РЕФЕРАТ

Магистерская диссертация содержит 33 страницы, 4 таблицы, 7 рисунков. Список использованных источников содержит 15 позиций.

ТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, КЛАСТЕРИЗАЦИЯ, АВТОКОДИРОВЩИК, АНАЛИЗ ДИАЛОГОВЫХ ДАННЫХ.

Магистерская диссертация посвящена построению системы анализа тематик на данных транкрибаций звонков колл-центра. На входе имеется коллекция документов, отобранная аналитиком по метаданным звонка, таким как дата, продукт, город звонящего, продолжительность диалога и другие. На выходе получаем разделение коллекции документов на группы с общей тематикой. Каждая группа описана словами, наиболее характерными для нее. Аналитик имеет возможность выявить основные проблемы, о которых заявляют клиенты и оценить объемы звонков по каждой из проблем.

Для решения поставленной задачи было проведено сравнение различных методов тематического моделирования, начиная от классических и заканчивая современными нейронными моделями. При построении модели учтена специфика предметной области, был разработан алгоритм подготовки данных, состоящий из семи этапов. Для валидации качества были привлечены эксперты в предметной области, которые оценили качество полученных тематик. В результате была построена модель, которая показывает высокую интепретируемость тем по мнению экспертов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Большинство компаний часто сталкиваются с задачей оптимизации расходов на колл-центры. Одним из направлений снижения затрат является улучшение клиентского сервиса и тем самым снижение количества звонков. Для выполнения этой задачи аналитикам необходимо изучить основные тематики обращений, выделить наиболее перспективные и предложить решения для уменьшения количества звонков. Процесс анализа тематик можно построить различными способами. Возможно прослушать небольшую часть записей звонков и по ней составить представление о тематиках всего объема звонков, однако это очень монотонный и трудозатратный процесс. Другим вариантом является обучение модели классификации, которая будет относить звонок к одной или нескольким из заранее заданным категориям. Недостаток данного подхода заключается в том, что список категорий ограничен и для добавление новой категории нужно собирать обучающую выборку и обновить модель классификации. В данной работе рассматривается применение тематического моделирования для выделения тематик обращений.

Тематическое моделирование – это методом обучения без учителя, что позволяет строить модели, не прибегая к ресурсам разметки. Модель делит набор данных на группы, объединенные общей темой. Для каждой группы выделяются слова, наилучшим образом характеризующие ее. Это позволяет аналитику быстро изучить, о чем чаще всего говорят клиенты, о каких проблемах они сообщают и какие вопросы задают, также оценить объем звонков по каждой из тем. Если аналитик не может понять, о чем тема по наиболее характерным словам, он может прочитать несколько наиболее вероятных диалогов.

Целями работы являются:

- исследование современных подходов использования тематического моделирования;

- разработка алгоритма применения тематических моделей к диалоговым данным;
- анализ качества полученных моделей;
- оценка применимости моделей на практике.

В результате практических экспериментов были найдены методы, позволяющие повысить качество диалоговых данных, лучше подготовить их перед использованием в модели, а значит улучшить результаты работы тематических моделей. Была определена модель, которая показывает высокое качество выделения тематик текста как по значениям метрик качества, так и по оценке экспертов. Был произведен анализ применимости моделей на практике путем оценки временных затрат на обучение одной модели, а также требуемых для этого мощностей.

В разделе 1.1 описана концепция тематического моделирования, а также основные модели, которые для этого используются. В разделе 1.2 описаны методы оценки качества тематических моделей. В разделе два описан процесс обучения тематической модели, приведены результаты сравнительных экспериментов. Также в этом разделе описан процесс валидации тематических моделей экспертами в предметной области, приведены и проанализированы результаты.

В заключении подведены итоги работы.

ОСНОВНАЯ ЧАСТЬ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Описание моделей

1.1.1. BigARTM

Дана коллекция документов D, в которой используется множество токенов W. Все документы $d \in D$ в коллекции состоят из последовательности токенов $(w_1, \dots, w_{n_d}) \in W$. Также для каждого токена сохраняется число вхождений n_{dw} в документ d.

Коллекция документов описывается матрицей частотных оценок вероятности встретить токен w_i в документе d_j:

$$F = \left(\frac{n_{dw}}{n_d}\right)_{W \times D} \tag{1.1}$$

При построении модели предполагается, что существует множество тем T, которые описывают множество документов D. Коллекция документов представляет как выборка троек $(w_i, d_i, t_i), i = 1 \dots n$ из дискретного распределения $p(w, d, t)$, заданном на конечном вероятностном пространстве $W \times D \times T$[14]. Причем токены и документы коллекции являются наблюдаемыми переменными, в то время как темы – скрытыми.

Определим дополнительные понятия: $p(w|t)$ – вероятность токена w в теме t, а $p(t|d)$ – вероятность темы t в документе d. Тогда можно описать модель двумя матрицами:

$$\Phi = p(w|t)_{W \times T} \tag{1.2}$$

$$\Theta = p(t|d)_{T \times D}, \tag{1.3}$$

где Φ – матрица токенов в темах, а Θ – матрица тем в документах.

Важным допущением является то, что в модели не учитывается порядок слов, так как используется «мешок слов». Также нет возможности учитывать порядок документов в коллекции.

Задача тематического моделирования определяется следующим образом: по коллекции документов D необходимо оценить параметры $\phi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$. Данную задачу можно решить с помощью матричного

разложения матрицы частот на произведение двух матриц: матрицы Φ токенов-тем и матрицы Θ тем-документов (рис. 1.1):

