

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  
кандидата в бакалавры

Научный руководитель:  
доцент, кандидат технических наук  
Клышинский Эдуард Станиславович

Консультант:  
старший преподаватель  
Волкова Лилия Леонидовна

Москва — 2019

Заменить эту страницу на подписанное ТЗ

Заменить эту страницу на подписанный календарный план

## Реферат

Сделать что бы тут был заголовок, но не включался в оглавление

Объект исследования и разработки

Цель и задачи работы

Метод и методология проведения работы

Результаты работы

Основные конструктивные, технологические и технико-эксплуатационные характеристики объекта исследования

Степень внедрения

Рекомендации по внедрению

Область применения

Экономическая эффективность или значимость работы

Прогнозы и предположения о возможных направлениях развития объекта исследования

## Перечень условных обозначений

Сделать что бы тут был заголовок, но не включался в оглавление

Добавить условные обозначения (только если встречается более 3 раз)

// Документ -

// Тема -

## Оглавление

<b>1</b>	<b>Введение . . . . .</b>	<b>8</b>
1.1	// актуальность выбранной темы . . . . .	8
1.2	// подвести к предметной области и задаче . . . . .	8
<b>2</b>	<b>Аналитический раздел . . . . .</b>	<b>9</b>
2.1	Постановка задачи . . . . .	9
2.2	Задачи тематического моделирования . . . . .	9
2.3	Существующие методы . . . . .	10
2.3.1	Основы кластеризации и классификации документов	11
2.3.2	Латентный семантический анализ (LSA) . . . . .	12
2.3.3	Вероятностный латентный семантический анализ (PLSA) . . . . .	13
2.3.4	Латентное размещение Дирихле (LDA) . . . . .	15
2.3.5	Аддитивная регуляризация тематических моделей (ARTM) . . . . .	17
2.3.6	Решение задачи максимизации регуляризованного правдоподобия . . . . .	18
2.3.7	Выбор алгоритма . . . . .	19
2.3.8	Формализованное описание проблемы . . . . .	19
2.4	// Функциональные требования к . . . . .	20
<b>3</b>	<b>Конструкторский раздел . . . . .</b>	<b>21</b>
3.1	Общие сведения . . . . .	21
3.2	Структура данных . . . . .	21
3.3	Сбор данных . . . . .	22
3.4	Обработка данных . . . . .	24
3.5	Обучение модели . . . . .	26
3.6	Использование модели . . . . .	27
3.7	Оценка модели . . . . .	27
3.8	Требования к реализации . . . . .	27
<b>4</b>	<b>Технологический раздел . . . . .</b>	<b>29</b>

4.1	Выбор средств программной реализации и разработка . . .	29
4.1.1	Выбор основного языка программирования . . . . .	29
4.1.2	Создание базы данных . . . . .	29
4.1.3	Сбор данных . . . . .	30
4.1.4	Обработка данных . . . . .	32
4.1.5	Обучение модели . . . . .	33
4.1.6	Использование модели . . . . .	34
4.1.7	Оценка модели . . . . .	34
4.2	Тестирование . . . . .	34
4.3	Подготовка к запуску . . . . .	34
<b>5</b>	<b>Экспериментальный раздел . . . . .</b>	<b>35</b>
5.1	// эксперименты и их результаты . . . . .	35
5.1.1	// проводим апробацию . . . . .	35
5.1.2	// анализируем результаты . . . . .	35
5.2	// качественное и количественное сравнение с аналогами .	35
5.3	// даём рекомендации о применимости метода/софта . . .	35
<b>6</b>	<b>Заключение . . . . .</b>	<b>36</b>
6.1	// отчитаться по каждому пункту тз/по каждой задаче и цели . . . . .	36
6.2	// сказать про перспективы (мы все уже не умрём) . . . .	36
<b>7</b>	<b>Список источников . . . . .</b>	<b>37</b>
7.1	// Разобрать . . . . .	37
7.2	// Датасеты . . . . .	37
<b>8</b>	<b>Приложения . . . . .</b>	<b>38</b>
8.1	// . . . . .	38

## 1 Введение

2 - 3 страницы

Выключить нумерацию введения (Ирина присылала как)

Костя пошарил свою работу - глянуть что тут должно быть

Пронумеровать все формулы как положено

1.1 // актуальность выбранной темы

1.2 // подвести к предметной области и задаче



## 2 Аналитический раздел

25 – 30 страниц

### 2.1 Постановка задачи

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

Для достижения этой цели необходимо выполнить следующие основные задачи:

- Анализ существующих решений и выбор базового алгоритма тематического моделирования для классификация/категоризация новостей на русском языке
- Разработка программного продукта для сбора новостей на русском языке и подготовки данных для последующего анализа
- Подбор методов улучшения алгоритма и значений их параметров
- Обучение модели
- проведение эксперимента

### 2.2 Задачи тематического моделирования

проводится анализ предметной области

выделяется основной объект исследования

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 и категоризация документов** - необходимо присвоить каждому документу соответствующие классы. Если классы имеют иерархическую структуру - говорят о категоризации.

