Санкт-Петербургский государственный университет

**Кафедра технологии программирования**

## Садреева Юлия Ильдаровна

**Выпускная квалификационная работа бакалавра**

# Автоматическая классификация новостей из коллекции Reuters в таксономию IPTC

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Добрынин В. Ю.

Санкт-Петербург
2016

# Содержание

**Введение** **3**

**Постановка задачи** **5**

**Обзор литературы** **6**

**1 Подготовка данных** **9**

1.1. Извлечение данных о таксономии и построение графа . . . . 9

1.2. Предварительная обработка . . . . . 10

1.3. Векторная модель данных . . . . . 11

1.4. Расширение описаний узлов . . . . . 12

**2 Классификация документов без учителя** **14**

2.1. Метод k-средних . . . . . 16

2.2. Латентное размещение Дирихле . . . . . 17

2.3. Построение отображения документов в таксономию . . . . . 19

2.3.1. Косинусная мера . . . . . 19

2.3.2. Дивергенция Дженсена–Шеннона . . . . . 20

**3 Реализация и эксперимент** **21**

3.1. Реализация автоматического классификатора . . . . . 21

3.1.1. Алгоритм классификации на основе k-means . . . . . 22

3.1.2. Алгоритм классификации на основе LDA . . . . . 24

3.2. Эксперимент . . . . . 26

3.3. Выводы из эксперимента . . . . . 30

3.4. Дальнейшее направление исследования . . . . . 31

**Заключение** **32**

**Список литературы** **33**

# Введение

В настоящее время количество информации в свободном доступе увеличивается колоссальными темпами. Для упрощения навигации среди этих данных используют информационно-поисковые системы, методы ранжирования, рекомендательные системы и многое другое. Одним из подходов для упорядочения данных является построение каталога тем, рубрикатора, предметного указателя — это всё синонимы в рассматриваемом контексте. В каждой книге для удобства поиска нужной информации существует оглавление. Подобная иерархическая структура часто используется и для навигации на сайтах.

В сфере новостей проблема автоматической классификации особенно актуальна. Для повышения удобства читателей новостные сайты встраивают разделение новостей на рубрики и регионы. Самым тривиальным способом присвоения темы документу является ручная разметка тем. При таком подходе появляется ряд проблем. Во-первых, такая разметка будет весьма субъективна даже в пределах одного новостного ресурса. Журналисты могут по-разному воспринимать темы, к тому же могут допустить ошибку. Во-вторых, у каждого новостного агентства могут быть свои собственные наборы рубрик, что приведет к сложностям при создании рубрикатора новостным агрегатором.

Для решения данной проблемы может применяться автоматическая классификация в единую систему классов. Одной из таких систем является таксономия IPTC — таксономия медиа-тематик, предназначенная для упрощения обмена новостными данными. Международный совет по прессе и телекоммуникациям (англ. International Press Telecommunications Council, IPTC) — консорциум крупнейших мировых новостных агентств и дру-

гих поставщиков новостей. IPTC выступает в качестве глобального органа стандартизации СМИ. Структура таксономии новостных тем является иерархической. Более детальное описание структуры можно увидеть в разделе 1.1. Существует два основных подхода для категоризации текстовых документов. Первый основан на изучении связей слов в предложении, использовании тезариуса языка и онтологии предметных областей. Второй подход рассматривает каждое слово или N-грамму как независимую единицу текста. В данной работе используется второй подход.

Новостные статьи и структуру таксономии сначала необходимо представить в пригодном для анализа виде. В работе используется модель векторного представления данных (англ. Vector Space Model, VSM). Об этом и о другой предварительной обработке данных речь пойдет в Главе 1.

Особенностью данной работы является отсутствие обучающей выборки. По этой причине становится невозможным использование классических классификаторов, таких как, наивный классификатор Байеса или метод k ближайших соседей. Для решения задачи используется кластеризация новостной коллекции с дальнейшим сопоставлением каждого кластера ближайшему в семантическом смысле узлу иерархической структуры. Перечисленные задачи рассматриваются в Главе 2.

В Главе 3 формально описываются разработанные алгоритмы автоматической классификации. Эксперимент и проверка качества проведенной работы также описываются в третьей главе. В качестве тестовой коллекции была выбрана широко известная коллекция Reuters-21578. Коллекция состоит из новостей, опубликованных агентством новостей Reuters в 1987 г. Reuters — британская организация, основанная в 1851 г., является одним из ведущих поставщиков финансовой информации, а также новостей на общественно-политические темы.

