Титульный лист

РЕФЕРАТ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

*\_*

Разраб.

Провер.

Реценз.

Н. Контр.

Утверд.

Здесь тоже что-то

Лит.

Листов

Пояснительная записка \_\_ стр., \_\_ рис., \_\_ табл.

КЛЮЧЕВЫЕ СЛОВА.

Цель работы – разработка программы автоматического сопоставления названий товаров для продуктивного использования продающими компаниями. Объектом исследования являются исторические данные о продажах компании. Рассматриваются основы моделей обработки естественного языка, приводятся основные источники и способы извлечения признаков в экспериментальном наборе данных. Рассмотрена гибридная схема предобработки данных с использованием нескольких моделей машинного обучения. Приведены алгоритмы работы моделей машинного обучения и результаты исследования. Оценена точность приведенных алгоритмов.

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc134295110)

[СОДЕРЖАНИЕ 3](#_Toc134295111)

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 4](#_Toc134295112)

[ВВЕДЕНИЕ 5](#_Toc134295113)

[ВЫВОДЫ 21](#_Toc134295114)

[ЗАКЛЮЧЕНИЕ 22](#_Toc134295115)

[СПИСОК ЛИТЕРАТУРЫ 23](#_Toc134295116)

[ПРИЛОЖЕНИЕ 1 24](#_Toc134295117)

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

*ML* - Machine Learning (машинное обучение)

*AI* - Artificial Intelligence (искусственный интеллект)

*ANN* - Artificial Neural Network (искусственная нейронная сеть)

*RF* - Random Forest (случайный лес)

*KNN* - K-Nearest Neighbors (метод ближайших соседей)

*NB* - Naive Bayes (наивный байесовский классификатор)

*DL* - Deep Learning (глубокое обучение)

*CNN* - Convolutional Neural Network (сверточная нейронная сеть)

*RNN* - Recurrent Neural Network (рекуррентная нейронная сеть)

*NLP* - Natural Language Processing (обработка естественного языка)

*NLTK* - Natural Language Toolkit (инструментарий для работы с естественным языком)

*POS* - Part of Speech (часть речи)

*NER* - Named Entity Recognition (распознавание именованных сущностей)

*LSA* - Latent Semantic Analysis (латентный семантический анализ)

*LDA* - Latent Dirichlet Allocation (латентное размещение Дирихле)

*SVM* - Support Vector Machine (метод опорных векторов)

*ANN* - Artificial Neural Network (искусственная нейронная сеть)

*MLP* - Multi-Layer Perceptron (многослойный персептрон)

ВВЕДЕНИЕ

Для современного бизнеса в России ежегодно проводятся реформы, направленные на увеличение эффективности компаний и роста их количества на территории государства. В условиях стремительно развивающегося бизнеса важно своевременно найти слабое место на пути к развитию и вовремя его устранить. И, если на уровне государства, можно быстро понять, что тормозит бизнес-процессы в целом, то на уровне отдельно взятой корпорации, при рассмотрении бизнес-процессов, все может быть неоднозначно. Множество факторов могут быть годами скрыты от глаз и их эффект в работе компании не явен. В частности, большое влияние оказывает осведомленность сотрудников компании о трудовых обязанностях друг друга, поскольку понять, что процесс выстроен неэффективно можно лишь зная чем занимается каждый участник процесса.

Поскольку бизнес строится людьми и в бизнес-процессах участвуют люди, на определенном уровне абстракции, можно считать, что развитие бизнеса ограничено именно человеческими способностями, например способностью долго сохранять концентрацию или эффективно вести коммуникацию в рамках рабочих процессов.

Все это подталкивает к тому, что необходимы способы увеличения человеческой продуктивности в задачах, где человека не может полностью заменить компьютер. Машинное обучение в задачах подобного рода наиболее органично встраивается в бизнес-процесс.

Одной из проблем, с которой сталкиваются многие компании, является задача учета товарооборота. Это многоуровневый процесс, и один из его этапов — это ручное сопоставление удобочитаемых названий продуктов с кодами из системы товарооборота (*1С, SAP*). Это необходимо, поскольку разные филиалы компании, а также дилеры могут продавать один и тот же товар, но описывать его по-разному, что приводит к несоответствиям в итоговых данных. В результате весь процесс может занимать много времени, быть подверженным ошибкам и чрезмерным временным затратам. Подобные процессы выгодно автоматизировать в первую очередь, поскольку автоматизация позволит не только исключить часть ошибок, но и сделает бизнес более прогнозируемым.