- **Автоматическое аннотирование документов** - составление краткого обзора на документ, используя наиболее важные фразы.
- **Автоматическая суммаризация коллекций** - решение предыдущей задачи для большой коллекции документов.
- **Тематическая сегментация документов** - разбиение длинного документа 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 так же 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 систем рекомендации на основании данных о просмотренных документах пользователем и его активности.

## 2.3 Существующие методы

обзор существующих путей/методов/решений и алгоритмов решения

? Графические модели

? pLDA

?JPM - Join Probabilistic Model, АНММ - Aspect Hidden Markov Model, АТМ - Autor-Topic Model, СТМ - Correlated Topic Model

? dvl.kiev.ua - Дмитрия Владимировича Ландэ

обосновывается необходимость разработки нового или адаптации существующего метода или алгоритма

выводы из обзора (лучше сравнительную таблицу) отсюда актуальность (никто не делал так/улучшаем то-то и то-то)

рассмотреть математику используемых регуляризаторов

добавить математику мультимодальности

### 2.3.1 Основы кластеризации и классификации документов

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.

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:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D),$$

где

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

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

пояснить что такое freq

Еще вариант из работы [1]:

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

где  $||\vec{d}||$  - номер вектора представляющего документ  $D$ . Еще варианты модификаций TF-IDF из работ [1]:

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

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

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

Для определения расстояния в таком представлении данных использовались различные метрики: дивергенция Кульбака-Лейблера, косинус и другие. В первых работах для решения таких задач использовались алгоритмы кластеризации: метод К-средних, инкрементальная кластеризация 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 выше примере про новость финансирования спорта мы увидели, что в одном документе затрагиваются сразу две темы и футбол и финансы. При таком подходе эти данные теряются.

### 2.3.2 Латентный семантический анализ (LSA)

Dumais et al [1] в 1988 году предложил метод LSA. Суть метода в том, что бы спроецировать документы и термины в пространство более низкой размерности. Для этого анализируется совместная встречаемость слов (терминов) в документах. Таким образом задача состоит в том, что бы часто встречающиеся вместе термины были спроецированы в одно и то же измерение семантического пространства.

Дописать что надо по минимуму, что бы был понятен PLSA

### 2.3.3 Вероятностный латентный семантический анализ (PLSA)

В 1999 году Томасом Хофманом был предложен метод вероятностного латентного семантического анализа (PLSA) [1]. 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 чем появляется возможность дообучения моделей и упрощается подбор параметров.

Для лучшего понимания алгоритма рассмотрим подробнее 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 затрагивает 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 себе любую новость как смесь разных тем. А каждое слово, встречающееся в новости как результат срабатывания события упоминания этого слова журналистом из тем, на которые он опирался создавая документ.

«процесс порождения текстового документа вероятностной тематической моделью.png»

Вставить картинку

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 не большое число образующих слов.
- В документе используется не большое 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  $n_t$  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  $d$
- $\hat{p}(w|d) = \frac{n_{dw}}{n_d}$  - 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:

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

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:

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

«матричное разложение.png»

Вставить картинку

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

#### Понятие стохастической матрицы

Теперь, воспользовавшись принципом максимума правдоподобия с ограничениями на элементы стохастических матриц, если максимизировать логарифм правдоподобия получается:

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

#### 2.3.4 Латентное размещение Дирихле (LDA)

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

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

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:

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

В 2003 году Дэвидом Блеем, Эндрю Энджи и Маклом Джорданом был предложен метод латентного размещения Дирихле (LDA) [1]. На дан-

ный момент это одна из самых цитируемых статей по тематическому моделированию. Они предложили решать задачу со следующим регуляризатором:

$$R(\Phi, \Theta) = \sum_{t,w} (\beta_w - 1) \ln \phi_{wt} + \sum_{d,t} (\alpha_t - 1) \ln \theta_{td},$$

$$\beta_w > 0,$$

$$\alpha_t > 0,$$

где  $\beta_w$  и  $\alpha_t$  - 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

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

Дивергенция Кульбака-Лейблера обладает следующими свойствами:

- Неотрицательность:

Добавить формулу

- Несимитричность:

Добавить формулу

Дивергенция Кульбака-Лейблера связана с максимумом правдоподобия:

Добавить формулу

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

Минимизация дивергенции Кульбака-Лейблера эквивалентна максимизации правдоподобия.



