МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

ГУЛИН
Кирилл Иванович

ИССЛЕДОВАНИЕ МЕТОДОВ ВЫДЕЛЕНИЯ СМЫСЛОВЫХ ЕДИНИЦ ИЗ ДЕЛОВОЙ ПЕРЕПИСКИ

Дипломная работа

Научный руководитель:
доцент Ю. В. Свирид

Допущена к защите
«____»_____ 2021 г.
Зав. кафедрой дискретной математики
и алгоритмики, доктор физ.-мат. наук,
профессор В. М. Котов

Минск, 2021

Содержание

РЕФЕРАТ

Дипломная работа, 49 с., 26 рис., 8 таблиц, 19 источников.

Ключевые слова: ОБРАБОТКА ТЕКСТОВ, МАШИННОЕ ОБУЧЕНИЕ, ЛАТЕНТНОЕ РАЗМЕЩЕНИЕ ДИРИХЛЕ, МЕТОДЫ ПОНИЖЕНИЯ РАЗМЕРНОСТИ, ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ ТЕКСТОВ, СТАТИСТИЧЕСКИЙ АНАЛИЗ, МЕТОДЫ КЛАСТЕРИЗАЦИИ, НЕЙРОННЫЕ СЕТИ.

Объект исследования — наборы данных деловых электронных переписок.

Цель работы — анализ деловых электронных переписок методами машинного обучения.

Методы исследования — латентное размещение Дирихле, методы кластеризации, методы обработки текстов, методы понижения размерности, методы получения векторных представлений.

Работа посвящена исследованию и анализу деловых электронных переписок, в частности, переписок Хиллари Клинтон и переписок сотрудников корпорации Enron. В результате работы был произведен статистический анализ электронных переписок. Были обнаружены закономерности в исходных данных. Также была разработана кластеризация содержаний электронных писем, в результате которой получились интерпретируемые результаты, что показало эффективность разработанных методов.

РЭФЕРАТ

Дыпломная праца, 49 с., 26 рыс., 8 табліц, 19 крыніц.

Ключавыя словы: АПРАЦОЎКА ТЭКСТАЎ, МАШЫННАЕ НАВУЧАННЕ, ЛАТЭНТНАЕ РАЗМЯШЧЭННЕ ДЫРЫХЛЕ, МЕТАДЫ ЗНІЖЭННЯ ПАМЕРНАСЦІ, ВЕКТАРНАЕ ПРАДСТАЎЛЕННЕ ТЭКСТАЎ, СТАТЫСТЫЧНЫ АНАЛІЗ, МЕТАДЫ КЛАСТАРЫЗАЦЫІ, НЕЙРОНАВЫЯ СЕТКІ.

Аб'ект даследавання — наборы дадзеных дзелавых электронных перапісак.

Мэта работы — аналіз дзелавых электронных перапісак метадамі машыннага навучання.

Метады даследавання — латэнтнае размяшчэнне Дырыхле, метады кластарызацыі, метады апрацоўкі тэкстаў, метады зніжэння памернасці, метады атрымання вектарных уяўленняў.

Праца прысвечана даследаванню і аналізу дзелавых электронных перапісак, у прыватнасці, перапісак Хілары Клінтан і перапісак супрацоўнікаў карпарацыі Enron. У выніку працы быў выраблены статыстычны аналіз электронных перапісак. Былі выяўлены заканамернасці ў зыходных дадзеных. Таксама была распрацавана кластарызацыя зместаў электронных перапісак, у выніку якой атрымаліся інтэрпрэтаваныя вынікі, што паказала эфектыўнасць распрацаваных метадаў.

ABSTRACT

Diploma thesis, 49 p., 26 fig., 8 tables, 19 sources.

Keywords: TEXT PROCESSING, MACHINE LEARNING, LATENT DIRICHLET ALLOCATION, DIMENSION REDUCTION METHODS, TEXT EMBEDDINGS, STATISTICAL ANALYSIS, CLUSTERIZATION METHODS, NEURAL NETWORKS.

The object of research is business e-mail datasets.

Objective: analysis of business e-mails using machine learning methods.

Research methods — latent Dirichlet allocation, clustering methods, text processing methods, dimension reduction methods, methods for obtaining text embeddings.

The work is devoted to the research and analysis of business e-mails, in particular, the e-mails of Hillary Clinton and the e-mails of employees of the Enron corporation. As a result of the work, a statistical analysis of emails was carried out. Patterns were found in the original data. Also, the clustering of the contents of e-mails was developed, as a result of which interpretable results were obtained, which showed the effectiveness of the developed methods.

Введение

С ростом доступности электронных документов и быстрым ростом всемирной паутины задача автоматической категоризации документов стала ключевым способом классификации и группирования информации и знаний любого рода. Для правильной классификации электронных документов, онлайн-новостей, блогов, электронной почты и электронных библиотек необходимы интеллектуальный анализ текста (англ. *Text Mining*), машинное обучение (англ. *Machine Learning*) и методы обработки текстов на естественном языке (англ. *Natural Language Processing, NLP*).

Современные системы обработки текстов на естественном языке могут анализировать неограниченные объемы текстовых данных. Они могут понимать суть сложных контекстов, расшифровывать двусмысленности языка, извлекать ключевые факты и взаимосвязи. Учитывая огромное количество неструктурированных данных, которые создаются каждый день, от электронных медицинских карт до сообщений в социальных сетях, обработка текстов на естественных языках стала критически важной для эффективного анализа текстовых данных. Интеллектуальный анализ текста выявляет факты, взаимосвязи и утверждения, которые в противном случае остались бы погребенными в массе текстовых больших данных. После извлечения эта информация преобразуется в структурированную форму, которая может быть дополнительно проанализирована или представлена непосредственно с помощью кластеризованных HTML-таблиц

В данной работе планируется провести анализ деловых электронных переписок с помощью методов машинного обучения. В частности, планируется произвести тематическое моделирование и кластеризацию содержания электронных писем с использованием методов машинного обучения, таких как латентное размещение Дирихле, метод k-средних и основанная на плотности пространственная иерархическая кластеризации для приложений с шумами и более новые алгоритмы тематического моделирования, которые были представлены буквально в прошлым году. Для каждой модели планируется провести оптимизацию гиперпараметров, чтобы добиться хороших результов. Также планируется тщательный и внимательный к деталям анализ писем и их метаданных с большим количеством визуализации. Однако, машинное обучение требует высокачественные данные, чтобы показывать хорошие результаты. Обычные переписки людей в свободной форме, конечно, не являются высокачественными данными. Поэтому этапу предварительной обработки текстов в данной работе будет уделяться много внимания.

Глава 1
Обзор наборов данных

1.1 Используемые данные

1.1.1 Электронные письма Хиллари Клинтон

В 2015 году Хиллари Клинтон (американский политик, государственный секретарь США в 2009-2013 гг., кандидат в президенты США в 2016 г.) была вовлечена в большое количество споров по поводу использования личных учетных записей электронной почты на негосударственных серверах во время ее пребывания на посту государственного секретаря США. Некоторые политические эксперты утверждают, что использование Клинтон личных учетных записей электронной почты для ведения дел госсекретаря является нарушением протоколов и федеральных законов, обеспечивающих надлежащий учет деятельности правительства.

Был подан ряд исков о свободе информации из-за того, что Государственный департамент США не опубликовал полностью электронные письма, отправленные и полученные на личные аккаунты Клинтон.

В июле расследование ФБР пришло к выводу, что никакой «разумный прокурор» не будет возбуждать уголовное дело против госпожи Клинтон, но что она и ее помощники «крайне небрежно» обращались с секретной информацией. Затем ФБР удивило всех, за 11 дней до выборов, объявив, что изучает недавно обнаруженные электронные письма, отправленные или полученные Хиллари Клинтон. Как позже выяснилось, она установила адреса электронной почты на государственном сервере для своего давнего помощника Хумы Абедина и начальника штаба Госдепартамента Шерил Миллс. На сегодняшний день Государственным департаментом США опубликовано почти 7000 страниц отредактированных электронных писем Клинтон.

Документы были опубликованы в формате *pdf*. Платформа *Kaggle* очистила и нормализовала выпущенные документы и разместила их для публичного анализа [1]. Мы будем основываться именно на датасете, опубликованном *Kaggle*.

1.1.2 Электронные письма корпорации Enron

Enron являлась государственной корпорацией штата Орегон со штабквартирой в городе Хьюстон. До объявления о банкротстве в декабре 2001 года *Enron* была седьмой по величине корпорацией в США. В феврале 2002 года Федеральная комиссия по регулированию в области энергетики (*Federal Energy Regulatory Commission, FERC*) начала всестороннее расследование торговой деятельности *Enron* на рынках электроэнергии Калифорнии. Со-

гласно *FERC*, *Enron* получила некоторую информацию о рынке, недоступную для ее конкурентов. Прибыль *Enron* превысила 500 миллионов долларов в 2000 и 2001 годах. Расследование пришло к выводу, что многие торговые стратегии, используемые *Enron*, нарушают рыночные отношения, утвержденные Федеральной комисиией для Калифорнии. С июня 2002 года Министерство юстиции США возбудило уголовные дела против 30 человек, включая Джеффри Скиллинга, бывшего президента и генерального директора *Enron*, а также других руководителей высшего звена. Обвинения включали сговор, мошенничество с ценными бумагами и инсайдерскую торговлю.

Исходный набор данных *Enron* был обнародован и размещен в Интернете Федеральной комиссией во время расследования энергетического кризиса в Западной Европе 2000-2001 годов. Позднее набор данных был приобретен Лесли Кельблинг из Массачусетского технологического института, где было выявлено несколько проблем с целостностью данных. Вскоре после этого группа исследователей из *SRI International*, некоммерческой корпорации, основанной Стэнфордским университетом, во главе с Мелиндой Гарвасио провела серьезную очистку и удаление вложений и отправила ее профессору Уильяму Коэну из Университета Карнеги-Меллона, который сделал публикацию на своей веб-странице. В документе с анализом базы данных *Enron*, представленном на конференции 2004 года, делается вывод о том, что набор данных *Enron* «подходит для оценки методов классификации электронной почты».

Используемый набор данных состоит из архивных электронных писем сотрудников *Enron*, в основном руководителей высшего звена и трейдеров, из которых были удалены вложения. Также некоторые из писем были удалены по требованию сотрудников корпорации. Общее количество писем – более полумиллиона.

1.2 Исследования, проведенные над наборами данных

1.2.1 Исследования над письмами Хиллари Клинтон

Анализ, проведенный центром Беркмана Кляйна по интернету и обществу в Гарвардском университете и центром Шоренштейна в Гарвардской школе Кеннеди, показывает, что споры по поводу электронной почты Клинтона получили больше освещения в основных средствах массовой информации, чем любая другая тема во время президентских выборов США 2016 года [2].

Например, исследование профессора Уэйн Олдфорда и студента бакалавра университета Уотерлу построили инструмент для анализа содержимого писем [3].

Инструмент предоставляет визуальные аналитические инструменты и демонстрирует, как много можно узнать о человеке из того, что ошибочно счи-

тается неинформативными метаданными. По словам Олдфорда, «общественность может воспроизвести наши анализы и сама увидеть, как они могут быть показательными, особенно в сочетании с другими общедоступными источниками».

Например, приложение показывает ежедневный объём электронной почты, отправленной и полученной Клинтон, а также слова, наиболее часто встречающиеся в электронных письмах за выбранный период времени.

Например, при анализе данных исследователи обнаружили 10 периодов отсутствия электронных писем от Клинтон, пока она была госсекретарем, включая значительный разрыв между 30 октября и 9 ноября 2012 года, который совпадает с первоначальным расследованием нападения в Бенгази и его последствий, а также президентских выборов в США в 2012 году.

Существует большое количество исследований электронных писем Клинтон, выполненные пользовалями *Kaggle*. К сожалению, качество данных работ не является высоким, поскольку они, во-первых, не являются большими проектами, а представляют собой скорее любительские исследования, а во-вторых, при этом часто выполнены с недостатками, например, с плохой предобработкой исходных данных.

1.2.2 Исследования над письмами корпорации Enron

По крайней мере два исследования, касающихся классификации текстов, были выполнены на наборе данных *Enron*. Один из них — автоматическая категоризация электронной почты по папкам, выполняемая факультетом компьютерных наук Массачусетского университета [4]. Другой связан с анализом социальных сетей. Используя корпус *Enron* и судебные документы, выпущенные судом США по делам о банкротстве, Джитеш Шетти и Джафар Адиби извлекли из электронной почты социальную сеть, состоящую из 151 сотрудника, соединив людей, которые обменивались электронными письмами [5].

Кроме того, команда *Legal Track* конференции по извлечению текстов («*Text Retrieval Conference Legal Track*») фокусировалась на методах крупномасштабного поиска текста. Команда ученых изучала следующие методы поиска: булевы, нечеткие модели поиска, вероятностные (байесовские) модели, статистические методы, подходы машинного обучения, инструменты категоризации и анализ социальных сетей. Исследователи *TREC Legal Track* пришли к выводу, что «всего от 22 до 57 процентов релевантных документов могут быть извлечены с помощью различных альтернативных методов поиска».

В статье Шетти и Адиби набор данных *Enron* использовался для классификации электронной почты на основе моделирования энтропии графа. Энтропия пыталась выбрать наиболее интересные вершины в графе, вершины которого представляют электронные письма, а ребра — сообщения между пользователями [6].

Что касается непосредственно тематического моделирования и кластеризации, проводимых в данной работе, основная масса исследований представлена небольшими проектами, выполненными пользователями *Kaggle*. Как и в случае с электронными письмами Клинтон, в силу аналогичных причин, качество данных работ не является высоким.