Методы обработки естественного языка (*NLP*) могут применяться для автоматизации задачи сопоставления описаний продуктов со штрих-кодами или любыми другими внутренними номерами, которые представлены в плохо читаемом формате.

* **Цели и задачи работы**

Цель этого проекта — создать модель машинного обучения для автоматического сопоставления описаний продуктов с внутренними кодами компании. Это поможет упростить процесс управления данными о продуктах и ​​запасами для компаний.

Конкретные цели проекта:

1. Подготовка набора данных описаний продуктов и соответствующих им кодов;
2. Предварительная обработка данных, очистка и подготовка к обучению;
3. Изучение и анализ данных, просмотр статистик;
4. Выбор подходящих методов NLP и машинного обучения для решения задачи;
5. Проектирование и обучение модели машинного обучения на подготовленном наборе данных;
6. Оценка производительности модели и ее модернизация;
7. Тестирование модели на новых данных для обеспечения ее точности и эффективности.

В процессе достижения этих целей, можно создать надежную и эффективную систему сопоставления описаний продуктов с их кодами, которая поможет компаниям сэкономить время и ресурсы, повысив точность и согласованность данных. Кроме того, этот проект может служить основой для дальнейших исследований и разработок в области NLP и машинного обучения, применяемых для управления запасами и обработки данных о продуктах.

**Обзор литературы по NLP и машинному обучению**

Обработка естественного языка (*NLP*) — это область искусственного интеллекта, которая занимается взаимодействием между компьютерами и человеческим языком. Она включает в себя разработку алгоритмов и моделей, которые могут обрабатывать и интерпретировать текстовые или речевые данные на естественном языке.

Машинное обучение — это область искусственного интеллекта, которая включает в себя использование статистических методов и алгоритмов, позволяющих компьютерам учиться на основе данных без явного программирования. Алгоритмы машинного обучения можно обучать на больших наборах данных, чтобы распознавать закономерности и отношения в данных, что позволяет им делать прогнозы или классифицировать новые данные [1].

*NLP* и машинное обучение (*Machine Learning* – *ML*) тесно переплетены, и алгоритмы машинного обучения часто используются в задачах *NLP* для анализа, классификации и генерации текста на естественном языке. Например, алгоритмы машинного обучения можно использовать для анализа тональности, классификации текста, распознавания именованных сущностей и машинного перевода.

Существует несколько методов и моделей, используемых в NLP и машинном обучении, включая модели глубокого обучения, такие как сверточные нейронные сети, рекуррентные нейронные сети и модели с архитектурой трансформеров. Эти модели можно обучать на больших наборах данных для выполнения сложных задач, таких как языковой перевод, определение смысла текста и ответы на вопросы.

В последние годы *NLP* и машинное обучение приобретают все большее значение в широком спектре приложений, включая чат-боты, виртуальных помощников, обслуживание клиентов, обнаружение мошенничества и медицинскую диагностику. По мере того, как данных становится все больше, а вычислительная мощность растет, возможности *NLP* и машинного обучения становятся шире, и мы можем ожидать еще большего прогресса в будущем [2].

**Основные методы машинного обучения в *NLP***

* 1. **Обучение с учителем**

Обучение с учителем — это популярный подход к обработке естественного языка, который использует алгоритмы обучения на размеченных данных. В этом типе обучения модель обучается на наборе пар ввода-вывода с целью изучения сопоставления между ними. Затем это сопоставление можно использовать для прогнозирования выходных данных для новых, неизвестных входных данных. На рисунке 1.1 изображена архитектура алгоритма обучения с учителем [7].

