МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ Н.Э. БАУМАНА

Маркин Кирилл Вадимович

Разработка метода тематического моделирования для новостей на русском языке

Специальность 2301050065 —
«Программное обеспечение вычислительной техники и автоматизированных систем»

Квалификационная работа бакалавра

Научный руководитель:
доцент, кандидат технических наук
Клышинский Эдуард Станиславович

Консультант:
старший преподаватель
Волкова Лилия Леонидовна

Москва — 2019

Реферат

Отчет страниц, 4 части, 8 рисунков, 2 таблицы, 10 источников.

Объектом исследования является метод тематического моделирования в применении к новостям на русском языке.

Цель работы: разработка метода тематического моделирования для новостей на русском языке.

В данной работе разрабатывается метод тематического анализа новостей на русском языке. Реализовано ПО для начиная первого этапа - сбора данных из сети Интернет. Рассмотрен процесс обработки и подготовки данных, создания модели. Сравниваются результаты различных модификаций, и создаются рекомендации для применения.

Расчетно-пояснительная записка содержит аналитический, конструкторский, технологический и исследовательский разделы.

В аналитическом разделе детально изучена предметная область. Проведен анализ принципов работы существующих методов тематического моделирования.

В конструкторском разделе создана структура данных для хранения коллекций новостей. Описаны принятые проектные решения. Рассмотрены варианты и выбраны способы реализации решения. Выделены функциональные требования к решению.

В технологическом разделе описан выбор средств разработки, описаны нетривиальные моменты реализации.

В исследовательском разделе продемонстрирован и проанализирован процесс классификации новостей на нескольких коллекциях. Разобран процесс подбора коэффициентов при регуляризации. Проведен сравнительный анализ обученных моделей.

Поставленная цель работы достигнута: разработан метод тематического моделирования для новостей на русском языке.

Содержание

Введение

Из-за огромного количества различных новостных потоков стало сложно выделять действительно актуальную для себя информацию.

Разрабатываемый мною метод тематического моделирования новостей будет применяться для распределения новостного потока на различные нужные для пользователя темы.

Это решение будет особенно актуально пользователям, интересующимся узкими темами, которые не распределяются журналистами на категории.

Целью данной работы является разработка метода тематического моделирования для новостей на русском языке.

Для достижения этой цели необходимо выполнить следующие основные **задачи**:

- анализ существующих методов тематического моделирования и выбор базового для классификации новостей на русском языке;
- разработка программного продукта для сбора новостей на русском языке и подготовки данных в виде отдельных сервисов;
- обучение тематической модели для новостных текстов;
- проведение параметризации метода;
- проведение апробации метода;
- составление рекомендаций о применимости предложенного метода.

1 Аналитический раздел

В аналитическом разделе подробно рассмотрена предметная область и проанализированы существующие методы. Описана формальная постановка задачи. А так же сформулированы функциональные требования к решению.

1.1 Задачи тематического моделирования

Задачи, для решения которых используется тематическое моделирование разбивают на 2 класса: **Автоматический анализ текста** и **систематизация больших объемов информации**.

В задачах автоматического анализа текста обычно выделяют следующие направления.

- **Классификация документов** - необходимо присвоить каждому документу метку соответствующих классов.

- **Автоматическое аннотирование документов** - составление краткого обзора документа на основании использования наиболее важных фраз, используя наиболее важные фразы.

- **Автоматическое реферирование или суммаризация коллекции** - решение предыдущей задачи для большой коллекции документов.

- **Тематическая сегментация документов** - разбиение длинного документа на части с различными темами.

В задачах систематизации больших объемов информации обычно выделяют следующие направления:

- **Семантический (разведочный) поиск информации** - поиск по коллекции документов на базе тематического моделирования позволяет использовать длинный документ в качестве поискового запроса, а также находить документы, близкие по смыслу, даже

если ключевые слова, используемые при поиске, отсутствуют в результатах поиска.

- **Визуализация тематической структуры коллекции** - все задачи, связанные с графическим представлением больших массивов документов.

- **Анализ динамики развития тем** - обычно используется при наличии данных о времени создания документов в коллекции.

- **Тематический мониторинг новых поступлений** - автоматический мониторинг настроенных ресурсов на наличие новых документов, схожих по тематике с настроенным целевым документом.

- **Рекомендация документов пользователям** - создание рекомендательных систем на основании данных о просмотренных пользователем документах и его активности.

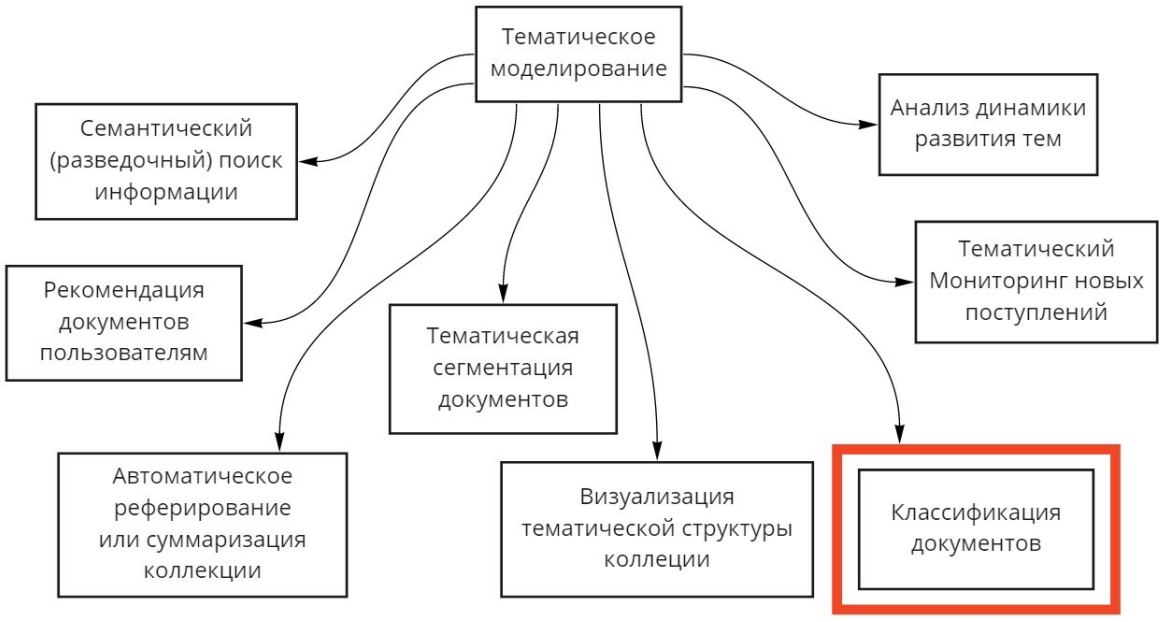

Рисунок 1.1: Задачи тематического моделирования.

1.2 Существующие методы

1.2.1 Основы кластеризации и классификации документов

Документы представляются векторной моделью (VSM, Vector Space Model). В такой модели каждому слову сопоставляется определенный вес, вычисляемый по весовой функции.

Базовый вариант весовых функций в таком представлении данных - частота слова (TF), которая равна отношению числа вхождения определенного слова к общему числу слов документа

$$TF(t,d) = \frac{freq(t,d)}{max_{W \in D} freq(w,d)}. \tag{1.1}$$

где

freq() - частота (frequency).

Также используется агрегирующий показатель

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D), \tag{1.2}$$

где

IDF — обратная частота документа, инверсия частоты, с которой определенное слово встречается в документах коллекции.

$$IDF(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{1.3}$$

где

$|D|$ — число документов в коллекции.

$|\{d \in D : t \in d\}|$ — число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

Выбор основания логарифма в формуле не имеет значения, поскольку изменение основания приводит к изменению веса каждого слова на постоянный множитель, что не влияет на соотношение весов.

В первый раз задача определения и отслеживания тем (TDT, Topic Detection and Tracking) встречается в работе "Topic Detection and Tracking Pilot Study. Final Report"[1]. Темой в этой работе называют событие или действие вместе со всеми непосредственно связанными событиями или

действиями. Задачей является извлечение событий.

Еще вариант из работы [1]:

$$w(t,D) = (1 + \log_2 TF(t,D)) \times \frac{IDF(t)}{||\vec{d}||,} \tag{1.4}$$

где $||\vec{d}||$ - номер вектора, представляющего документ D. Еще варианты модификаций TF-IDF из работ [2]:

$$TF' = \frac{TF}{TF + 0.5 + 1.5\frac{l_d}{l_{avg}}}, \tag{1.5}$$

где l_d - длинна документа d, а l_{avg} - средняя длинна документа.

$$IDF' = \frac{\log(IDF)}{\log(N+1)} \tag{1.6}$$

Для определения расстояния в таком представлении данных использовались различные метрики: дивергенция Кульбака-Лейблера, косинусная мера и другие. В первых работах для решения таких задач использовались алгоритмы кластеризации - выделение групп близких объектов без обучающей выборке и без сведений о классах: метод К-средних, инкрементальная кластеризация, на основе анализа плотности точек [3] и т. д. Каждый кластер описывал то или иное событие.

Главным недостатком такого подхода является однозначность отношения документ-тема. То есть один документ относится к одной теме (событию). В рассматриваемом ниже примере про новость финансирования спорта будет продемонстрировано, что в одном документе могут затрагиваться сразу две темы и футбол и финансы. При таком подходе эти данные теряются.