$$F_{W\times D} \approx \Phi_{W\times T} \times \Theta_{T\times D} \quad (1.4)$$

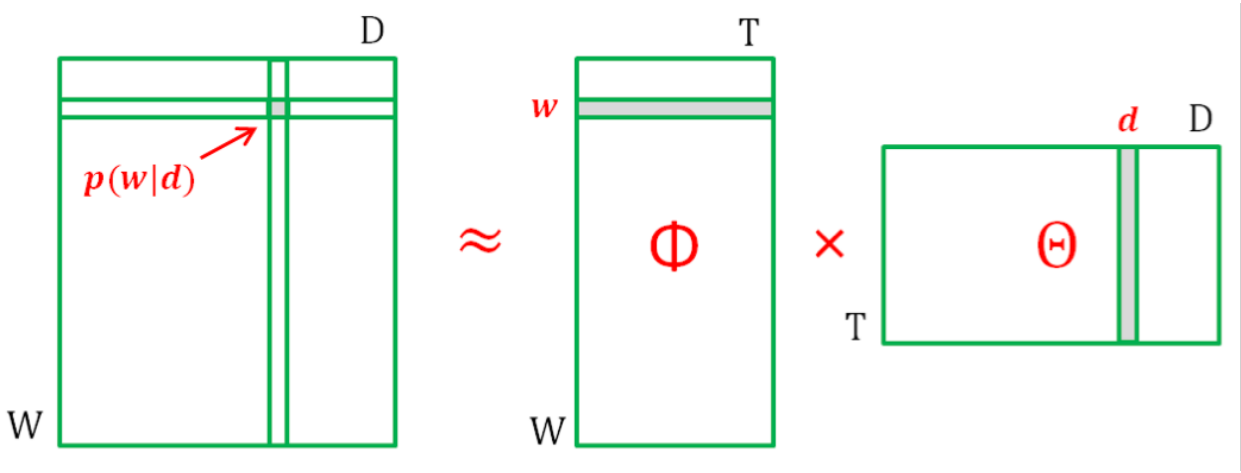

Рис. 1.1 Разложение матрицы токенов документов

1.1.2. Автокодировщик

Одна из архитектур, которую применяют при обучении без учителя – автокодировщик[10] или автоэнкодер. В простейшем случае модель состоит из входного, промежуточного и выходного слоя. При этом выходной слой содержит столько же нейронов, сколько и входной.

Модель автокодировщика состоит из двух частей: энкодера g и декодера f. Энкодер переводит входные данные x в скрытое представление $h = g(x)$ меньшей размерности, а декодер восстанавливает входные данные по скрытому представлению $x' = f(h)$. Параметры энкодера и декодера выучиваются одновременно, так, чтобы восстановленные данные были максимально похожи на входные. В качестве функции «похожести» может использоваться квадратичная функция ошибки или кросс-энтропия с сигмоидной функцией активации на последнем слое.

Однако у автокодировщика есть одна проблема: нам ничего не известно о распределении переменной скрытого состояния, а значит невозможно генерировать объекты с высоким качеством. Обозначенную проблему решает модификация архитектуры – вариационный автокодировщик (variational autoencoder, VAE).

Вариационный автокодировщик представляет собой генеративную модель — он оценивает плотность вероятности (PDF) обучающих данных. Он является естественным выбором для тематических моделей, поскольку обучается сеть, которая непосредственно сопоставляет документ с приближенным апостериорным распределением без необходимости выполнения дальнейших вариационных обновлений. Это привлекательно, потому что в тематических моделях мы ожидаем, что отображение из документов в апостериорные распределения будет хорошо себя вести, то есть что небольшое изменение в документе приведет только к небольшому изменению в темах.

Модель, аналогично автоэнкодеру, состоит из двух частей:

1. Энкодер отображает гауссовское распределение в истинное распределение по скрытому пространству
2. Декодер восстанавливает исходные данные из скрытого пространтства.

Общая схема работы вариационного автокодировщика приведена на рис. 1.2.

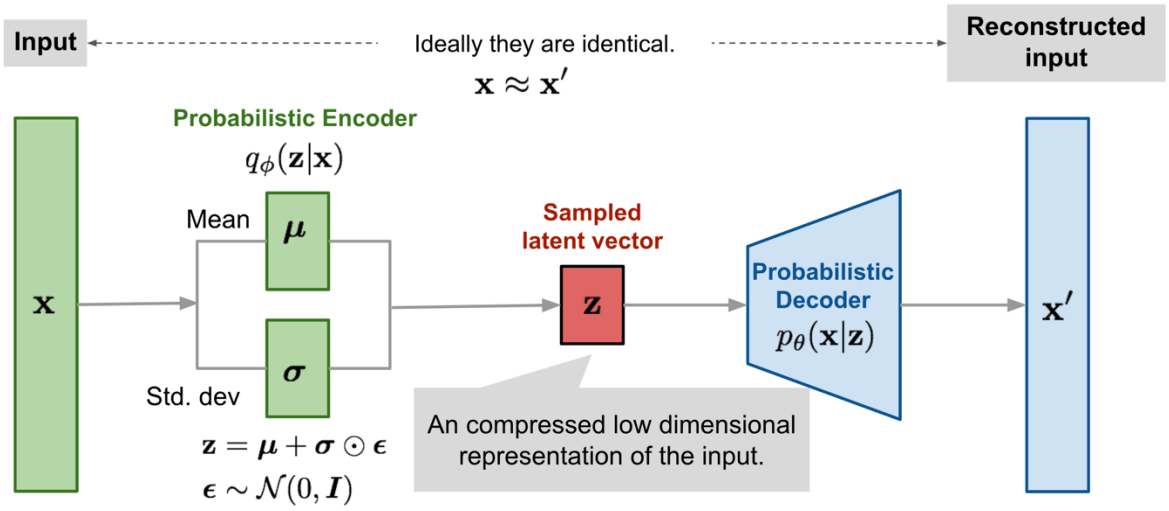

Рис. 1.2 Схема вариационого автокодировщика

Сложность обучения такой модели состоит в том, что операция сэмплирования из распределения – это стохастический процесс, а значит невозможно сделать обратное распространение ошибки. Чтобы побороться с

этой проблемой предлагается трюк репараметризации. Зачастую возможно заменить генерацию точек из параметрического распределения на генерацию точек из распределения без настраиваемых параметров. В случае многомерного гауссовского распределения получаем следующее выражение:

$$h \sim N(z|\,\mu, \sigma^2) \tag{1.5}$$

$$h = \mu + \sigma \cdot \epsilon, \tag{1.6}$$

где $\epsilon \sim N(0, I)$.

Данный трюк применим не только к нормальному распределению, но и к другим видам. В случае гауссовского распределения модель выучивает параметры μ и σ, среднее и дисперсию распределения, которые затем используются в трюке репараметризации. Схема трюка репараметризации изображена на рис. 1.3.