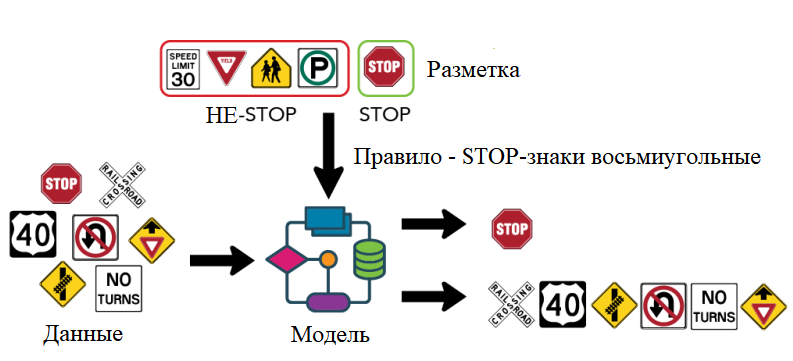


Рисунок 1.1 – Образная схема работы ML с учителем

Одним из распространенных применений обучения с учителем является анализ тональности текста. Эта задача подразумевает обучение модели классифицировать текст как положительный, отрицательный или нейтральный на основе его тональности. Для этого в обучении модели используется размеченный набор данных с текстами. Модель использует полученные закономерности для прогнозирования тональности нового текста.

Еще одно популярное применение называется распознаванием именованных сущностей (*Name Entity Recognition – NER*). Оно включает в себя обучение модели идентифицировать и классифицировать именованные сущности, такие как люди, организации и местоположения, в тексте. Эта задача особенно актуальна в тех случаях, когда целью является извлечение структурированной информации из неструктурированного текста, например обширный поиск по документам, с целью восстановления логики документооборота. Примером области, где необходимо использовать такой подход может быть аудит и консалтинг. В работе аудитора главной целью является валидация деятельности компании, и главный способ проверки – непредвзятое восстановление последовательности документооборота от начала до конца цикла. Если программа поможет найти именованные сущности (фамилии, должности и т.д.), то процесс существенно ускорится, а его продуктивность вырастет.

Чтобы обучить модель машинного обучения с учителем, обычно требуется несколько шагов. Во-первых, необходимо найти или создать набор размеченных данных. Во-вторых, требуется предварительная обработка: токенизация, выделение корней и извлечение признаков. И затем модель можно обучить с использованием одного алгоритмов ML, таких как логистическая регрессия или нейронные сети [3]. Выбор метода токенизации и алгоритма обучения модели может варьироваться в зависимости от результатов на каждом шаге.

* 1. **Обучение без учителя**

Обучение без учителя — еще один популярный подход к обработке естественного языка, который включает в себя алгоритмы обучения на неразмеченных данных. В этом типе обучения модели не даются явные метки или частично размеченные данные, вместо этого она должна самостоятельно находить закономерности и структуру в данных (рисунок 2.1) [7].

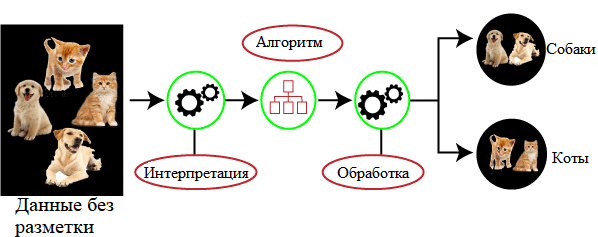


Рисунок 2.1 – Образная схема работы ML без учителя

Одним из распространенных применений моделей обучения без учителя является создание заголовков к текстам. Эта область включает в себя обучение модели для определения тем, присутствующих в наборе документов, без явных меток для этих тем. В модели используются такие методы, как кластеризация и семантический анализ, для группировки похожих документов и выявления общих тем.

Процесс обучения модели без учителя менее структурирован, чем для обучения с учителем. Вместо размеченных данных модель обучается на необработанных, неразмеченных данных. Предварительная обработка по-прежнему является необходимой, требуется токенизация и выделение корней, однако извлечение признаков здесь обычно не используется. Модель обучается с использованием таких алгоритмов, как кластеризация или автоэнкодеры. Обученная модель оценивается на основе ее способности идентифицировать полезные закономерности или выделять структуру в данных.

* 1. **Обучение с подкреплением**

Обучение с подкреплением — это частный случай машинного обучения с учителем, в котором агенты (системы машинного обучения) обучаются принимать решения на основе обратной связи из их среды. В обработке естественного языка обучение с подкреплением можно использовать для обучения моделей генерации текста, это могут быть чат-боты или системы машинного перевода.