Глава 2
Обзор используемых технологий

2.1 Тематическое моделирование

Тематическое моделирование — это метод классификации документов, аналогичный кластеризации по числовым данным, который находит некоторые естественные группы элементов (темы), даже в случаях, когда пользователь не имеет конкретной цели касательно нахождения определенных тем.

Документ может быть частью нескольких тем, как в нечеткой (мягкой) кластеризации, в которой каждый элемент данных принадлежит более чем одному кластеру.

Тематическое моделирование предоставляет методы для автоматической организации, понимания, поиска и обобщения больших электронных архивов. Это может помочь в следующих случаях:

- обнаружение скрытых (неочевидных) тем в корпусе (т.е. общем наборе слов) данных,
- классификация документов по обнаруженным темам,
- использование классификации для, собственно, организации, обобщения либо поиска интересующих документов.

Например, предположим, что документ относится к темам *еда*, *собаки* и *здоровье*. Таким образом, если пользователь запрашивает «*корм для собак*», вышеупомянутый документ может определиться как релевантный, поскольку, помимо других тем, он охватывает и эти темы. Другими словами, его релевантность по отношению к запросу может быть выяснена даже без просмотра всего документа, а только на основании уже известных тем.

Получается, аннотируя документ на основе тем, предсказанных методом моделирования, становится возможно оптимизировать выполняемый процесс поиска.

2.2 Латентное размещение Дирихле

Латентное размещение Дирихле (*Latent Dirichlet Allocation*, *LDA*) — один из самых популярных методов тематического моделирования. Каждый документ состоит из разных слов, и к каждой теме также относятся разные слова. Цель *LDA* — найти темы, к которым принадлежит документ, на основе содержащихся в нем слов [7].

2.2.1 Определение модели

Допустим, имеется несколько документов, каждый из которых содержит присутствующие в нем слова, отсортированные по частоте встречаемости. Целью метода является определение соответствий между словами и темами. Это можно отобразить, например, с помощью таблицы ниже. Каждая строка в таблице представляет отдельную тему, а каждый столбец — отдельное слово в корпусе. Каждая ячейка содержит вероятность того, что слово (столбец) принадлежит теме (строке).

	$слово_1$	$слово_2$	$слово_3$	$слово_4$	...
$тема_1$	0.01	0.23	0.19	0.03	
$тема_2$	0.21	0.07	0.48	0.02	
$тема_3$	0.53	0.01	0.17	0.04	
...					

Таблица 2.1 — Пример представления модели латентного размещения Дирихле

2.2.2 Поиск слов, репрезентативных по отношению теме

Чтобы описать, какие слова описывают тему, можно поступить одним из следующих способов.

- Сортировка слов в зависимости от их вероятностей.

  Из каждой темы выбирается какое-то количество лучших слов для представления темы. Этот шаг не всегда может быть необходим, потому что, если корпус небольшой, можно хранить все слова, отсортированные по их вероятностям.

- Установление *порога* вероятностей.

  Все слова в теме, получившие вероятность выше порогового значения, могут быть использованы как ее представители (в порядке сортировки).

2.2.3 Принятые предположения

- Каждый документ — это просто набор слов. Порядок слов и грамматическая роль слов (субъект, объект, глаголы и т.д.) в модели не учитываются.

- Такие слова, которыми, к примеру, в английском языке, являются *am*, *is*, *are*, *of*, *a*, *the*, *but* и др. не несут никакой информации о темах и поэтому могут быть удалены из документов на этапе предварительной

обработки. Фактически, мы можем удалить слова, которые встречаются как минимум в $80 - 90\%$ документов, не теряя при этом в итоговом результате.

Например, если корпус содержит только медицинские документы, такие слова, как *человек*, *тело*, *здоровье* могут присутствовать в большинстве документов и, следовательно, могут быть удалены, поскольку они не добавляют никакой конкретной информации, которая описывала бы документ.

- Заранее известно, на какое количество тем будет происходить распределение слов.

- На каждом шаге алгоритма, при рассмотрении текущего слова, распределения всех предыдущих слов на темы верны, а обновление новым словом происходит с использованием уже имеющейся текущей модели.

2.2.4 Алгоритм

- Взять каждый документ и случайным образом определить каждому слову в документе одну из k тем (напомним, что k выбирается заранее).

- Для каждого документа d взять каждое слово w и вычислить:

  1. $p(\text{тема } t \mid \text{документ } d)$ — долю слов в документе d, которые относятся к теме t. С помощью этого значения производится попытка определить, сколько слов принадлежит теме t для данного документа d, исключая текущее слово w. Чем больше слов из d принадлежит t, тем более вероятно, что слово w принадлежит t.

  2. $p(\text{слово } w \mid \text{тема } t)$ — долю распределений темы t по всем документам, которые содержат слово w. С помощью этого значения производится попытка определить, сколько документов относятся к теме t, опираясь на слово w.

     LDA представляет документы как неупородяченный набор тем. Точно так же тема — это неупорядоченный слов. Если слово имеет высокую вероятность отношения к теме, все документы, содержащие w, также будут более прочно связаны с t. Точно так же, если w относится к t с небольшой вероятностью, документы, содержащие w, будут иметь очень низкую вероятность отношения к t, потому что остальные слова в d будут принадлежать какой-то другой теме и, следовательно, d будет иметь более высокую вероятность по отношению к той теме. Таким образом, даже если w будет добавлено к t, оно не принесет много похожих документов к t.

- Обновить вероятность приналежности слова w теме t следующим образом:

$$p(w \in t) = p(\text{тема } t \mid \text{документ } d) \cdot p(\text{слово } w \mid \text{тема } t)$$

2.3 Кластеризация

В общих чертах, цель кластеризации — найти различные группы внутри элементов данных. Для этого алгоритмы кластеризации находят такую структуру данных, чтобы элементы одного и того же кластера (или группы) были более похожи друг на друга, чем на элементы из разных кластеров. Для примера можно представить, что имеется набор данных фильмов и мы хотим их кластеризовать:

Рисунок 2.1 — Пример работы кластеризации

Модель машинного обучения сможет сделать вывод о существовании двух разных классов, ничего не зная о природе данных.

Эти алгоритмы т.н. обучения без учителя имеют невероятно широкий спектр приложений и весьма полезны для решения реальных проблем, таких как обнаружение аномалий, рекомендации систем, группировка документов или поиск клиентов с общими интересами на основе их покупок.

В данной работе мы группируем похожие по смыслу слова с помощью векторного представления слов, полученных с помощью *Word2Vec* [8].

Word2Vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он генерирует словарь корпуса, а затем вычисляет векторное представление слов, «обучаясь» на входных текстах. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по косинусному расстоянию) векторы. Полученные векторные представления слов могут быть использованы для обработки естественного языка и машинного обучения.

2.4 Метод k-средних

Алгоритм k-средних нацелен на поиск и группировку в классы точек данных, которые имеют большое сходство между собой. В терминах алгоритма это сходство понимается как противоположность расстояния между точками данных. Чем ближе точки данных, тем больше они будут похожи и с большей вероятностью принадлежат одному кластеру [9].

Алгоритм k-средних чрезвычайно прост в реализации и очень эффективен с точки зрения вычислений. Это основные причины, по которым он так популярен. Однако они не очень хороши для идентификации классов при работе с группами, не имеющими сферической формы распределения.

2.4.1 Ключевые идеи

- Квадрат евклидова расстояния

  Наиболее часто используемое расстояние в методе k-средних — это квадрат евклидова расстояния. Пример расстояния между двумя точками x и y в m-мерном пространстве:

  $$d(x, y)^2 = \sum_{j=1}^{m} (x_j - y_j)^2 = ||x - y||_2^2$$

  Здесь индекс j — это j-я координата точек x и y.

- Кластерная инерция

  Кластерная инерция — это имя, данное сумме квадратов ошибок (*Sum of Squared Errors*, *SSE*) в контексте кластеризации, которое представляется следующим образом:

  $$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \cdot ||x^{(i)} - \mu^{(j)}||_2^2$$

  Где $\mu^{(j)}$ — это центроид (среднее значение) для кластера j, а $w^{(i,j)}$ равно 1, если $x^{(i)}$ находится в кластере j, и 0 в противном случае.

  Метод k-средних можно понимать как алгоритм, который пытается минимизировать инерции кластеров.

2.4.2 Алгоритм

1. Сперва нужно выбрать k — количество кластеров, которые мы хотим найти.

2. Затем алгоритм случайным образом выберет центроиды каждого кластера.

3. Для каждой точки будет вычислен выбран ближайший центроид (с использованием евклидова расстояния).

4. Вычисляется инерция кластера.

5. Новые центроиды будут вычисляться как среднее значение точек, принадлежащих центроиду предыдущего шага.

6. Вернуться к шагу 3.

2.4.3 Гиперпараметры алгоритма

- Количество кластеров: количество создаваемых кластеров и центроидов.

- Максимальное количество итераций: количество итераций, которое будет пытаться совершить алгоритм за один запуск.

- Количество случайных запусков: количество запусков алгоритма с разными начальными значениями центроидов. Конечным результатом работы алгоритма будет лучший (с точки зрения инерции) из всех запусков.

2.4.4 Проблемы алгоритма

- Выходные данные работы алгоритма не всегда будут одинаковыми, потому что начальные центроиды устанавливаются случайным образом, соответственно, это будет влиять на весь процесс алгоритма.

- Как было сказано ранее, из-за природы евклидова расстояния этот алгоритм не подходит для работы с кластерами, которые принимают несферические формы.

2.5 BERTopic

BERTopic — это метод моделирования тем, который использует трансформеры и *c-TF-IDF* для создания плотных кластеров, позволяющих легко интерпретировать темы, сохраняя при этом важные слова в описаниях тем [10].

2.5.1 Алгоритм

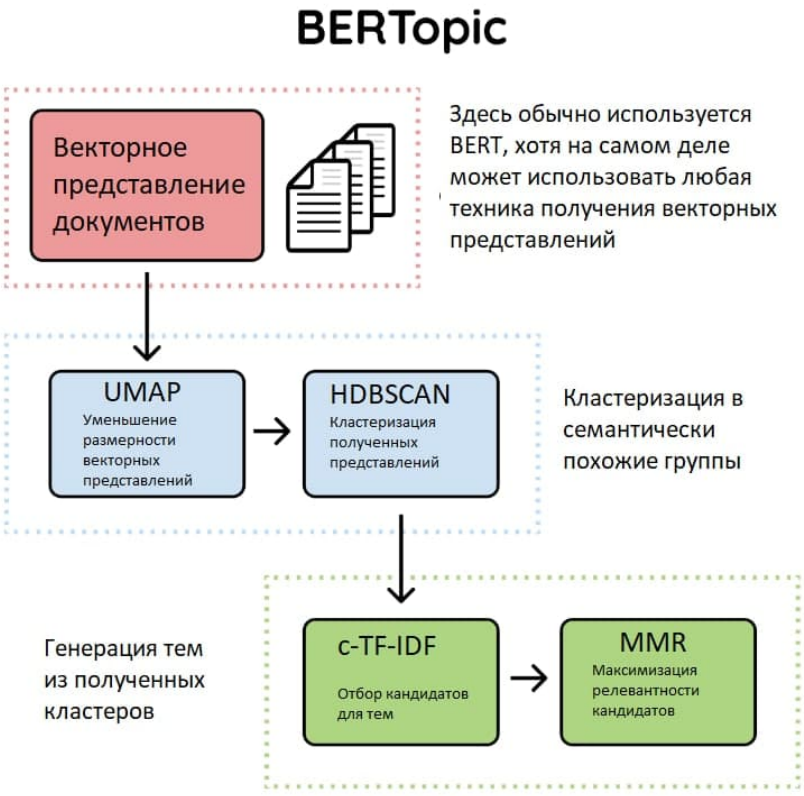

Рисунок 2.2 — Стадии работы алгоритма *BERTopic*

Алгоритм состоит из 3 шагов:

1. Первый шаг, который требуется сделать – это преобразовать документы в числовые данные. Для этой цели используется *BERT*, который извлекает различные векторные представления слов в зависимости от контекста. Это одна из лучших моделей в настоящее время для многих задач, связанных с текстом. *BERT* получил награду за лучшую работу на ежегодной конференции североамериканского отделения компьютерной лингвистики 2019 года [11] [12].

2. Далее необходимо убедиться, что документы с похожим смыслом сгруппированы вместе, чтобы мы могли найти темы в этих кластерах. Перед этим нам сначала нужно снизить размерность векторных представлений слов, поскольку многие алгоритмы кластеризации плохо справляются с высокой размерностью. *UMAP* — это один из немногих алгоритмов уменьшения размерности, он является наиболее эффективным, поскольку он сохраняет значительную часть многомерной локальной структуры в более низкой размерности.

   После уменьшения размерности встраиваемых документов мы можем кластеризовать документы с помощью *HDBSCAN* (*Hierarchical Density-*

Based Spatial Clustering of Applications with Noise). *HDBSCAN* основан на алгоритме *DBSCAN* и, как и другие алгоритмы кластеризации, используется для группировки данных [13].

Помимо того, что он обычно показывает лучшее качество, он также быстрее, чем обычный *DBSCAN*. Ниже приведен график нескольких алгоритмов кластеризации. При отметке в 200 000 объектов *DBSCAN* занимает примерно вдвое больше времени, чем *HDBSCAN*. Стоит отметить, что по мере увеличения количества объектов разница в производительности будет и дальше увеличиться в пользу *HDBSCAN*:

Рисунок 2.3 — Сравнение производительности алгоритмов кластеризации

HDBSCAN — алгоритм кластеризации, который довольно хорошо работает с *UMAP*, поскольку *UMAP* поддерживает большую локальную структуру даже в пространстве меньшей размерности. Более того, *HDBSCAN* не переносит отдельные точки в кластеры, поскольку считает их выбросами.