# Постановка задачи

Целью данной работы является построение автоматического классификатора текстовых документов в заданную иерархическую структуру классов.

Поставленную задачу можно разбить на следующие подзадачи:

1. сделать предварительную обработку текстов коллекции и описаний классов;
2. разбить данные на кластеры известными методами кластеризации;
3. построить отображение кластеров на заданные классы;
4. оценить качество классификации;
5. выявить достоинства и недостатки построенного классификатора.

## Формализация задачи

Имеется коллекция новостных статей $D = (d_1, \ldots, d_n)$. Также имеется таксономия медиа-тематик IPTC, представляющая собой ациклический направленный граф $G = (V, E)$, где $V$ — множество всех тем, $E$ — связи между ними. Данный граф состоит из 1154 узлов; на верхнем самом общем уровне содержит 17 узлов. Пусть множество $L \in V$ — множество концевых узлов (листов). Необходимо каждому документу $d \in D$ поставить в соответствие одну или несколько тем из множества $L$.

# Обзор литературы

Хороший обзор существующих методов рубрикации документов предметной области можно увидеть в работе [1]. Также автор рассматривает различные алгоритмы отбора признаков для векторного представления документа. Для проведения эксперимента по рубрикации текстовых документов автор использует 4 коллекции документов: Ohsumed, Krapivin, Reuters-21758 и 20NewsGroups. Первые две коллекции – статьи по медицине и информатике соответственно, последние две – документы из новостных лент. Для классификации используются методы машинного обучения с учителем, такие как метод опорных векторов (англ. Support Vector Machine, SVM), метод наименьших квадратов (англ. Linear Least Squares, LLSF), метод k ближайших соседей (англ. k-Nearest Neighbors, kNN).

В работе [2] авторы представили автоматический метод категоризации новостей в таксономию IPTC. В распоряжении авторов было 2700 вручную размеченных новостей на хорватском языке, предоставленных Хорватским Агентством Новостей (HINA). Для классификации авторы использовали метод k ближайших соседей. Алгоритм тестировался на случайных новостях, не входящих в обучающую выборку. Точность классификации определялась как вероятность того, что категория, присвоенная случайному документу, верна. Наилучший результат показал классификатор с 10-ю ближайшими соседями – 84,6% микроусредненной F-меры. Также результат классификации был оценен профессионалами из HINA. Специалистам предоставили 150 автоматически классифицированных новостей, из них 137 были оценены, как удачно классифицированные, т.о. точность классификации составила 91,4%.

Одной из работ по автоматической категоризации текста на основе

онтологий является [3]. Авторы предлагают подход, не требующий обучающей выборки. Категоризация ведется с использованием онтологии, основанной на англоязычной Википедии. Анализируемый документ преобразуется в структуру, согласующуюся с онтологией. Затем определяется семантическая близость между документом и фрагментом описания объекта онтологии и формируется семантический граф. На следующем этапе выделяются компоненты связности полученного графа и формируется доминирующий (тематический) подграф. На основе предопределенного набора классов строится отображение на классы понятий онтологии. И затем каждому понятию доминирующего подграфа ставятся в соответствие подходящие рубрики, после чего определенным образом выбираются рубрики для целого подграфа. Данный метод тестировался на статьях Википедии и на корпусе новостей CNN, а затем сравнивался с результатами наивного байесовского классификатора. Последний показал большую эффективность: 94,21% точности, тогда как рассмотренный метод – 80,77% для новостей CNN, и 67,28% против 83,29% для статей Википедии.

В работе [4] авторы поставили задачу классификации новой коллекции Reuters, состоящей из 806.791 новости. Метод основывается на комбинации самоорганизующейся карты Кохонена (Selforganizing map, SOM) и семантической сети WordNet. При помощи отношений между словами, взятых из WordNet, одним словом можно представить множество синонимов, однокоренных слов или других релевантных слов. Эксперименты проводились на 100000 новостных статей, столько же использовалось в качестве тестовой выборки. Авторы сначала тестировали метод, основываясь только на заголовках статей, затем и на полном тексте. Выяснилось, что несмотря на то, что в тексте статьи содержится гораздо больше информации, чем в заголовке, нет большой разницы для точности классификации.

Определенная тема считалась корректной, если совпадала с одной из предопределенных тем Reuters. В результате исследований авторы выяснили, что комбинация нейронных сетей и семантических отношений дает возможность правильно классифицировать более 98% статей.

