МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РЕАЛИЗАЦИЯ ПОДХОДА РАСПОЗНАВАНИЯ ИМЕНОВАННЫХ СУЩНОСТЕЙ НА ЗАДАЧЕ СТАНДАРТИЗАЦИИ АДРЕСОВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Столыпиной Анастасиии Михайловны

Научный руководитель
доцент, к. ф.-м. н. _____ О. А. Мыльцина

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы. В современном мире чаще и чаще искутсенный интеллект заменяет человечкский труд. Машинное обучение - класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. В данной работе будет рассмотрен один из методов машинного обучения - метод распозонвания именованных сущностей. Имена людей, названия организаций, книг, городов, и другие имена собственные называют «именованные сущности» (named entities), а саму задачу — «распознавание именованных сущностей». По-английски «Named entity recognition» или коротко NER; это сокращение регулярно используется и в русскоязычных текстах.

За одной задачей NER, на самом деле, стоит две: 1) обнаружить, что какая-то последовательность слов — это именованная сущность; 2) понять, к какому классу (имя человека, название организации, город и т.п.) эта именованная сущность относится. На каждом из этапов возникают свои сложности.

Данная работа рассмотривает метод в контексте задачи стандартизации адресов.

Целью бакалаврской работы является рассмотрение метода распознования именованнх сущностей в контексте задачи стандратизации адресов.

Объект исследования: система стандартизации адресов, использующая модель машинного обучения, базирующаяся на подходе распознавания именованных сущностей.

Предмет исследования: адреса, получаемые в систему стандартизации.

Для достижения поставленной цели в работе необходимо решить следущие задачи:

- ознакомиться с темой машинного обучения, изчусить понятие именнованных сущностей;

- изучить алгоритм построения модели машинного обучени;

- построить подель машинного обучения.

Структура и содержание бакалврской работы: работа состоит из введения, 3 разделов, заключения, списка использованных источников,

содержащего 20 наименований, и двух приложений. Общий обьем раобты составляет 40 страниц и без приложений.

1 ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В введении обосновывается актуальность работы, формулируется цель работы и решаемые задачи, отмечается практическая значимость полученных результатов.

В первом разделе приводятся основыне понятия машинного обучения и подхода именованных сущностей.

Машинное обучение (англ. machine learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Различают два типа обучения:

Обучение по прецедентам, или индуктивное обучение, основано на выявлении эмпирических закономерностей в данных. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний.

Имена людей, названия организаций, книг, городов, и другие имена собственные называют «именованные сущности» (named entities), а саму задачу — «распознавание именованных сущностей». По-английски «Named entity recognition» или коротко NER;

За одной задачей NER, на самом деле, стоит две:

обнаружить, что какая-то последовательность слов — это именованная сущность; понять, к какому классу (имя человека, название организации, город и т.п.) эта именованная сущность относится.

Распознавание именованных сущностей (NER) (также известное как идентификация (именованных) сущностей , разделение сущностей и извлечение сущностей) — это подзадача извлечения информации , которая направлена на поиск и классификацию именованных сущностей, упомянутых в неструктурированном тексте , в заранее определенные категории, такие как человек имена, организации, местоположения, медицинские коды , выражения времени, количества, денежные значения, проценты и т. д.

Задача NER – выделить спаны сущностей в тексте (спан – непрерывный фрагмент текста).

Во втором разделе разбираетмя предметная область и соотвествущиц ей BPMN-процесс.

BPMN (англ. Business Process Model and Notation, нотация и модель бизнес-процессов) — система условных обозначений (нотация) и их описания в XML для моделирования бизнес-процессов. Разработана Business Process Management Initiative (BPMI.org) и поддерживается Object Management Group, после слияния обеих организаций в 2005 году.

В третьем разделе изложения практическая часть работы, где будет описан код полученной модели.

Из исходного датасета необходимо нагенерить больше адресов. Для этого можем менять порядок следования полей, а также перемешивать одинаковые поля для разных строк между собой. Также можно самим изменять некоторые слова: ул -> ул./улица и т.д.

Т.е. из начального датасета вида: region: Ленинский, Октябрьский; Town: Маркс, Саратов. Можно получить 4 строки: г Саратов р-н Ленинский; г Саратов р-н Октябрьский; г Маркс р-н Ленинский; г Маркс р-н Октябрьский. Обозначения примерные.

