МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дифференциальных уравнений и математической экономики

Методы построения векторных моделей для кластеризации текстовых данных

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 441 группы

Направления 09.03.03 Прикладная информатика

механико-математического факультета

Булатова Никиты Владимировича

Научный руководитель
профессор, д.э.н., профессор _____ В.А. Балаш
дата, подпись

Заведующий кафедрой
зав. каф., д.ф.-м.н., профессор _____ С.И. Дудов
дата, подпись

Саратов 2024 год

Введение. В эпоху быстрого роста информации текстовые данные стали одним из наиболее распространенных видов информации. Это привод к необходимости автоматического взаимодействия с большим количеством текста. В частности, разбиения текста на группы по смыслу – кластеризации. Чтобы это качественно реализовать необходима предобработка текста, а также преобразование текста в числовое представление – скаляр или вектор. Такое преобразование текста – векторизация.

Целью данной бакалаврской работы является изучение методов векторизации и кластеризации текстовых данных, рассмотрение различных методов векторизации и кластеризации с математической точки зрения, а также объединение в группы новостных статей по теме.

Объект исследования – векторные модели и методы кластеризации, использующиеся для анализов текстов. Предмет исследования – коллекция русскоязычных новостей.

Бакалаврская работа состоит из введения, четырех основных разделов, заключения, списка использованных источников и приложения. В первом разделе рассматриваются основные методы предобработки и векторизации текстов на естественном языке, а также их влияние на точность кластеризации. Во втором разделе рассматриваются методы кластеризации векторных представлений текста для выявления тем, методы их работы, математические формулы, достоинства и недостатки представленных моделей. В третьем разделе описываются анализ существующих программных решений, их возможности, достоинства и недостатки. Четвертый раздел посвящен архитектурам и реализациям программных решений для тематического моделирования на основе векторных моделей. В нем описываются архитектурные подходы приложений, использующих кластеризацию текста, их сильные и слабые стороны.

Основное содержание работы. В разделе 1 описываются популярные методы предобработки и векторизации текстовых данных.

Предобработка текстовых данных состоит из нескольких этапов:

- Токенизация
- Удаление стоп слов
- Нормализация

Рассмотрим каждый из этих этапов:

Токенизация – это процесс разбиения текста на компоненты. Токенами могут выступать отдельные слова, пары слов (биграммы) или несколько идущих подряд символов. Существуют различные методы разделения текста на токены каждый из которых имеет свои преимущества и недостатки. Вот некоторые из них:

- Разделение по пробелам
- Разделение с помощью регулярных выражений
- Разделение по символам (удаляются все символы кроме нужных)
- Готовые токенизаторы такие как NLTK, SpaCY

Удаление стоп-слов. Стоп-слова – это такие слова, которые не несут смысловой нагрузки и не влияют на тематику текста. К стоп словам обычно относят союзы, предлоги, артикли, местоимения и другие служебные слова. Удаление стоп слов помогает сильно снизить размерность текста благодаря чему повышается производительность. Так же это помогает повысить качество кластеризации текста. Существует несколько методов составление словаря стоп – слов:

- Взять готовый словарь стоп слов
- Составить словарь стоп-слов с помощью статистических методов

Нормализация текста – это процесс преобразования текста к нормальной форме. Существует два основных метода нормализации слов – лемматизация и стемминг.

Лемматизация позволяет поставить слово в нормальную форму. Достоинство этого метода в том что слова которые меняются уникальным образом (например при помощи чередования) преобразуются к своей нормальной форме. Недостатком лемматизации является низкая производительность

Стемминг – метод нормализации который приводи слово к нормальной форме удаляя суффикс и окончание. Плюсом стемминга является его быстродействие. Недостатком стемминга является то, что он плохо приводит к нормальной форме супплетивизмы.

Так же к предобработке текста можно отнести другие преобразования текста такие как: приведение к нижнему регистру, удаление пунктуационных знаков, удаление диакритических знаков.

Следующем этапом идет векторизация текста. Векторизация текста – это процесс преобразования текста в числовой формат. Существует множество способов представить текст в виде векторов, но мы рассмотрим только следующие способы:

- TF – IDF
- Word2Vec
- FastText

TF – IDF – это метод векторизации текста, который учитывает частоту встречаемости слова в документе и важность слова для всего набора документов.

Для вычисления TF-IDF вычисляются две метрики TF и IDF. TF – рассчитывается для каждого документа в корпусе и считает важность слова для документа. TF = (количество вхождений слова в документ) / (число слов в документе). IDF – рассчитывает важность слова для всего корпуса документов. IDF – (количество документов в корпусе) / (количество документов содержащих это слово). TF-IDF = TF * log(IDF).