Примером русскоязычной работы на тему кластеризации текстовых документов может послужить статья [5]. Авторы предлагают использование FRiS-алгоритма (Function of Rival Similarity) для решения задачи кластеризации текстов из электронной базы публикаций. Указанный алгоритм сначала разбивает множество документов на линейно разделимые кластеры, а затем объединяет их в класс с более сложными формами. Мерой схожести объектов между собой является функция конкурентного сходства с центральным «профильным» объектом кластера. Также авторы представили распараллеленную версию алгоритма, которая показала большую эффективность.

# 1 Подготовка данных

## 1.1. Извлечение данных о таксономии и построение графа

Таксономия медиа-тем IPTC находится в открытом доступе на официальном сайте организации [6]. Существует сервер, хранящий все метаданные о каждом из разделов таксономии, также имеется наглядное представление древовидной структуры. Дополнительно есть возможность скачать структуру целиком в разных форматах: NewsML-G2 Knowledge Item, RDF/XML или RDF/Turtle. Для удобства извлечения данных было решено использовать формат XML.

Рассмотренной таксономии соответствует связный ориентированный ациклический граф $G = (V, E)$, где $V$ — множество всех тем, $E$ — связи между ними. Данный граф состоит из 1154 узлов, на верхнем уровне содержит 17 узлов, максимальная глубина вложенности — 5, количество концевых узлов — 919. Для удобной работы с графом добавляется еще одна фантомная вершина, являющаяся предком тематик верхнего уровня.

О каждом узле в XML-файле хранится следующая информация:

| | |
|---|---|
| conceptId qcode | уникальный код темы |
| created | дата и время создания темы |
| modified | дата и время последнего изменения темы |
| name | заголовок темы |
| definition | описание темы |
| related qcode | код темы-потомка (может быть несколько) |
| related uri | универсальный идентификатор ресурса |

## 1.2. Предварительная обработка

Для дальнейшей работы с данными необходимо провести предобработку описания узлов графа и текста новостей. Она состоит из нескольких этапов.

### Токенизация

Токенизация — разбиение текста на отдельные слова. Полученные слова переводятся в нижний регистр, из слов удаляются цифры, специальные символы и знаки препинания. Например, из предложения

*"Showers continued in the Bahia cocoa zone."*

получаем следующий набор слов:

*['showers', 'continued', 'in', 'the', 'bahia', 'cocoa', 'zone']*

### Удаление стоп-слов

Стоп-слова — это слова, относящиеся к «связующим» частям речи, таким как союз, предлог, местоимение и т.д. Они не несут смысловой нагрузки и могут быть удалены с целью уменьшения размерности пространства признаков. Из предыдущего примера получим:

*['showers', 'continued', 'bahia', 'cocoa', 'zone']*

### Стемминг

Стемминг (англ. stemming) — это процесс нахождения основы слова, учитывая морфологию исходного слова. Основа слова необязательно совпадает с морфологическим корнем. Алгоритм, реализующий стемминг называется стеммер. Существует несколько разновидностей стеммеров, но

большинство из них работают на усечение окончаний, суффиксов и префиксов, основываясь на особенностях языка. Классической реализацией данного процесса является стеммер Портера. Он находится в свободном доступе на сайте автора [7]. В рассматриваемом примере получаем такой результат:

*['shower', 'continu', 'bahia', 'cocoa', 'zone']*

**Создание словаря коллекции**

Все термы, содержащиеся в обрабатываемых текстах (в том числе из таксономии), после обработки добавляются в единое множество $T$, которое не содержит повторяющихся слов. Указанное множество принято называть «мешком слов» (англ. bag of words). Пусть $|T| = n$.

## 1.3. Векторная модель данных

Для кластеризации данных необходимо, чтобы документы были выражены в числовом формате. В данной работе используется векторная модель данных (англ. Vector Space Model, VSM) — математическая модель представления текстовых данных в едином векторном пространстве $\mathbb{R}^n$. Каждый документ представляется в виде вектора $(a_1, \ldots, a_n)$, где $a_i$ — вес $i$-го терма, отражающий «важность» терма для документа. Вес $a_i$ можно находить различными способами. Самый простой из них – определение вхождения терма в документ. Если $i$-ый терм встречается в документе, то $a_i = 0$, иначе $a_i = 1$. Такой способ нахождения весов не имеет широкого распространения из-за недостаточной информативности. Наиболее популярной функцией взвешивания является tf-idf (англ. term frequency, частота терма; англ. inverse document frequency, обратная частота документа).