Из исходного датасета необходимо нагенерить больше адресов. Для этого можем менять порядок следования полей, а также перемешивать одинаковые поля для разных строк между собой. Также можно самим изменять некоторые слова: ул -> ул./улица и т.д.

Т.е. из начального датасета вида: region: Ленинский, Октябрьский; Town: Маркс, Саратов. Можно получить 4 строки: г Саратов р-н Ленинский; г Саратов р-н Октябрьский; г Маркс р-н Ленинский; г Маркс р-н Октябрьский. Обозначения примерные.

Выполнить лемминг, стемминг, стоп-слова.

Стемминг – это грубый эвристический процесс, который отрезает «лишнее» от корня слов, часто это приводит к потере словообразовательных суффиксов.

Лемматизация – это более тонкий процесс, который использует словарь и морфологический анализ, чтобы в итоге привести слово к его канонической

форме – лемме.

Таким образом, далее, мы используем только лемматизацию для преобразования исходного текста к её нормальной (словарной) форме.

Переменная patterns нужна чтобы избавиться от всех некириллических символов и используется в функции re.sub. Переменная morph – это морфологический анализатор, который используется для нахождения нормальной формы слова.

Функция lemmatize:

Избавиться от букв латинского алфавита, чисел, знаков препинания и всех символов, например, символ @; Разобим пост на токены; Проведести лемматизацияю, получив нормальную (начальную) форму слова; Удалить стоп-слова. Возвращает список слов (токенов), причем только тех адресов, который содержат более 1 слов.

Для поиска ближайших предложений испольуется векторизация по триграмам (используем готовый алгоритм из библиотеки fuzzy-match).

Если в предложении нет запятых, то пытаемся искать сразу весь адрес (к вектору входной строки ищем ближайший из колонки Address в датасете).

Если в колонке есть запятые, то разделяем строку по ним (Саратовский район, Саратв, улица Чемодурова -> [Саратовский район, Саратв, улица Чемодурова]). Далее будем называть каждый элемент (например, Саратовский район) в полученном массиве - токеном.

Заранее заводим список из тех типов, которые мы ещё не посетили: "region "district "town "street". Стоит заметить, что здесь нет TownType и StreetType, поскольку их мы определяем после того, как найдём город / улицу.

Теперь для вектора каждого токена ищем ближайший вектор среди каждого непосещённого типа из нашей базы данных. Фиксируем тип и имя (значение до векторизации) ближайшего вектора.

Например: токен = "Саратв". Непосещенные типы: ["district "town "street"], тогда сначала смотрим все значения районов в нашей базе данных и ищем ближайший вектор к вектору для "Саратв". Допустим, что ближайшим к вектору "Саратв"будет вектор, соответствующий "Советский а значение похожести - 0.1. Аналогично сделаем для "town": ближайшим окажется "Са-

ратов а значение похожести - 0.8. Для "street": "Чемодурова значение - 0.2. Выбираем тип и имя с набольшим значением похожести, для нас это будет "Саратов а его тип - "town".

Имя записываем в массив результата для выбранного типа, а сам тип вычеркиваем из непосещенных. И продолжаем эту операцию либо пока не останется непосещённых типов, либо все токены в массиве закончатся.

Теперь мы нашли что-то из: регион, район, город, улицу. Зная улицу, мы можем найти город: смотрим в нашем базе данных города, которые идут в парке с заданной улицей и выбираем наиболее частотный. Аналогично мы делаем с районом (угадываем его по городу), а также с приставками TownType и StreetType - выбираем наиболее частотные для уже известной улицы / городу.

Если в названии улицы были числа, то также добавляем их в ответ (делаем так, поскольку в нашей базе данных почти нет улиц с номерами).

Также, отдельно из базы данных выделяем именованные улицы (они содержат "им "). И для нашей найденной улицы находим ближайшую из списка именованных, а затем смотрим значение похожести и если он выше некоторого порога (мы используем 0.3), то заменяем обычное название улицы именованным вариантом (Чемодурова -> им Чемодурова В.И.). Это необходимо делать, поскольку в базе данных есть и "Чемодурова"и "им Чемодурова В.И.".

2 ОСНОВНОЕ РЕЗУЛЬТАТЫ

1. Раскрыты основные понятия машинного обучения, метода именованных сущностей.

2. Разработана модель машинного обучения.

3. Разработан графический интерфейс.