3. Теперь мы сгруппировали похожие документы вместе, которые должны представлять темы, из которых они состоят. Что мы хотим узнать из созданных нами кластеров — это то, что отличает один кластер по своему содержанию от другого. Как мы можем извлечь темы из сгруппированных документов? Чтобы решить эту проблему, используется классовый вариант *TF-IDF* (*c-TF-IDF*), который позволил бы извлечь то, что делает каждый набор документов уникальным по сравнению с другим.

Интуиция, лежащая в основе метода, заключается в следующем: когда мы применяем *TF-IDF* как обычно к набору документов, мы в сравниваем важность слов среди всех документов, а в классовом варианте теперь у нас есть одно значение важности для каждого слова в кластере, которое можно использовать для создания темы. Если мы возьмем несколько самых важных слов в каждом кластере, то получим хорошее представление о кластере и, следовательно, теме.

Чтобы создать эту оценку классового *TF-IDF*, сначала нужно создать один документ для каждого кластера документов. Затем мы применяем *TF-IDF* на основе классов:

$$c\text{-}TF\text{-}IDF_i = \frac{t_i}{w_i} \cdot \log \frac{m}{\sum_j^n t_j}$$

Где частота каждого слова t извлекается для каждого класса i и делится на общее количество слов w в классе. Это действие можно рассматривать как форму регуляризации частых слов в классе. Затем общее количество документов m делится на общую частоту слова t по всем n классам.

Теперь у нас есть одно значение важности для каждого слова в кластере, которое можно использовать для создания темы. После обучения нашей модели мы можем итеративно пройти, возможно, сотню тем, чтобы получить хорошее представление о темах, которые были извлечены. Однако это занимает некоторое время и не имеет глобального представления. Вместо этого мы можем визуализировать темы.

2.5.2 Интерпретация работы алгоритма

Для визуализации работы алгоритмы используется представление тем в *2D* с помощью *UMAP*, который создает двумерную проекцию всех точек и затем визуализирует эти два измерения, причем в интерактивном виде, что позволяет нам получить представление, понятное человеку. Пример визуализации от авторов алгоритма [14]:

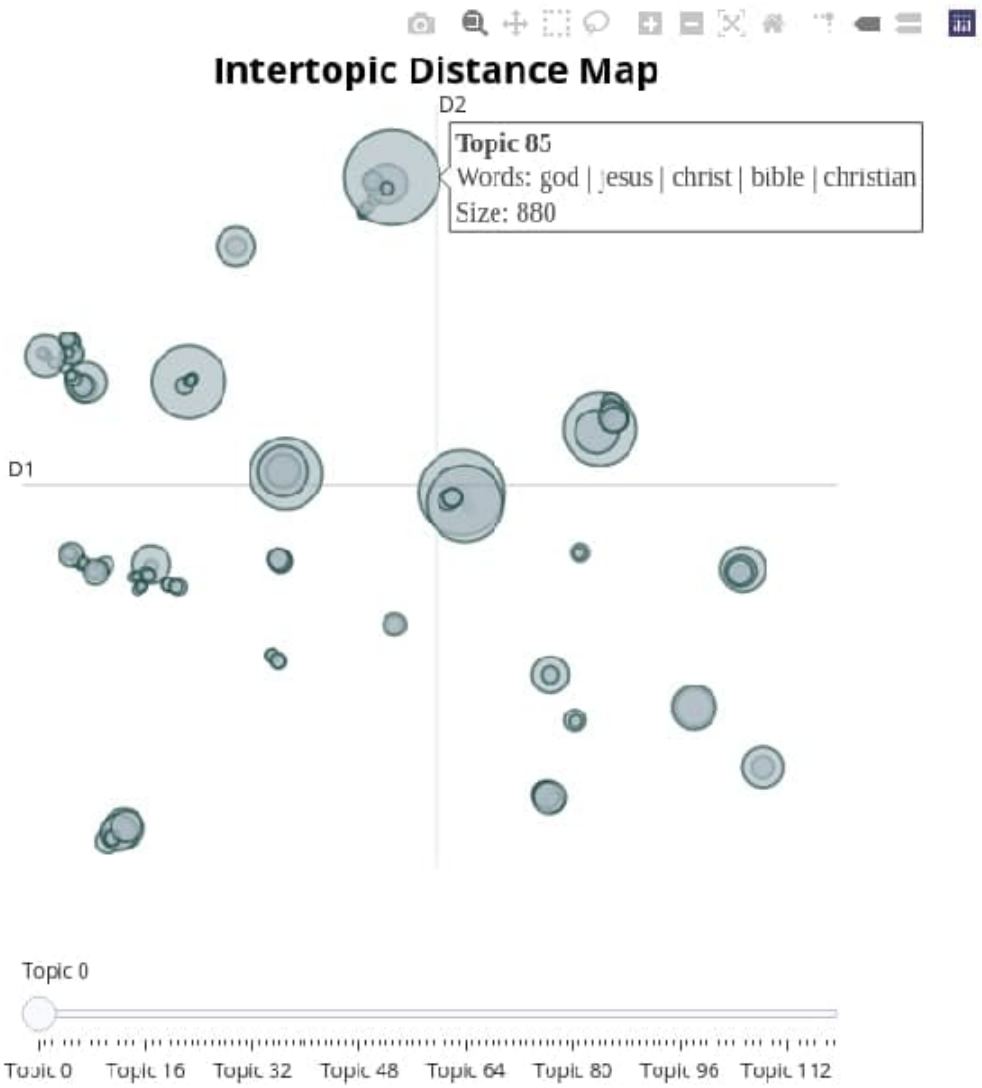

Рисунок 2.4 — Пример визуализации тем в *BERTopic*

На картинке видно, что кластеры довольно равномерно распределились по пространству и кластера действительно очень интерпремируемы: например, на рисунке показан кластер религиозных слов.

Мы также можем рассчитать вероятность того, что темы могут быть найдены в документе. Эти вероятности означают, насколько *BERTopic* уверен в том, что определенные темы могут быть найдены в документе [14]:

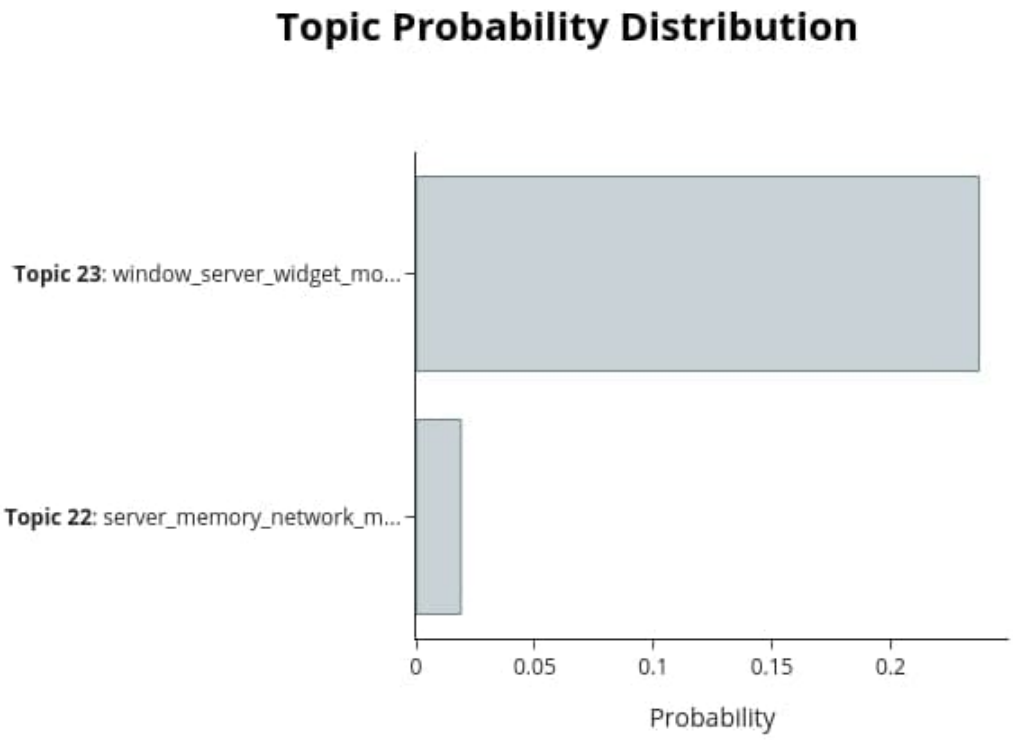

Рисунок 2.5 — Пример распределения вероятностей тем в *BERTopic*

Важно понимать, что распределение вероятностей не указывает на распределение частотности тем в документе. Это просто показывает, насколько *BERTopic* уверен в том, что в документе можно найти определенные темы.

Глава 3
Предобработка данных

Предварительная обработка данных является неотъемлемым шагом в машинном обучении, поскольку качество данных и полезная информация, которая может быть получена из них, напрямую влияют на способность нашей модели к обучению, поэтому чрезвычайно важно, чтобы мы предварительно обработали наши данные, прежде чем вводить их в нашу модель.

3.1 Предобработка электронных писем Хиллари Клинтон

В отличие от набора данных корпорации *Enron*, набор данных Клинтон не содержит метаданных письма — всё уже разделено на дату отправки письма, информацию о получателях и так далее, поэтому в этом случае нам будет немного проще.

Посмотрим на пример содержания письма, до его обработки:

```
Nice\nForgot to tell you about our harrowing circling
and attempted landings last night ...
```

Этап предварительном обработки данных можно разбить на 7 шагов:

1. Для начала нужно содержание каждого документа (он может быть формата *docx*, *pdf* и так далее) представить в виде последовательности слов (или нескольких последовательностей), чтобы далее можно было применять методы машинного обучения. Этот шаг уже произведен платформой *Kaggle*. Общее количество электронных писем — 7945.

2. Может оказаться, что исходные документы были плохого качества (например, «битые» *pdf*-файлы, или же версия документа давно уже не поддерживается). Поэтому в данных у нас могут встречаться пропуски. Такие данные мы просто удаляем из датасета. После осуществления этого шага остается 6742 писем.

3. Следующим шагом идёт нормализация, то есть удаляются знаки препинания, подряд идущие пробелы, переносы строк, выделяются конкретные слова. Это помогает уменьшить количество различной информации, с которой приходится иметь дело компьютеру, и, следовательно, повышает качество и эффективность алгоритмов. Для этого была использована библиотека *nltk* [15], содердащая большое количество инструментов для обработки текстов.

4. Далее каждые слово приводится к нижнему регистру. Это делается для чтобы, например, слова *Hello* и *hello* модель воспринимала как одно слово, ведь по смыслу они означают одно и то же.

5. Потом удаляются стоп-слова — слова в языке, которые имеют малое смысловое значение (местоимения, предлоги и так далее). Они встречаются в изобилии, поэтому практически не предоставляют уникальной информации, которая может быть использована для классификации или кластеризации данных.

6. Следующий шаг — лемматизация. При обработке естественного языка может наступить момент, когда вы захотите, чтобы программа распознала, что слова *ask* и *asked* — это просто разные времена одного и того же глагола. Это идея сведения различных форм слова к основному корню. Такая процедура заметно уменьшает количество уникальных слов в тексте, что является большим плюсом, также модель будет понимать, что это одно и то же слово, что заметно повышает качество работы многих моделей машинного обучения. Для реализации такой процедуры была использовано библиотека *spaCy* [16]. На данный момент почти всё готово к тому, чтобы давать полученные данные на вход модели.

7. Последним шагом является индексация слов. Нужно слова привести в формат, понятный компьютеру, и над полученными представлениями проводить некоторые арифметические операции. Для этого каждому слову достаточно сопоставить уникальный числовой идентификатор.

Посмотрим на это же письмо после обработки (не включая этап индексации):

```
nice forget tell harrow circle attempt landing last night
```

Видно, что все этапы прошли успешно. Теперь всё готово к тому, чтобы данные передавать на вход модели.

3.2 Предобработка электронных писем корпорации Enron

3.2.1 Выделение метаданных из сырого текста писем

Набор данных *Enron* также требует предварительной обработки. К примеру, так выглядит необработная информация одного электронного письма:

```
Message-ID: <18782981.1075855378110.JavaMail.evans@thyme>
Date: Mon, 14 May 2001 16:39:00 -0700 (PDT)
From: phillip.allen@enron.com
```

```
To: tim.belden@enron.com
Subject:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: Tim Belden <Tim Belden/Enron@EnronXGate>
X-cc:
X-bcc:
X-Folder: \\Phillip_Allen_Jan2002_1\\Allen, Phillip K.\\\'Sent Mail
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst

Here is our forecast
```

Конечно, анализировать данные (в том числе метаинформацию о письме) в таком формате бессмысленно. Для обработки мы будем использовать библиотеку *email* [17]. Данная библиотека позволяет из сырых данных выделить вспомогательную информацию о письме, в частности, библиотека позволяет выделить следующие интересные нам атрибуты:

- полное содержание письма,
- дата отправки,
- адрес получателя,
- адрес отправителя,
- тема письма,
- логин отправителя.