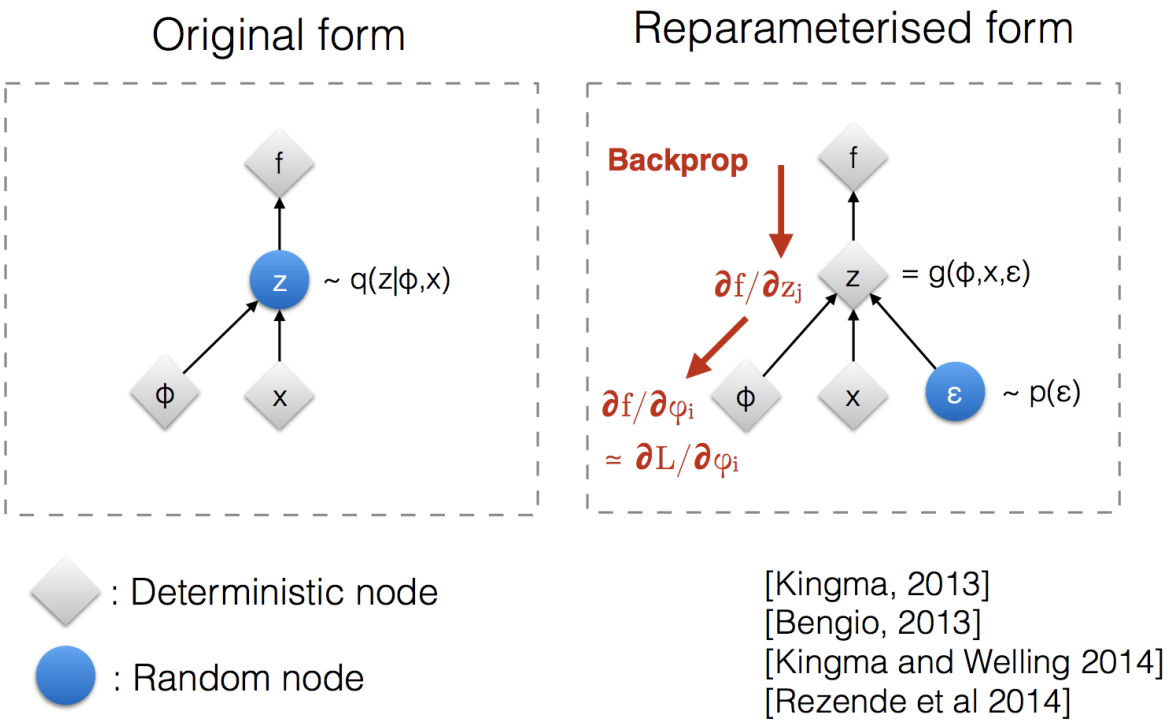

Рис. 1.3 Схема трюка репараметризации

Для обучения модели применяется комбинированная функция ошибки. С одной стороны мы хотим, чтобы модель хорошо восстанавливала входные данные, а с другой чтобы распределение скрытой переменной было близко к многомерному гауссовому распределению. Мы хотим максимизировать логарифм правдоподобия генерации реальных данных $\log\left(p_\phi(x)\right)$, а также

минимизировать разницу между реальным и предполагаемым апостериорным распределением $D_{\mathrm{KL}}\left(q_{\phi}(\mathbf{z}|\mathbf{x})|p_{\theta}(\mathbf{z}|\mathbf{x})\right)$. Функция ошибки модели принимает следующий вид:

$$L_{\mathrm{VAE}} = -\log p_{\theta}(x) + D_{\mathrm{KL}}\left(q_{\phi}(z|x)|p_{\theta}(z|x)\right) \tag{1.7}$$

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{1.8}$$

В статьях [4] и [8] предлагается использовать вариационный автоэнкодер без каких-либо модификаций для тематического моделирования. В качестве входных данных подается представление документов в виде мешка слов. Энкодер и декодер состоят из полносвязных слоев.

1.1.3. ProdLDA

ProdLDA – нейронная тематическая модель, основанная на вариационном автокодировщике. Энкодер модели обучается отображать представленные в виде мешка слов документов в непрерывное скрытое представление. Декодер, в свою очередь, восстанавливает мешок слов из полученного представления.

ProdLDA явно аппроксимирует априорное распределение Дирихле, используя распределение Гаусса, вместо использования гауссовского априорного распределения, как другие модели, предложенные ранее. Более того, ProdLDA заменяет полиномиальное распределение по отдельным словам в стандартном LDA произведением экспертов[13] (отсюда и название ProdLDA).

1.1.4. SBERT

Модель SBERT не является тематической моделью, однако считаю важным рассмотреть ее, так как она является составной частью трех моделей, которые будут описаны далее. SBERT является модификацией модели BERT и используется для кодирования текста документа в семантический вектор. Полученный вектор может использоваться для поиска перефразирований текста, семантического поиска, поиска текстов, похожих по смыслу на заданный.

Для обучения SBERT использовались веса предобученной модели BERT использовалась сиамская структура сети и косинусная мера близости в качестве функции ошибки:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\| \mathbf{A} \| \| \mathbf{B} \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (1.9)$$

где A и B – вектора документов.

При использовании SBERT не требуется предобработка текста, так как модель обучена работе с исходными текстами, без какой-либо подготовки. Проведение предобработки даже может повредить качеству семантического вектора,

1.1.5. Contextualized Topic Model

Contextualized Topic Model (CTM) – одна из наиболее современных моделей, вышедшая в 2020 году. Она совмещает в себе нейронную модель тематического моделирования, показывающую наивысшее качество на момент публикации, и наработки из других областей анализа текста.

Модель состоит из двух частей: тематической модели ProdLDA и модели контекстной векторизации документов SBERT (рис. 1.4)

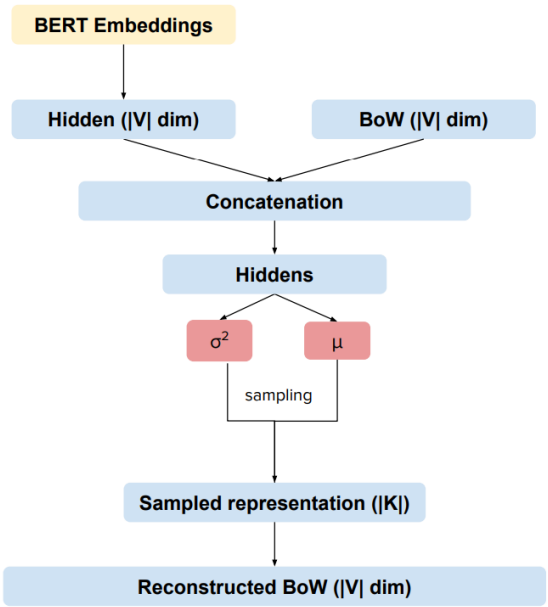

Рис. 1.4 Архитектура модели CTM

Входные данные для модели также состоят из двух частей: семантического вектора документа, полученного с помощью модели SBERT и представления документа в виде мешка слов. Перед формированием мешка слов производится стандартная предобработка текста, такая как удаление стоп слов, самых частых и самых редких слов.

Модель SBERT используется лишь для генерации семантических векторов документов и не учувствует напрямую в процессе обучения, ее веса не настраиваются согласно функции ошибки. Это значит, что процесс обучения модели не отличается от стандартного автокодировщика.

1.1.6. Top2Vec

Модель Top2Vec[11] отличается от моделей, предложенных выше, тем, что состоит из нескольких компонентов, которые обучаются друг за другом:

- векторизация документов и токенов;
- уменьшение размерности;
- кластеризация.

