Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

ГРУППОВАЯ КУРСОВАЯ РАБОТА

Исследовательский проект на тему "Классификация запросов клиентов для автоматизации обслуживания"

Выполнили студенты:
Охрименко Дитрий Андреевич, группа 172, 3 курс,
Федоров Павел Сергеевич, группа 172, 3 курс,

Руководитель КР:
Руководитель Цифровые сервисы Системы взаимодействия с клиентами
Бондарев Иван Николаевич

Консультант:
Заместитель руководителя департамента, Старший преподаватель
Соколов Евгений Андреевич

Москва 2020

Аннотация

В рамках курсовой работы изучались различные методы получения векторного представления текстовых документов и кластеризации в контексте истории запросов пользователей к чат-боту ПАО "МегаФон". Главная задача работы состояла в том, чтобы найти новые темы для последующей обработки ботом, и тем самым повысить долю запросов, которые обработал бот в общем объеме. Так был произведен сравнительный анализ разных методов предобработки текста для получения признаков и анализ методов их дальнейшей кластеризации. В итоге была написана библиотека с применением различных методов для определения тематики обращений клиентов.

Ключевые слова: Обработка естественного языка, кластеризация текста, "МегаФон", машинное обучение, тематическое моделирование, нейронные сети

Содержание

1 Введение

Данный раздел был написан совместно

1.1 Описание предметной области

В ПАО "МегаФон" существует контактный центр, специалисты которого нужны для обработки запросов клиентов и помощи в решении их проблем. Сейчас на часть вопросов умеет отвечать бот Елена. Но чтобы понять, сможет ли она решить проблему пользователя, нужно уметь классифицировать тему обращения - то есть общую характеристику, которая характеризует смысл вопроса абонента. Каждое обращение относится к одной определенной теме. Кластеризация запросов пользователей предназначена для снижения нагрузки на специалистов обслуживания контактного центра ПАО "МегаФон" и повышения доступности информационных сервисов для абонентов за счёт автоматизации решения запросов потенциальных клиентов и действующих абонентов.

В работе рассматривается история сообщений из чат-бота ПАО "МегаФон" Елена начиная с 2017 года. Для каждого сообщения присутствует много различной информации, в том числе дата и время отправки, направление сообщения (от клиента боту или наоборот), соответственно данные отправителя/получателя и т. д.

1.2 Актуальность и значимость работы

Работы в этой области остаются актуальными, так как любое улучшение точности предсказаний выгодно компании. Актуальность задачи для ПАО "МегаФон" обуславливается количеством денег, которые компания сможет сэкономить путем уменьшения количества специалистов контактного центра, так как бот сможет обрабатывать больше запросов, научившись определять большее количество тем.

1.3 Задачи работы

В рамках данной курсовой работы была поставлена задача изучить и применить методы предобработки текста и алгоритмы кластеризации для того чтобы в дальнейшем построить классификатор новых обращений клиентов. А также реализовать библиотеку для кластеризации, обеспечивающий приемлемый уровень производительности и точности. Планируется работа с неразмеченными данными и задача состоит в том, чтобы научиться кластеризовать их. Следует отличать классификацию текстов от кластеризации, так как в последнем случае тексты тоже группируются по некоторым критериям, но заранее заданные категории (в нашем случае тематики обращений) отсутствуют. Это нужно для того, чтобы понять, какие темы существуют и каких тем сейчас не хватает в размеченных по темам данным. Целью работы является автоматизация обработки обращений действующих абонентов по авторизованным каналам обращения (SMS и чат в личном кабинете)

2 Обзор литературы

Данный раздел был написан совместно

2.1 Постановка задачи

Пусть X — множество объектов, Y — множество номеров (имён, меток) кластеров. Задана функция расстояния между объектами $\rho(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \ldots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i.

Алгоритм кластеризации — это функция $a : X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y

в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Кластеризация (обучение без учителя) отличается от классификации (обучения с учителем) тем, что метки исходных объектов y_i изначально не заданы, и даже может быть неизвестно само множество Y. [1]

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин:

- Не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд эвристических критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты.
- Число кластеров неизвестно заранее и устанавливается в соответствии с некоторым критерием.
- Результат кластеризации зависит от метрики, выбор которой субъективен и определяется экспертом.

Далее задача кластеризации текстов состоит из этапов

1 Предобработка текста:

   1.1. Удаление редких/частотных слов

   1.2. Стэмминг или лемматизация

   1.3. Удаление стопслов

2 Извлечение признаков из текста

   2.1. TF-IDF

   2.2. Вероятностные тематические модели (Латентное размещение Дирихле, далее LDA)

2.3. Text2vec модели (Doc2vec, Word2vec и fastText)

2.4. BERT embeddings

3 Обучение алгоритма кластеризации

3.1. K-means

3.2. DBSCAN

3.3. Affinity propagation

3.4. Агломеративные методы

3.5. BIRCH

3.6. Нейронная сеть

4 Создание словаря

5 Подбор оптимального количества тем

Разберем существующие подходы кластеризации текстовых документов:

2.1.1 Ручной перебор

Первый, и самый очевидный, это кластеризации без помощи компьютера. Идея данного метода в том, чтобы вручную присваивать текстовым файлам тематические рубрики. Например, так поступают библиотекари в библиотеках. Очевидно, это самый простой способ, но в то же время самый долгий и времязатратный. Он не подходит для случаев, когда нам дороги время, или же мы имеем очень большое количество документов.