При обучении с подкреплением агент совершает действия в окружающей среде и получает обратную связь в виде вознаграждения или наказания. Цель агента — изучить политику, которая максимизирует ожидаемое совокупное вознаграждение с течением времени (рисунок 1.3). В текстовых моделях действия агента могут включать генерацию ответа на запрос пользователя или перевод предложения с одного языка на другой.

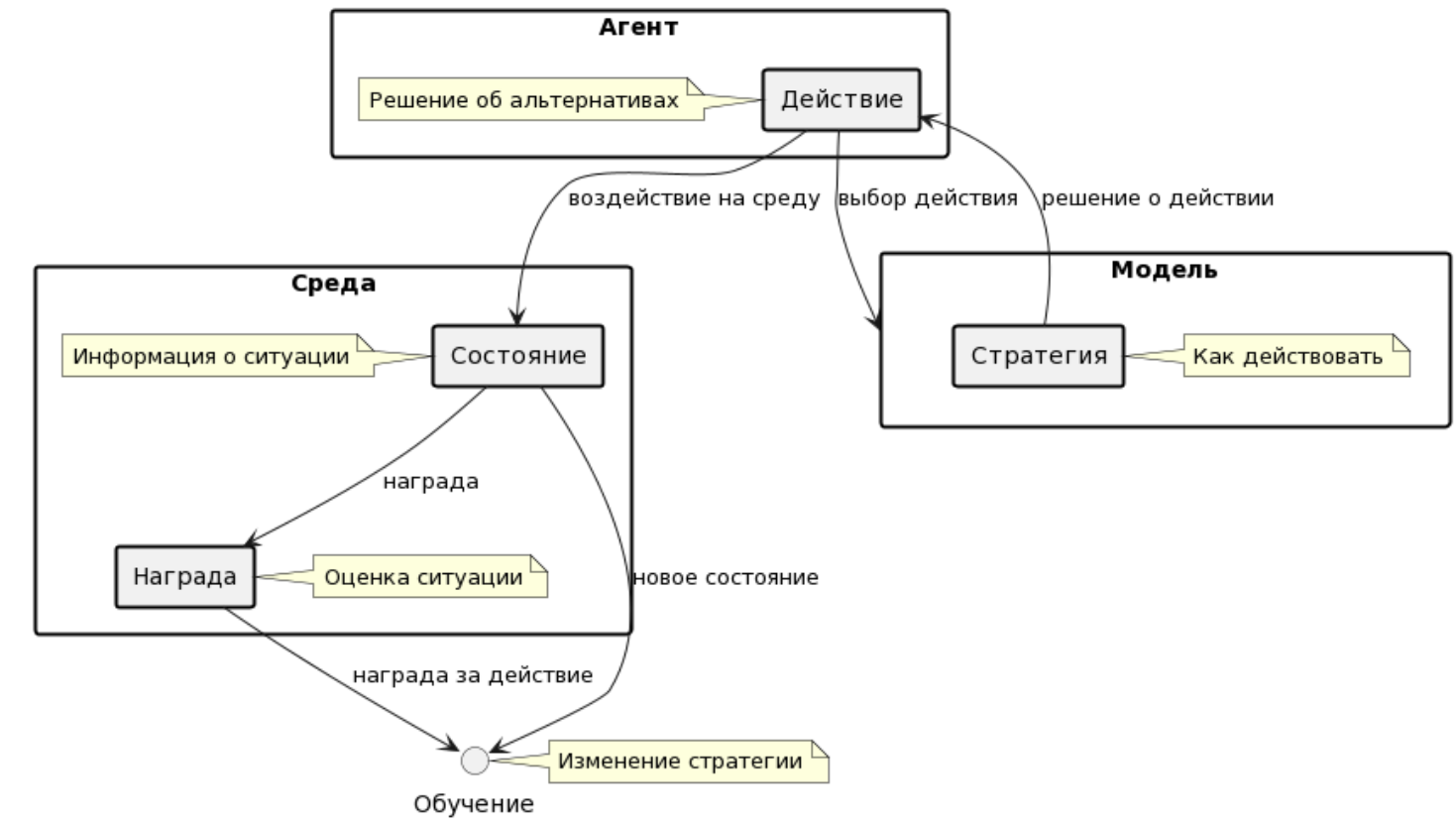


Рисунок 1.3 – Схема работы модели при обучении с подкреплением

Еще одним из распространенных применений обучения с подкреплением в NLP является создание диалогов. В этой задаче агент должен генерировать ответ на запрос пользователя, который является информативным, релевантным и естественным. Агент обучается с использованием сигнала вознаграждения, который измеряет качество сгенерированных им ответов на основе таких показателей, как беглость, актуальность и согласованность [4].