Рассмотрим каждый компонент подробнее. В первую очередь для построения модели такого типа необходимо создать совместное пространство векторов токенов и документов, которое будет обладать некоторыми свойствами. Семантически похожие документы должны располагаться близко друг к другу в данном пространстве, а непохожие документы должны располагаться дальше друг от друга. Кроме того, слова должны быть близки к документам, которые они лучше всего описывают. С помощью векторов документов и слов в одном пространстве можно вычислять векторы тем. Это пространственное представление слов и документов называется семантическим пространством. Авторы Top2Vec утверждают, что семантическое пространство с описанными свойствами является непрерывным представлением тем.

Для создания такого векторного пространства используется модель doc2vec. Архитектура doc2vec очень похожа на модель word2vec skip-gram, которая использует контекстное слово для прогнозирования окружающих

слов в контекстном окне. Единственное отличие заключается в том, что doc2vec меняет контекстное слово на вектор документа, который используется для прогнозирования окружающих слов в контекстном окне. Это сходство позволяет одновременно обучать векторы документов и слов, которые находятся в одном пространстве.

Векторы слов, наиболее близкие к вектору документа, являются наиболее семантически описательными для темы документа. В семантическом пространстве плотная область документов может быть интерпретирована как область очень похожих документов. Эта плотная область документов указывает на основную тему, которая является общей для документов. Поскольку векторы документов представляют темы документов, можно вычислить центроид или среднее значение этих векторов. Этот центроид является вектором темы, который наиболее репрезентативен для плотной области документов, из которой он был рассчитан. Слова, наиболее близкие к этому вектору темы — это слова, которые лучше всего описывают его семантически. Основное предположение, лежащее в основе top2vec, заключается в том, что количество плотных областей векторов документов равно количеству значимых тем. Для поиска плотных областей документов в семантическом пространстве векторов документов используется кластеризация на основе плотности, в частности метод HDBSCAN [25, 26, 27].

Однако "проклятие размерности", возникающее в результате многомерных векторов документов, создает две основные проблемы. Многомерное семантическое пространство из 300 измерений очень разряженное, это затрудняет поиск плотных кластеров. Чтобы решить эту проблему, выполняется уменьшение размерности векторов документа с помощью алгоритма UMAP.

1.1.7. BERTopic

Алгоритм BERTopic во многом похож на Top2Vec, в нем также используется кластеризация семантических векторов с помощью HDBSCAN, однако есть и отличия. На первом этапе создаются только векторизуются

только документы, а не токены. Затем, аналогично с Top2Vec, с помощью UMAP уменьшается размерность векторов документов и затем они кластеризуются методом HDBSCAN.

Для слов, которые образуют тему кластера, используется модифицированная формула $TF - IDF$:

$$c - TF - IDF_i = \frac{t_i}{w_i} \times \log \frac{m}{\sum_j^n t_j}, \quad (1.10)$$

где w_i — количество слов в кластере i, t_i — частота слова t в кластере i, m — количество слов в кластере.

Схема работы алгоритма представлена на рис. 1.5

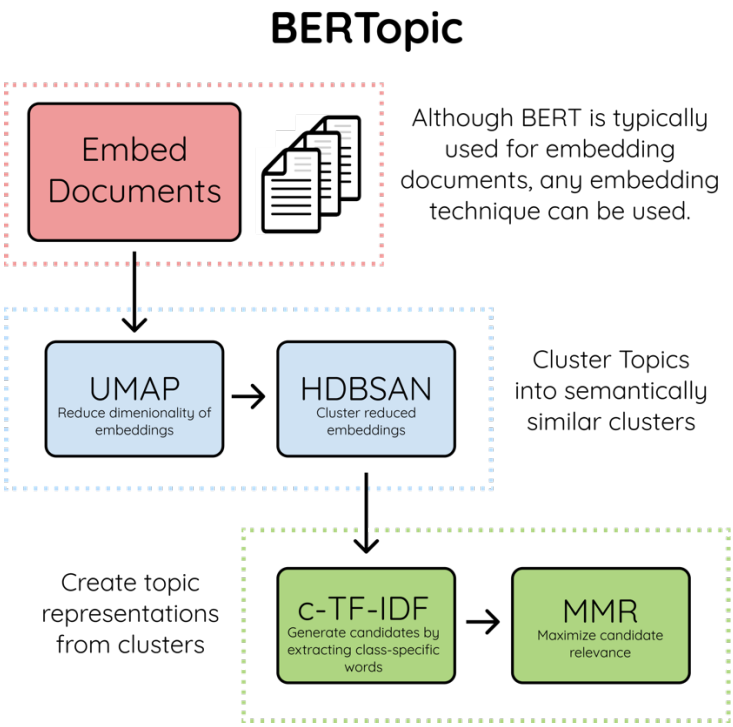

Рис. 1.5 Схема работы алгоритма BERTopic

1.2. Оценка качества тематических моделей

Зачастую использование тематической модели сводится к построению выводов по темам, без глубокого анализа документов, входящих в каждую из тем. Как правило исследователи изучают несколько наиболее вероятных токенов для каждой тематики. Это значит, что оценке качества получаемых тематик нужно уделить особое внимание. Процесс оценки можно разбить на два подхода: с помощью экспертной оценки и с использованием метрик качества.

1.2.1. Экспертная оценка тематических моделей

В работе [1] авторы предлагают оценивать темы по шкале «хорошая — средняя — плохая». Если оценки экспертов не совпадают, то они должны прийти к консенсусу через обсуждение. Дополнительно эксперты классифицировали темы низкого качества, относя их к классам тем-химер, случайных тем, тем с посторонними словами и несбалансированных тем. Описание каждой темы состояло из 30 наиболее значимых слов, отсортированных по убыванию их вероятности.

Для отнесения темы к списку «хороших» от экспертов требовалось, чтобы ее можно было соотнести с единой целостной концепцией. Также в ряде случаев дополнительно требовалось, чтобы среди наиболее вероятных слов содержались слова, напрямую связанные с концепцией темы.

Большинство последующих работ использовало похожий процесс, в котором темы непосредственно оценивались экспертами по их качеству.

1.2.2. Когерентность

В работах [2; 3] впервые была предложена метрика когерентности, основанная на метрике из теории информации и статистики PMI.

$$PMI(u,v) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}, \tag{1.11}$$

где $\mathrm{P}(w_i, w_j)$ – совместная вероятность токенов w_i и w_j, согласно некоторой вероятностной модели, $P(w_i)$ и $P(w_j)$ – вероятность отдельно токена w_i и токена w_j.

Метрика вычисляется по формуле:

$$C_{\mathrm{UCI}} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathrm{PM\,I}(w_i, w_j) \tag{1.12}$$

Обычно вероятности токенов вычисляются на основе частоты встречаемости токенов в корпусе текстов. Вероятность совместной встречаемости оценивается по частоте встречаемости токенов в рамках окна заданной ширины.