2.1.2 Написание правил

Данный способ основывается на придумывании правил, по которым тексты можно будет кластеризовать. Очевидный пример - использовать ключевые слова. Если, например, текст содержит слово "уравнение" или "производная", то логично отнести его к категории текстов о математике. При

хорошем уровне понимания предметной области и использовании написания регулярных выражение можно составить правила, которые затем будут автоматически применяться к новым поступающим документам для кластеризации. Очевидно, процесс автоматизируется, в связи с чем данный способ гораздо лучше предыдущего, ведь число документов для обработки может быть практически любым. И при хорошем построении правил результат может быть даже лучше, чем при применении алгоритмов машинного обучения. Однако данный метод требует постоянного обновления правил для поддержания их актуальности, что требует постоянных затрат. Поэтому этот метод не является универсальным.

2.1.3 Обучение с учителем

Третий существующий метод использует алгоритмы машинного обучения. Идея в том, что набор правил пишется не человеком, а вычисляется автоматически на основе обучающих данных. В качестве обучающих данных используется набор текстовых документов, являющихся хорошими примерами для каждого класса. Для этого потребуется вручную разметить небольшой объем данных. Однако, это проще, чем вручную писать правила, поэтому данный метод более предпочтителен. Кроме того, разметку можно проводить в процессе использования системы. Примером может служить обычная электронная почта, в которой у пользователя есть возможность помечать некоторые письма как спам. Таким образом формируется обучающее множества для спам-фильтра. В итоге классификация текстов, основанная на машинном обучении, является типичным примером обучения с учителем. Но в условиях нашей задачи нужны люди, которые предварительно разметят обучающее множество и создадут набор классов.

2.1.4 Обучение без учителя (кластеризация)

В данном подходе не используется размеченная выборка, а используются алгоритмы кластеризации, которые учатся разделять векторные представ-

ления объекты на отдельные кластеры. Данный способ и будет изучаться в работе, так как важно найти новые тематики обращений.

На текущий момент в ПАО "МегаФон" используется второй способ с ручным написанием правил.

2.2 Метрики качества кластеризации тем

- **Silhouette coefficient**.

  Рассчитывается с использованием среднего расстояния внутри кластера (a) и среднего расстояния до ближайшего кластера (b) для каждого объекта. Коэффициент для него равен $\frac{b-a}{max(a,b)}$, где b - это расстояние между объектом и ближайшим кластером, частью которого он не является, что значит (среднее расстояние между каждым объектом ближайшего соседнего кластера - среднее расстояние к другому объекту того же кластера) / max(среднее расстояние к объекту ближайшего соседнего кластера, среднее расстояние к другому объекту того же кластера). Значение метрики показывает, насколько объект похож на свой кластер по сравнению с другими кластерам . [2]

- **Topic coherence**.

  Рассчитывается путем измерения семантического сходства между высоко оцененными словами в одной теме. Связность (coherence) означает, что набор фактов поддерживает друг друга, например, набор фактов «игра - это командный вид спорта», «игра проводится с мячом», «игра требует больших физических усилий» - связный. Каждая сгенерированная тема состоит из слов, и согласованность темы применяется к наиболее популярным N словам из темы. Coherence определяется как среднее значение / медиана метрики попарного сходства слов и попу-

лярных слов в теме. [3].

$$C = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}$$

3 Основная часть

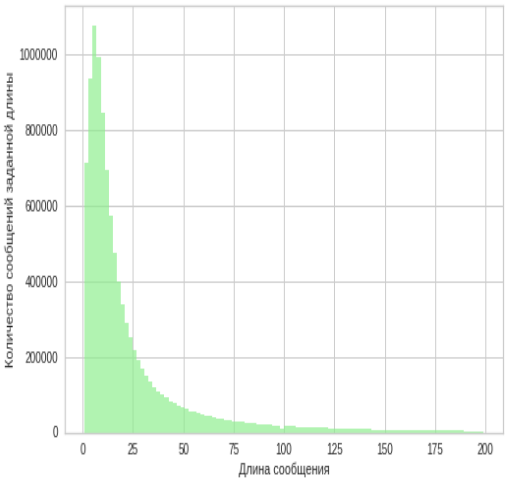
3.1 Выгрузка данных (выполнил Федоров Павел)

Для начала нужно было получить данные для анализа. Для этого была изучена документация баз данных ПАО "МегаФон", чтобы понять, из каких таблиц получать данные. И затем был написан SQL-запрос, чтобы скачать историю сообщений пользователей, начиная с начала 2017 года. Для экспериментов были использованы обращения за 2019 год.