Используется векторное представление текста, как было сказано выше. Координатой документа может быть частота термина или иных конструкций, полученных при анализе текста. Текст подлежит четырем ключевым этапам анализа - морфологическому, синтаксическому, семантическому [4], графематическому. В качество координат документа в данной

работе будем рассматривать частоты употребления в нем слов, представленных леммами - начальными формами слова.

Семантика это раздел лингвистики, изучающий смысловое значение единиц языка.

1.2.2 Латентный семантический анализ (LSA)

Dumais и другие [5] в 1988 году предложили метод LSA. Суть метода в том, чтобы спроецировать документы и термины в пространство более низкой размерности. Для этого анализируется совместная встречаемость слов (терминов) в документах. Таким образом задача состоит в том, чтобы часто встречающиеся вместе термины были спроецированы в одно и то же измерение семантического пространства.

Этот метод использует мешок слов (или Bag of Words). Это модель текстов на натуральном языке, в которой каждый документ или текст выглядит как неупорядоченный набор слов без сведений о связях между ними. Его можно представить в виде матрицы, каждая строка в которой соответствует отдельному документу или тексту, а каждый столбец — определенному слову.

1.2.3 Вероятностный латентный семантический анализ (PLSA)

В 1999 году Томасом Хофманом был предложен метод вероятностного латентного семантического анализа (PLSA) [6] [7]. В вероятностных тематических моделях, в отличие от рассмотренных выше методов, сначала задается модель, а после с помощью матрицы слов в документах оцениваются ее скрытые параметры. В связи с этим появляется возможность дообучения моделей и упрощается подбор параметров.

Для лучшего понимания алгоритма рассмотрим детальнее процесс написания новости журналистом. Для начала работы он выбирает тему своей новостной статьи. Это, в свою очередь, влияет на то, какие слова он будет использовать. Очевидно, что если журналист решил написать новость про футбол, то слово «мяч» в таком документе появится с большей вероятностью, чем слово «антиматерия». При этом если статья затраги-

вает финансовую сторону вопроса, то вероятности возникновения слов «мяч» и слово «бюджет» могут сравняться. В таком случае мы можем сказать, что такая новость имеет минимум две темы - «спорт» и «финансы», которые в свою очередь и породили слова «мяч» и «бюджет».

Продолжая эту аналогию, можно представить любую новость как смесь различных тем, которые в свою очередь породили слова.

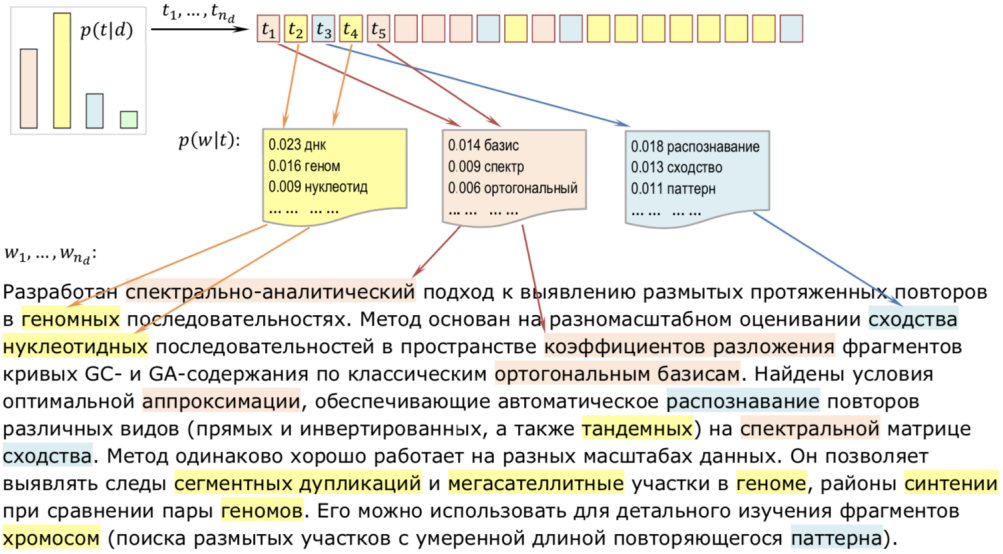

Рисунок 1.2: Процесс порождения текстового документа вероятностной тематической моделью. Иллюстрация из лекции К.В. Воронцова

Приняты следующие допущения.

- Порядок слов в документе не важен (bag of words).
- Слова в документах генерируются темой, а не самим документом.
- Порядок документов в коллекции не важен.
- Каждое отношение документ-слово (d,w) связано с некоторой темой $t \in T$.
- Коллекция представляет собой последовательность троек документ-слово-тема (d,w,t).
- В теме невелико число образующих слов.

- В документе используется небольшое число тем.

Пусть

D - коллекция документов размера n_d с документами d,

W - словарь терминов размера n_w со словами w,

T - список тем размера размера n_t с темами t,

n_{dw} - количество использований слова w в документе d,

каждый документ состоит из слов: $d \subset W$,

$p(w|d)$ - вероятность появления слова w в документе d,

$p(w|t)$ - вероятность появления слова w в теме t,

$p(t|d)$ - вероятность появления темы t в документе d,

$\hat{p}(w|d) = \frac{n_d w}{n_d}$ - наблюдаемая частота слова w в документе d.

Требуется найти параметры вероятностной порождающей тематической модели, то есть представить вероятность появления слов в документе $p(w|d)$ в виде:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d). \tag{1.7}$$

Запишем вероятности $p(w|t)$ в матрицу $\Phi = (\phi_{wt})$, а вероятности $p(t|d)$ - в матрицу $\Theta = (\theta_{td})$. Тогда вероятность появления слов в документе можно представить в виде матричного разложения:

$$p(w|d) = \sum_{t \in T} \phi_{wt}\theta_{td}. \tag{1.8}$$

То есть решается задача, обратная к генерации текста (работе журналиста). Необходимо по имеющийся коллекции документов понять, какими распределениями матриц ϕ_{wt} и θ_{td} она могла быть получена.

При этом так как речь идет о вероятностных тематических моделях каждый столбец матриц ϕ_{wt} и θ_{td} представляет собой дискретное распределение вероятностей. То есть значения не отрицательны и сумма по каждому столбцу равна 1. Такие матрицы называют стохастическими.

Теперь, воспользовавшись принципом максимума правдоподобия с ограничениями на элементы стохастических матриц, если максимизиро-

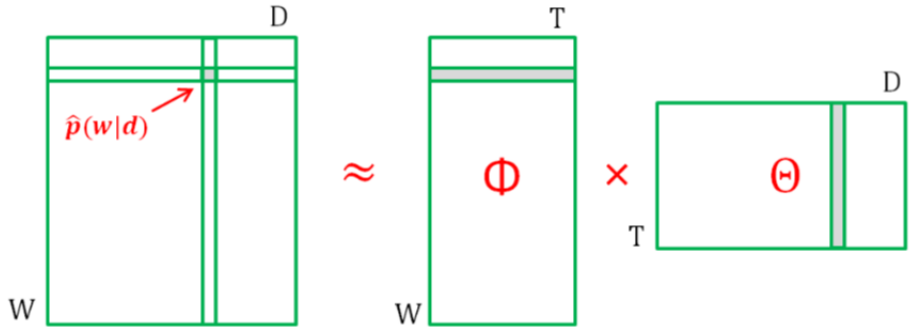

Рисунок 1.3: Матричное разложение. Иллюстрация из лекции К.В. Воронцова .

вать логарифм правдоподобия,получается:

$$\begin{cases} \sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \phi_{wt}\theta_{td} \to max_{\Phi,\Theta}; \\ \sum_{w\in W} \phi_{wt} = 1; & \phi_{wt} \geq 0; \\ \sum_{t\in T} \theta_{td} = 1; & \theta_{td} \geq 0. \end{cases} \tag{1.9}$$

1.2.4 Латентное размещение Дирихле (LDA)

Задача в таком виде поставлена не корректно, так как существует больше одного решения этой системы:

$$\Phi\Theta = (\Phi S)(S^{-1}\Theta) = \Phi'\Theta'. \tag{1.10}$$

То есть результаты будут зависеть от стартовых значений параметров модели и при кадом обучении будут различаться. Но так же это означает, что есть возможность модифицировать алгоритм, сужая пространство решений. Введем для этого критерий регуляризации $R(\Phi,\Theta)$ - некоторый функционал, соответствующий прикладной задаче, для которой обучается модель. Рассмотрим задачу максимизации регуляризованного

правдоподобия:

$$\begin{cases} \sum_{d\in D}\sum_{w\in d} n_{dw}\ln\sum_{t\in T}\phi_{wt}\theta_{td} + R(\Phi,\Theta) \to max_{\Phi,\Theta}; \\ \sum_{w\in W}\phi_{wt} = 1; & \phi_{wt} \geq 0; \\ \sum_{t\in T}\theta_{td} = 1; & \theta_{td} \geq 0. \end{cases} \tag{1.11}$$

В 2003 году Дэвидом Блеем, Эндрю Энджи и Маклом Джорданом был предложен метод латентного размещения Дирихле (LDA) [8]. На данный момент это одна из самых цитируемых статей по тематическому моделированию. Они предложили решать задачу со следующим регуляризатором:

$$\begin{gathered} R(\Phi,\Theta) = \sum_{t,w}(\beta_w - 1)\ln\phi_{wt} + \sum_{d,t}(\alpha_t - 1)\ln\theta_{td}, \\ \beta_w > 0, \\ \alpha_t > 0, \end{gathered} \tag{1.12}$$

где β_w и α_t - параметры регуляризатора.