Метрика когерентности принимает большое значение если наиболее вероятные слова встречаются вместе чаще, чем в случайном распределении. Было отмечено, что метрика когерентности коррелирует с экспертной оценкой.

Существуют также другие способы вычисления метрики когерентности. В статье [9] предлагается использовать асимметричную меру согласия между наиболее вероятными парами слов

$$C_{\text{UMass}} = \frac{2}{N \cdot (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \tag{1.13}$$

Параметр ϵ необходим для предотвращения взятия логарифма от 0. В статье [6] авторы определили, что метрика когерентности лучше оценивает качество тем, если ϵ равно небольшому значению, а не 1, как предлагалось в оригинальной статье.

В статье «Evaluating topic coherence using distributional semantics» [5] предлагается формула когерентности, основанная на контекстных векторах наиболее вероятных слов.

Когерентность вычисляется как среднее между попарными расстояниями контекстных векторов темы

$$C_{npmi} == \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{sim}(w_i, w_j), \tag{1.14}$$

где $\text{sim}(w_i, w_j)$ - симметричная мера близости слов. Для ее вычисления можно использовать:

- косинусную близость:

$$\text{sim}_{\cos}(\vec{w}_i, \vec{w}_j) = \frac{\vec{w}_i \cdot \vec{w}_j}{\|\vec{w}_i\| \|\vec{w}_j\|}, \tag{1.15}$$

- коэффициент Сёренсена (Dice)

$$sim_{\text{Dice}}(w_i, w_j) = \frac{2 \times \sum_{k=1}^{N} min(\overrightarrow{w_{ik}}, \overrightarrow{w_{jk}})}{\sum_{k=1}^{N} (\overrightarrow{w_{ik}} + \overrightarrow{w_{jk}})}, \tag{1.16}$$

- коэффициент Жаккара

$$\text{sim}_{\text{Jaccard}}\left(w_i, w_j\right) = \frac{\sum_{k=1}^{N} min\left(\overrightarrow{w_{ik}}, \overrightarrow{w_{jk}}\right)}{\sum_{k=1}^{N} max\left(\overrightarrow{w_{ik}}, \overrightarrow{w_{jk}}\right)} \quad (1.17)$$

Контекстные вектора $\overrightarrow{w_{ik}}$ создаются с помощью подсчета совстречаемости слов в окне ±5 токенов вокруг слова w. Наибольшая корреляция с экспертными оценками достигается при взвешивании векторов с помощью метрики NPMI:

$$\overrightarrow{w_{ik}} = \text{NPM I}\left(w_i, w_j\right)^{\gamma} = \left(\frac{\text{PMI}\left(\text{w}_\text{i}, w_j\right)}{-\text{lo g}\left(P\left(w_i, w_j\right) + \epsilon\right)}\right)^{\gamma} \quad (1.18)$$

Параметр γ необходим для повышения значимости больших значений метрики NPMI. Авторы предлагают использовать $\gamma = 2$, так как это значение показало наилучшие результаты в экспериментах.

1.2.3. Тематическое разнообразие

Зачастую для экспертов важна не только интерпретируемость тем, но и их различность. При анализе диалогов эксперту необходимо понять, на какие группы можно разбить набор данных. В статье [12] для оценки различности тем предлагается использовать метрику тематического разнообразия:

$$td = \frac{|\{W_1, \dots, W_n\}|}{N \cdot M}, \quad (1.19)$$

где M – количество тем, $|\{W_1, \dots, W_n\}|$ - количество уникальных среди наиболее вероятных слов всех тем.

1.2.4. Inverted rank biased overlap

Авторы статьи [3] используют метрику Inverted rank biased overlap (IRBO) для оценки различности тем.

$$IRBO\left(W_i, W_j, p, d\right) = 1 - \left(\frac{W_i}{N} \cdot p^N + \frac{1-p}{p} \sum_{i=1}^{N} \frac{w_i}{i} \cdot p^i\right), \quad (1.20)$$

где d – количество слов в одной теме, p – параметр силы влияния порядка слов на метрику.

В отличие от тематического разнообразия, эта метрика учитывает также порядок слов в темах. IRBO принимает значение 0 когда все слова в темах

совпадают и одинаково упорядочены и значение 1 когда все слова различны. Когда $p = 1$, порядок слов не имеет значения, учитывается только их пересечение. Меньшие значения p дают больший вес порядку слов. В качестве оптимального значения предлагается использовать $p = 0.9$. Это значения предлагается авторами метрики RBO [1].

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Описание исходных данных

Эксперименты проводились на наборе из 27281 диалогов, коллекции текстов входящих звонков по кредитной тематике. Каждый документ в коллекции является транскрибацией аудиозаписи диалога с помощью нейронной модели машинного обучения, что затрудняет анализ текста. Результат модели транскрибации отличается от письменного текста отсутствием знаков препинания, заглавных букв, а также наличием ошибок, специфичной для данной модели. Также тексты представляют из себя диалог оператора и клиента, для каждой фразы есть отметка о том, кому она принадлежит: оператору или клиенту. На рис 2.1 представлена диаграмма распределения длины речи клиента и сотрудника в диалогах. Видно, что диалоги преимущественно длинные и состоят из более чем 400 слов.