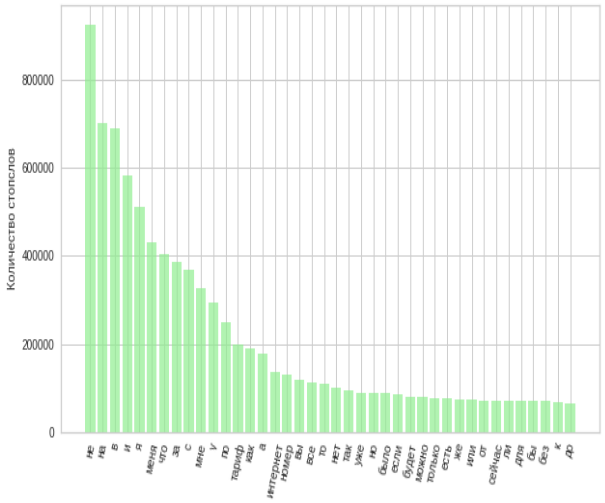
3.2 Предобработка данных (выполнил Охрименко Дмитрий)

Очень важно было разделить данные на важные и неважные признаки и очистить данные от вторых. Учитывая, что изначально для каждого сообщения был 61 признак различной информации, было очевидно, что часть из них была бесполезна. После анализа имеющихся данных было принято решение оставить 15 признаков с действительно полезной информацией. В их число вошли сообщения пользователей, а так же данные о времени сообщения, успешности или неуспешности решения вопроса ботом, и так далее. Некоторые данные пришлось удалить, как, например, сообщения от бота людям, а так же электронные почты, устройства отправления и прочие данные, не способствующие успешному обучению. После нескольких экспериментов с оставшемся признаками, были оставлены только сообщения пользователей, где каждый документ для кластеризации был сгруппирован по id обращений. Так размер выборки составил приблизительно 10 миллионов объектов.

Также предварительно нужно было предобработать текст (для этого использовались библиотеки gensim и nltk). Были удалены все стоп слова (то есть слова, не несущие особого смысла в одиночку, например, предлоги или союзы, в том числе часто встречающиеся слова), а также применить стэмминг (процесс нахождения основы слова для исходного слова, т. е. отсечение от слова окончаний и суффиксов). Иначе же слова различные формы одного слова (например, падежи) воспринимались алгоритмом как разные, в связи с чем получался довольно бессмысленный результат. Например, для того же слова "МегаФон" близкими словами считались "МегаФона", "МегаФону", и т. д.



(a) Распределение количества слов в сообщениях



(b) Распределение частот стопслов

Рис. 1: Графики распределения слов в сообщениях

3.3 Генерация выборки для кластеризации

3.3.1 TF-IDF (исследовал Федоров Павел)

Статистическая мера TF-IDF [4] - используется для оценки важности слова в контексте каждого сообщения из всех сообщений. Вес слова пропорционален частоте употребления этого слова в сообщении и обратно пропорционален частоте употребления слова во всех сообщениях.

$$TfIdf(t, d, D) = tf(t, d) \times idf(t, D)$$

TF — это частотность слова, которая измеряет, как часто термин встречается в документе.

IDF = $log(\frac{1+n}{1+df(d,t)}) + 1$, где n - это количество документов, $df(d, t)$ - количество документов, где встречается слово t. Это обратная частотность документов, она измеряет непосредственно важность слова.

Таким образом каждое обращение пользователя представляет собой разряженный вектор длины количества слов в словаре, где 0 соответствует отсутствию слова в документе, а числа - значения метрики TF-IDF для слов, которые есть в документе.

3.3.2 LDA (исследовал Федоров Павел)

Тематическая модель - это модель коллекции текстов, определяющая, к каким темам относится каждый документ. Алгоритм построения получает на входе коллекцию документов и на выходе для каждого документа выдаётся численный вектор, который состоит из оценок степени принадлежности каждого документа каждой из тем. Порядок слов в документе не важен, документ это "мешок слов" (bag of words).

LDA - это тематическая модель [5]. В работе авторы показали, что модель может захватывать семантическую информацию из набора документов и продемонстрировали превосходство модели по сравнению с существующими на тот момент моделями, включая модель полиномиальной смеси [**?**] и вероятностный латентно-семантический анализ [7]. Алгоритм LDA придерживается такой точки зрения, согласно которой тексты представлены как векторы счетчиков слов, и полагается на двухэтапную генерацию. Ключевое предположение состоит в том, что каждый документ представлен определенным распределением тем, и каждая тема имеет свое распределение слов. Модель LDA выявляет скрытые связи между словами посредством тем. Она

также позволяет присваивать вероятности новым документам, не входившим в обучающую выборку, используя алгоритм вариационного вывода

Раньше, до популярности нейронных сетей, для задач кластеризации текстов по темам (или тематическое моделирование) обычно использовались методы тематического моделирования. Поэтому далее рассмотрим более современные методы, с помощью которых можно получить векторное представление документа (или его эмбеддинг), а также те, которые используют контекст слов в сообщениях, т. е. порядок слов в документе будет важен.