Для понимания метода введем понятие дивергенции Кульбака-Лейблера для дискретных распределений.Пусть даны два дискретных распределения $P = (p_i)_{i=1}^n$ и $Q = (q_i)_{i=1}^n$, тогда дивергенция Кульбака-Лейблера выражается так

$$KL(P||Q) = \sum_i p_i \log\frac{p_i}{q_i}. \tag{1.13}$$

Дивергенция Кульбака-Лейблера обладает следующими свойствами.

- неотрицательность:

$$KL(P||Q) \geq 0;$$

$$KL(P||Q) = 0 \Leftrightarrow P = Q$$

- несимитричность:

$$KL(P||Q) \neq KL(Q||P)$$

Дивергенция Кульбака-Лейблера связана с максимумом правдоподобия:

$$\sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i(\alpha)} \to \min_{\alpha} \Leftrightarrow \sum_{i=1}^{n} p_i \ln q_i(\alpha) \to \max_{\alpha} \tag{1.14}$$

Минимизация дивергенции Кульбака-Лейблера эквивалентна максимизации правдоподобия. Пусть P - эмпирическое распределение. Q - параметрическая модель распределения с параметром α. При минимизации дивергенции Кульбака-Лейблера (максимизации правдоподобия) определяется такое значение α, при котором P как можно лучше соответствует модели.

Пусть $\beta = (\beta_w)$ - некоторый вектор над словарем W со словами w.

При $\beta_w > 1$ вероятность ϕ_{wt} этого слова по темам будет сглаживаться, приближаясь к β_w^+ :

$$KL(\beta^+ || \phi_t) \to min, \tag{1.15}$$

$$\beta_w^+ = \underset{w \in W}{norm}(\beta_w - 1) \tag{1.16}$$

При $\beta_w < 1$ значение ϕ_{wt} наоборот будут разреживаться, удаляясь от β_w^- к нулю :

$$KL(\beta^- || \phi_t) \to max, \tag{1.17}$$

$$\beta_w^- = \underset{w \in W}{norm}(1 - \beta_w) \tag{1.18}$$

то есть в матрице Ф будет больше нулевых элементов или близких к нулю.

1.2.5 Аддитивная регуляризация тематических моделей (ARTM)

Неединственность решения максимизации регуляризованного правдоподобия позволяет накладывать сразу несколько ограничений на модель, этот метод называется аддитивной регуляризацией тематических моделей (ARTM).

То есть

$$\sum_{d,w} n_{dw} \ln \sum_{t} \phi_{wt}\theta_{td} + \sum_{i=1}^{k} \tau_i R_i(\Phi,\Theta) \to \max_{\Phi\Theta} \tag{1.19}$$

где τ_i - коэффициенты регуляризации, а $R_i(\Phi,\Theta)$ - регуляризаторы.

При таком подходе возникает проблема поиска коэффициентов, которая обычно решается добавлением регуляризаторов в модель по одному и оптимизации соответствующих коэффициентов в ходе пробных запусков моделей.

1.2.6 Решение задачи максимизации регуляризованного правдоподобия

Решение задачи в общем виде аналитическими методами слишком сложно. Однако, если выбирать гладкие регуляризаторы, то можно воспользоваться условием Крауша-Куна-Таккера. Получится система уравнений:

$$\begin{cases} p_{tdw} = \underset{t\in T}{norm}(\phi_{wt}\theta_{td}) \\ \phi_{wt} = \underset{w\in W}{norm}\left(\sum\limits_{d\in D} n_{dw}p_{tdw} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right) \\ \theta_{td} = \underset{t\in T}{norm}\left(\sum\limits_{w\in d} n_{dw}p_{tdw} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right) \end{cases} \tag{1.20}$$

где

$$\underset{t\in T}{norm}(x_t) = \frac{max\{x_t, 0\}}{\sum\limits_{s\in T} max\{x_s, 0\}} \tag{1.21}$$

Такую систему можно решить численным методом простых итераций. В данном случае его называют EM-алгоритм.

Для получения результата необходимо итерационно выполнять E-шаг и M-шаг до достижения требуемой точности.

E-шаг :

$$p_{tdw} = \underset{t\in T}{norm}(\phi_{wt}\theta_{td}) \tag{1.22}$$

M-шаг :

$$\phi_{wt} = \underset{w\in W}{norm}\left(\sum_{d\in D} n_{dw}p_{tdw} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right) \tag{1.23}$$

$$\theta_{td} = \underset{t \in T}{norm}\left(\sum_{w \in d} n_{dw} p_{tdw} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right) \tag{1.24}$$

Этот процесс можно организовать параллельно, если обновлять матрицу Φ по порциям, после анализа очередного пакета документов. Обычно уже после просмотра нескольких первых десятков тысяч документов, матрица Φ получается уже устоявшиеся, и остается только тематизировать остальные документы. Подробнее с EM алгоритмом можно ознакомится в работе Frei O., Apishev M. [9].

1.2.7 Выбор алгоритма

В данной работе рассматривается задача классификации документов. В качестве документов выступают новости на русском языке. Необходимо с помощью выбранного метода и способов его усовершенствования разбить коллекцию новостей на темы, интерпретируемые человеком и получить возможность оценивать новый документ (новость) на принадлежность этим темам.

Особенностью тематического моделирования является возможность не использовать в процессе построения модели размеченные данные. То есть темы, на которые разбивается коллекция, также создаются в процессе формирования модели.

Для дальнейшей работы принято решение использовать ARTM в качестве базового алгоритма так как он оставляет исследователю много свободы в выборе регуляризаторов, из комбинации и коэффициентов.

1.3 Формализованное описание проблемы

Входные данные:

- коллекция новостей на русском языке на разные темы в сети интернет.

Выходные данные:

- обученная тематическая модель с настроенными регуляризатора-

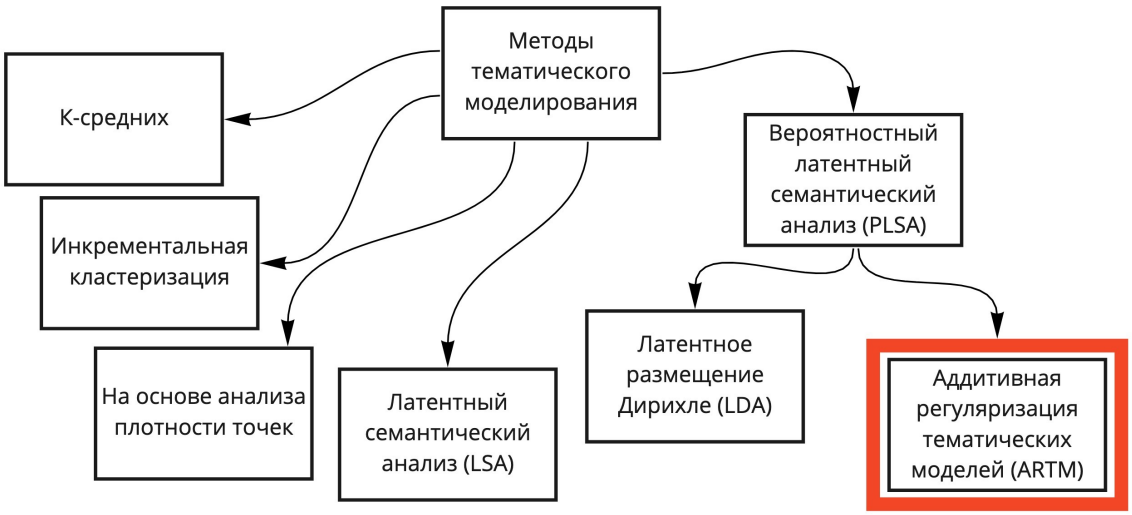

Рисунок 1.4: Методы тематического моделирования.

ми;

- список тем с образующими их словами.

Получение данных:

- парсинг новостных агрегаторов;
- парсинг крупных новостных сайтов.

Подготовка данных:

- удаление форматирования текста;
- исправление опечаток;
- слияние слишком коротких текстов;
- выделение терминов;

- приведение слов к нормальной форме (лемматизация);
- удаление слишком частых слов;
- удаление слишком редких слов.

1.4 Функциональные требования

Для решения задачи классификации и категоризации новостей на русском языке необходимо следующее:

- собирать новости из ресурсов сети Интернет;
- преобразовывать их в необходимый формат;
- создавать и обучать модель;
- провести параметризацию: подобрать наилучший комплект регуляризаторов, их параметров и коэффициентов;
- иметь возможность последующего повторного использования и дообучения модели.

2 Конструкторский раздел

Для проектирования программной реализации необходимо разбить весь процесс на этапы. При разбиении следует руководствоваться теми или иными, уже существующими, решениями для хранения и обработки информации.

В данной работе под коллекцией документов здесь и далее понимается массив новостей на русском языке. В сети Интернет документы хранятся в виде html-файлов. Для обучения модели необходимо получить данные в виде мешков слов. Кроме того, из-за большого объема данных необходимо разделить предварительную обработку каждого документа и последующий сбор обработанных данных в общую коллекцию для обучения.

Отдельным пунктом была рассмотрена структура анализируемых данных, чтобы при выборе средств реализации можно было использовать эту информацию.