Пусть

Добавить формулу

- некоторый вектор над словарем  $W$ . Со словами  $w$ .

При

Добавить формулу

вероятность

Добавить формулу

этого слова по темам будет сглаживаться приближаясь к

Добавить формулу

:

Добавить формулу

При

Добавить формулу

значение

Добавить формулу

наоборот будут разреживаться, удаляясь от

Добавить формулу

к нулю :

Добавить формулу

то есть в матрице  $\Phi$  будет больше нулевых элементов или близких к нулю.

### 2.3.5 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 :

Добавить формулу

Где :

Добавить формулу

- коэффициенты регуляризации, а

Добавить формулу

- 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.

### 2.3.6 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:

Добавить формулу

Где

Добавить формулу

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

Для получения результата необходимо итерационно выполнять Е-шаг и М-шаг до достижения требуемой точности.

Е-шаг :

Добавить формулу

Где

Добавить формулу

М-шаг :

Добавить формулу

Где

Добавить формулу

Этот процесс можно организовать параллельно, если обновлять мат-

рицу  $\Phi$  по порциям, после анализа очередного пакета документов. Обычно уже после просмотра нескольких первых десятков тысяч документов матрица  $\Phi$  получается уже устоявшаяся и остается только тематизировать остальные документы

### 2.3.7 Выбор алгоритма

Добавить выбор алгоритма

В данной работе рассматривается задача классификации и категоризации 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 так же создаются по ходу формирования модели.

### 2.3.8 Формализованное описание проблемы

Откуда брать данные и какие они бывают

описание критериев сравнения нескольких реализаций метода или алгоритма

Входные данные:

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

Выходные данные:

- Обученная тематическая модель с настроенными регуляризаторами.
- Список тем с образующими их словами
- Названия тем

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

## 2.4 // Функциональные требования к

Что мы хотим получить (это и будет "мостиком" к конструкторской)

Для решения задачи классификации и категоризации новостей на русском языке необходимо, чтобы программная реализация собирала новости из ресурсов сети Интернет, обрабатывала их в формат, необходимый для работы модели, создавала и обучала модель. При обучении необходимо подобрать наилучший комплект регуляризаторов, их параметров и коэффициентов. Также должна быть возможность последующего повторного использования и дообучения модели.

## 3 Конструкторский раздел

25 – 30 страниц

написать что такое коллекции

### 3.1 Общие сведения

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

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 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

### 3.2 Структура данных

добавить высокоуровневый тип данных, ограничения

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 обрабатываются данные по документам, 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 так же может быть усовершенствован или изменен. Следовательно так же 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

### 3.3 Сбор данных

как будем извлекать данные (без кода пока)

Мой написанный код для парсинга

Уже предварительно собранные открытые данные

<https://newspaper.readthedocs.io/en/latest/> - возможный инструмент для парсинга

25 500 новостей (там суммарно 9 000 000 слов - я посчитал) за все время существования media.zone (я сам написал парсер, могу его же натравить на любой другой новостной ресурс) - уже скачены и лежат на моем компьютере

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

Можно сделать сервис на РИА новости

Можно сделать сервис на агрегаторы новостей

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 остается проблема слишком коротких текстов, слова в новостях не приведены к нормальной форме, не выделены словосочетания, много слишком часто используемых слов и слишком редко используемых слов. Также отдельной проблемой является то, что 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 дан-

ных, 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 в перемешку 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 сложно 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 так же все ссылки, на которые ведут уже скаченные 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 в последствии 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.

## **3.4 Обработка данных**

рассказать, что такое лемматизация

Предобработка текста: лемматизация, удаление стоп-слов, ngrams



Используем модальности (дата публикации, ссылки на другие документы, авторы)

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.** Например, рисунков, графиков, таблиц.
- **Приведение слов к нормальной форме.** Для английского языка используется **стемминг**. Для исследования 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 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 меньше десяти раз за коллекцию) также не несут собой обычно никакой информации о принадлежности документа к той или иной теме.

## Формирование коллекции в формате, удобном для модели

Если будут использоваться модальности - дописать сюда про них

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 поддокументно так как происходит непосредственно перед обучением модели и может зависеть от целей исследователя.