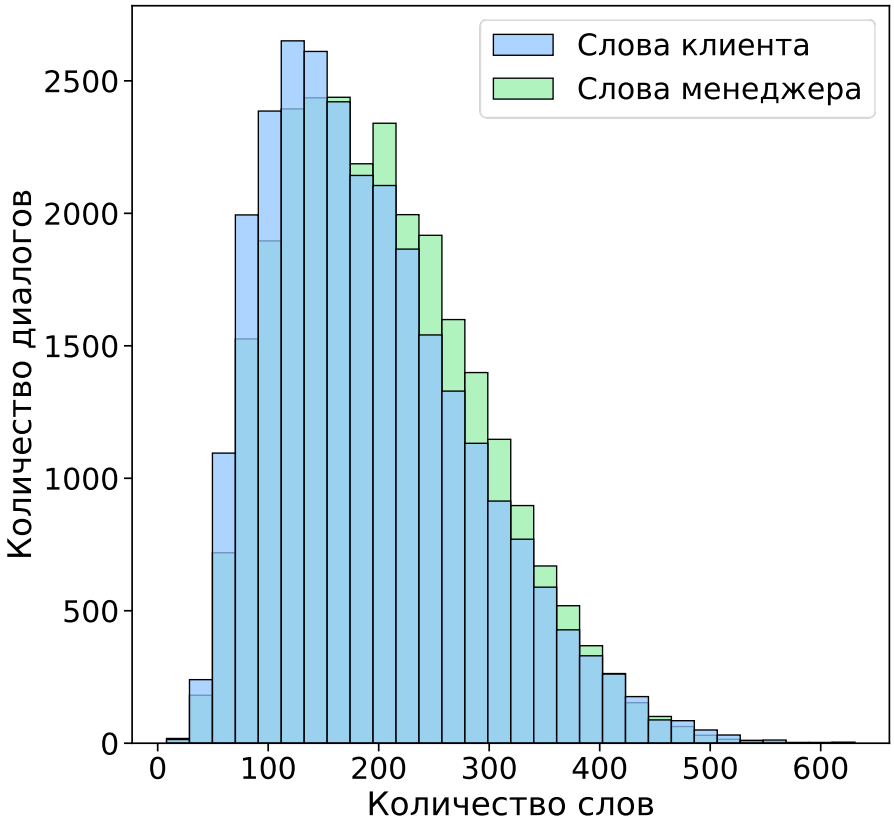

Рис. 2.1 Диаграмма распределения количества слов в диалогах

Слова оператора зачастую сильно шаблонизированы, особенно в начале диалога. Это объясняется наличием инструкций, по которым операторы

должны вести диалог с клиентом. Также в диалогах присутствуют реплики, которые напрямую не относятся к тематике разговора, например приветственные и вводные слова, определение личности клиента. Для решения указанных выше проблем был разработан механизм подготовки данных, который будет описан в следующем разделе.

2.2. Предварительная обработка данных

Обработка данных – это очень важный этап при работе с транскрибациями, потому что он позволяет повысить качество данных, приблизить его к качеству письменного текста. Первый этап в подготовке данных – удаление квазислов («псевдослов»).

Квазислова — искусственные конструкции, построенные по аналогии с лексическими единицами языка[14]. В данном случае под квазисловами понимаются слова, которые не существуют в русском языке, но зачастую похожи по написанию или созвучны с существующими словами.

Список слов, которые могут быть на выходе модели транскрибаций фиксирован и составляет около 500.000 слов. Для выделения из него квазислов были отобраны те слова, леммы которых не входят в известные словари русского языка (Русский орфографический словарь (Лопатин), Словарь современного русского литературного языка (Большой академический словарь), Викисловарь). Проверяется наличие леммы, а не самого слова, так как в словарях есть только слова в начальной форме. Затем полученный список кандидатов в квазислова был провалидирован вручную, из него были убраны слова русского языка, не представленные в используемых словарях.

Так как квазислова чаще всего созвучны с словами русского языка, то для подбора слова-замены было использовано редакционное расстояние, которое равно минимальному количеству вставок, удалений и замен букв для получения второго слова из первого. Для каждого квазислова была подобрана слово-замена из оставшегося после удаления квазислов словаря транскрибатора и затем полученный список замен слов был проверен вручную.

Вторым этапом подготовки стало исправление нарушений беглости речи. Во время разговора люди могут ошибаться и поправлять себя, повторяя эту же фразу заново. Также человека может отвлечь другой человек, либо фоновый шум. Все это приводит к появлению подряд идущих одинаковых слов или словосочетаний. Для устранения этой проблемы было написано несколько регулярных выражений, которые удаляют нарушения беглости речи.

Третий этап – удаление фоновых, шаблонных фраз, не несущих в себе смысла. При создании списка фраз необходимо исходить из цели построения тематической модели. В данном случае целью является выявление о каком продукте или о какой проблеме идет речь в диалоге. Следовательно, к фоновым фразам можно отнести приветствие, прощание, выражение благодарности и определение личности звонящего. На основе инструкций для операторов был составлен справочник фоновых фраз. Для поиска и удаления шаблонных фраз в речи клиента были написаны регулярные выражения.

Четвертый этап – лемматизация, то есть приведение словоформы к лемме – её нормальной (словарной) форме. Лемматизация необходима для уменьшения размера словаря и проводилась с помощью библиотеки Pymorphy2.

Пятый этап – удаление шумовых слов. Шумовые слова увеличивают размер словаря и снижают общее качество модели, также они могут попадать в список наиболее вероятных слов темы – в название темы, что снижает ее интерпретируемость экспертами. В качестве шумовых были выбраны слова следующих частей речи:

- числительные;
- компаративы (лучше, получше, выше);
- местоимения;
- предикативы (некогда);
- предлоги;
- союзы;
- частицы;

- междометия.

Часть речи определяется с помощью библиотеки Pymorphy2 на этапе лемматизации текста.

Помимо фильтрации слов по части речи используются также заранее определенные списки шумовых слов, который был собран из двух источников:

- общие шумовые слова русского языка;
- шумовые слова, специфичные для текстов рассматриваемого домена.

К последним можно отнести такие слова, как «инн», «организация» и т.д.

Последний этап – векторизация и конвертация текста в формат, подходящий для тематической модели. Этот этап отличается в зависимости от модели, так как они имеют разную архитектуру и требуют различные форматы данных на вход. По формату входных данных модели можно разделить на следующие группы:

1) BigARTM;
2) LDA, NVDA, ProdLDA;
3) Top2Vec, Bertopic;
4) CTM.

Рассмотрим каждую группу по отдельности.

Модель BigARTM принимает на вход данные в двух форматах: Vowpal Wabbit или UCI. В данной работе был выбран формат Vowpal Wabbit из-за его простоты в использовании. Формат представляет из себя текстовый файл, в каждой строке которого содержится один документ. В начале строки указывает уникальный идентификатор документа, а затем текстовые поля, разделенные между собой символами «|@».

Для моделей LDA, NVDA и ProdLDA на вход необходимо подать представление текста в виде мешка слов. Для его создания используются методы библиотеки анализа текстов genism.

Методы Top2Vec и Bertopic основаны на кластеризации векторов предложений, а значит для каждого документа необходимо составить семантический вектор. Наиболее современным подходом для решения этой

задачи является использование языковой модели BERT, основанной на архитектуре трансформер. Обучение модели BERT является вычислительно затратной задачей и требует наличия GPU, поэтому используются предобученные на большом объеме данных модели. В данной работе была использована модель rubert-base-cased-sentence от компании DeepPavlov. Для ее обучения использовались тексты из русской части Википедии, а также с новостных сайтов.

Вектора модели BERT дают высокое качество при решении задач классификации и извлечения сущностей, однако при кластеризации они плохо разделимы и зачастую образуют один кластер. Для решения этой проблемы уже обученную модель дообучают на наборе данных «понимания предложений». Один пример обучающей выборки такого набора данных состоит из двух предложений и метки класса. Задачей является предсказать связь между первым и вторым предложением (подразумевает, противоречит или связь отсутствует). Оба этих способа обучения были использованы в модели rubert-base-cased-sentence.