Процесс создания тематической модели разбивается на следующие этапы:

- сбор данных;
- обработка данных;
- обучение модели;
- использование модели;
- оценка модели.

2.1 Структура анализируемых данных

Очевидно, что для работы решения необходимо хранить коллекцию новостей, где о каждом документе известны тема новости, текст новости, ссылка на html-файл новости в сети Интернет.

Так как данные обрабатываются подокументно, будет удобно иметь данные в обработанном виде рядом с сырыми, чтобы иметь возможность обрабатывать коллекцию по частям.

При описании структуры данных желательно предоставить возможность обновлять данные, так как со временем html документы на выбранном ресурсе могут меняться. Для этого необходимо хранить дату сохранения документов.

Кроме того, процесс обработки данных также может быть усовершенствован или изменен. Следовательно, также необходимо хранить дату обработки данных.

Так как все данные текстовые и однородные, для хранения выбрана таблица в базе данных со следующими полями:

- тема новости;
- текст новости;
- ссылка на html файл новости в сети Интернет;
- обработанный текст новости;
- дата сохранения документа;
- дата обработки данных.

Для организации сохранения всех новостей с выбранного ресурса необходимо отслеживать на какие страницы ресурса ведут уже обработанные страницы. Для этого создается еще одна таблица с данными: какая ссылка, на какую другую ссылку ведет. То есть создается таблица со следующими полями:

- ссылка-родитель;
- ссылка-ребенок.

Ограничения:

- Новости на русском языке.

2.2 Сбор данных

В аналитическом разделе были выделены несколько типов данных:

- предварительно подготовленные массивы новостей;
- новостные сайты;
- новостные агрегаторы.

Рассмотрим их детальнее.

Предварительно подготовленные массивы новостей

Обычно в таких массивах данных текст новостей и их заголовки уже очищены от форматирования и переносов, опечатки исправлены, а также удалена нетекстовая информация. При этом остаются следующие проблемы:

- слишком короткие тексты;
- слова в новостях не приведены к нормальной форме;
- не выделены словосочетания;
- много часто используемых слов и редко используемых слов;
- каждый такой массив данных оформлен по-своему, поэтому для работы с ним необходимо писать код, преобразующий коллекцию в удобный для модели формат.

Так как часть обработки уже выполнена, получить такой массив данных предпочтительнее, чем добывать данные из сети Интернет. Но стоит учесть, что найти такие массивы данных достаточно сложно. Необходимо опрашивать специалистов в этой области, изучать платформы сообществ по обработке естественного языка, анализировать архивы конференций.

Новостные сайты и агрегаторы

У данных, хранящихся в сети Интернет, существует большое количество недостатков: они не обработаны, текст хранится вперемешку с html кодом, содержит опечатки. Также из-за неорганизованности владельцев новостных сайтов, зачастую важные для последующего анализа данные (например, дата публикации, имя документа и т.д.) хранятся в разном виде за разные периоды времени, и поэтому их сложно извлечь.

С другой стороны, такой подход предоставляет практически безграничные возможности выбора тематики для последующего анализа.

Для извлечения таких данных необходим специальный софт, который анализирует указанный интернет ресурс, а также все ссылки, на которые ведут уже скачанные страницы. Отдельно стоит отметить,и что часть ссылок зачастую на новостных сайтах появляются динамически, после того, как посетитель сайта нажимает специальную кнопку или перематывает страницу до конца.

Также учитывая технические ограничения автора работы и то, что документов на выбранном ресурсе может быть много, необходимо, чтобы процесс анализа и сохранения новостей можно было остановить в любой момент и впоследствии продолжить с места остановки.

Так как данная задача довольно распространена существует библиотеки, частично или полностью решающие проблему получения данных. Однако, часто данные на сайтах хранятся в таком виде, что приходится модифицировать существующие решения.

2.3 Обработка данных

После того, как получены сырые данные, перед началом обучения модели, данные необходимо подготовить. Подготовка данных разбивается на два этапа:

- обработка документа (новости);
- формирование коллекции в формате, удобном для модели.

Обработка документа (новости)

В рамках этого этапа подготовки данных производится обработка по документам. В связи с техническими ограничениями необходим хранить дату обработки, чтобы иметь возможность при изменении алгоритма выполнить процесс подготовки текста повторно. Кроме того, так же, как и в случае с сохранением страницы сети Интернет, из-за того, что данных много, необходимо реализовать возможность подготовки новостей коллекции по частям, останавливая и запуская процесс в любой момент времени.

Подготовку данных по документам можно разбить на следующие этапы.

- **Очистка от форматирования и переносов.** В сыром виде текст новости часто перемешан с html кодом, специальными символами pdf файлов, часть слов разделены дефисов для переноса на новую строку.

- **Исправление опечаток.** Журналисты и редакторы могут не уследить за орфографической ошибкой, и обучаемая модель воспримет слово с ошибкой как отдельное редкое слово в коллекции.

- **Удаление нетекстовой информации:** рисунков, графиков, таблиц.

- **Приведение слов к нормальной форме.** Для английского языка используется **стемминг** (выделение неизменной части слова). Для данной работы лучше подходит **лемматизация**, так как новости на русском языке.

- **Выделение словосочетаний.** По умолчанию модель воспринимает каждое слово в тексте новости как отдельный термин. При выделении словосочетаний появляется возможность, обучая модель, относиться к ним как к цельным многословным терминам.

- **Удаление часто используемых слов.** Часто используемые слова встречаются в большом количестве тем, и их наличие в документе не может стать признаком того, какие именно темы затрагиваются в новости.

- **Удаление редко используемых слов.** Редко используемые слова (обычно встречающиеся меньше десяти раз за коллекцию) также не несут обычно никакой информации о принадлежности документа к той или иной теме.

Формирование коллекции в формате, удобном для модели

После того, как каждый документ обработан и представлен в виде мешка слов необходимо собрать все документы в одну коллекцию. В связи с техническими ограничениями потребуется возможность собирать такую коллекцию по частям на основании источника документа, даты скачивания, даты обработки. В зависимости от выбранной реализации модели так же следует привести данные в формат, необходимый для обучения модели. Конкретное представление данных выбрано в технологическом разделе. Формирование коллекции должно выполняться отдельно от обработки подокументно так как происходит непосредственно перед обучением модели и может зависеть от целей исследователя.

2.4 Обучение модели

Согласно выбранному алгоритму сначала модель обучается на подготовленных данных без регуляризаторов (как в рассматриваемом варианте алгоритма PLSA) до того момента как перплексия перестанет изменяться. Это будет означать, что слова достаточно хорошо и однозначно распределились по темам и осталась только задача тематизирования документов.

После этого в модель добавляются регуляризаторы по одному. Добавляя регуляризатор исследователь подбирает параметры регуляризатора, таким образом, что бы перплексия уменьшалась, разреженность матриц Φ и Θ увеличивалась, но при этом, что бы не достигала единицы. То же

самое касается остальных параметров. Частота тем должна стремиться к величине равное количество слов, поделенное на количество тем.

В данной работе были рассмотрены три регуляризатора. Ниже приводятся их формальные описания и интерпретации. В первых двух регуляризаторах реч идет о разреживании или сглаживании, так как при положительных коэффициентах τ при соответствующих регуляризаторах значения матриц сглаживаются, а при отрицательных разреживаться.

Разреживающий или сглаживающий регуляризатор матрицы словотема Φ:

$$R(\Phi) = \tau \sum_{t \in T} \sum_{w \in W} \ln \phi_{wt} \to max. \tag{2.1}$$

Разреживающий или сглаживающий регуляризатор матрицы тема-документ Θ:

$$R(\Theta) = \tau \sum_{d \in D} \sum_{t \in T} \ln \theta_{td} \to max. \tag{2.2}$$

Максимизация регуляризатора декоррелирования тем приводит к тому, что темы как столбцы матрицы Φ как можно меньше коррелируют друг с другом:

$$R(\Phi) = -\frac{\tau}{2} \sum_{t \in T} \sum_{s \in T \setminus t} \sum_{w \in W} \phi_{wt} \phi_{ws} \to max. \tag{2.3}$$

После анализа функционала решения была разработана диаграмма метода (рисунки 2.1 и 2.2).

2.5 Использование модели

При выборе реализации алгоритма в технологической части необходимо удостовериться, что любую построенную модель можно сохранить, а также загрузить в дальнейшем для последующего использования.

После загрузки модели, также должна быть возможность дообучения, чтобы сократить время на создание сложных моделей, также необходим функционал по оценке нового документа (новости) с помощью загруженной модели.