3.2.2 Выделение содержания писем

После этого требуется также привести содержание письма в приемлимый для дальнейшего обучения вид. Например, для письма с содержанием ниже мы хотим выделить только единицы, имеющие отношение к сути письма.

```
Forwarded by Phillip K Allen/HOU/ECT on 09/12/2000 11:22 AM

Michael Etringer

09/11/2000 02:32 PM
```

```
To: Phillip K Allen/HOU/ECT@ECT
cc:
Subject: Contact list for mid market

Phillip,
Attached is the list. Have your people fill in the columns
highlighted in yellow. As best can we will try not to overlap
on accounts.
Thanks, Mike
```

Выделение этих единиц происходит в соответствии со следующей последовательности шагов:

1. Перевод всех символов в нижний регистр.
2. Удаление всех слов, содержащих цифры. Такие слова не несут смысловой нагрузки и, соответственно, влияют на качество обучения в худшую сторону (подавляющее большинство таких слов было получено отправителями по ошибке).
3. Удаление единиц, соответствующих информации о пересланных письмах.
4. Удаление единиц, соответствующих информации о вложениях в письме.
5. Удаление единиц, соответствующих почтовым адресам, присутствующим в тексте писем.
6. Удаление единиц, соответствующих корпоративным именам пользователей.
7. Удаление единиц, соответствующих ссылкам в сети Интернет.
8. Удаление единиц, не несущих смысловой составляющей, в заголовке письма.

Шаги 3-7 выполняются с использованием регулярных выражений. Регулярные выражения являются стандартом сопоставления шаблонов для анализа и замены строк. В языке программирования *Python* регулярные выражения реализованы в библиотеке *re*, с помощью неё получилось довольно компактно и быстро «почистить» электронные письма:

```
# text inside <>, some forwarded info
text = re.sub('<.*>', '', text)
```

```
# text inside [], some attachements
text = re.sub('\[.*\]', '', text)

# remove email addresses
text = re.sub('\S*@\S*\s?', '', text)

# some corporation usernames
text = re.sub('\S*/hou/\S*s?', '', text)

# remove links
text = re.sub('http[s]?://\S+', '', text)

# remove links
text = re.sub('www\.\S+', '', text)
```

Далее нужно выделить основной текст письма (часть после `Subject:`), это можно сделать при помощи соответствующего регулярного выражения:

```
text = re.sub('.*subject:', '', text)
```

Однако, используя библиотеку *re*, такой вариант регулярного выражения работает слишком долго, поэтому пришлось реализовать его самостоятельно:

```
msg_text = 'subject:'
idx = text.find(msg_text)
if idx != -1:
    text = text[idx + len(msg_text):].strip()
```

Такой вариант на имеющихся данных работает в десятки раз быстрее. Таким образом, теперь всё готово для быстрой обработки текста.

В результате, для примера выше, дальнейшая работа будет производиться со следующим текстом:

```
contact list for mid market. phillip, attached is the list.
have your people fill in the columns highlighted in yellow.
as best can we will try not to overlap on accounts. thanks, mike'
```

Глава 4
Предварительный анализ электронных писем

Прежде чем приступать к исследованию данных методами машинного обучения, может быть полезно посмотреть на различного статистики.

4.1 Анализ электронных писем Хиллари Клинтон

4.1.1 Длины слов и писем

- На основании содержаний писем можно посчитать различные статистики, например, распределение длин слов по письмам:

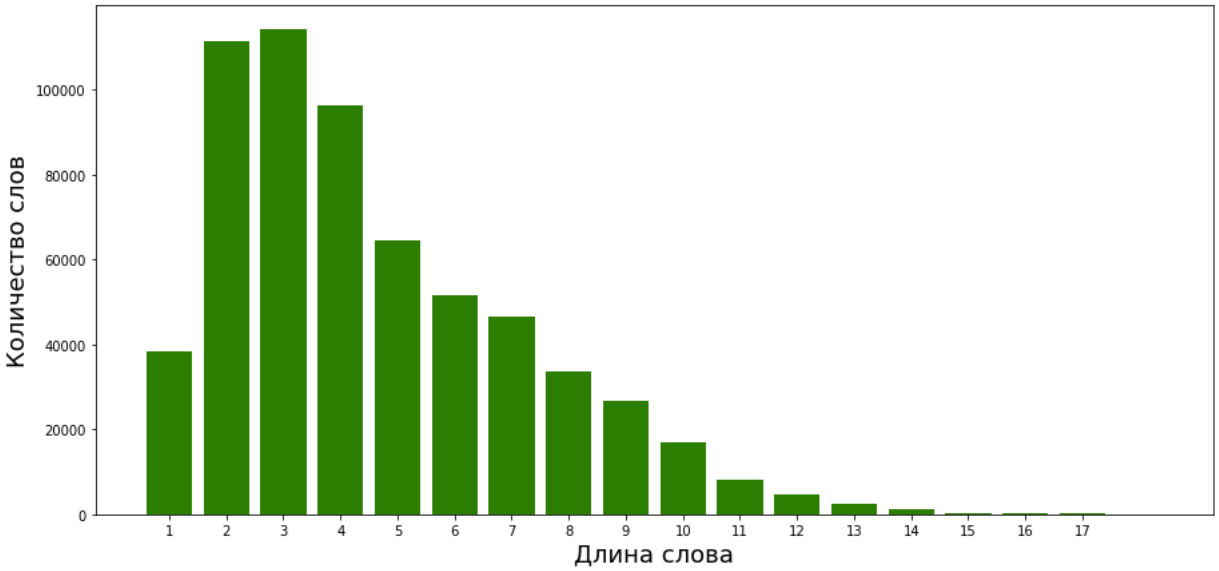

Рисунок 4.1 — Гистограмма распределения длин слов в наборе Клинтон

  Гистограмма выглядит вполне естественным образом, много коротких слов, что естественно для английского языка.

- Аналогично мы можем посчитать распределения длин писем (по количеству слов):

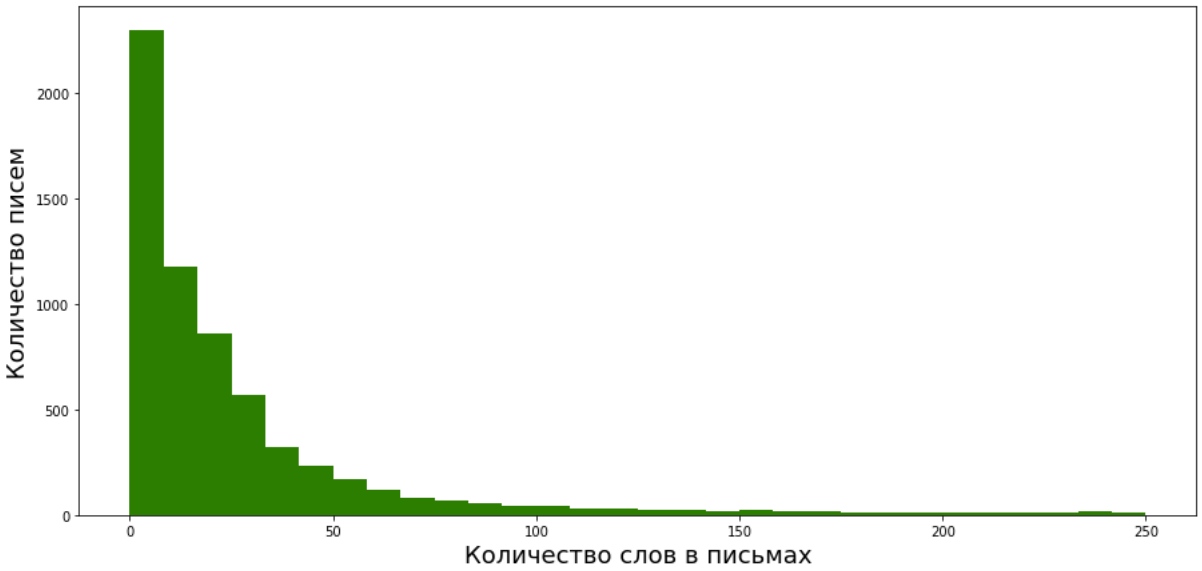

Рисунок 4.2 — Распределение количества слов в письмах в наборе Клинтон

Гистограмма соответствует интуитивным ожиданиям – более длинные письма пишутся реже.

4.1.2 Время отправки писем

- В метаданных писем содержится информация об дате отправки писем, на основании этого можно проанализировать, например, в какие года были отправлены письма:

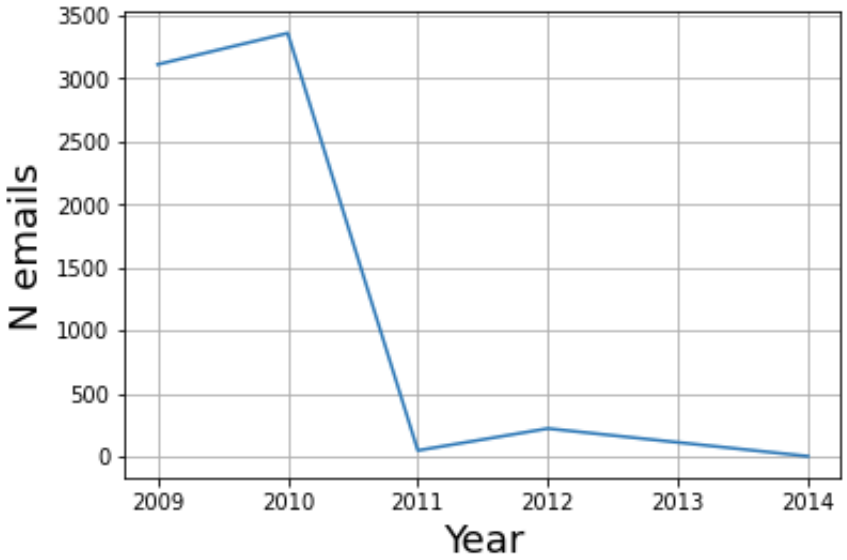

Рисунок 4.3 — График количества отправленных писем по годам в наборе Клинтон

На графике можно заметить странную аномалию с нулем писем в 2011

году. Вероятнее всего, это связано с особенностями набора данных.

- Аналогично можно проанализировать дни недели отправки писем:

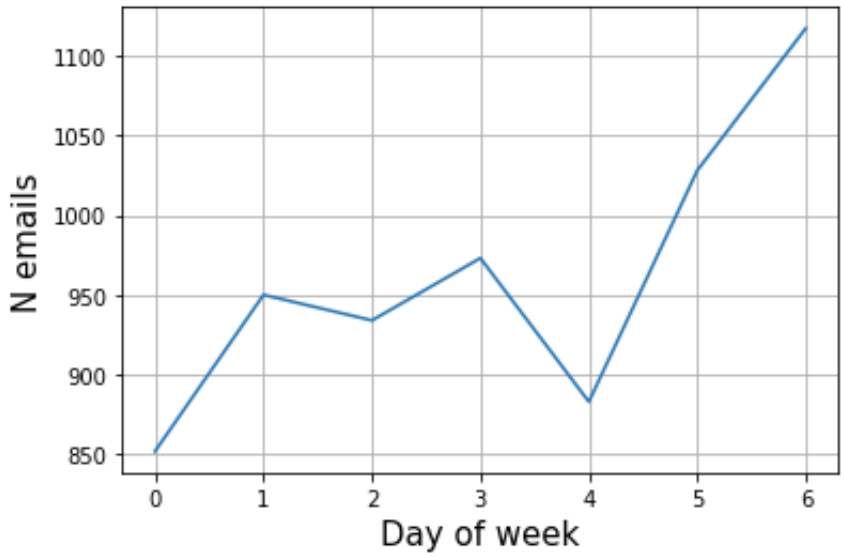

Рисунок 4.4 — График количества отправленных писем по дням недели в наборе Клинтон

График выглядит слегка неестественно (в отличие от *Enron*). Можно попытаться интерпретировать это как особенности одного отдельного человека, занимающего специфичным видом деятельности.

4.2 Анализ электронных писем корпорации Enron

4.2.1 Длины писем

Аналогично тому, как было сделано для набора Клинтон, можно проанализировать распределение длин писем (по количеству слов). Гистограмма, как и в случае писем Клинтон, соответствует интуитивным ожиданиям – более длинные письма пишутся реже:

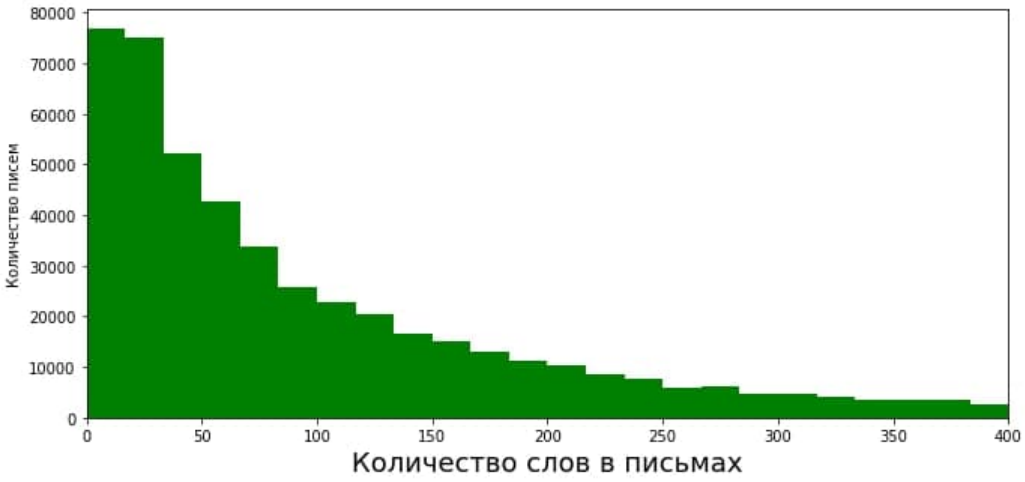

Рисунок 4.5 — Распределение количества слов в письмах в наборе *Enron*

4.2.2 Время отправки писем

- Как и в наборе Клинтон, в метаданных писем содержится информация об дате отправки писем, поэтому можно проанализировать различную полезную информацию, например в какие года были отправлены письма:

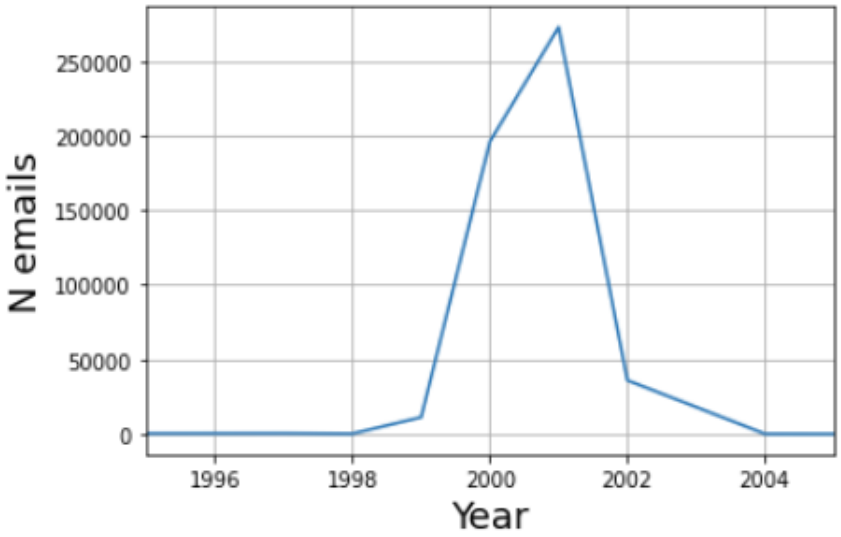

Рисунок 4.6 — График количества отправленных писем по годам в наборе *Enron*

  График соответствует наибольшей активности компании в 2000-2001 годах и банкротству к концу 2001 года.