3.3.3 Text2vec подходы (исследовал Охрименко Дмитрий)

3.3.3.1 Word2vec

В качестве модели, используемой для анализа предоставленных данных, было решено рассмотреть Word2vec [8]. Суть этого алгоритма в том, что каждому слову из множества всех слов, используемых в сообщениях клиентов и бота за исследуемый период, присваивается в соответствие вектор. Чем ближе значения векторов, тем ближе по смыслу слова. Хотелось бы отметить, как сильно на итоговый результат влиял общий объем обучающей выборки. Когда в целях экономии времени модель обучалась на маленькой выборке, результаты получались довольно бессмысленными, однако после обучения на полном наборе данных итог стал куда лучше. В качестве поверхностного анализа успешности алгоритма использовалась функция similarity(), которая по предложенному ей слову выдает n самых близких согласно алгоритму w2v. И действительно, при запросе слова "телефон" результатом работы были такие слова, как "смс", "сообщение", "связь", а при запросе слова "МегаФон" - "мтс", "билайн", "теле2". Для получения наилучшего результата остается лишь подобрать гиперпараметры модели. Так, например, увеличение значения гиперпараметра, отвечающего за размер вектора со 150 до 300, немного улучшает точность алгоритма. Существует две вариации word2vec - основанная на Continous bag of words (в данном случае слово предсказывается по

контексту) и Skip-gram (наоборот, контекст предсказывается по слову). Мы использовали первый вариант. Далее, чтобы получить вектор документа, мы складывали вектора слов, составляющих его.

word	cosine similarity
билайн	0.537915
сохранен_номер	0.483131
мтс	0.483092
компан	0.480208
сво	0.473821
сотов_связ	0.451853
знают	0.436874
относ	0.436347
другов	0.431213
обидн	0.429039

Рис. 2: Слова, схожие со словом "мегафон" в контексте модели Word2vec

3.3.3.2 fastText

Так же был рассмотрен алгоритм fastText [9]. Главный принципе его работы заключается в использовании морфологической структуры слова, которая несет важную информацию о его смысле. Это вдвойне актуально для языков, в которых одно слово может иметь множество различных морфологических форм (множественное число, падежи, роды), в число которых, безусловно, входит и русский язык. В этом заключается ключевое отличие fastText от традиционных эмбеддингов, например, word2vec. Идея заключается в рассматривании каждого слова как совокупность его подслов. Для простоты, "подслова"рассматриваются как н-граммы букв, соответственно вектор - сумма всех векторов составляющих его н-граммы букв. Таким образом, fastText значительно лучше справляется с синтаксическими задачами по сравнению с word2vec, однако слегка уступает ему на семантических задачах. Тем не менее, при увеличении обучающей выборки разница в результате становится все меньше. При всем при этом, fastText так же требует значительно меньше времени на обучение.

word	cosine similarity
«мегафон	0.717442
мегафонтв	0.674227
мегафонпр	0.672903
мегафоновск	0.597955
мегафонбанк	0.592686
мегафондолг	0.584288
мегаф	0.545466
«мегафон»	0.514315
магафон	0.432386
мигафон	0.419846

Рис. 3: Слова, схожие со словом "мегафон" в контексте модели fastText.

3.3.3.3 Doc2vec

Другой рассмотренной моделью для извлечения признаков был Doc2vec [10]. По сути, это усовершенствованный word2vec. Отличие заключается в том, что в данном алгоритме используются вектора абзаца и документа. Существует два метода реализации doc2vec - distributed memory и distributed bag of words. DM прогнозирует следующее слово по предыдущим слова и вектору абзаца, в то время как DBOW основывает предсказания случайных групп слов только на по вектору абзаца. В нашем случае мы использовали distributed bag of words. Аналогично word2vec, здесь мы тоже пользовались мешком слов (DBOW), и в этот раз результат был значительно лучше, чем при использовании word2vec.

3.3.4 BERT (исследовал Федоров Павел)

BERT [11] — двунаправленная языковая модель с transformer-архитектурой [12], которая заменила собой рекуррентные нейронные сети с более быстрым подходом на основе механизма attention. Transformer - это такая архитектура нейросети, которая учитывает весь контекст. Каждый ее слой делает много параллельных связей между определёнными словами, игнорируя остальные. Такие древовидные репрезентации предложений дают трансформерам воз-

можность моделировать смысл контекста и эффективно изучать связи между словами, которые могут стоять далеко друг от друга в длиных предложениях. Нейронная сеть такого рода может на выходе создать векторные представления для слов или целых фраз. BERT предобучен на двух задачах без учителя — моделирование языковых масок и предсказание следующего предложения. Это позволяет использовать предобученную модель, настраивая её под нужные задачи, например для получения эмбеддингов слов и предложений, что как раз применимо к исследуемой проблеме. Такой подход был применен в статье 2020 года «Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!» [13], где исследовались разные подходы к получению эмбеддингов текстов.

3.3.5 BERT + LDA (исследовал Федоров Павел)

Информация из "мешка слов" (LDA или TF-IDF) эффективна для определения тем путем нахождения часто встречающихся слов, когда тексты связаны между собой. С другой стороны, когда тексты непоследовательны (с точки зрения выбора слов или значения предложения), необходима дополнительная контекстная информация, чтобы наиболее полно представить идею текстов. Тогда почему бы не объединить и "мешок слов", и контекстную информацию. Комбинируя LDA, BERT [11] и кластеризацию, можно сохранить семантическую информацию и создать контекстную идентификацию темы. Так поступили авторы статьи «Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence» [14]. Они использовали нейронную тематическую модель ProdLDA [15], основанную на вариационном автоэнкодере [16], который обучается для отображения "мешка слов" всех документов в скрытое представление и затем сеть декодера восстанавливает "мешок слов", генерируя слова из скрытого представления документов. Затем используются эмбеддинги документов с помощью предобученной модели BERT, которые проецируются через скрытый слой автоэнкодера с той же размерностью, что и размер словаря и потом объединяются с представлениями

"мешка слов".