* 1. **Глубокое обучение**

Нейронные сети — это мощный класс алгоритмов машинного обучения, которые можно использовать для широкого круга задач обработки естественного языка. Простейший пример схемы работы модели глубокого обучения приведен на рисунке 1.4 [7].

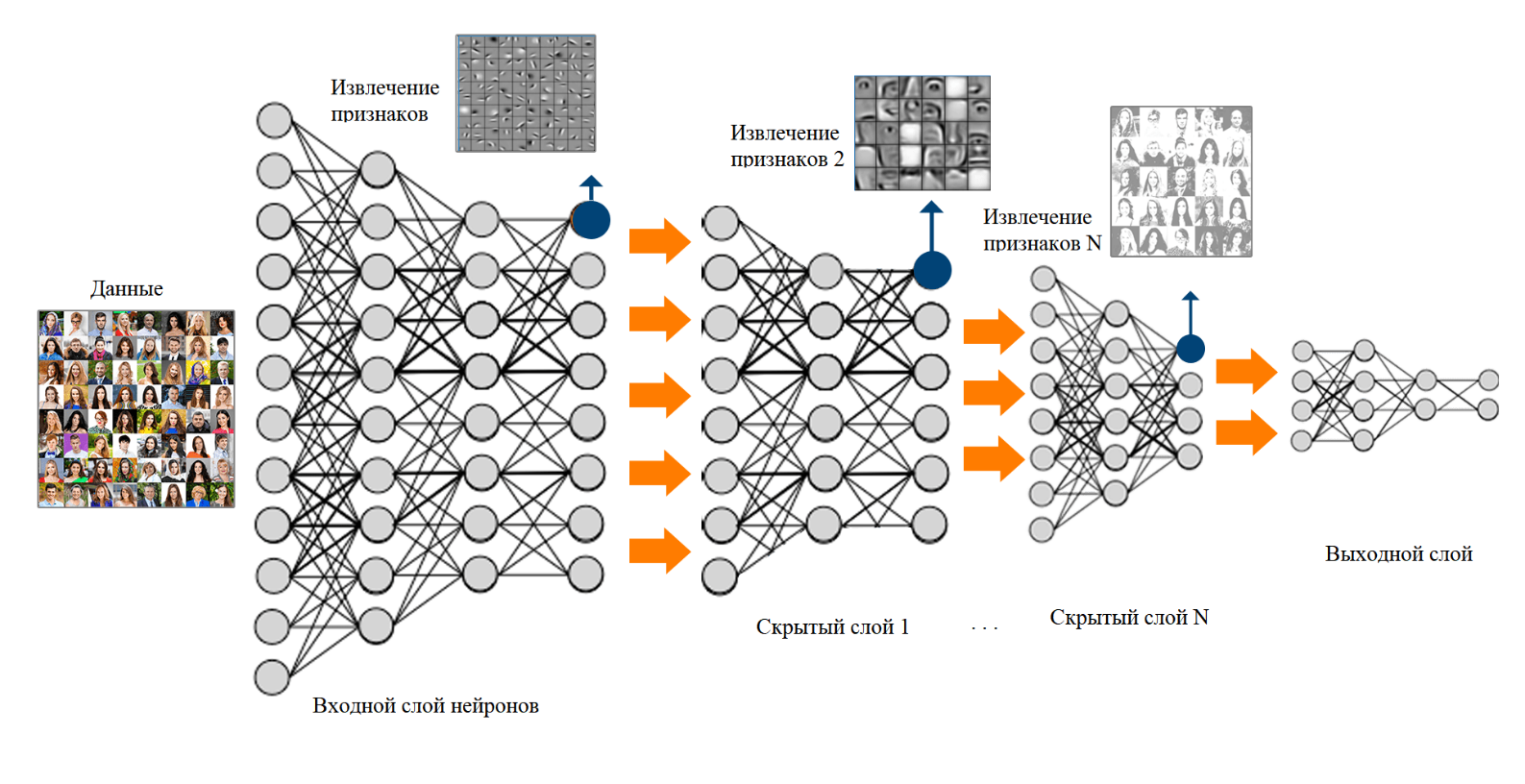


Рисунок 1.4 – Схема работы модели с глубоким обучением

Распространенным применением нейронных сетей является классификация текста. В этой задаче цель состоит в том, чтобы классифицировать текст по одной или нескольким категориям, таким как анализ настроений или классификация тем, наличие или отсутствие в тексте токсичных высказываний. Нейронные сети можно использовать для изучения соответствия между входным текстом и выходными категориями, для этого используют такие алгоритмы, как сверточные нейронные сети (*CNN*) или рекуррентные нейронные сети (*RNN*).

Другое применение нейронных сетей — языковое моделирование. Для этого обучают модель предсказывать распределения вероятностей по последовательностям слов в языке. Затем это используют для создания нового текста, а также для повышения производительности других задач обработки текста, таких как распознавание речи и машинный перевод.

**Техники предобработки текстовых данных**

Методы предварительной обработки текста имеют решающее значение в задачах обработки естественного языка, поскольку они помогают преобразовывать необработанные текстовые данные в формат, который может быть легко понят и проанализирован моделями машинного обучения. Методы предварительной обработки — это ряд шагов, которые очищают (удаляют лишнее, например символы и знаки препинания) и подготавливают текстовые данные для дальнейшего анализа. Большинство современных методов предварительной обработки текста включают токенизацию, удаление символов и стоп-слов, выделение корней и лемматизацию.

Токенизация — это процесс разбиения текста на отдельные слова (предложения, словосочетания) или, по-другому, токены. Этот шаг важен, поскольку он позволяет модели машинного обучения анализировать текст по одному слову или предложению, а не целиком.

Удаление стоп-слов — это процесс удаления общих слов, таких как «и», «тот» и «есть», которые не вносят большого вклада в смысл текста, а также удаление любых других слов и символов, не влияющих на смысл, а носящих орфографический характер. Этот метод помогает уменьшить размерность данных и, в дальнейшем, повысить эффективность модели.