- Аналогично можно проанализировать дни недели отправки писем:

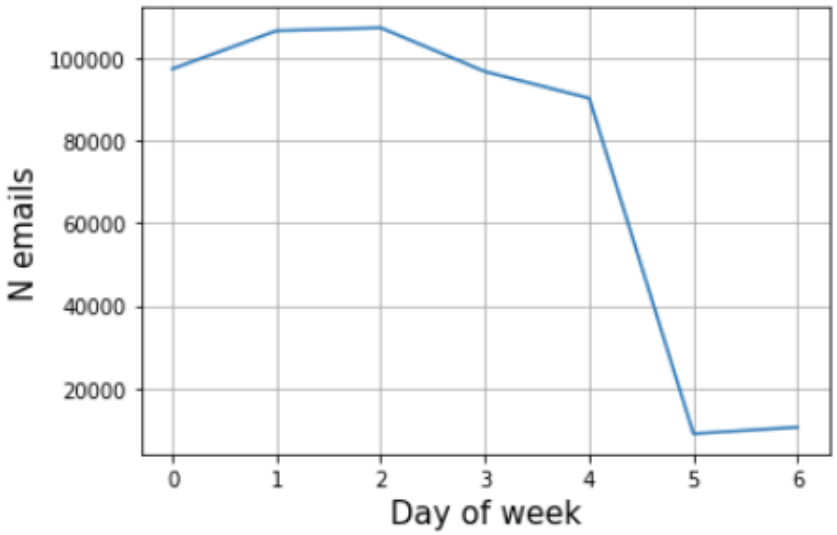

Рисунок 4.7 — График количества отправленных писем по дням недели в наборе *Enron*

Этот график также выглядит естественно — наибольшее число писем во вторник и среду, в середине рабочей недели, наименьшее — в выходные дни.

- В отличие от набора Клинтон, в наборе *Enron* дополнительно содержится информация о том, в какое время суток было отправлено письмо:

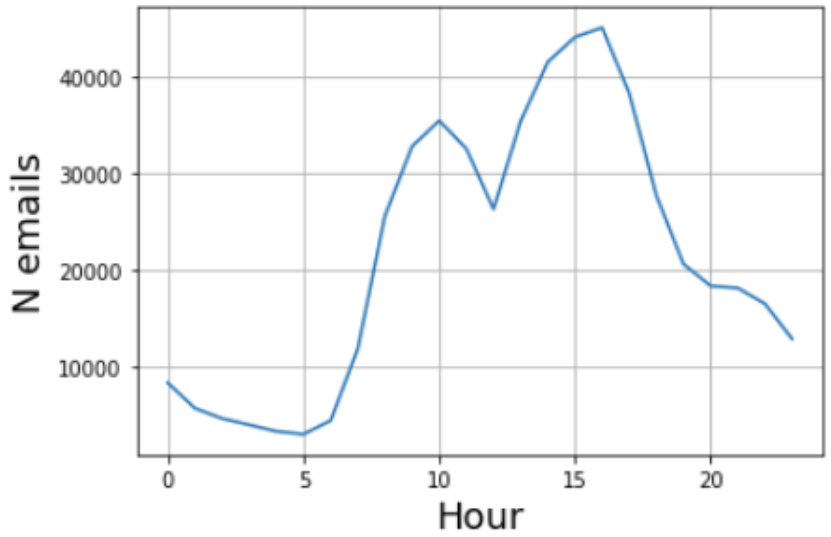

Рисунок 4.8 — График количества отправленных писем по времени суток в наборе *Enron*

На графике вышем видим наибольшую продуктивность во вторую половину дня, низкую активность в ночные часы, а также аномалию в

самом разгаре дня, объясняющуюся обеденным перерывом.

4.2.3 Частотность слов

Для интерпретации самых часто используемых слов использовались так называемые облака слов.

Облако слов — это метод визуализации данных, используемый для представления текстовых данных, в которых размер каждого слова указывает его частоту или важность. Важные точки текстовых данных могут быть выделены с помощью облака слов. Облака слов широко используются для анализа данных с веб-сайтов социальных сетей.

- В нашем случае можно, например, построить облако слов, используя информацию о темах писем:

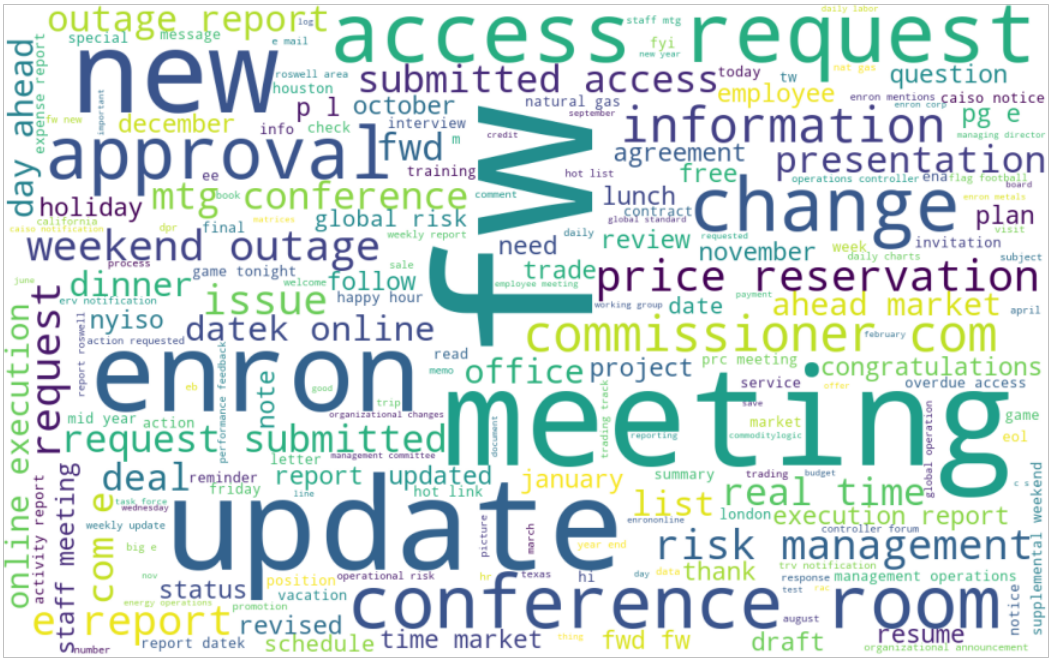

Рисунок 4.9 — Облако слов по словам из названий тем писем в наборе *Enron*

Слова, встречающиеся в темах писем: `meeting`, `conference room`, `update`, `approval`,`access request`, `enron`.

- Аналогично можно построить облако слов по информации из содержаний писем:

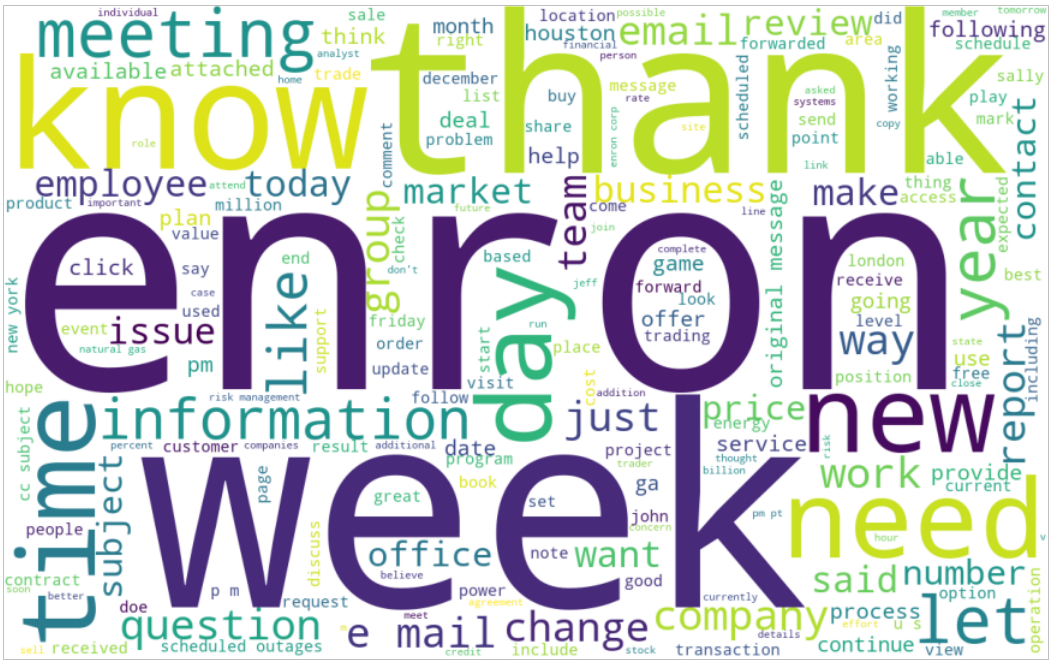

Рисунок 4.10 — Облако слов по словам из содержаний писем в наборе *Enron*

Слова, встречающиеся в содержании писем: `enron`, `week`, `thank`, `know`, `new`.

4.2.4 Отправители и получатели писем

- На основании отправителей и получателей писем можно посчитать некоторую интересную статистику. Например, можно найти 20 адресов, с которых было отправлено наибольшее количество электронных писем:

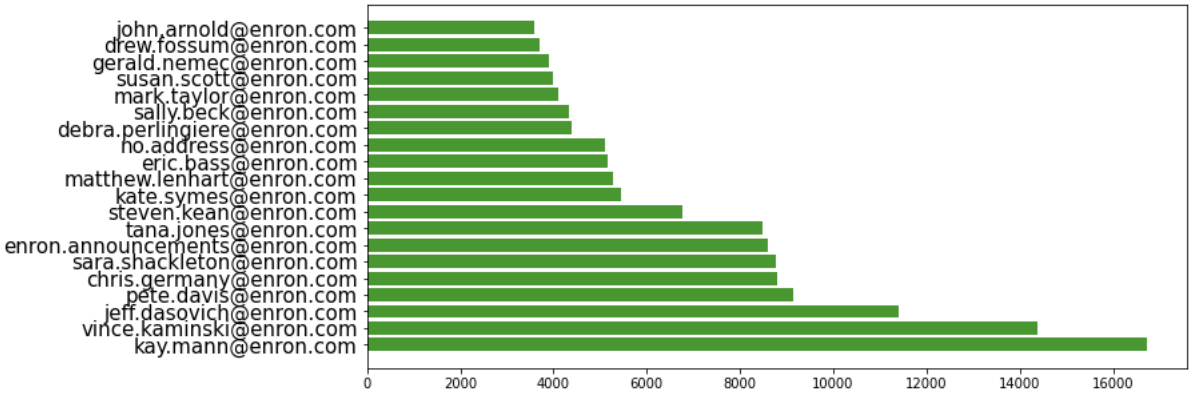

Рисунок 4.11 — Распределений количества писем по 20 адресам, с которых было отправлено наибольшее количество электронных писем в наборе *Enron*

- Аналогично, можно найти 20 адресов, на которые было отправлено наибольшее количество электронных писем:

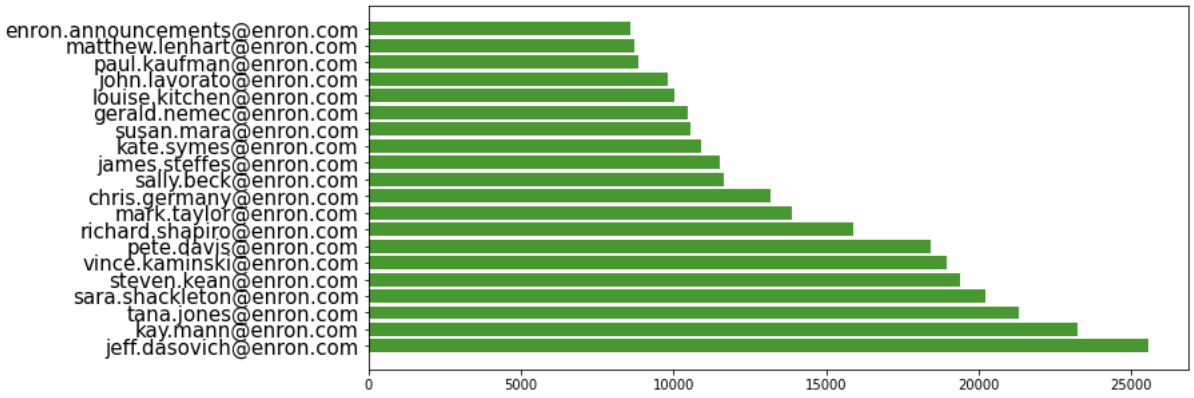

Рисунок 4.12 — Распределений количества писем по 20 адресам, на которые было отправлено наибольшее количество электронных писем в наборе *Enron*

Как видим, в графиках распределения получателей и отправителей писем много различий — некоторые люди пишут писем меньше, чем получают и наоборот.