Рассмотрим документ $d_i \in D$ и представим его в виде вектора из пространства $\mathbb{R}^n$. Этот вектор будет иметь вид: $d_i = (tf_{i1}, \ldots, tf_{in})$, где $tf_{ij}$ — число, с которым терм $t_j$ встречается в документе $d_i$. Получаем, что если терм встречается в документе в $n$ раз больше другого терма, то он в $n$ раз важнее. Смягчим этот показатель следующим образом:

$$wtf_{ij} = \begin{cases} 1 + \ln tf_{ij}, & \text{если } tf_{ij} > 0; \\ 0, & \text{если } tf_{ij} = 0. \end{cases}$$

Слишком часто встречающиеся в текстах слова, как правило, не являются важными. Чтобы учитывать этот нюанс, находят обратную частоту документа:

$$idf_j = \ln \frac{|D|}{|\{d : t_j \in d \wedge d \in D\}|}.$$

В числителе имеем количество документов в коллекции, а в знаменателе — число документов множества $D$, в которых встречается терм $t_j$. Одной из модификаций формулы является «смягченный» $idf$:

$$smooth\_idf_j = \ln \frac{|D| + 1}{|\{d : t_j \in d \wedge d \in D\}| + 1}.$$

Благодаря добавленной единичке можно избежать деления на ноль.

Итоговая формула tf-idf обычно выглядит как $tf\text{-}idf_{ij} = tf_{ij} \cdot idf_j$. В данной работе мы будем использовать модифицированный вид данной формулы:

$$smooth\_tf\text{-}idf_{ij} = wtf_{ij} \cdot (1 + smooth\_idf_j).$$

## 1.4. Расширение описаний узлов

При классификации новостей мы будем ориентироваться на название узла и его описание. Оказалась, что во многих случаях описание является весьма скудным.

Рассмотрим пример темы IPTC:

| Название | jewellery (украшения) |
|---|---|
| Описание | accessories to clothing (аксессуары к одежде) |

Видно, что в названии и описании суммарно содержится всего 4 слова, одно из которых, слово «to», является стоп-словом. Получается, что при составлении вектора значений tf-idf получим всего 3 элемента, отличных от нуля.

Для более качественной классификации было принято решение расширить описание узлов при помощи сервиса Яндекс.XML. Данный сервис принимает на вход запрос к поисковой базе Яндекс и возвращает ответ в формате XML. Запрос формируется из названия узла с добавлением слова «definition». В результате получаем список релевантных страниц, из которых выбираем 5, отбрасывая pdf-файлы, презентации и страницы сайта «www.iptc.org», являющимся оригинальным сайтом таксономии. Далее из найденных страниц извлекаем видимый текст. Страницы могут содержать очень много текста, поэтому для нормирования результатов вводим ограничение на количество символов, считываемых с каждой страницы.

Таким образом, для приведенного выше примера набор наиболее релевантных ключевых слов будет иметь вид:

*['jewelleri', 'jewelri', 'bracelet', 'necklac', 'adorn', 'ring', 'gem', 'precious', 'metal', 'ornament', 'brooch', 'made', 'person', 'jewel', 'wear', 'gold', 'word', 'diamond', 'imit', 'set', 'stone']*

# 2 Классификация документов без учителя

Распределение документов по заранее заданным классам — это задача классификации. Методы классификации требуют участия эксперта (учителя), который присваивает документам классы. Но так как у нас отсутствует обучающая выборка (т.е. нет учителя), нельзя воспользоваться стандартными классификаторами. Для решения задачи разобьем документы на кластеры. Кластерами называются подмножества множества $D$, которые однородны внутри, но максимально отличаются друг от друга [8]. После разбиения документов на кластеры будем либо находить ближайший к кластеру узел графа $G$, либо на полученных в результате разбиения данных присваивать каждому документу соответствующий класс (здесь и далее множество узлов $V$ графа $G$ будем называть классами).