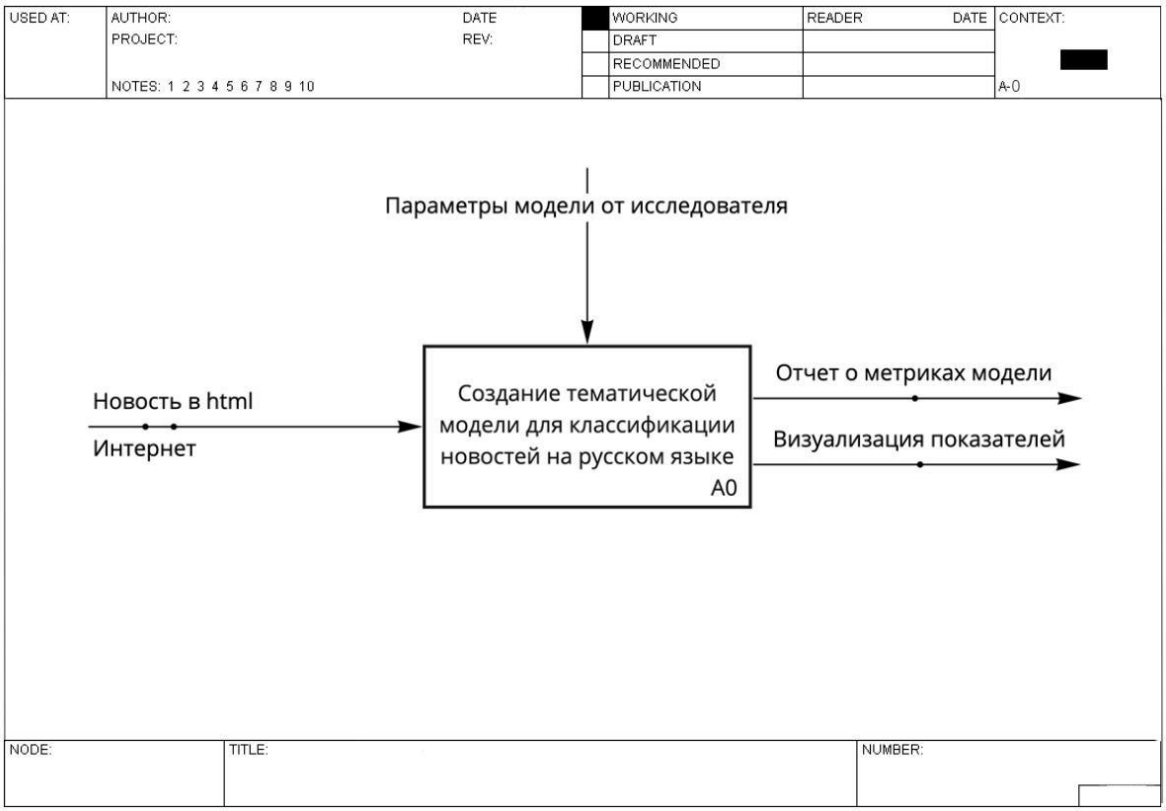

Рисунок 2.1: Концептуальная схема 1

2.6 Оценка модели

Для оценки полученных результатов были использованы следующие метрики:

- перплексия;
- разреженность матрицы слово-тема Φ;
- разреженность матрицы тема-документ Θ;
- средний контраст тем;
- средняя чистота тем;
- средний размер тем;
- 10 наиболее вероятных слов в теме.

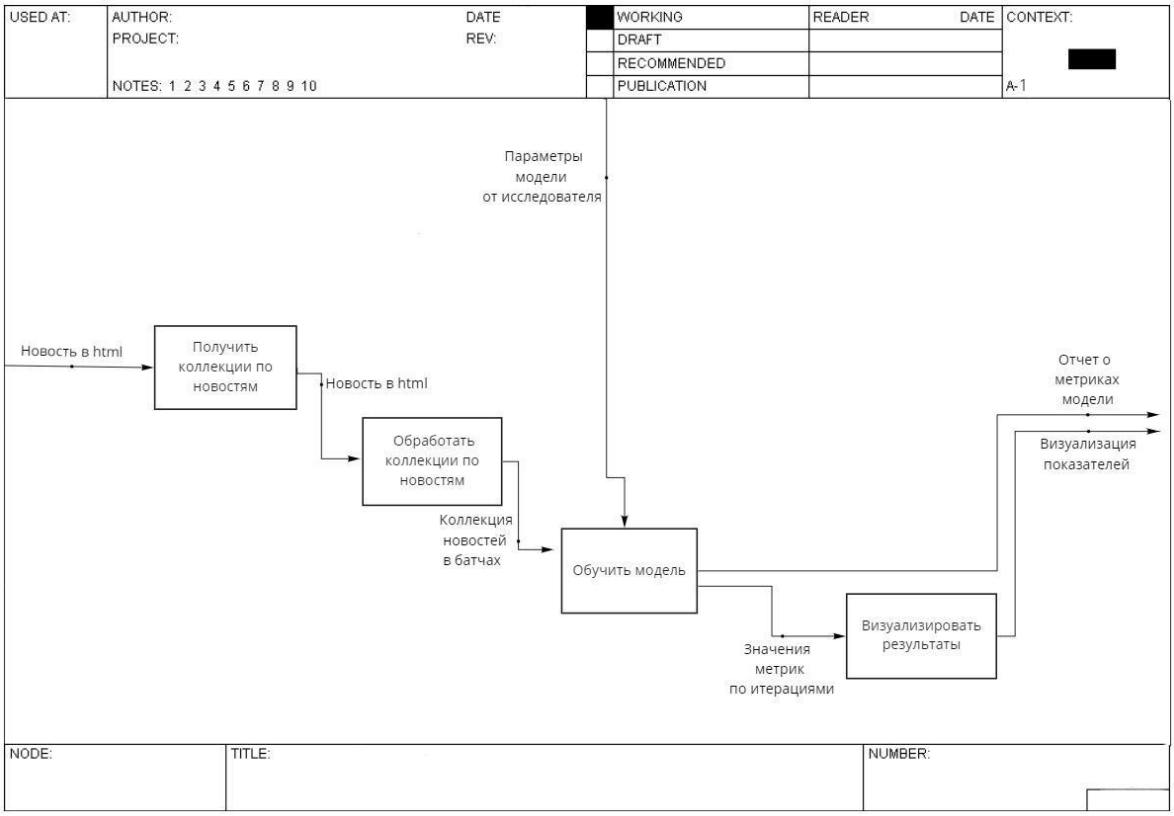

Рисунок 2.2: Концептуальная схема 2

Меньшая перплексия чаще всего означает, что темы более интерпретируемы, но сама по себе эта метрика ничего не значит. Она принимает значения от нуля до бесконечности и подходит только для сравнения моделей на дном и тех же данных, с одним и тем же количеством тем. Перплексия вычисляется по следующей формуле:

$$P(D;\Phi,\Theta) = \exp\Big(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw}\ln\sum_{t\in T}\phi_{wt}\theta_{td}\Big). \tag{2.4}$$

Разреженность матрицы слово-тема Φ изменяется от 0 до 1 и интерпретируется как процент значений в матрице Φ близких к нулю. Чем ближе значение этой метрики к единице, тем меньшим количеством слов описана тема и тем лучше становится интерпретируемость тем.

Разреженность матрицы тема-документ Θ изменяется от 0 до 1 и интерпретируется как процент значений в матрице Θ близких к нулю. Чем ближе значение этой метрики к единице, тем меньшим количеством тем описан документ и тем лучше становится интерпретируемость тем.

Следующие три метрики строятся на основании ядра темы. Пусть ядро темы определяется следующей формулой:

$$W_t = \{w \in W | p(t|w) > 0\} \tag{2.5}$$

Чев выше контраст у темы, тем меньше будет пересечений. То же самое справедливо для средней величины по всем темам. Контраст темы определяется формулой:

$$\frac{1}{|W_t|} \sum_{w \in W_t} p(t|w) \tag{2.6}$$

Чем выше чистота темы, тем легче человека ее интерпретировать. То же самое справедливо для средней величины по всем темам. Данная метрика определяется формулой:

$$\sum_{w \in W_t} p(w|t) \tag{2.7}$$

Средний размер тем определяется значением $|W_t|$ и в оптимальном случае стремится к отношению количества слов в коллекции к количеству тем.

10 наиболее вероятных слов в теме необходимы для того, что бы проверить интерпретируемость тем исследователем.

Так же для того, что бы облегчить анализ данных исследователю, при разработке решения необходимо написать процедуру вывода графического представления рассмотренных выше статистик.

2.7 Требования к реализации

На основании проведенного анализа в конструкторской части были сформированы следующие требования к реализации:

- необходимо использовать реляционную базу данных для хранения документов,

- необходимые сущности и поля в базе данных:

  - сущность страницы:
    * тема новости,
    * текст новости,
    * ссылка на html-файл новости в сети Интернет,
    * обработанный текст новости,
    * дата сохранения документа,
    * дата обработки данных;
  - сущность ссылки:
    * ссылка родитель,
    * ссылка ребенок;

- программный комплекс должен состоять из следующих отдельных процедур:
  - сбор информации
  - обработка информации подокументно
  - формирование коллекции для обучения
  - обучение модели
  - использование модели
  - вывод параметров модели для оценки

3 Технологический раздел

3.1 Выбор основного языка программирования

В качестве основного языка программирования, на котором был произведен сбор данных, подготовка данных, и управление моделями, был выбран Python3. Тематическом моделировании почти у всех реализаций есть интерфейс для этого языка. Язык прост в освоении. Исследователь уже знаком с его синтаксисом. Недостатком Python3 обычно называют его скорость, но при использовании верных библиотек все сложные вычисления выполняются с помощью C/C++, и проблемы со скоростью отсутствуют. Также из-за распространенность языка, следующим исследователям будет легко воспользоваться результатами данной работы.

3.2 Создание базы данных

Для данной работы рассматривается несколько самых известных реализаций реляционных баз данных:

- MySQL;
- SQLite;
- PostrgreSQL.