- Теперь посмотрим количество писем между фиксированной парой собеседников. Рассмотрим только электронные письма, отправленные на один адрес электронной почты, так как они могут быть более важными личными сообщениями.

Отправитель	Получатель	Количество
pete.davis	pete.davis	9141
vince.kaminski	vkaminski@aol.com	4308
enron.announcements	all.worldwide	2206
enron.announcements	all.houston	1701
kay.mann	suzanne.adams	1528
vince.kaminski	shirley.crenshaw	1190
steven.kean	maureen.mcvicker	1014
kay.mann	nmann@erac.com	980
kate.symes	evelyn.metoyer	915
kate.symes	kerri.thompson	859

Таблица 4.1 — Таблица количества писем по парам собеседников в наборе *Enron*

Здесь интересно, что некоторые люди отправляют сами себе много электронных писем.

Глава 5
Исследование данных

5.1 Исследование электронных писем Хиллари Клинтон

5.1.1 Тематическое моделирование

Для тематического моделирования в качестве модели используется латентное размещение Дирихле.

Для оценки качества данной модели используется перплексия (англ. *perplexity*) — оценка того, насколько хорошо вероятностная модель предсказывает выборку. Низкая перплексия указывает на то, что распределение вероятностей хорошо предсказывает выборку.

В зависимости от параметра модели, отвечающего за количество тем у распределения текстов, получилась следующая зависимость значения перплексии от количества тем:

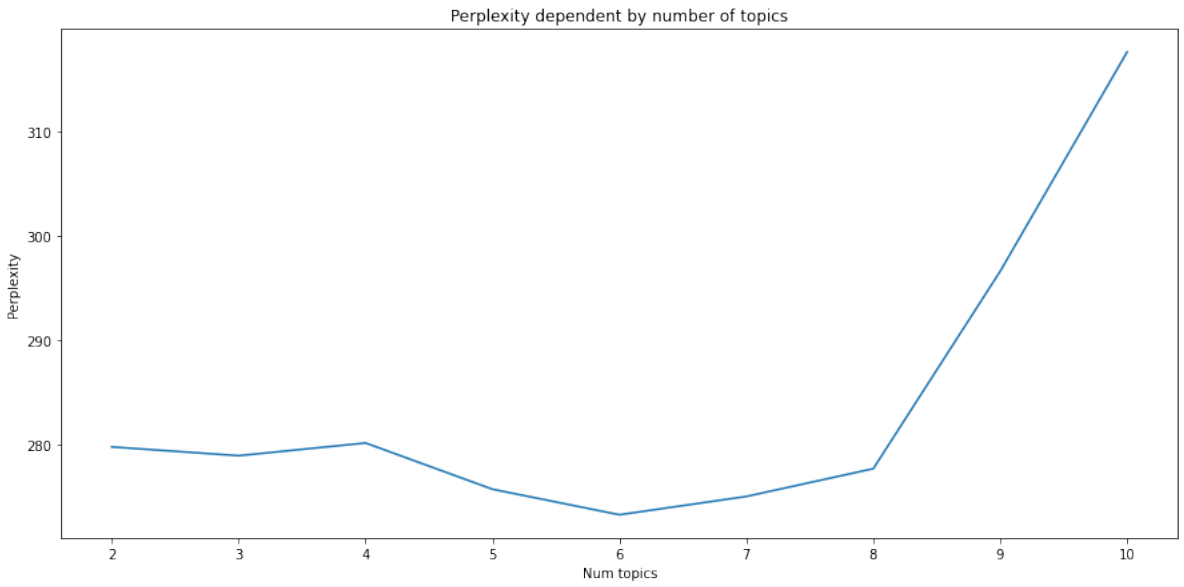

Рисунок 5.1 — Перплексия модели в зависимости от количества тем в наборе Клинтон

Ниже приведены примеры слов, принадлежащие каждой из 6 (с оптимальным значением перплексии) тем:

Номер темы	Слова
1	obama, state, president, government, american, israel, policy, country
2	woman, say, work, health, year, senate, group, government, support, company
3	call, get, work, see, want, know, good, also, think, tomorrow
4	secretary, office, state, meet, room, department, arrive, route, depart, private
5	state, information, benghazi, department, doc, case, subject, iran, agreement, house
6	cheryl, gov, fyi, sullivan, state, friday, sunday, branch, wednesday, april, january

Таблица 5.1 — Полученное описание тем в наборе Клинтон

Для визуализации работы алгоритмы используется представление тем в $2D$, то есть вычисляется двумерная проекция всех точек и затем визуализируются эти два измерения, причем в интерактивном виде, что позволяет нам получить представление, понятное человеку:

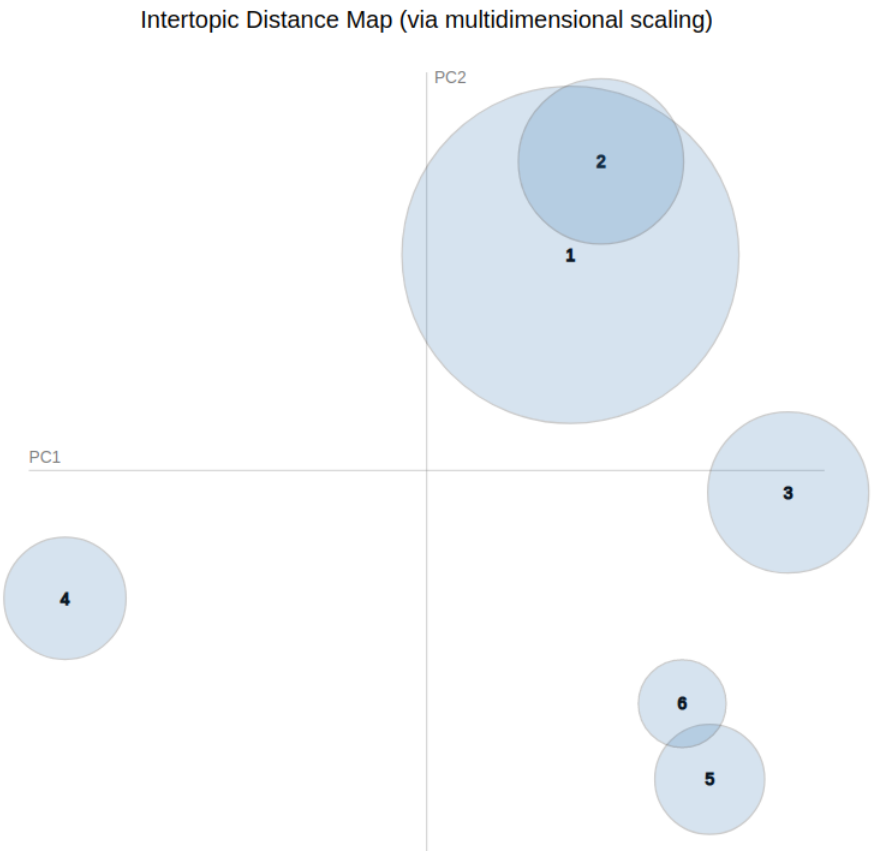

Рисунок 5.2 — Визуализации тем для модели латентного размещения Дирихле в наборе Клинтон

5.1.2 Кластеризация слов из электронных писем

Над содержанием электронных писем Хиллари Клинтон была также произведена кластеризация.

Текстовый корпус, состоящий из слов из электронных писем, оказался недостаточно большим, чтобы получить хорошие результаты. Поэтому для получения векторных представлений слов мы использовали *Word2Vec*, предобученный на датасете, полученный из постов в *Twitter* [18], а затем он был «дообучен» на словах из электронных писем.

Теперь, когда модель обучена, можем, например, для каждого слова посмотреть на ближайшие ему слова (ближайшие по векторному представлению):

- Возьмём в качестве исходного слова «*obama*»:

Слово	Расстояние
romney	0.9429854154586792
barack	0.9073218107223511
president	0.8986026048660278
clinton	0.8913119435310364
hillary	0.8597259521484375
say	0.8407208323478699
hovv	0.8315389752388

Таблица 5.2 — Ближайшие в векторном смысле слова к слову «*obama*», на основе векторных представлений *Word2Vec*, обученном на наборе Клинтон

- Аналогично, можем взять и любое другое слово, например, «*trump*»:

Слово	Расстояние
appropriator	0.7439741492271423
infighter	0.7368026971817017
zappos	0.7316897511482239
perkins	0.7260088920593262
donald	0.7180437445640564
buffett	0.7113708853721619
bloomberg	0.7067334651947021
clinton	0.7052138447761536

Таблица 5.3 — Ближайшие в векторном смысле слова к слову «*trump*», на основе векторных представлений *Word2Vec*, обученном на наборе Клинтон

Так же была построена интерактивная проекция точек на *2D*-плоскость с помощью алгоритма *t-SNE* [19]. *t-SNE* — это техника нелинейного сниже-

ния размерности, хорошо подходящей для вложения данных высокой размерности для визуализации в пространство низкой размерности (двух- или трехмерное). В частности, метод моделирует каждый объект высокой размерности двух- или трёхмерной точкой таким образом, что похожие объекты моделируются близко расположенными точками, а непохожие точки моделируются с большой вероятностью точками, далеко друг от друга отстоящими.

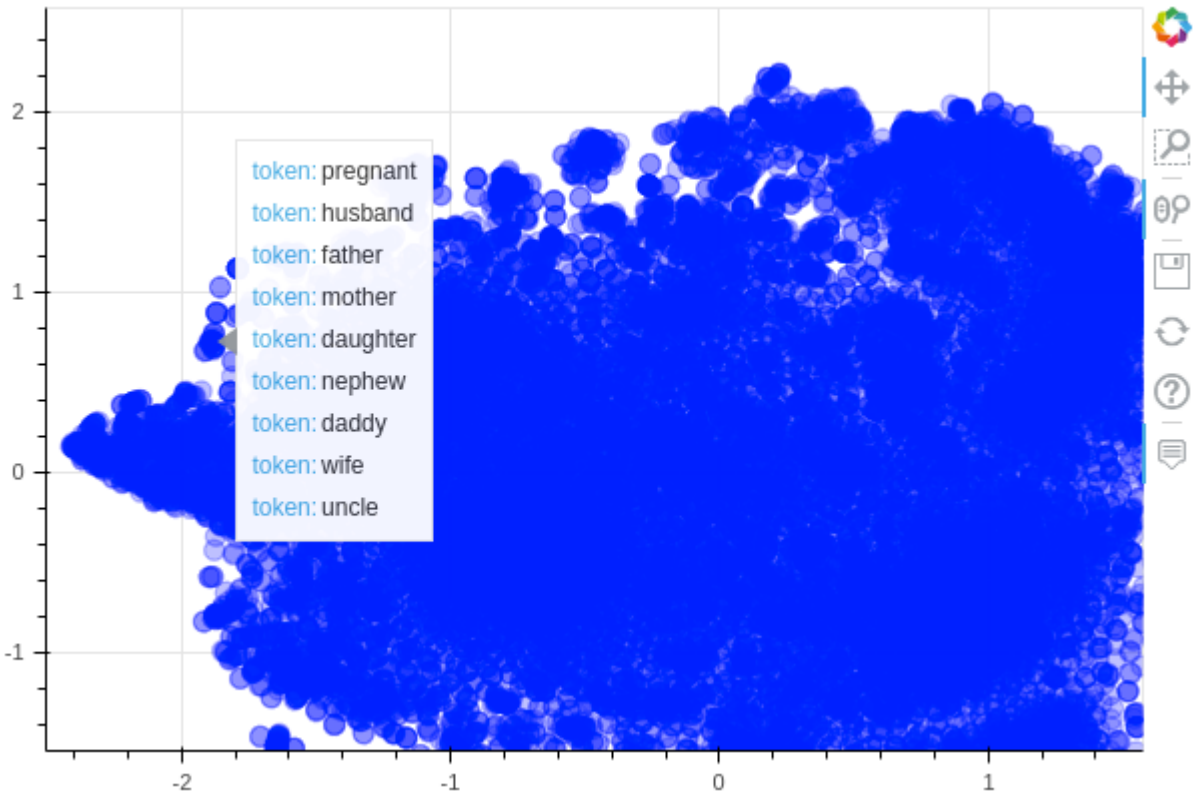

Рисунок 5.3 — Визуализации полученных векторных представлений слов набора Клинтон

5.2 Исследование электронных писем корпорации Enron

5.2.1 Тематическое моделирование

Как и в случае с электронными письмами Клинтон, в качестве модели используется латентное размещение Дирихле и такая же оценка качества — перплексия.