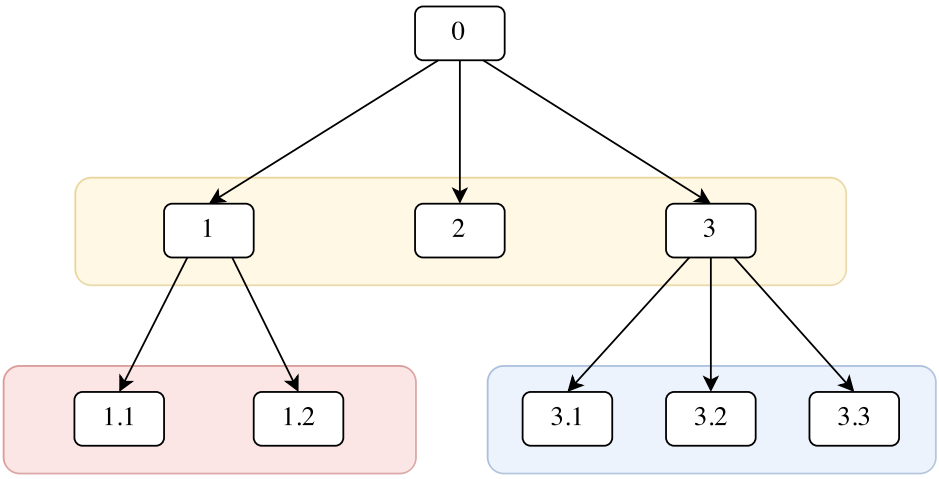



Рис. 1: Разбиение в соответствии с уровнями таксономии.

Сначала будет кластеризована вся коллекция $D$, и каждому документу коллекции будет определен один или несколько соответствующих классов верхнего уровня (на Рис. 1 помечен бледно-оранжевым цветом). Пусть подмножество $D_1 \in D$ — документы, попавшие в узел 1. Далее на основе этого подмножества будет вестись кластеризация на следующий дочерний уровень (помечен розовым цветом), и так далее. Таким образом,

все документы будут распределены по листам графа $G$.

Задачей кластеризации документов является разбиение множества документов на плотные, удаленные друг от друга подмножества. Алгоритмом кластеризации называют функцию, которая каждому объекту ставит в соответствие метку кластера. Кластеризация может преследовать различные цели:

1. Выделение одного наиболее типичного представителя из каждого кластера и, как следствие, уменьшение объема данных.

2. Изучение структуры множества объектов.

3. Обнаружение нетипичных или «шумовых» объектов.

В нашем случае кластеризация ведется с целью группировки документов схожей тематики.

Алгоритмы кластеризации разделяют на два вида:

- плоская (англ. flat clustering)
- иерархическая (англ. hierarchical clustering)

При плоской кластеризации предполагается разбиение всего множества на кластеры, не имеющие взаимосвязей. Примеры плоской кластеризации: метод k-средних (англ. k-means), EM-алгоритм (англ. Expectation-maximization algorithm), алгоритмы теории графов. Различают четкую и нечеткую кластеризацию. При четкой кластеризации каждому объекту присваивается одна и только одна метка кластера. Примером такого алгоритма является k-means. При нечеткой кластеризации, напротив, документ может относиться к нескольким кластерам. Например, c-means.

Иерархическая кластеризация, работая в несколько приемов, создает иерархию кластеров. Под несколькими приемами следует понимать последовательное разбиение или объединение кластеров, тем самым образуя

древовидную структуру. Алгоритмы, которые занимаются объединением, называют агломеративными, разделением — дивизионными.

В данной работе будут опробованы два метода: четкая плоская кластеризация методом k-средних и тематическая вероятностная модель LDA.

## 2.1. Метод k-средних

Самым известным и популярным методом кластеризации является метод k-средних, где k — число кластеров, которое задается заранее.

Алгоритм начинается со случайного выбора начальных центров кластеров $\mu_1, \ldots, \mu_k$, $\mu_i \in \mathbb{R}^n$. Искомые кластеры обозначим $C_1, \ldots, C_k$. Следующие шаги повторяются до тех пор, пока $\mu_i$ и $C_i$ не перестанут меняться:

1. Каждый документ $d \in D$ относится к ближайшему центру, в результате получаем кластеры:

$$C_i = \left\{ d \in D \;\middle|\; \operatorname*{argmin}_{j \in \{1,\ldots,k\}} \rho(d, \mu_j) = i \right\}.$$

2. Вычисляется новое положение центров кластеров:

$$\mu_i = \operatorname*{argmin}_{z \in \mathbb{R}^n} \sum_{d \in C_i} \|d - z\|^2.$$

Метод k-means имеет значительный недостаток. Начальные центры кластеров инициализируются случайным образом, что сильно влияет на результат кластеризации. Для оптимального выбора начальных центров существует модификация данного метода, которая называется k-means++.