3.4 Алгоритмы кластеризации

3.4.1 K-means (исследовал Охрименко Дмитрий)

K-means - один из самых простых алгоритмов кластеризации [17]. Его суть заключается в стремлении минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров. Вначале выбираются точки - центры кластеров, после чего плоскость разбивается на кластеры при помощи диаграммы Воронова (то есть каждая точка плоскости принадлежит к кластеру, образованному ближайшим центром кластеров. После этого среди вновь образованных кластеров ищутся центры масс, которые теперь считаются новыми центрами кластеров. После чего алгоритм повторяет все действия, кроме первого, для тех пор, пока не сойдется. К сожалению, данный метод не лишне недостатков, среди которых можно отметить тот факт, что вовсе не гарантированно достижение глобального минимума суммарного квадратичного отклонения, а так же зависимость результата от выбора начальных центров кластеров. Последнюю проблему, однако, решают более современные модернизации алгоритма, которые мы и использовали в экспериментах.

3.4.2 Агломеративная иерархическая кластеризация (исследовал Охрименко Дмитрий)

Иерархическая кластеризация подразумевает построение иерархии кластеров. Различают два вида иерархической кластеризации - при агломеративной иерархической кластеризации каждый новый кластер - объединение более мелких кластеров, в то время как при дивизионном, наоборот, более крупные кластеры делятся на мелкие подкластеры. В нашем исследовании мы использовали агломеративный подход.

Есть несколько методов связи кластеров, как, например, метод одиночной связи, когда межкластерное расстояние полагается равным минималь-

ному расстоянию между элементами двух кластеров или метод полной связи, где, наоборот, межкластерное расстояние приравнивается максимальному расстоянию между элементами разных кластеров. В нашей работе мы использовали метод средней связи. В таком случае расстояние между кластерами равно среднему расстоянию между элементами взятых кластеров: $\frac{1}{|A|\cdot|B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$

3.4.3 BIRCH (исследовал Федоров Павел)

Алгоритм BIRCH используется для иерархической кластеризации больших данных с ограниченными техническими возможностями, который сначала создает небольшую и компактную подвыборку большого набора данных, которая сохраняет как можно больше информации. Затем кластеризуется эта меньшая подвыборка, вместо кластеризации большего набора данных. BIRCH часто используется для дополнения других алгоритмов кластеризации, создавая подвыборки набора данных, которые теперь может использовать другой алгоритм кластеризации.

Для каждого кластера вычисляются свойства $CF = (N, LS, SS)$, где N – количество документов в кластере, LS – их линейная сумма, SS – сумма их квадратов. Для всей коллекции данных и их разбиения на кластеры строится CF-дерево – сбалансированное дерево с параметрами: фактор ветвления B и пороговое значение T. Каждый промежуточный узел содержит не более B вхождений вида $[CF_i, child_i]$, где $child_i$ – указатель на его i-й дочерний узел, CF_i – подкластер, представленный этим дочерним узлом. Листовой элемент содержит не более L вхождений вида $[CF_i]$. Размер дерева является функцией от T: чем больше T, тем меньше дерево. Значения B и L определяются P. На первом шаге алгоритма сканируется вся коллекция документов и выстраивается первоначальное CF-дерево. Потом просматриваются все листья для построения нового дерева меньшего размера, за счет удаления шума (сильно удаленных документов) и слияния тесно расположенных подкластеров. Затем с помощью одного из статических алгоритмов кластеризуются листовые

элементы. Например, можно использовать иерархический агломеративный метод на подкластерах, представленных их CF-векторами, так как он позволяет выбрать количество кластеров, или желаемый порог для диаметров кластеров.

3.4.4 Автоэнкодер (исследовал Федоров Павел)

Автоэнкодер - это нейронная сеть, обучающейся без учителя, которая учится эффективно сжимать и кодировать данные, а затем учится восстанавливать данные обратно из представления пониженной размерности в представление, максимально приближенное к исходным данным. [19] Таким образом автоэнкодер уменьшает размерность данных, научившись игнорировать в них шум.

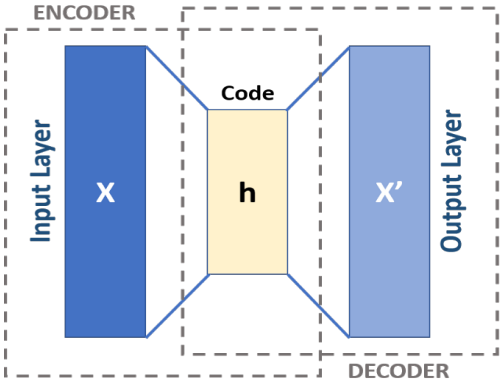


Рис. 4: Архитектура Автоэнкодера

Архитектура:

Архитектура для автоэнкодеров может варьироваться от простой сети прямого распространения, сети LSTM [20] или сверточной нейронной сети [21] в зависимости от варианта использования. В нашей работе использовалась архитектура сети прямого распространения.