Стемминг — это процесс приведения слова к его корневой форме, который помогает объединить слова, имеющие одинаковое значение, но разные формы. Например, «ходьба», «ходили» и «поход» будут сокращены до «ходить». Также существует схожий метод – лемматизация. Лемматизация похожа на стемминг, но включает в себя приведение слова к его базовой форме в зависимости от его части речи. Например, глагол «были» будет сокращен до «быть», а существительное «мыши» будет сокращен до «мышь».

Автоматы – в машинном обучении автоматы используются для моделирования поведения системы на основе входных данных. Автомат может быть описан как набор состояний и переходов между этими состояниями в зависимости от входных данных. Автоматы могут использоваться для сжатия текста, например, создавая конечный автомат для определения повторяющихся фрагментов в тексте и заменяя их на одну ссылку на этот фрагмент. Это позволяет сократить объем текста без потери информации.

Существуют и другие методы предварительной обработки текста, которые могут быть нацелены на удаление пунктуации, преобразование всего текста в нижний регистр и обработку специальных символов и числовых данных (если числовые данные не влияют на смысл текста в задаче). Конкретные используемые методы зависят от характера текстовых данных и требований задачи машинного обучения. В целом, предварительная обработка текста является важным шагом в *NLP*, который может существенно повлиять на точность и производительность моделей машинного обучения.

**Алгоритмы классификации текстовых данных**

Классификация текста предполагает присвоение предопределенных категорий неструктурированным текстовым данным на основе их содержания. Существует несколько алгоритмов, которые были разработаны для классификации текста, каждый из которых имеет свои сильные и слабые стороны. Одним из популярных алгоритмов классификации текста являются метод опорных векторов (SVM).

SVM — это тип алгоритма машинного обучения, который используется для классификации и регрессионного анализа. SVM работает путем нахождения наилучшей границы между классами точек данных в многомерном пространстве объектов. SVM является популярным выбором для классификации текста из-за его способности обрабатывать многомерные данные и достигать высокой точности с небольшими наборами данных [5].