MySQL

Решение от компании Oracle. Очень популярное и мощное решение для малых и средних приложений, распространяемое под лицензией GNU General Public License. Преимущества этого решения - популярность и богатый функционал. Из недостатков можно отметить требовательность к ПО и относительно медленная разработка.

SQLite

Компактная встраиваемая СУБД. Движок SQLite представляет собой библиотеку, а не отдельно работающий процесс. При работе с этой СУБД обращения происходят напрямую к файлам. Среди недостатков можно отметить небольшое количество типов данных, доступных по умол-

чанию, отсутствие системы пользователей. Среди преимуществ хранение всей базы одним файлом.

PostrgreSQL

Самое профессиональное из всех трех рассмотренных решений. Обладает богатым функционалом. PostrgreSQL это не только реляционная СУБД, но также и объектно-ориентированная. К недостаткам можно отнести низкую производительность на простых операциях.

Выбр СУБД

Исходя из технических требований для этой работы выбор был остановлен на SQLite. Использование данного решения позволяет хранить все в одном файле и упрощает стартовую настройку решения. Ограниченность функционала и типов данных не будет проблемой в связи с простой структурой данных.

В качестве дополнительного функционала был реализован подсчет рейтинга страниц, который становится тем больше чем больше ссылок ведет на рассматриваемую страницу. Данный подход часто используется при сортировке страниц в поисковой выдаче. Эти данные могут пригодиться для процесса сохранения html-файлов. Можно модифицировать решение и в первую очередь скачивать страницы с наибольшим рейтингом.

3.3 Сбор данных

В работе используется два источника данных: новостные сайты и агрегаторы и предварительно подготовленные открытые массивы новостей. Работа с агрегатором новостей ничем не отличается от работы с сайтом новостного агентства.

Предварительно подготовленные массивы новостей

Самое сложное в получении готовых массивов данных - найти их. Для того, что бы поработать с большим объемом информации были проанализированы переписки

В сообществе Open Data Science были найдены ссылки на два массива данных:

- statmt.org - это не совсем подходит нам, тут новости короткие совсем. Но тоже скачал на всякий случай поиграться - тут суммарно 8,4 гигабайта чистого текста - уже скачены и лежат на моем компьютере;

- webhose.io - 290 000 новостей - уже скачены и лежат на моем компьютере.

После посещения конференции "Диалог"стало понятно где найти еще два массива данных:

- Lenta.ru

- Россия сегодня (РИА новости)

Новостные сайты и агрегаторы

Для начала сбора данных необходимо убедиться, что в базе данных присутствуют все необходимые сущности и поля для скачивания. Поэтому в начале программы реализован анализ состояния базы и если база не соответствует требованиям программы для сбора html-страниц - программа создает нужные сущности и поля.

Существует множество библиотек для анализа html страниц. Было принято решение воспользоваться самой популярной из них - «BeautifulSoup». Данная библиотека позволяет разобрать html файл на теги и производить операции по ним.

Так как на вход программа получает только корневую ссылку ресурса - необходимо, что бы все внутренние ссылки главной html страницы новостного ресурса так же добавлялись в список на проверку. Для того, что бы избежать смещения скаченных данных к определенной дате или теме - ссылки из списка запланированных на скачивание страниц

должны выбираться случайным образом.

Кроме того часть новостей может скрываться за кнопками вида «Показать еще» и действиями пользователя (например перемотка страницы новостей). Для того, что бы выполнить требование, по которому программу сбора данных можно остановить в любой момент, что бы потом продолжить с того же места необходимо записывать в базу html-файл каждой обработанной страницы.

Для того, что бы пользователю было понятно, что процесс протекает нормально принято решение каждые 50 обработанных страниц выводить промежуточную статистику в терминал. При каждом сохранении новости записывается дата сохранения, что бы в последствии данные в базе можно был сравнивать с данными по ссылке и обновлять при необходимости.

3.4 Обработка данных

Обработка данных разделена на два этапа: подокументная обработка и подготовка коллекции для обучения модели. В обработке по документам необходимо из html файла получить мешок слов и сохранить его в базе в соответствующем поле. При подготовке коллекции к обучению необходимо собрать из базы и приготовить данные в том виде, в котором требует реализация выбранного алгоритма (выбор реализации алгоритма приведен ниже).

Обработка подокументно

Обработка документа содержит следующие этапы:

- преобразование html кода в текст;
- леммирование слов;
- преобразование текста в формат vowpal wabbit.

где vowpal wabbit - тип представления данных в виде мешков слов по документам [10].

При преобразовании html кода в текст используется рассмотренная

выше популярная библиотека «BeautifulSoup». Исследователем устанавливается какие теги новостной ресурс использует для хранения заголовка и текста статьи. Программа настраивается в соответствии с этим выявленным шаблоном. Все что находится внутри настроенных тегов очищается от html разметки и сохраняется в виде текста в базу с документами в соответствующие записи. Этот процесс вынесен в отдельную процедуру и так же как и процесс сохранения страниц может быть в любой момент остановлен и в последствии запущен снова.

После того как получены данные в виде текста на русском языке производится леммирование слов и преобразование в формат vowpal wabbit. В процессе удаляются все слова на английском языке, как не несущие большой значимости для модели. Слова, прошедшие леммирование сохраняются в соответствующее поле в базе через пробел. Для леммирования выбран решение pymystem3 от Yandex так как но хорошо зарекомендовало себя в ранних исследованиях автора работы.

Подготовка коллекции

Подготовка коллекции содержит следующие этапы:

- выгрузка из базы документов в формате vowpal wabbit в текстовый файл;
- преобразование текстового файла в формате vowpal wabbit в батчи;
- удаление слов, которые использовались меньше n_{fmin} раз во всей коллекции;
- удаление слов, которые использовались больше n_{fmax} раз за всю коллекцию;
- удаление слов, которые использовались в n_{fpd} проценте документов.

Величины n_{fmin}, n_{fmax} и n_{fpd} определяются исследователем.

Перед следующим этапом необходимо выгрузить все необходимые

для обучения документы, прошедшие подокументную обработку в отдельный текстовый файл в формате vowpal wabbit. После чего этот файл преобразуется в батчи методом класса ARTM, встроенным в выбранную реализацию алгоритма ARTM (рассмотрена ниже).

3.5 Обучение модели

Из доступных реализаций ARTM была выбрана библиотека BigARTM с открытым кодом и эффективной потоковой параллельной реализацией.

После инициализации модели

```
model_artm.initialize(dictionary=dictionary)
```

добавляются необходимые статистики для оценки и проводится ее обучение до тех пор пока перплексия не перестанет изменяться

```
model_artm.fit_offline(
    batch_vectorizer=batch_vectorizer,
    num_collection_passes=50
)
```

В данном примере по коллекции будет сделано 50 проходов, после чего необходимо проанализировать результаты. Если перплексия не сошлась - необходимо продолжить обучать модель.

После того как обучение на PLSA закончено - можно добавлять регуляризаторы. Первым регуляризатор разреживает матрицу Φ то есть увеличивает количество нулевых и почти нулевых значений в этой матрице.

```
if (
    params['SparsePhi']['name']
    not in
    list(model_artm.regularizers.data)
):
    model_artm.regularizers.add(
        artm.SmoothSparsePhiRegularizer(
            name=params['SparsePhi']['name']
        )
```

```
10      )
model_artm.regularizers[params['SparsePhi']['name']].tau
12      = params['SparsePhi']['tau']
```

Процесс обучения повторяется до тех пор пока не сойдется перплексия. Исследователь изучает результаты и, если получил достаточный прирост по параметру, добавляется следующий регуляризатор схожим образом. Если результат не достаточно хорош - исследователь увеличивает по модулю коэффициент при соответствующем регуляризаторе. Если оцениваемый параметр привел к тому, что модель выродилась - значение коэффициента при регуляризаторе уменьшается по модулю.

Таким же образом последовательно добавляются еще два, рассмотренных выше регуляризатора. Один из них увеличивает разреженность матрицы Θ, а второй делает темы более различными.

3.6 Использование модели

После того, как все три регуляризатора добавлены и модель обучена - ее можно использовать для оценки новых документов (например новую написанную новость на сайте).

```
model_artm.save("news_model_0_0")
```

Так же ее можно сохранить методом, встроенным в реализацию BigARTM для последующей загрузки.

```
1 test_theta_matrix = model_artm.transform(
      batch_vectorizer=test_batch_vectorizer
3 )
```

3.7 Оценка модели

Для оценки модели была реализован функционал, выводящий всю необходимую статистику в графическом представлении. Пример вывода на рисунке 3.1.

3.8 Подготовка к запуску

Для запуска решения необходимо установить:

- ipython notebook версии 7.5.0 и выше;
- библиотеку для тематического моделирования BigARTM для python3 версии 0.9.0 и выше;
- библиотеку для работы с html кодом BeautifulSoup версии 4.7.1 и выше;
- библиотеку для лемматизации слов pymystem3 для python версии 0.2.0 и выше.