В зависимости от параметра модели, отвечающего за количество тем у распределения текстов, получилась следующая зависимость значения перплексии от количества тем:

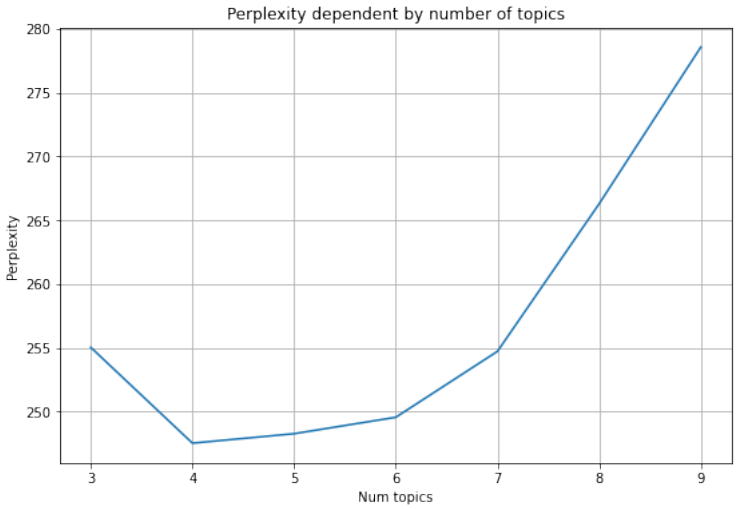

Рисунок 5.4 — Перплексия модели в зависимости от количества тем в наборе *Enron*

Оптимальным значением получилось количество тем, равное 4. Ниже приведены примеры слов, принадлежащие каждой из 4 тем:

Номер темы	Слова
1	page, com, mail, please, new, click, order, free, receive, email
2	please, enron, meet, thank, time, schedule, contact, attach, information, report
3	get, know, send, subject, would, thank, week, message, let, think
4	enron, company, market, page, would, state, issue, say, year, price

Таблица 5.4 — Полученное описание тем в наборе *Enron*

Видим, что темы получились интерпретируемые, слова связаны между собой смыслом.

Как и ранее, для визуализации работы алгоритмы используется представление тем в *2D*:

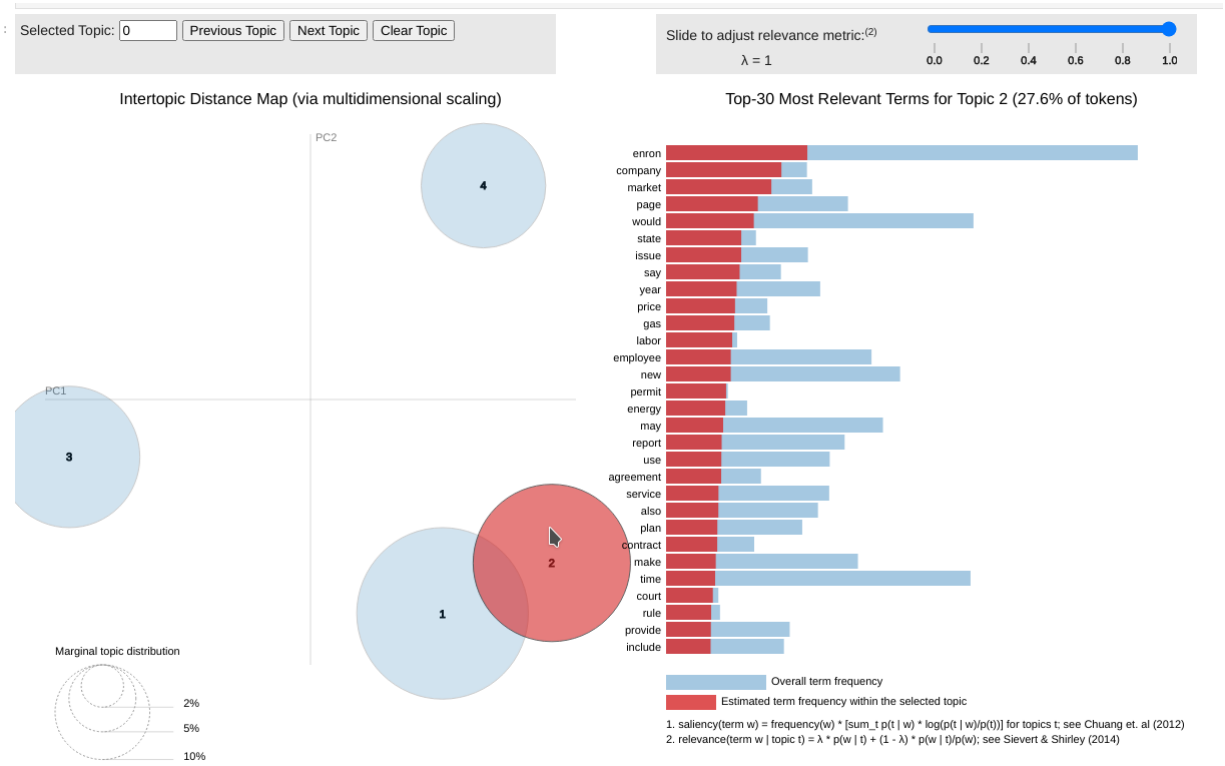

Рисунок 5.5 — Визуализации тем для модели латентного размещения Дирихле в наборе *Enron*

К примеру, во второй теме сосредочены слова об *Enron* и релевантные к деятельностьи компании термины.

5.2.2 Кластеризация слов из электронных писем

Аналогично кластеризации над письмами Клинтон, была произведена кластеризация над электронными письмами *Enron*. Как и ранее, для улучшения качества результатов, использовалась модель $Word2Vec$, предобученная на датасете постов из *Twitter*, которая затем была «дообучена» на корпусе писем *Enron*.

Полученная модель удовлетворяет таким же свойстам, что и ранее: если для слова взять его векторное представление и найти ближайшие векторные представления других слов, мы получим близкие по смыслу слова.

Так же была построена интерактивная проекция точек на *2D*-плоскость, однако в данном случае лучших результатов удалось добиться, используя в качестве алгоритма уменьшения размерности $UMAP$, а не *t-SNE*, который использовался для набора Клинтон:

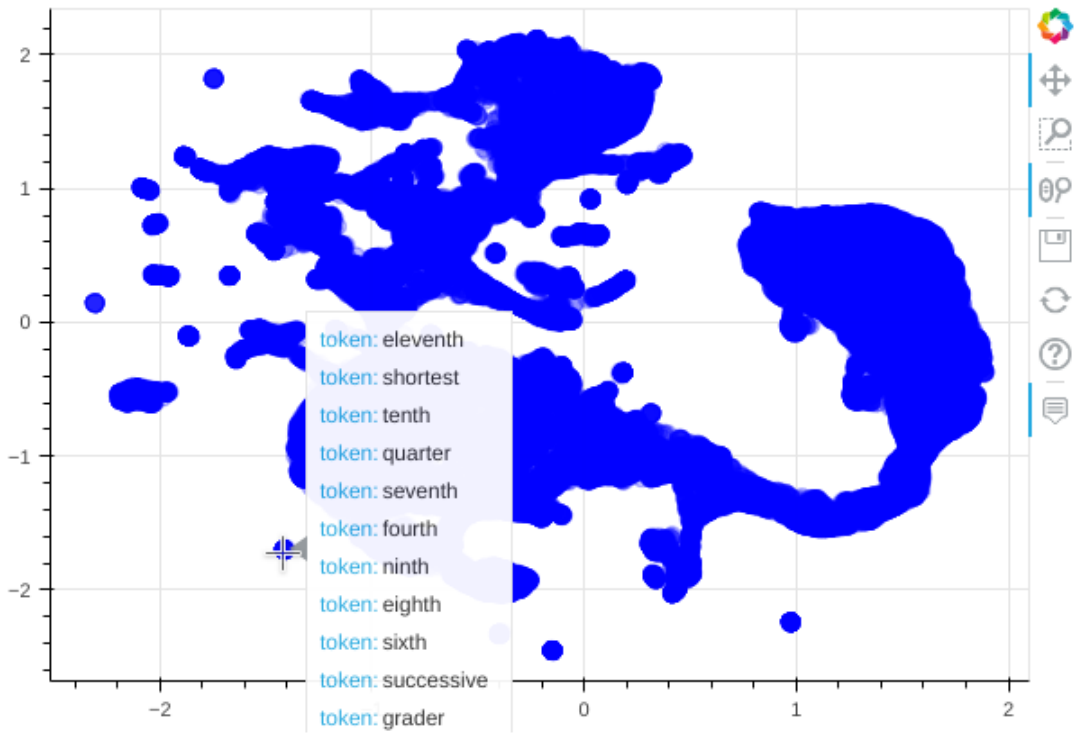

Рисунок 5.6 — Визуализации полученных векторных представлений слов набора *Enron*

Как можно заметить, сжатие координат до двумерных прошло успешно, об этом можно судить по равмерности распределения полученных точек и выделенному на рисунке скоплению точек — этот кластер содержит в себе названия чисел, также есть много других интерпретируемых кластеров.

Для набора *Enron* также была разработана кластеризация другим алгоритмом: вместо того, чтобы брать векторные представления слов и их кластеризовать, для каждого письма были усреднены векторные представления всех его слов (по каждой координате), которые далее были кластеризованы с помощью алгоритма $HDBSCAN$ (таким образом количество кластеров определится самостоятельно), предварительно уменьшив размерность векторных представлений до 5 с помощью $UMAP$. Теперь, для каждого такого кластера можно брать несколько самых близких слов (их векторных представлений). Таким образом получилась кластеризация исходных слов, которая, как оказалось, показывает результаты лучше наивной кластеризации векторных представлений слов. Интуиция, стоящая за этим, может заключаться в том, что смысл той или иной темы определяется всеми предложениями, описывающими эту тему.

Ниже приведены примеры слов, принадлежащие некоторым кластерам (из 1335 полученных):

Номер кластера	Слова
110	text, follow, txt, plz, reply, pls, please
3	hawaii, disneyland, jamaica, alaska, beach, resort, skyline, harbor, atlantic, caribbean
54	angel, queen, princess, prince, savage, poetic, explicit, remix, prod, barbie
194	satisify, vain, endless, definition, quite, priceless, compete, successfully, genius

Таблица 5.5 — Полученные кластера слов в наборе *Enron*

Как видно, кластера получились довольно интерпретируемые, например, в кластере 110 некоторые слова написаны по-разному (с опечатками, с сокращением), но модель смогла понять что они по сути означают одно и то же, а в кластере 3 содержатся слова, содержащие названия некоторых географических объектов.

5.2.3 BERTopic

Также было произведено тематическое моделирование с использованием алгоритма *BERTopic*.

В отличие от всех предыдущих методов, здесь требуется немного другая предобратка текста: ранее из предложений удалялись редко встречаемые слова и стоп-слова (артикли, предлоги и так далее), в данном случае этот этап предобработки будет пропущен. Для получения векторного представления предложений вместо того, чтобы усреднять векторные представления всех входящих в него слов, будет использоваться сразу языковая модель *BERT*, предобученная на огромном корпусе предложений. В обучаем корпусе не удалялись такие редкие слова и стоп-слова, так как при их удалении теряется некоторая часть структуры предложения, и также теперь векторные представления извлекаются у слов, учитывая их контекст внутри предложения. Поэтому достаточно сделать всю ту очистку текста, которая была проведена ранее, за исключением удаления редких и стоп-слов.

В результате применения алгоритма получилось 1750 тем, из которых 13 состоят более чем из 80 слов, для остальных тем ниже приведен рисунок распределения количества слов в темах:

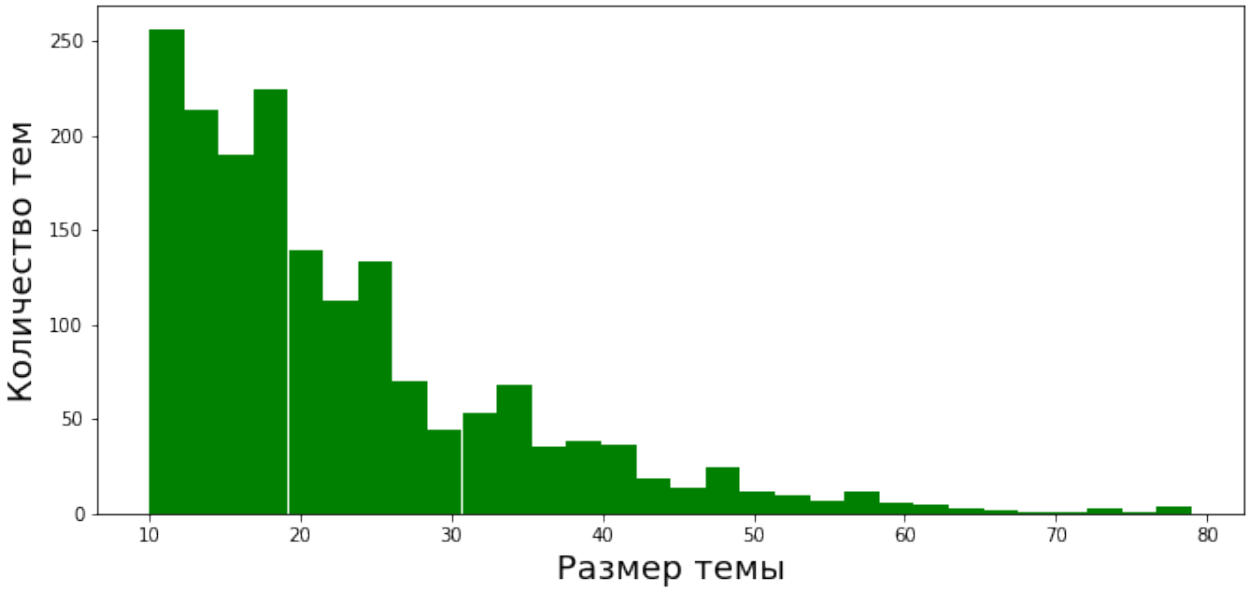

Рисунок 5.7 — Гистограмма распределения размеров тем в результате работы алгоритма *BERTopic* на наборе *Enron*