Совершенно другой подход используется в методах Word2Vec и FastText. Word2Vec – это метод преобразования теста в числовой вектор основанный на нейросетях. Основная идея данного подхода заключается в том, что рассчитываются не все слова в корпусе, а только те слова что находятся рядом (в каноническом случае 5 – граммы). Существует два подхода к обучению Word2Vec:

- При подходе со skip – граммами контекст слов (входные слова) предсказываются на основе интересующего нас (входного) слова.
- При подходе с непрерывным мультимножеством слов (CBOW) целевое (выходное) слово предсказывается по близлежащим (входным) словам.

Преимуществом этого метода векторизации является то, что полученные этим методом векторы будут находиться рядом если значения слов будут похожи.

Похожий на Word2Vec подход применяется в FastText. В отличии от Word2Vec FastText обучается не на 5 граммах, а на последовательности из нескольких символов. Благодаря такому подходу fast text может предугадывать формы слова, которые он раньше не видел.

В разделе 2 описываются методы кластеризации для выделения тем в корпусе текстов.

Кластеризация – это процесс объединения объектов в группы (кластеры) по общим характеристикам, значению или темам. Существует большое количество методов кластеризации, но мы рассмотрим следующие:

- Kmeans
- Kmeans++
- DBSCAN
- Спектральная кластеризация

Метод кластеризации kmeans или метод k-средних, является одним из самых популярных и широко используемых алгоритмов кластеризации. Особенностью алгоритма k-means является то, что мы должны заранее определить количество кластеров. Данный алгоритм берет k случайных точек из нашего набора данных – эти точки становятся центройдами кластеров. Все остальные точки попадают в тот кластер центройд которого находится к ним ближе всего. К преимуществам k-means относятся простота реализации, эффективность для больших наборов данных. К недостаткам k-means можно отнести необходимость предварительного определения числа кластеров и зависимость качества кластеризации от выбора начальных центройдов.

Метод Kmeans++ является улучшенной версией k-means, которая использует улучшенный подход к выбору первых центройдов. Это помогает строить более качественные кластеры.

DBSCAN – это алгоритм кластеризации который основывается на плотности. У данного алгоритма имеется два начальных параметра: eps – максимальное расстояние между двумя точками для того чтобы они считались соседями и minPts – минимальное число соседей необходимое для того чтобы точка считалась точкой ядра. К преимуществам DBSCAN можно отнести то что данный алгоритм не требует заранее задавать количество кластеров, устойчив к выбросам и способен находить кластеры не правильной формы. Недостатками DBSCAN являются высокая чувствительность к параметрам eps и minPts, и данный алгоритм плохо подходить для данных с высокой размерностью.

Спектральная кластеризация – это алгоритм кластеризации который использует спектр матрицы подобия данных для группировки точек данных.

Третья часть содержит анализ существующих программных решений векторизации текстов. Существует множество готовых программных решений для векторизации текста, каждое из которых имеет свои преимущества и недостатки. В этом анализе я рассмотрю некоторые из наиболее популярных решений для векторизации текста.

GloVe – это метод векторизации текста, который был разработан Стэндфордским университетом. Он похож на Word2Vec, но использует статистическую модель для обучения представлениям слов. GloVe может быть более точным, чем Word2Vec, для некоторых задач, особенно для задач которые требуют понимания семантических отношений между словами. Достоинствами GloVe является лучшая чем у Word2Vec точность. Недостатком является меньшая по сравнению Word2Vec скорость.

Word2Vec – это метод векторизации текста, который был разработан Томасом Миколовым и его коллегами из Google. Он использует нейронную сеть для обучения представлениям слов на основе их окружения в тексте.

К преимуществам Word2Vec относятся простота, скорость работы и эффективность. Недостатками может являться неточность для некоторых задач.

FastText – это метод векторизации текста, который был разработан Facebook. Он похож на Word2Vec, но может обрабатывать подстроки и морфемы. Это делает FastText более подходящим для задач, которые требуют обработки ненормативного или неизвестного текста. Преимуществом FastText является предсказание незнакомых для модели форм слова. Недостатком модели является низкая по сравнению с Word2Vec или GloVe точность.