Автоэнкодеры состоит из четырех основных частей:

- **Энкодер**: в котором модель учится уменьшать входной размер и сжимать данные в закодированное представление.

- **Bottleneck**: слой, содержащий сжатое представление входных данных. Это минимально возможные размеры входных данных.

- **Декодер**: в котором модель учится восстанавливать данные из закодированного представления, чтобы они были как можно ближе к исходному вектору.

- **Реконструкция потерь**: это метод, который измеряет, насколько хорошо работает декодер и насколько близок выходной сигнал к входящим данным.

Затем обучение происходит за счет обратного распространения ошибки, чтобы минимизировать потери при восстановлении входных данных.

3.4.5 Остальные подходы (исследовал Охрименко Дмитрий)

3.4.5.1 DBSCAN

Суть данного подхода[22] заключается в разделении кластеризуемых точек на основные точки, достижимые точки и выбросы. В качестве основной точки выбирается любая точка, если в ее ϵ-окрестности лежит не менее minpts точек, она признается основной, а все точки внутри ее окрестности - достижимыми и добавляются к кластеру, иначе она признается выбросом. После этого для каждой точки из ее окрестности строится своя ϵ-окрестность, и новые точки, лежащие в ней, добавляются к кластеру. Таким образом, повторяя данный процесс для всех точек, мы построим полносвязный кластер. После чего выбирается новая точка из уже не определенных к какому-либо кластеру и весь алгоритм повторяется для нее.

3.4.5.2 Affinity propagation

В этом подходе[23] используется две матрицы, а так же функция сравнения похожести двух точек x_i и x_k, например, $s(i,k) = -||x_i - x_k||^2$ Матрица

"ответственности" R, где r(i, k) показывает, насколько хорошо x_k стыкуется с x_i по сравнению с другими вариантами x_i. Матрица "доступности" A, где a(i, k) показывает, насколько вероятно было бы x_i выбрать x_k в качестве пары по сравнению с остальными x_k. Обе матрицы инициализируются нулями, после чего происходит обновление, сначала матрицы R:

$$r(i,k) \longleftarrow s(i,k) - max_{k \neq k'} a(i,k') + s(i,k')$$

А затем матрицы A:

$$a(i,k) \longleftarrow min(0, r(k,k) + \sum_{i' \notin i,k} max(0, r(i',k)))$$

$$a(k,k) \longleftarrow \sum_{i' \neq k} max(0, r(i',k))$$

Итерации продолжаются, пока кластеры не остаются неизменными в течение некоторого количества итераций, либо до достижения максимального заранее обговоренного количества итераций. Из матриц извлекаются объекты, для которых $r(i,i) + a(i,i) > 0$.

Мы опробовали последние два подхода на наших данных, но они не дали положительных результатов.

3.5 Эксперименты

Из-за технических ограничений все исследования проводились на небольшой выборке данных размером сто тысяч обращений.

3.5.1 Бенчмарк модель (выполнил Федоров Павел)

В качестве бенчмарк модели для извлечения признаков из текста была использована статистическая мера TF-IDF реализация была взята из библиотеки sklearn, и разные алгоритмы кластеризации для разбиения сообщений

по темам. При этом, т. к. нули в матрице представлений документов не несут важной информации, предварительно была понижена размерность векторов с помощью метода главных компонент. Такой подход был применен в статье «Text document dimension reduction using Principal Component Analysis» [24]. Результаты работы сравнивались по метрикам Silhouette coefficient [2] и Topic coherence [3], оптимальному количеству тем, а также по времени работы, как и в дальнейших экспериментах.

3.5.2 Text2vec подходы (выполнил Охрименко Дмитрий)

Здесь рассматривались такие подходы для анализа данных, как word2vec, fastText и doc2vec. После предобработки данных, описанной выше, мы обучили text2vec подходы, экспериментируя с гиперпараметрами аналогично тому, как мы делали в предыдущем блоке, мы подобрали их оптимальные значения. Отдельно отмечу, что для получения вектора документа в методе word2vec мы суммировали значения слов, составляющих этот самый документ. В итоге лучший результат показал алгоритм doc2vec.

3.5.3 LDA (выполнил Федоров Павел)

Реализация алгоритма была взята из библиотеки gensim, с поддержкой многопоточной работы. Для этого метода не требовалась предварительная векторизация сообщений (только стандартная чистка и стэмминг). Входными данными для тематической модели LDA являются словарь (все встречающейся слова из сообщений) и корпус, который представляет собой документы с номерами слов из словаря вместо самих слов. По итогу получилось неплохо кластеризовать сообщения по тематикам, где каждый кластер может представлять смесь тем. Ниже представлена проекция векторных представлений тем на плоскость, которая была сделана с помощью библиотеки pyLDAvis, используя алгоритм понижения размерности T-SNE.