Другим часто используемым алгоритмом классификации текста является наивный байесовский классификатор. Наивный Байесовский классификатор — это вероятностный алгоритм, который хорошо работает с большими наборами данных. Это простой и эффективный алгоритм, который предполагает, что каждая функция независима от любой другой функции, что и называют "наивным" предположением. Несмотря на это, наивный Байес часто хорошо справляется с задачами классификации текстов [5].

Деревья решений — это еще один алгоритм, который может быть использован для классификации текста. Деревья решений — это тип модели, который использует древовидный граф для моделирования решений и их возможных последствий. Деревья решений поддаются интерпретации, что означает, что легче понять, как алгоритм пришел к своему решению, по сравнению в SVM или другими алгоритмами. Это может быть полезно в областях, где важны интерпретируемость и объяснимость [6].

Методы, основанные на глубоком обучении, такие как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), также показали высокую производительность в задачах классификации текстов, особенно в задачах, связанных с последовательными данными, таких как анализ настроений и языковое моделирование. CNN особенно полезны в задачах распознавания изображений, но их также можно адаптировать для классификации текста, рассматривая слова как пиксели на одномерном изображении. В то же время, RNN полезны в задачах, связанных с последовательными данными, таких как языковое моделирование и машинный перевод. RNN способны учитывать контекст каждого слова в предложении или документе, что приводит к повышению производительности в задачах классификации и генерации текста [6].

Есть множество алгоритмов, которые могут быть использованы для классификации текста, и выбор алгоритма зависит от характера данных и конкретной поставленной задачи. Исследователи и практики часто сравнивают производительность различных алгоритмов на одинаковых наборах данных, чтобы определить наиболее подходящий подход для данной задачи [6].

**Задача поиска штрихкода по описанию товара**

**Анализ существующих методов и моделей**

Классические подходы к поиску штрих-кода на основе описания продукта используют примитивные сопоставления строк или регулярных выражений для идентификации потенциальных объектов со штрих-кодами. Эти методы часто основаны на правилах и определенных шаблонах или форматах, которым обычно следуют алгоритмы поиска и генераторы штрих-кодов.

Классический метод заключается в поиске одномерных штрих-кодов, соответствующих стандартам *EAN-13 (European Article Number)* или *UPC-A (Universal Product Code)* (рисунок 1.6) и коды «Честный ЗНАК» (рисунок 1.7). Эти штрих-коды состоят из 12, 13 или 40 цифр и символов соответственно, и могут быть легко распознаны по их характерным особенностям начала и конца. Как только потенциальный объект штрих-кода идентифицирован, происходит сверка с помощью контрольной суммы.



Рисунок 1.6 – Пример кодов *EAN* и *UPC*



Рисунок 1.7 – Пример кода «Честный ЗНАК»

Нетрудно заметить, что функциональность цифробуквенных кодов высокая, в них может содержаться много информации о товаре, однако вся информация никак не интерпретируется человеком без специального оборудования. Это означает, что даже очень информативный штрих-код или *qr*-код, не избавит бизнес от текстового описания товара.

Другой подход заключается в использовании методов нечеткого сопоставления, которые допускают некоторую степень вариации или ошибки в описании объекта. Например, если описание объекта содержит слово с ошибкой или синоним фактического названия продукта, нечеткое сопоставление все же может идентифицировать правильный штрих-код, принимая во внимание сходство между фактическим названием продукта и словом с ошибкой или синонимом.

Нечеткое сопоставление основано на концепции нечеткой логики, которая допускает наличие степеней истинности вместо строгих значений *true/false*. Это метод сопоставления строк, который допускает некоторую степень сходства или отклонения между двумя строками. Существует несколько технологий для реализации нечеткого сопоставления, включая алгоритм расстояния Левенштейна или алгоритм расстояния Яро-Винклера [6].



https://www.pvsm.ru/programmirovanie/115402

Одним из примеров использования нечеткого сопоставления для сопоставления штрих-кодов с продуктами является использование базы данных продуктов, которая содержит все названия продуктов (словарь) и штрих-коды, а также их варианты и орфографические ошибки. Алгоритм сравнивает отсканированный штрих-код со штрих-кодом в базе данных и возвращает совпадение, если есть определенная степень сходства. Если точного совпадения нет, алгоритм может выполнить поиск похожих названий продуктов или синонимов в базе данных и вернуть наиболее близкое совпадение на основе степени сходства.