Данная модела сработала хорошо, полученные темы получились интерпретируемы, ниже приведены некоторые из них:

Номер кластера	Слова
261	congrats, congratulations, exciting, promotions, jeff
327	superstar, choate, wonderful, success, gifts, huge, nice, informative, kimberly, glad
1164	drop, pick, damon, butch, dropping, picking, richie, rid, neal, kirby
394	vpn, pending, applications, requested, id, natgas, jeffrey, phillip, application, jeff

Таблица 5.6 — Примеры полученных тем в результате работы алгоритма *BERTopic* на наборе *Enron*

Как и ранее, можно визуализировать темы в $2D$:

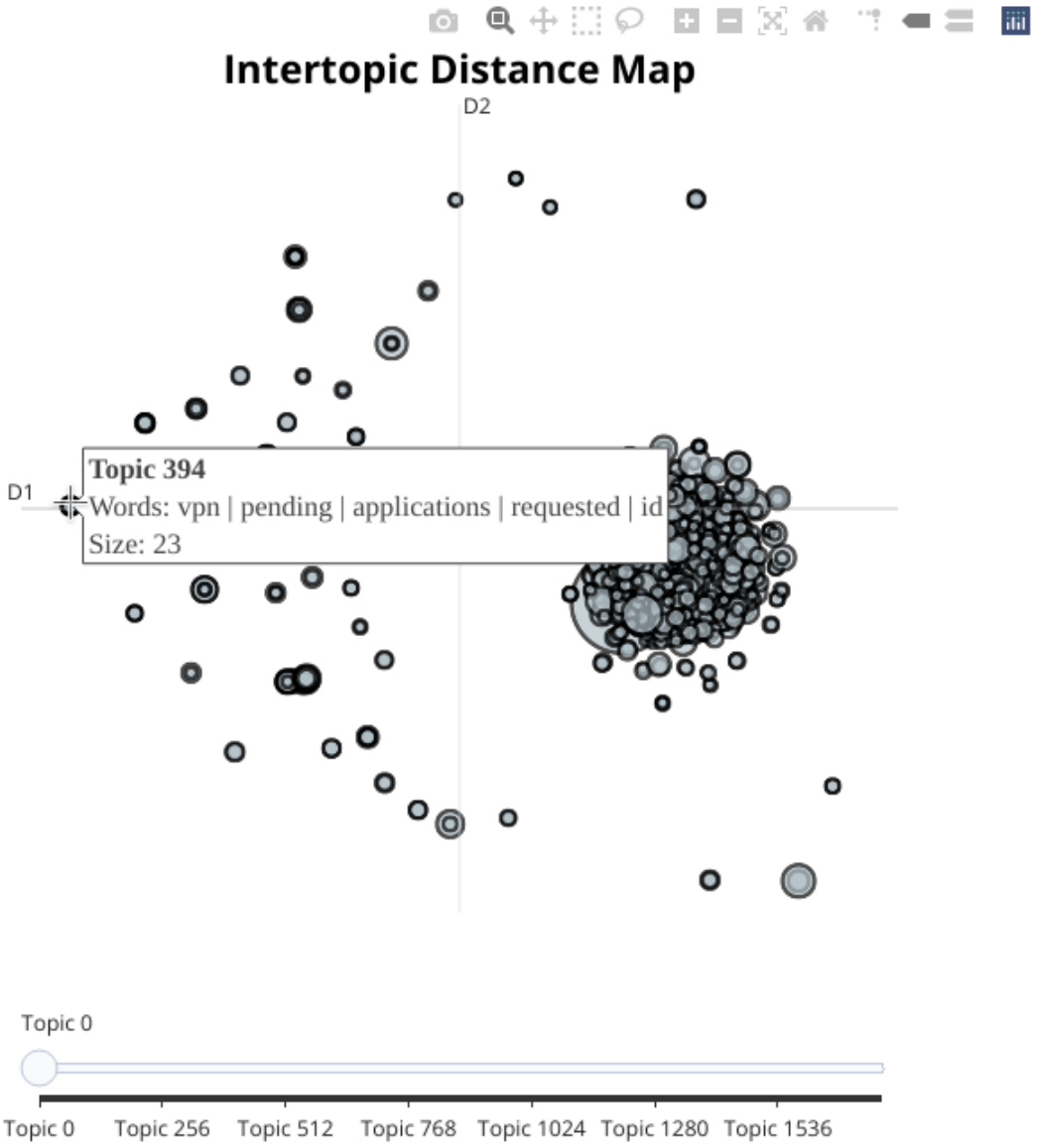

Рисунок 5.8 — Визуализации тем для модели *BERTopic* на наборе *Enron*

В данном случае точки распределились не совсем равномерно, есть темы, расположенные далеко от большого скопления точек, однако, как можно видеть на рисунке ниже, внутри этого скопления точки распределены равномерно:

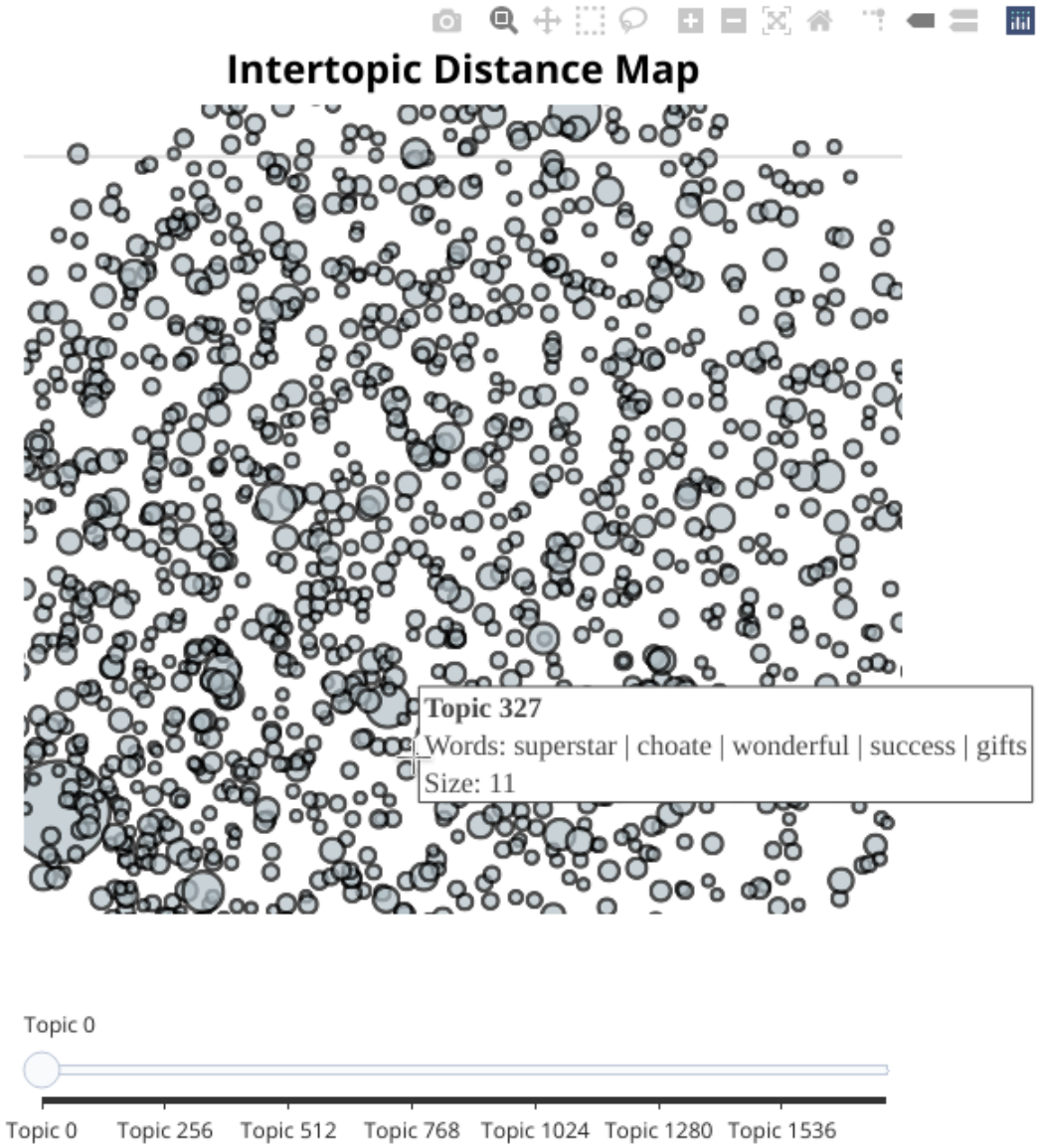

Рисунок 5.9 — Визуализация части скопления тем в $2D$ пространстве для модели $BERTopic$ на наборе $Enron$

Заключение

В данной работе были проведены эксперименты с исследованием электронных писем Хиллари Клинтон и корпорации *Enron*.

Основная проблема в исследовании писем Клинтон — недостаточно большой размер датасета. Это приводит к проблеме недостаточного уровня обученности моделей. Она решается с помощью моделей, предобученных на датасетах большего размера. Здесь и тематическое моделирование, и кластеризация показали неплохие интерпретируемые результаты, о чем можно судить по представленным таблицам в соответствующих разделах работы.

Что касается набора *Enron*, он содержит гораздо больше данных, однако содержимое писем представлено не очень качественно (например, много автоматически генерируемого текста, полученного сервисами для отправки писем, который не несёт никакой смысловой информации), поэтому была проведена более серьёзная работа по очистке текста. Как и в случае набора данных Клинтон, модели тематического моделирования и кластеразация векторных представлений слов показали хорошие интерпретируемые результаты. Для набора *Enron* также использовалась техника кластеризации слов, основанная на векторных представлениях целых предложений, а не отдельных слов — такой метод показал ещё более интерпретируемые результаты. Также для писем *Enron* был применен алгоритм *BERTopic*, который сгенерировал хорошо интерпретируемые темы, но, в сравненении с моделью Латентного размещения Дирихле, тем получилось намного больше.

На обоих датасетах был произведен тщательный и внимательный анализ писем и их метаданных, построено большое количество визуализаций и распределений различной информации о письмах.

Для лучшей работы всех используемых моделей была произведена оптимизация гиперпараметров и техника передачи обучения (англ. *Transfer Learning*), позволяющая использовать предобученные модели, которые «видели» большое количество данных при обучении и были впоследствии дообучены на исходных электронных переписках. С использованием этой техники результаты получились лучше, чем при обучении модели «с нуля», то есть в случае, когда все параметры модели инициализируются константами или же случайным образом.

Если сравнивать результаты с результатами работ пользователей *Kaggle*, исследование данных проведено более тщательно и результаты данной работы получились более интуитивными и интерпретируемыми.

Исходный код работы доступен по ссылке: `https://github.com/kefaa/emails-research-diploma`.

Литература

1. Hillary Clinton's Emails, https://www.kaggle.com/kaggle/hillary-clinton-emails.

2. Partisanship, Propaganda, and Disinformation: Online Media and the 2016 U.S. Presidential Election // Berkman Klein Center — December 7, 2017.

3. Analyzing the meta data of Hillary Clinton's emails. https://uwaterloo.ca/statistics-and-actuarial-science/news/analyzing-meta-data-hillary-clintons-emails.

4. Ron Bekkerman et al., "Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora, Technical Report, University of Massachusetts. https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1217&context=cs_faculty_pubs.

5. Jitesh Shetty and Jafar Adibi, "The Enron Email Dataset: Database Schema and Brief Statistical Report,"Technical Report, Information Sciences Institute, 2004. http://www.isi.edu/~adibi/Enron/Enron_Dataset_Report.pdf.

6. Jitesh Shetty, Jafar Adibi, 2005. Discovering Important Nodes through Graph Entropy the Case of Enron Email Database, KDD'2005, Chicago, Illinois.

7. Ria Kulshrestha. A Beginner's Guide to Latent Dirichlet Allocation (LDA). https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2

8. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // In Proceedings of Workshop at ICLR. — 2013a.

9. Victor Roman. Unsupervised Machine Learning: Clustering Analysis. https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e.

10. BERTopic. https://maartengr.github.io/BERTopic/.

11. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805.

12. Lena Voita. (Introduction to) Transfer Learning. https://lena-voita.github.io/nlp_course/transfer_learning.html.

13. Brendan Bailey. Lightning Talk: Clustering with HDBScan. https://towardsdatascience.com/lightning-talk-clustering-with-hdbscan-d47b83d1b03a.

14. BERTopic, Topic Visualization. https://maartengr.github.io/BERTopic/tutorial/visualization/visualization.html.

15. Natural Language Toolkit. https://www.nltk.org.

16. Библиотека spaCy. https://spacy.io/.

17. Email — An email and MIME handling package. https://docs.python.org/3/library/email.html.

18. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/.

19. van der Maaten L.J.P., Hinton G.E. Visualizing Data Using t-SNE // Journal of Machine Learning Research. — 2008. — Ноябрь (т. 9).