Topic Coherence для 110 кластеров был равен 0.58, что уже лучше результата бенчмарк модели. Так на ней видно, что например 31 тема связана с

отключением услуг связи больше, чем с другими тематиками.

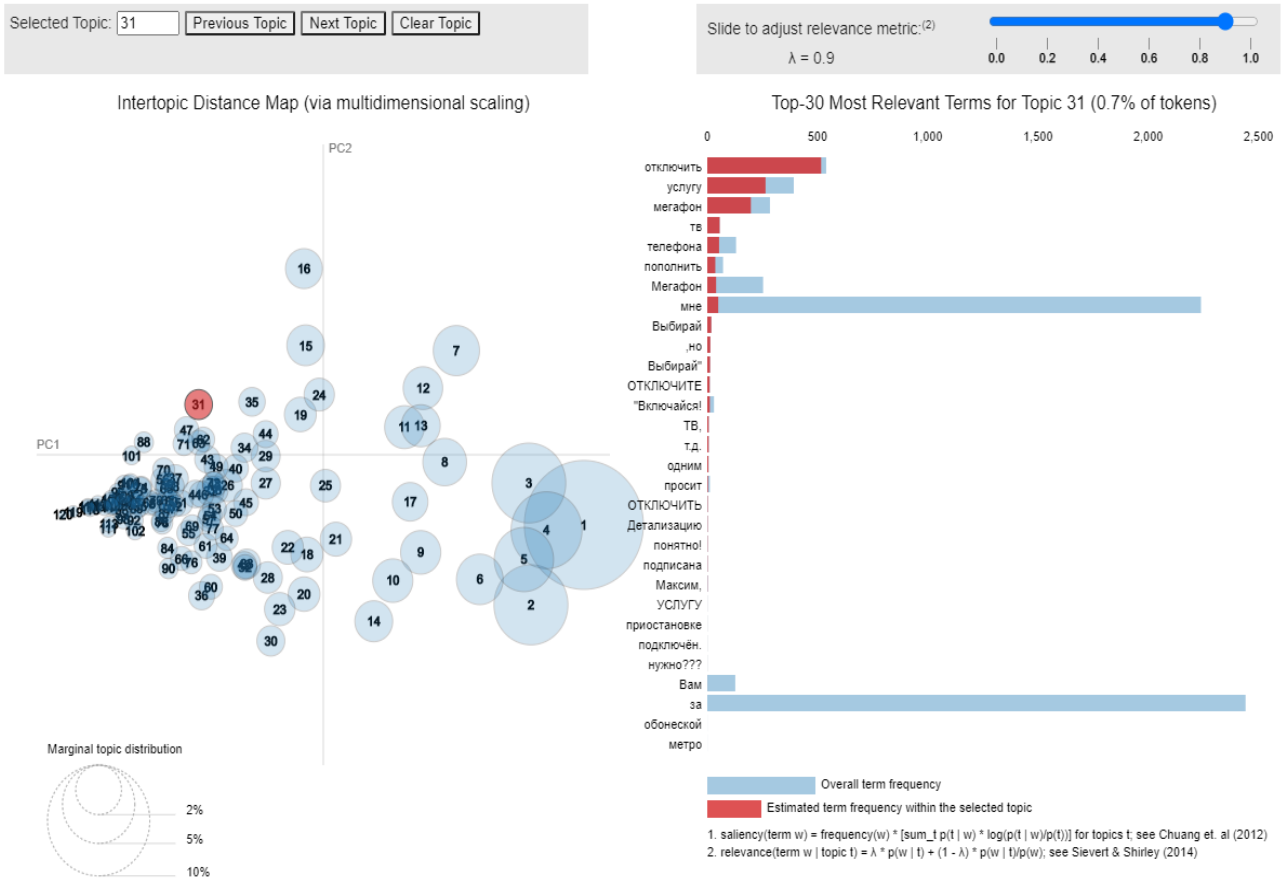


Рис. 5: Проекция векторных представлений тематик.

3.5.4 BERT (выполнил Федоров Павел)

В качестве предобученной модели для получения эмбеддингов сообщений использовалась русскоязычная разговорная модель BERT [?] из библиотеки deeppavlov: "Conversational RuBERT, Russian".

3.5.5 LDA + BERT (выполнили совместно)

Поскольку полученный вектор находится в многомерном пространстве, где информация разрежена и коррелирована, использовался автоэнкодер, чтобы изучить представление векторов в скрытом пространстве более низкой размерности. Предполагая, что он должен иметь форму многообразия в пространстве большой размерности, получились представления более низкой размерности с более сжатой информацией. И для была обучена нейросеть

с архитектурой автоэнкодера на представлениях скрытого пространства и потом получены контекстные темы. Но по итогу такте представления слов оказались хуже, чем эмбеддинги из BERT.

4 Заключение

Данный раздел был написан совместно

4.1 Результаты

При поборе числа кластеров в среднем для всех алгоритмов наилучшее количество кластеров получилось равно 110, поэтому во всех экспериментах использовалось именное такое число кластеров. В таблице ниже метрики расшифровываются как SC - Silhouette coefficent и TC - Topic coherence.