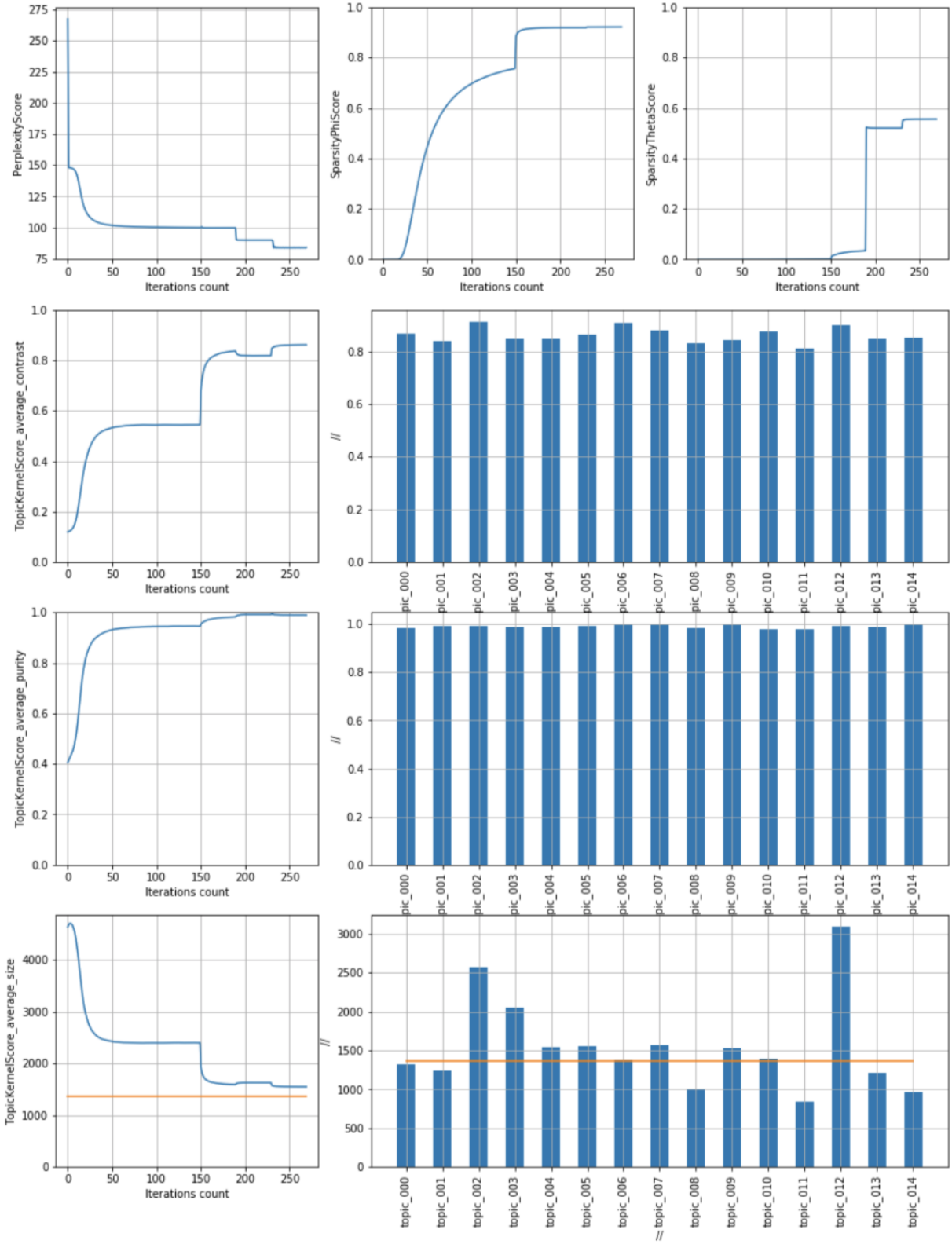

Рисунок 3.1: Пример визуализации результатов модели.

4 Исследовательский раздел

В рамках исследования была проведена апробация метода и были разработаны рекомендации.

4.1 Апробация метода

Подготовка данных

Для исследования было принято решение использовать 3 блока данных:

- 23 000 записей с сайта zona.media
- 1 000 000 записей с сайта ria.ru
- 24 000 случайно выбранных записей из 1 000 000 с сайта ria.ru

Подготовка двух блоков данных по 23 и 24 тысячи записей происходила в один поток. Для обработки 1 миллиона записей была использована многопоточность. Записи были обработаны в 16 параллельных потоков.

Так же были удалены слова, которые использовались меньше 10 раз и больше 2000 раз во всей коллекции, а так же слова, которые встречаются только в 0.01% документов.

Подбор базовых значений коэффициентов регуляризации

Что бы получить первые результаты исследования необходимо определить хотя бы примерно центры диапазонов коэффициентов при регуляризаторах для последующего их уточнения. После каждой попытки анализируется результат. Проверяется достиг ли исследователь своей цели по соответствующему параметру и не выродилась ли при этом модель.

Так как регуляризаторы добавляются последовательно, поиск базовых значений так же реализован последовательно. Сначала на коллекции обучается модель PLSA до тех пор, пока модель не сойдется. После этого добавляется первый регуляризатор, разреживающий матрицу слова-темы Φ.

После обучения нескольких десятков первых моделей на 23 тысячах

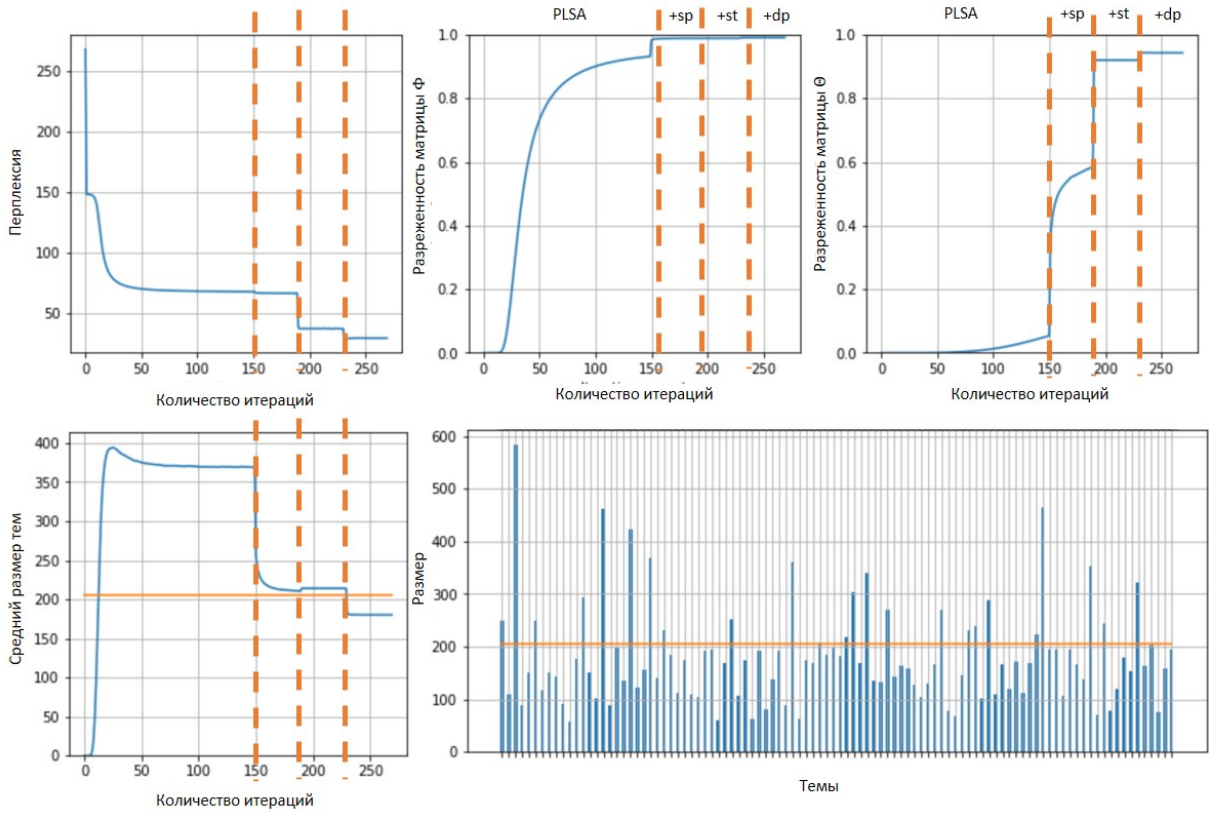

Рисунок 4.1: Метрики для модификации.

записей с zona.media были определены базовые значения коэффициентов регуляризации:

- регуляризатор, разреживающий матрицу слова-темы Φ: -3;
- регуляризатор, разреживающий матрицу темы-документы Θ: -5;
- регуляризатор, увеличивающий разницу между ядрами тем: 25 000 000.

Название	Перплексия	Разреженность матрицы слово-тема	Разреженность матрицы тема-документ	Средний контраст тем	Средняя чистота тем	Средний размер тем
0_0_zona_23000_15t_plsa	5,8766	0,5250	0,0013	0,5531	0,9571	210
0_0_zona_23000_15t_plsa+sp	5,8674	0,8797	0,1112	0,6403	0,9842	187
0_0_zona_23000_15t_plsa+sp+st	5,5030	0,8798	0,7139	0,5578	0,9987	217
0_0_zona_23000_15t_plsa+sp+st+dp	5,4354	0,9472	0,7943	0,9830	0,9995	93
0_0_zona_23000_15t_plsa+(sp+st+dp)	5,4455	0,9423	0,7993	0,9523	0,9990	99
...	...	...	...	...	...	...
1_1_zona_23000_10t_plsa	6,0201	0,4458	0,0005	0,5395	0,9670	326
1_1_zona_23000_10t_plsa+sp+st+dp	8,6023	0,9540	0,8853	1,0000	1,0000	83
...	...	...	...	...	...	...
3_1_zona_23000_15t_plsa	1355,7546	0,6934	0,0000	0,5372	0,9306	2339
3_1_zona_23000_15t_plsa+sp+st+dp	1208,1539	0,9220	0,3538	0,8647	0,9861	1473
3_1_zona_23000_15t_plsa+(sp+st+dp)	1250,5999	0,9124	0,3297	0,3297	0,9767	1601

Рисунок 4.2: Стадия исследования коэффициентов регуляризации.