## 3.5 Обучение модели

разработка метода

Базовый алгоритм: ARTM ([bigartm.readthedocs.io](http://bigartm.readthedocs.io))

Используем модальности (дата публикации, ссылки на другие документы, авторы)

? Используем производные от статьи данные по различным алгоритмам (записываем в модальности) - алгоритмы еще не выбраны

IDEF0 метода

### 3.6 Использование модели

Можно попробовать обучаться на месяце/неделе/дне (и это в теории можно вынести в эксперимент) и выдавать как меняются темы

решить иерархически ли хотим строить темы или многое ко многим

Дообучение

Оценка документа

### 3.7 Оценка модели

как будем оценивать (без кода)

Перплексия - написать что плохая метрика

Разбиение на 2 части и замеры разницы оценки - устойчивость - Через предложение разбивать статью можно попробовать

Толока - описание теста - выбрать лишнее слово, подумать что еще можно

### 3.8 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

## 4 Технологический раздел

20 - 25 страниц

обоснованный выбор средств программной реализации

описание основных (нетривиальных) моментов разработки

### 4.1 Выбор средств программной реализации и разработка

#### 4.1.1 Выбор основного языка программирования

Прописать сюда питон, сравнение и преимущетсва

#### 4.1.2 Создание базы данных

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

- MySQL
- SQLite
- PostgreSQL

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

## PostgreSQL

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

## Выбор 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.

### 4.1.3 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 со-

всем. 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 три массива данных:

- Lenta.ru
- Россия сегодня
- РИА новости

Было принято решение дальше работать именно с этими массивами данных.

## **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 «По-

казать 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.

#### 4.1.4 Обработка данных

Добавить обрезание часто и редкоиспользуемых слов и посмотреть что там еще есть в списке

Рассказать что такое формат vowpal wabbit

Рассказать что такое батчи

Обработка данных разделена на два этапа: поддокументная 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 поддокументно

Обработка документа содержит следующие этапы:

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

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.

### 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

Перед следующим этапом необходимо выгрузить все необходимые для обучения документы, прошедшие поддокументную 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).

#### 4.1.5 Обучение модели

Добавить выбор реализации - `BigARTM`

Прописать какие именно регуляризаторы и в какой последовательности были добавлены и какие коэффициенты получились

Согласно выбранному алгоритму сначала модель обучается на подготовленных данных без регуляризаторов (как в рассматриваемом вариан-

те алгоритма PLSA) до того момента как сойдется матрица  $\Phi$ . 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, что бы .

#### 4.1.6 Использование модели

Когда модель обучена ее можно сохранить методом , встроенным в реализацию BigARTM для последующей загрузки.

Добавить пример

#### Оценка документа

Новый поступивший документ (например новую написанную новость на сайте) можно тематизировать обученной моделью методом .

Добавить пример

#### Дообучение

Дообучить модель можно методом .

Добавить пример

#### 4.1.7 Оценка модели

Добавить сюда примеры из выбранных оценок выше

### 4.2 Тестирование

методики тестирования созданного программного обеспечения, примеры

### 4.3 Подготовка к запуску

информация, необходимая для сборки и запуска разработанного программного обеспечения

## 5 Экспериментальный раздел

10 - 15 страниц

### 5.1 // эксперименты и их результаты

Можно поиграть с периодом обучения и сравнения данных (месяц/неделя/день) и посмотреть где лучше (?что лучше)

Можно поиграть с размером новости и посмотреть как от этого зависят результаты

#### 5.1.1 // проводим апробацию

#### 5.1.2 // анализируем результаты

### 5.2 // качественное и количественное сравнение с аналогами

оцениваем адекватность и качество

### 5.3 // даём рекомендации о применимости метода/софта

## **6 Заключение**

**6.1 // отчитаться по каждому пункту тз/по каждой задаче и цели**

**6.2 // сказать про перспективы (мы все уже не умрём)**

## 7 Список источников

### 7.1 // Разобрать

Ссылка на записи с datafest

Воронцов - книги и лекции

Ученики Воронцова - доклады и статьи

Анастасия Янина - работала с Воронцовым - посмотреть ее доклады и статьи

Потапенко Анна - работала с Воронцовым - посмотреть ее доклады и статьи

"Диалог NLP Конференция

курсы на курсере

dwl.kiev.ua - Дмитрия Владимировича Ландэ

Обзор

Topic Detection and Tracking Pilot Study. Final Report.

### 7.2 // Датасеты

25 500 новостей (там суммарно 9 000 000 слов - я посчитал) за все время существования media.zone (я сам написал парсер, могу его же натравить на любой другой новостной ресурс) - уже скачены и лежат на моем компьютере

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

Можно сделать сервис на РИА новости

Можно сделать сервис на агрегаторы новостей

## 8 Приложения

добавить схемы, листинги программного кода, наборы тестов и др

8.1 //