слова.

2. Способ по п.1, характеризующийся тем, что векторное представление последовательности токенов на основе набора векторов формируют посредством расчета взвешенной суммы или средних значе-

ний показателей векторов, содержащихся в наборе векторов.

3. Способ по п.1, характеризующийся тем, что дополнительно содержит этап, на котором корректируют размерность векторов, содержащихся в полученном векторном представлении последовательности токенов, для получения векторного представления последовательности токенов с заданной размерностью.

4. Способ по п.1, характеризующийся тем, что дополнительно содержит этапы, на которых:

на основе показателей векторов, содержащихся в векторном представлении последовательности токенов, и показателей предыдущих векторов в упомянутой последовательности посредством рекуррентного слоя нейронной сети определяют зависимости показателей векторов токенов;

добавляют информацию о зависимостях показателей векторов токенов к представлению последовательности токенов посредством формирования нового векторного представления последовательности токенов.

5. Способ по п.1, характеризующийся тем, что дополнительно содержит этапы, на которых:

определяют для каждого вектора в векторном представлении последовательности токенов зависимости его показателей от показателей других векторов;

генерируют на основе показателей векторного представления последовательности токенов и данных о зависимостях показателей векторов, определенных на предыдущем этапе, новое векторное представление последовательности токенов.

6. Способ по п.1, характеризующийся тем, что финальная метка слова может быть определена посредством нейронных сетей методом голосования.

7. Способ по п.1, характеризующийся тем, что дополнительно содержит этапы, на которых:

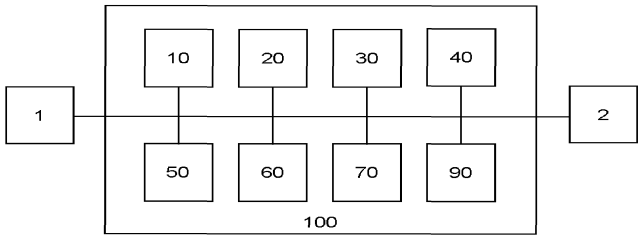
на основе показателей векторного представления последовательности токенов определяют максимальные значения этих показателей по всем векторам и на их основе формируют вектор, содержащий максимальные значения показателей полученного векторного представления последовательности токенов;

на основе показателей векторного представления последовательности токенов определяют средние значения этих показателей по всем векторам и на их основе формируют вектор, содержащий средние значения показателей полученного векторного представления последовательности токенов;

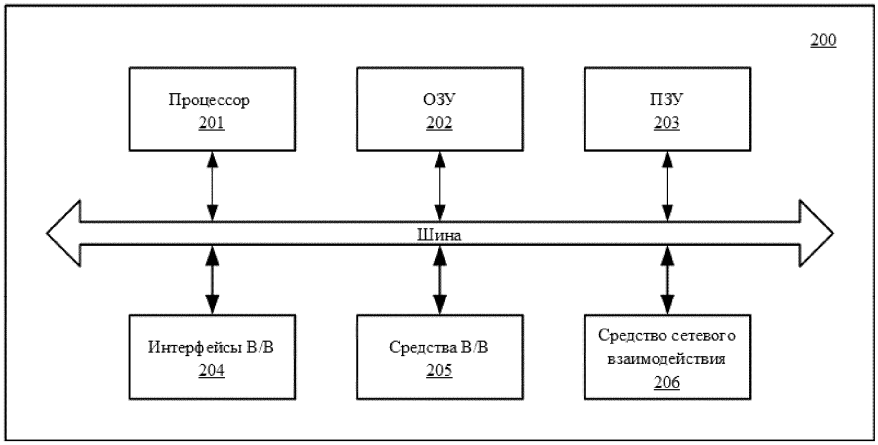
на основе показателей вектора, содержащего максимальные значения показателей векторного представления последовательности токенов, вектора, содержащего средние значения показателей полученного векторного представления последовательности токенов, и показателей последнего вектора в векторном представлении последовательности токенов формируют результирующий вектор;

осуществляют классификацию результирующего вектора для определения класса векторного представления последовательности токенов.

8. Система извлечения именованных сущностей, содержащая по меньшей мере одно вычислительное устройство и по меньшей мере одно устройство памяти, содержащее машиночитаемые инструкции, которые при их исполнении по меньшей мере одним вычислительным устройством выполняют способ по любому из пп.1-7.



Фиг. 1


200
Процессор
201
ОЗУ
202
ПЗУ
203
Шина
Интерфейсы В/В
204
Средства В/В
205
Средство сетевого взаимодействия
206

Фиг. 2