Из рисунка 4.2 видно, что одновременное включение всех регуляризаторов приводит к худшим результатам, чем стратегия с подключением регуляризаторов по одному, и обучения последовательно до сходимости.

Определение оптимального количества тем

После того, как были найдены приблизительные значения коэффициентов регуляризации было принято решение сменить коллекцию новостей на 24 тысячи случайно отобранных документов из 1 миллиона с сайта ria.ru, что бы темы стали еще более различны.

Было проведено исследование и построены следующие модели (рисунок 4.3):

Название	Перплексия	Разреженность матрицы слово-тема	Разреженность матрицы тема-документ	Средний контраст тем	Средняя чистота тем	Средний размер тем
...	...	...	...	...	...	...
4_0_ria_24000_15t_plsa	100,0766	0,7567	0,0018	0,5444	0,9453	2397
4_0_ria_24000_15t_plsa+sp+st+dp	84,1027	0,9225	0,5561	0,8611	0,9899	1549
4_1_ria_24000_50t_plsa+sp+st+dp	33,3470	0,9817	0,9034	0,9891	0,9999	372
4_2_ria_24000_100t_plsa+sp+st+dp	29,1980	0,9912	0,9436	0,9996	0,9999	180
4_3_ria_24000_150t_plsa+sp+st+dp	25,3347	0,9941	0,9611	1,0000	1,0000	120
...	...	...	...	...	...	...

Рисунок 4.3: Стадия исследования коэффициентов регуляризации.

По каждой из модели были проанализированы топ 10 слов каждой из тем и проведена оценка. Каждая тема было отнесена к одной из двух категорий: тема состоит из характерных слов, понятно о чем она и для нее можно придумать название и тема состоит из несвязанных между собой слов, название придумать сложно.

Пример хороших тем:

- **Наука**: ученый, исследование, коллега, примерно, лаборатория, эксперимент, изучение, анализ, изучать, метод.

- **Футбол**: футбольный, футболист, зенит, спартак, динамо, цска, поле, локомотив, болельщик, забивать.

- **Литература**: книга, автор, писатель, написать, название, поэт, литература, библиотека, рождаться, литературный.

- **Деньги**: продажа, кредит, капитал, сделка, актив, сбербанк, доля,

кредитный, банковский, прибыль.

Пример плохой темы:

- подробно, памятник, письмо, наследие, **охрана**, **спецслужба**, справка, реставрация, **сноудена**, запрос.

4.2 Рекомендации

В ходе работы были выделены рекомендации.

- Процесс обработки данных желательно выстраивать в несколько потоков, при этом необходимо решить проблему с заблокированной базой при одновременном обращении.

- При редукции словаря необходимо обрезать минимальное количество слов, чтобы не вырезать ключевые слова.

- При обучении модели ARTM следует сначала пройти по коллекции до того, как сойдется (перестанет изменяться) перплексия, и только после этого начать добавлять регуляризаторы по одному.

- Регуляризаторы стоит добавлять по одному. После чего продолжать обучение, пока модель не сойдется.

- Если у темы слишком большой размер по сравнению с другими темами – необходимо проверить ее на фоновые слова, характерные для любой темы.

Заключение

В результате данной работы был разработан метод тематического моделирования новостей на русском языке

Были решены следующие задачи:

- Проанализированы существующие методы тематического моделирования и выбран базовый для классификации новостей на русском языке;
- разработан программный продукт для сбора новостей на русском языке и подготовки данных в виде отдельных сервисов;
- обучена тематическая модели для новостных текстов;
- проведена параметризация метода;
- проведена апробации метода;
- составлены рекомендаций о применимости предложенного метода.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД СЕРВИСА ОЧИСТКИ

поправ
й

```
import sqlite3
import datetime
import multiprocessing as mp
import re
import time
import random as rnd

from pymystem3 import Mystem

WORKER_NUM = 2

#MODE = 'body_text'
MODE = 'html'

DB_ADRESS = 'application/data_raw/spider.sqlite'
#DB_ADRESS = 'spider.sqlite'
PART_LIMIT = 400

def my_rnd():
    return rnd.uniform(60.0, 120.0)

def clean_one_part(i):

    mystem = Mystem()

    conn = sqlite3.connect(DB_ADRESS)
    cur = conn.cursor()

    data = [0]
    while len(data) != 0:
        try:
```

```
            cur.execute("SELECT count() FROM Pages")
33          print("process_{0:0>3}: ".format(num) +
                '{0:30}'.format("rec:"),
                        '{0:7}'.format(cur.fetchone()[0]))
35          cur.execute("SELECT count() FROM Pages
              WHERE vowpal_wabbit_date is not NULL")
            print("process_{0:0>3}: ".format(num) +
                '{0:30}'.format("rec:"),
37                      '{0:7}'.format(cur.fetchone()[0]))
            #print()
39
            cur.execute(
41              "SELECT url, "+MODE+", vowpal_wabbit,
                   vowpal_wabbit_date, title_text FROM
                   Pages "
                + "WHERE vowpal_wabbit_date is NULL "
                   + "and "+MODE+" is not NULL "
43              + "and id IN (SELECT id FROM Pages
                   ORDER BY RANDOM() "
                + "LIMIT "
45              + str(PART_LIMIT)
                + ")"
47          )
            data = cur.fetchall()
49      except:
            data = [0]
51          time.sleep(my_rnd())
            conn = sqlite3.connect(DB_ADRESS)
53          cur = conn.cursor()
            continue
55
        for i in range(len(data)):
57
```

```
            data[i] = list(data[i])
59
            if MODE == 'html':
61              mode_text = data[i][1].decode()
            else:
63              mode_text = data[i][1]

65          # body_text/html + title_text
            text = mode_text + ' ' + data[i][4]
67
            if len(text) > 30:
69


71              s = text

73              if len(s) < 30:
                    raise Exception()
75              res = ''

77              res = '|text '

79              lemmas = mystem.lemmatize(s)

81              for l in lemmas:
                    l_strip = l.strip()
83
                    if (
85                      len(l_strip) > 3
                        and re.match("^[A-za-z]*$",
                            l_strip)
87                      and len(l_strip) < 30
                    ):
89                      res = res + l_strip + ' '
```

```
91                      vw_text = res

93                      data[i][2] = vw_text

95                  data[i][3] =
                        int((datetime.datetime.utcnow() -
                                              datetime.datetime(1970,
                                                  1,
                                                  1)).total_seconds())
97
            try:
99                  for i in range(len(data)):
                        cur.execute("UPDATE Pages SET
                            vowpal_wabbit=?,
                            vowpal_wabbit_date=? WHERE url=?", (
101                             data[i][2],
                                data[i][3],
103                             data[i][0],
                            ))
105                 conn.commit()
            except:
107                 data = [0]
                    time.sleep(my_rnd())
109                 conn = sqlite3.connect(DB_ADRESS)
                    cur = conn.cursor()
111                 continue

113 try:

115     print()
        print()
117     print()
```

```
        print()
119
        for num in range(WORKER_NUM):
121         mp.Process(target=clean_one_part, args=(
                num,
123         )).start()
            time.sleep(rnd.uniform(1.0, 2.0))
125
    except KeyboardInterrupt:           # Ctrl+C
127     print()
        pass
```

Список литературы

[1] Yang James Allan Jaime Carbonell George Doddington Jonathan Yamron Yiming. Topic Detecion and Tracking Pilot Study. Final Report. Proceedings of the Broadcast News Transcription and Understanding Workshop, 1998. 167 с.

[2] R. Allan J. Lavrenko F. Malin D. Swan. Detections, bounds and timelines: UMass and TDT-3. In Proceedings of Topic Detecion and Tracking Workshop, pages 167-174, 2000. 177 с.

[3] С. Клышинский Э. Метод кластеризации на основе анализа плотности точек. МИЭМ НИУ ВШЭ, 2014. 151 с.

[4] Большакова Е. И. Клышинский Э. С. Ландэ Д. В. и др. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. МИЭМ НИУ ВШЭ, 2011. 106 с.

[5] Dumais S. T. Furnas G. W. Landauer T. K., S. Deerwester. Using latent semantic analysis to improve information retrieval. In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 1988. 281 с.

[6] Hofmann Thomas. Probabilistic Latent Semantic Analysis. UAI, 1999. 195 с.

[7] Hofmann Thomas. Probabilistic Latent Semantic Indexing. SIGIR, 1999. 196 с.

[8] Jordan D. Blei A. Ng M. Latent Dirichlet allocation. Journal of Machine Learning Research, 2003. 187 с.

[9] Frei O. Apishev M. Parallel non-blocking deterministic algorithm for online topic modeling. Analysis of Images, Social networks and Texts., 2016. 132 с.

[10] VowpalWabbit. Vowpal Wabbit is a machine learning system [Электронный ресурс]. URL: `https://github.com/VowpalWabbit/vowpal_wabbit`. 2018. [дата обращения: 12.06.2019].