Важно также отметить, что перед подачей в модель BERT текст не проходит некоторые операции подготовки, такие как удаление стоп слов и лемматизация, так как для BERT нет необходимости сокращать размер словаря, так как модель предобучалась на текстах без предобработки.

Для модели CTM в качестве входных данных подается два вектора, описанных ранее: семантический вектор документа и представление документа в виде мешка слов.

2.3. Обучение тематической модели

В результате обучения моделей, которые были описаны в разделе 1.1 получились значения метрик, указанные в таблице 2.1.

Таблица 2.1. Метрики когерентности тематических моделей

Модель	c_v	c_npmi	c_uci
BigARTM wo stopwords	0,4819	0,0365	0,1416
LDA	0,5471	0,0669	0,3765

Продолжение таблицы 2.1

Модель	c_v	c_npmi	c_uci
CTM w stopwords	0,6121	-0,0023	-1,7427
NVDM GSM w stopwords	0,6165	0,0768	0,0633
CTM wo stopwords	0,6173	0,0089	-1,6610
NVDM GSM wo stopwords	0,6182	0,0687	-0,3302
AVITM w stopwords	0,6358	-0,0176	-
AVITM wo stopwords	0,6808	-0,0064	-
Bertopic distiluse	0,7185	0,1438	1,0209
Bertopic doc2vec	0,7401	0,1429	1,0521
Top2Vec	0,7796	0,1639	1,0760

Анализируя полученные результаты, можно сказать, что лучше всего себя показывают модели, основанные на кластеризации семантических векторов (BERTopic и Top2Vec), они лидируют с значимым отрывом. Это можно объяснить тем, что модель использует информацию из семантических векторов, что позволяет ей хорошо разделить документы на группы.

Рассмотрим проекцию семантических векторов документов, используемых в моделях BERTopic и Top2Vec, на 2-х мерное пространство (рис. 2.2).

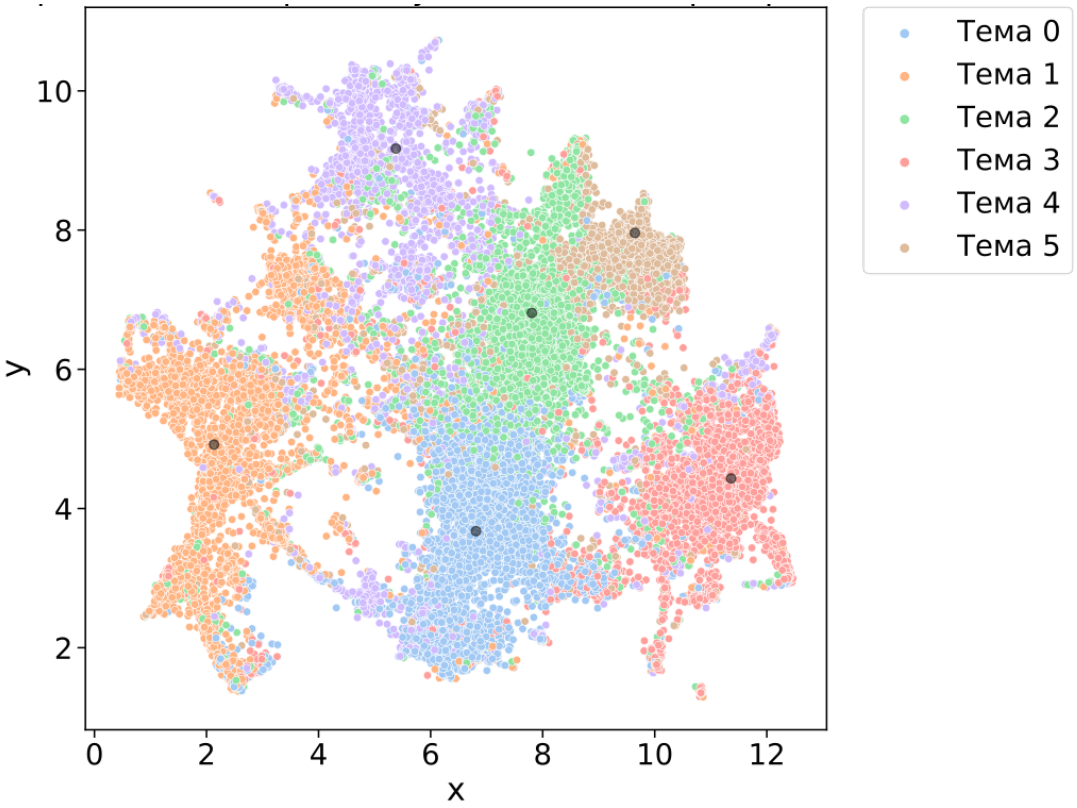

Рис. 2.2. Проекция сематинтических векторов документов на 2-х мерное пространство

Каждая точка на графике – один документ, цвет точки определяет принадлежность документа к определенной теме. Темными точками показаны центроиды каждой из тем. Визуально видно, что темы достаточно хорошо разделимы, области разных цветов практически на пересекаются. Это помогает убедиться в том, что используются качественные семантические векторные представления документов, а также что тематическая модель может верно разделить документы по кластерам.

Кроме оценки качества тем, было измерено и тематическое разнообразие, то есть различность полученных тематик. Результаты отражены в таблице 2.2.

Таблица 2.2. Метрики тематического разнообразия моделей

Модель	Topic diversity	Inversed rank biased overlap
BigARTM wo stopwords	0,8667	0,0369
LDA	0,8833	0,0458
CTM w stopwords	1,0000	0,0000

Продолжение таблицы 2.2

Модель	Topic diversity	Inversed rank biased overlap
NVDM GSM w stopwords	0,8556	0,0290
CTM wo stopwords	1,0000	0,0000
NVDM GSM wo stopwords	0,9444	0,0123
AVITM w stopwords	0,8222	0,0662
AVITM wo stopwords	0,8667	0,0575
Bertopic distiluse	1,0000	0,0000
Bertopic doc2vec	1,0000	0,0000
Top2Vec	1,0000	0,0000

Анализируя таблицу 2.2 можно сказать, что наиболее разнообразные темы выдают модели Top2Vec, Bertopic, а также CTM. В их темах нет повторяющихся слов, каждое слово встречается только в одной теме.