BERT – это метод векторизации текста, который был разработан Google AI. Он использует двунаправленную трансформерную нейронную сеть для обучения представлениям слов, которые учитывают контекст слова во всем предложении. BERT может быть более точным, чем остальные модели, для некоторых задач, особенно для задач, которые требуют понимания сложных отношений между предложениями. Но при этом модель может быть вычислительно сложной.

Так же нельзя обойти стороной метод векторизации TF IDF, который реализован в библиотеке sklearn. Хоть на наших данных данный метод векторизации не дает хороших результатов при кластеризации, данный метод показывает хороший результат при поиске одинаковых новостей из различных источников. Но он не совсем удобен так как появляется необходимость хранить IDF вектор, а также периодически его пересчитывать при добавлении новых данных. Так же имеется необходимость где-то хранить и TF вектора для быстрого расчета TF-IDF и поиска косинусного расстояния между векторами документов.

Четвертая часть содержит в себе описание, диаграммы, сравнения, а также плюсы и минусы различных архитектурных подходов, связанных с обработкой текста на примере новостного агрегатора. Новостной агрегатор состоит из различных сервисов – модулей. В данной работе рассматриваются только сервисы, отвечающие за кластеризацию и обработку текста. Сервисы, отвечающие за кластеризацию, состоит из нескольких модулей:

- Модуль предобработки текста: этот модуль принимает текстовые данные, разбивает текст на токены, удаляет стоп слова, нормализует текст
- Модуль векторизации: этот модуль принимает нормализированный текст и создает векторное представление текста
- Модуль кластеризации: данный модуль разбивает все новости на кластеры, добавляет новые новости в уже существующие кластеры, а также находит самые важные слова в кластере
- Модуль оценки: этот модуль рассчитывает различные метрики оценки для определения того насколько хорошо получились кластеры

Модуль предобработки текста принимает текстовые данные, разбивает текст на токены с помощью токенизатора из библиотеки NLTK. Данный токенизатор не разделяет слова и символы, которые имеют смысл только находясь рядом, например, такие как время, записанное через двоеточие или физические величины. После токенизации модуль удаляет стоп слова. Он это делает благодаря русскому словарю стоп слов NLTK. Это помогает сильно уменьшить количество токенов и как следствие увеличить производительность на следующих этапах. После удаления стоп слов наступает черед нормализации. В некоторых ситуациях нормализация ухудшает результат, так как теряется форма слов, но в нашем случае результат получается лучше. Мы нормализуем токены с помощью библиотеки pymorpy2. Эта библиотека помогает нам с лемматизацией токенов. В

результате многие одинаковые слова, которые стояли в разной форме будут учитываться одним и тем же словом, что позволяет нам уменьшить размер словаря.

Следом идет модуль векторизации. Этот модуль принимает на вход список токенов. Лучше всего на нашем наборе данных показала себя модель Word2Vec, поэтому мы используем предобученную модель word2vec-rucorpora-300 из библиотеки genism. По итогу работы для каждого документа мы получаем набор векторов, которые соответствуют нашим документам. Но для кластеризации нам нужен какой-то один общий для всего документа вектор, поэтому мы просто берем медианы координат и из них составляем вектор для документа.

После того как мы получили вектор для всех документов идет очередь разбиения на кластеры. Лучше всего на нашем наборе данных показала себя спектральная кластеризация. Но перед тем как кластеризировать необходимо понизить размерность наших векторов. Это повысит производительность следующих операций и повысит точность, так как мы оставим координаты, которые имеют наибольшее значение. Лучше всего для этих целей подошел метод LSA, который взяли из библиотеки sklearn. По итогу мы получили 10 мерные векторы, которые и будем кластеризировать. Наши данные успешно разбились на 10 кластеров, которые мы опишем с помощью TF-IDF векторов.

Когда мы разбили наши новости на кластеры, мы хотим узнать насколько правильные кластеры получились. Наши данные не размечены, поэтому мы не знаем в тот ли кластер попала новость или нет. Но мы можем посмотреть насколько далеко друг от друга кластеры и насколько похожи точки внутри кластера друг на друга. Для этих целей нам доходит метрика силуэт.

Заключение. В данной работе были исследованы методы векторизации и кластеризации текстовых данных. Были рассмотрены различные подходы к векторизации, такие как TF-IDF и Word2Vec, а также различные алгоритмы кластеризации, такие как K-means, DBSCAN и спектральная кластеризация.

Было показано, что выбор метода векторизации и алгоритма кластеризации оказывает значительное влияние на результаты кластеризации. Наиболее точные результаты были получены с использованием метода Word2Vec и алгоритма спектральной кластеризации.