Этот метод сопоставления штрих-кодов не основан на машинном обучении и может быть реализован с использованием традиционных методов программирования.

Оба подхода относительно просты и не требуют алгоритмов машинного обучения или специализированного оборудования. Однако они могут быть не такими точными, как подходы, основанные на машинном обучении, и могут иметь ограничения при обработке сложных описаний продуктов или нестандартных форматов штрих-кодов.

**IV. Методология работы**

**Выбор и описание архитектуры модели машинного обучения**

**Подготовка данных для обучения**

Была собрана база данных из более чем 600 000 строк, которая пополнялась более 10 лет путем ручного сопоставления продуктов с кодами SAP. Эта база данных послужила основой для модели машинного обучения. Далее набор данных был объединен со справочником по всем продуктам компании. После очистки и удаления дубликатов был получен набор данных из двух столбцов и 912 000 строк. Первый столбец содержит названия SAP, а второй столбец содержит их текстовые описания.

Процесс подготовки набора данных включал в себя несколько ключевых этапов. Первым шагом был сбор исходных данных путем ручного сопоставления продуктов с кодами SAP. Этот процесс требовал пристального внимания к особенностям продуктов и досконального понимания каталога продукции компании.

Далее, набор данных был объединен со справочником по всем продуктам компании. Этот шаг включал в себя объединение данных из нескольких источников и удаление дубликатов одного товара для разных стран с соответствующими кодами. В данном случае коды стран не несут смысловую нагрузку и могут быть отброшены.

Набор данных был разделен на два столбца, с кодами SAP в первом столбце и их соответствующими текстовыми описаниями во втором столбце. Этот формат обычно используется в задачах обработки естественного языка и классификации текста. Пример набора данных приведен в таблице 3.1.

Итоговые параметры датасета

Таблица 3.1 – Пример набора данных

|  |  |
| --- | --- |
| **ITEM** | **DESC** |
| CR1216BEA | CR1216 Элемент питания SONY BL5 |
| CR1616B1A | CR1616 Элемент питания SONY BL5 |
| CR1620BEA | CR1620 Элемент питания SONY BL5 |
| CR2016B1A | CR2016 Элемент питания SONY BL5 |
| CR2025BEA | CR2025 Элемент питания SONY BL5 |
| CR2032BEA | CR2032 Элемент питания SONY BL5 |
| DVPSR370B.RU3 | DVD-плеер SONY DVP-SR370 |
| DVPSR760HPB.RU3 | DVD-плеер SONY DVP-SR760HPB |
| XSFB1320E.RU2 | Автоакустика SONY XS-FB1320 |

Подготовка набора данных это сложный и отнимающий много времени процесс, помимо того, что требуется объединение, очистка и форматирование данных, процесс осложняется территориальной особенностью данных и наличием нескольких языков в текстовых описаниях товаров. Однако это важнейший шаг в разработке точной и надежной модели машинного обучения. Полученный набор данных служит основой для модели машинного обучения, позволяя ей делать прогнозы на основе взаимосвязей в данных.

**Описание параметров модели**

**- подбор кандидатов поиска (сколько на 1 продукт?)**

**1. Автоматы**

* 1. **Полное совпадение**
  2. **Частичное совпадение**

**2. TF IDF векторы**

**3. BERT эмбединги**

**Сколько фич?**

* **Предиктор**

**Catboost**

* **Inference?**

**Обучение и оценка модели**

**V. Результаты работы**

* Описание полученной модели машинного обучения
* Анализ полученных результатов (метрики, пример распознавания)
* Возможности дальнейшего улучшения модели

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

1. wiki
2. <https://huggingface.co/learn/nlp-course/chapter1/2?fw=pt>
3. Wiki
4. <https://sysblok.ru/nlp/8-glavnyh-proryvov-v-nejrosetevom-nlp/>
5. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Springer.
6. Zhang, Y., & Wallace, B. (2017). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1510.03820.
7. <https://vc.ru/newtechaudit/598664-nechetkoe-sravnenie-strok-kak-metod-obnaruzheniya-i-ispravleniya-oshibok>

ПРИЛОЖЕНИЕ 1