2.4. Оценка времени обучения модели

При сравнении моделей машинного обучения важно оценивать не только качество модели, но и скорость обучения и ресурсы, необходимые для этого. Современные нейросетевые модели зачастую требуют ускорения обучения на GPU, что может стать серьезным ограничением при внедрении модели с промышленные системы. В таблице 1 приведено время, необходимое для обучения каждой из тематических моделей. Сравнение всех моделей производилось на одном и том же устройстве с 6-ядерным процессором Intel Core i7 с частотой 2,6 ГГц и 16 Гб оперативной памяти. При необходимости использовалась видеокарта Nvidia V100 32Gb, это указано в столбце «Устройство». Время, затраченное на подготовку данных (лемматизацию, удаление стоп слов) не учитывалось, так как данные процедуры не отличаются от модели к модели и были выполнены один раз до обучения всех моделей. Полученные результаты указаны в таблице 2.3.

Таблица 2.3. Время обучения тематических моделей

Модель	Время обучения (минут)	Устройство
LDA	2,8	CPU
BigARTM wo stopwords	6,58	CPU
NVDM GSM w stopwords	30,28	CPU
NVDM GSM wo stopwords	29,52	CPU
AVITM w stopwords	33,45	CPU
AVITM wo stopwords	32,26	CPU
CTM w stopwords	10,15	GPU
CTM wo stopwords	10,26	GPU
Bertopic BERT	15,74	GPU
Bertopic doc2vec	18,46	CPU
Top2Vec	7,18	GPU

Стоит отметить, что видеокарта использовалась только в моделях, в которых используются семантические вектора документов BERT, так как получение векторов на центральном процессоре заняло бы существенное количество времени.

Модели NVDM GSM и AVITM имеют поддержку ускорения на GPU, так как являются нейросетевыми алгоритмами и реализованы с помощью библиотеки PyTorch, в таком случае время обучения составляет около 6 минут, что в 5 раз быстрее, чем на CPU. Однако эти модели состоят из последовательных полносвязных слоев, которые плохо поддаются распараллеливанию на GPU, из-за этого загрузка графической карты во время обучения не превышает 10%. Для обучения этих моделей более целесообразно использовать CPU или видеокарту младших поколений с небольшим количеством памяти.

По результатам сравнения можно сделать вывод, что наилучшее соотношение между качеством и скоростью работы показывает модель

Top2Vec, обучение занимает около 7 минут, однако требуется наличие GPU. Если рассматривать случай отсутствия GPU, то оптимальным выбором является BigARTM, данная модель имеет приемлемое качество и высокую скорость обучения на CPU.

2.5. Экспертная валидация моделей

Метрики для тематических моделей оценивают качество тем лишь косвенно, так как само понятие качества темы трудно формализуется. Чтобы убедиться в правильности выводов, сделанных ранее, была проведена разметка качества каждой из тем экспертами, хорошо разбирающимися в предметной области, для этого было привлечено 5 экспертов. Методика проведения разметки описана в главе 1.10. Результаты приведены в таблице 2.4. Максимально возможное значение оценки равно 2.

Таблица 2.4. Оценка качества моделей экспертами

Модель	Средняя оценка
Bertopic doc2vec	1.375
Bertopic BERT	1.125
NVDM GSM с удалением стоп слов	1.125
Top2Vec	1.0
CTM с удалением стоп слов	0.917
CTM без удаления стоп слов	0.917
NVDM GSM без удаления стоп слов	0.917
BigARTM	0.625
AVITM	0.583
LDA	0.417
Случайная	0.167

Чтобы убедиться в корректности эксперимента была добавлена тема, состоящая из случайных слов, которые выбирались с вероятностью, прямо пропорциональной их частоте встречаемости в коллекции.

Анализируя результаты, можно сказать, что модели Bertopic и Top2Vec не потеряли своих лидирующих позиций, что показывает верный выбор метрик для анализа качества моделей.

ЗАКЛЮЧЕНИЕ

Тематическое моделирование позволяют существенно ускорить работу аналитика, занимающегося оптимизацией звонков в колл-центр. Оно позволяет с высоким качеством разделять тексты транскрибаций диалогов на группы, а также выделять тему для каждой группы. Это освобождает аналитика от необходимости читать множество звонков с целью изучения проблем клиентов и позволяет во многом оптимизировать его работу.

В данной диссертации разработан набор шагов по подготовке данных, а также проведено обширное сравнение различных моделей тематического моделирования. В результате была получена модель, которая имеет хорошее качество как по значениям метрик, так и по мнению экспертов. Также была проанализирована практическая применимость модели, которая показала, что модель может быть обучена за приемлемое время и не требует для этого затраты существенных ресурсов.

СПИСОК ИСПОЛЬЗВАННЫХ ИСТОЧНИКОВ

[1] Webber, William, Alistair Moffat, Justin Zobel. A similarity measure for indefinite rankings. ACM Transactions on Information Systems, pages 1-38, 2010.

[2] Newman, David. Automatic evaluation of topic coherence."Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. 2010.

[3] Bianchi, Federico, Silvia Terragni, Dirk Hovy. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. arXiv preprint arXiv:2004.03974, 2020.

[4] Miao, Yishu, Edward Grefenstette, Phil Blunsom. Discovering discrete latent topics with neural variational inference. International Conference on Machine Learning. PMLR, 2017.

[5] Aletras, Nikolaos, Mark Stevenson. Evaluating topic coherence using distributional semantics. Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers. 2013.

[6] Stevens, Keith. Exploring topic coherence over many models and many topics. Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. 2012.

[7] Newman, David, Sarvnaz Karimi, Lawrence Cavedon. External evaluation of topic models. Australasian Doc. Comp. Symp. 2009.

[8] Miao, Yishu, Lei Yu, Phil Blunsom. Neural variational inference for text processing. International conference on machine learning. PMLR, 2016.

[9] Mimno, David. Optimizing semantic coherence in topic models. Proceedings of the 2011 conference on empirical methods in natural language processing. 2011.

[10] Hinton, Geoffrey E., Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. science 313.5786, pages 504-507, 2006.

[11] Angelov, Dimo. Top2Vec: Distributed Representations of Topics. arXiv preprint arXiv:2008.09470, 2020.

[12] Dieng, Adji B., Francisco JR Ruiz, David M. Blei. Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics 8, pages 439-453, 2020.

[13] Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. Neural computation 14.8, pages 1771-1800, 2002.

[14] Валявко Светлана Михайловна, Шулекина Юлия Александровна Особенности смыслового восприятия слова детьми с нарушениями речевого развития // Специальное образование. 2013. №3. URL: https://cyberleninka.ru/article/n/osobennosti-smyslovogo-vospriyatiya-slova-detmi-s-narusheniyami-rechevogo-razvitiya (дата обращения: 20.05.2021).

[15] Янина А. О., Ромов П. А., Воронцов К. В. Рекомендация статей коллективного блога на основе тематической модели. Машинное обучение и анализ данных, 2014. Т. 1, № 8.