	K-means		AHC		BIRCH		Autoenc.	
	SC	TC	SC	TC	SC	TC	SC	TC
TF-IDF	0.51	0.48	0.53	0.49	0.29	0.31	0.09	0.45
Word2Vec	0.39	0.40	0.43	0.41	0.13	0.05	0.35	0.41
FastText	0.53	0.49	0.57	0.51	0.09	0.34	0.54	**0.62**
Doc2Vec	0.55	**0.54**	0.58	**0.55**	0.33	0.19	0.56	0.50
BERT	**0.58**	0.50	**0.60**	**0.55**	**0.37**	**0.43**	**0.64**	**0.62**
BERT+LDA	0.50	0.48	0.52	0.51	0.35	0.33	0.51	0.60

Так из приведенной таблице метрик видно, что наилучшие представления дали эмбеддинги, полученные с помощью предобученной языковой модели BERT [11] независимо от выбора алгоритма кластеризации, а также представления из Doc2vec [10]. В том числе тематическая модель LDA [5] тоже достаточно хорошо интерпретирует тематики.

Также, в задаче кластеризации текстов, цифры показывают, что наилучшие результаты демонстрируют алгоритмы Агломеративной иерархической кластеризации и нейросеть с архитектурой Автоэнкодер.

При этом удалось обучить модели лучше, чем бенчмарк модель, и интерпретировать из них темы, которые не использовались ранее.

4.2 Перспективы исследования

Существуют другие неизученные алгоритмы для тематического моделирования, такие, как например BigARTM [25], который довольно популярен в индустрии, или нейросетевые тематические модели, как в работе [15], изучением которых можно заняться в будущем. Также в планах доработать библиотеку с использованными моделями для классификации новых текстовых запросов.

Список литературы

1. Мандель И. Д. *Кластерный анализ. — М.: Финансы и статистика*, ISBN 5-279-00050-7, 1988

2. Peter J.Rousseeuw. *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, University of Fribourg, ISES, 1986

3. Michael Roeder, Andreas Both and Alexander Hinneburg. *Exploring the space of topic coherence measures*, WSDM'15 399–408 p., 2015

4. Jones K. S. *A statistical interpretation of term specificity and its application in retrieval*, In MCB University Press 2004. pp. 493-502, 2004

5. David M Blei, Andrew Y Ng, and Michael I Jordan. *Latent dirichlet allocation.* Journal of machine Learning research, 3(Jan):993–1022, 2003

6. Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T. M. (2000). *Text classification from labeled and unlabeled documents using EM.* Machine Learning, 39(2/3), pp. 103–134, 2000

7. Thomas Hofmann. *Unsupervised Learning by Probabilistic Latent Semantic Analysis*. Machine Learning Journal, 42(1), pp. 177–196., 2001

8. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. In International Conference on Learning Representations (ICLR), 2013

9. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. *Enriching Word Vectors with Subword Information*. Transactions of the Association for Computational Linguistics, 2016

10. Quoc V. Le, Tomas Mikolov. *Distributed Representations of Sentences and Documents*. Proceedings of the 31st International Conference on Machine Learning, 2014

11. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805., 2018

12. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. *Attention Is All You Need*, arXiv preprint arXiv:1706.03762., 2017

13. Suzanna Sia, Ayush Dalmia, Sabrina J. Mielke. *Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!*, arXiv preprint arXiv:2004.14914., 2020

14. Federico Bianchi, Silvia Terragni, Dirk Hovy. *Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence*, arXiv preprint arXiv:2004.03974., 2020

15. Akash Srivastava and Charles Sutton. *Autoencoding variational inference for topic models*, arXiv preprint arXiv:1703.01488., 2017

16. Diederik P Kingma, Max Welling. *Auto-Encoding Variational Bayes*, arXiv preprint arXiv:1703.01488., 2013

17. Marco Capó, Aritz Pérez, Jose A. Lozano. *An efficient K-means clustering algorithm for massive data*, arXiv preprint arXiv:1801.02949., 2018

18. Tian Zhang, Raghu Ramakrishnan, Miron Livny. *BIRCH: an efficient data clustering method for very large databases*, ACM SIGMOD Record, 1996

19. Baldi, P. *Autoencoders, Unsupervised Learning, and Deep Architectures*, Proceedings of Machine Learning Research, vol. 27, pp. 37–49. PMLR, 2012

20. Sepp Hochreiter, Jürgen Schmidhuber. *Long Short-Term Memory*, Neural Computation 9(8):1735-80, 1997

21. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems 25, pp. 1097-1105, 2012

22. Daren Wang, Xinyang Lu, Alessandro Rinaldo *DBSCAN: Optimal Rates For Density Based Clustering*, WHOA-PSI 2018

23. Kaijun Wang, Junying Zhang, Dan Li, Xinna Zhang, Tao Guo *Adaptive Affinity Propagation Clustering*, 2008

24. Soufiene Jaffali, Salma Jamoussi *Text document dimension reduction using Principal Component Analysis*, 7th. International Conference on Information Systems and Economic Intelligence, 2012

25. Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, Marina Dudarenko. *BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections*, Conference: International Conference on Analysis of Images, Social Networks and Texts, 2015