### k-means++

Пусть $M$ — множество найденных центроидов. Тогда алгоритм выбора начальных центров кластеров будет выглядеть так:

1. Случайным образом выбрать центроид $\mu_1$ из множества $D$. $M = \{\mu_1\}$.

2. Выбрать новый центроид $\mu_i$ из множества $D$ с вероятностью

$$\frac{F(d)^2}{\sum_{d \in D} F(d)^2},$$

где F(d) — расстояние от документа $d$ до ближайшего центроида из множества $M = \{\mu_1, \ldots, \mu_{i-1}\}$

3. Повторять шаг 2, пока $|M| < k$

Остальные шаги совпадают с основным алгоритмом k-means.

## 2.2. Латентное размещение Дирихле

Латентное размещение Дирихле (latent Dirichlet allocation, LDA) — математический метод определения тем, основанный на предположении, что слова, составляющие текст документа, берутся из некоторого множества слов, т.н. темы. Сама же тема представляет собой вероятности, с которыми каждое слово из словаря принадлежит этой теме. Обозначим множество тем буквой $Z$. Количество тем является задаваемой величиной. Поставим задачу определить $k$ тем ($|Z| = k$) коллекции $D$. Каждый документ рассматривается как смесь найденных тем. Далее, чтобы не отклоняться от обозначений, принятых в литературе, термы будем называть «словами» и обозначать $w$, и, соответственно, множество всех термов в коллекции $W$.

В основе LDA лежит вероятностная модель существования пары документ-слово. Эту вероятность можно расписать по формуле умножений вероятностей:

$$p(d, w) = p(w|d) \cdot p(d).$$

Рассмотрим матрицу

$$F = [p(w|d)]_{\substack{w \in W \\ d \in D}}.$$

Необходимо представить эту матрицу в виде произведения матрицы вероятностей тем в документах $\Theta_{|D| \times k}$ с элементами $\theta_{dz} = p(z|d)$ и матрицы вероятностей слов в темах $\Phi_{k \times |W|}$ с элементами $\phi_{zw} = p(w|z)$. Тогда получим вероятностную модель [9]:

$$p(d, w) = \sum_{z \in Z} \underbrace{p(w|z)}_{\theta} \cdot \underbrace{p(z|d)}_{\phi} \cdot p(d),$$

Модель LDA имеет два настраиваемых гиперпараметра: $\alpha \in \mathbb{R}^k$ и $\beta \in \mathbb{R}^{|W|}$. Гиперпараметр $\alpha$ отвечает за выраженность тем в документах. Чем меньше будет $\alpha$, тем более разреженным окажется вектор распределения. Обычно берут $\alpha_z = \frac{50}{k}$, но по более тонким исследованиям рекомендуют оптимизировать вектор $\alpha$ [10]. Параметр $\beta$ определяет разреженность вектора, описывающего распределение слов в теме. Рекомендуемое значение $\beta_z = 0,01$.

Предполагается, что процесс порождения документов в модели LDA проходит следующим образом [11]:

1. Для каждого документа $d \in D$ выбирается случайный вектор $\theta_d$, подчиняющийся закону распределения Дирихле с коэффициентом $\alpha$.

2. Выбирается тема $z_{di}$ из полиномиального распределения с параметром $\theta_d$.

3. Выбирается слово $w_{di}$ из распределения $\phi_{z_{di}}$, являющимся распределением Дирихле с коэффициентом $\beta$.

Функция плотности вероятности для распределения Дирихле случайной величины $x = (x_1, \ldots, x_k)$ с параметром $\alpha = (\alpha_1, \ldots, \alpha_k)$ имеет вид [12]:

$$f(x|\alpha) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_k)} x_1^{\alpha_1 - 1} \ldots x_k^{\alpha_k - 1}.$$

Для идентификации параметров модели обычно применяется сэмплирование по Гиббсу (англ. Gibbs sampling) — алгоритм для получения выборки из совместного распределения нескольких случайных величин [13].

## 2.3. Построение отображения документов в таксономию

### 2.3.1. Косинусная мера

В случае k-means, представителем каждого класса можно считать центроид $\mu \in \mathbb{R}^n$. Векторное представление описания класса представляет из себя также вектор из пространства $\mathbb{R}^n$. Обозначим этот вектор буквой $a$. Множество узлов на уровне классификации обозначим $\tilde{V}$, такое, что $\tilde{V} \in V$. Для того, чтобы найти ближайший к кластеру узел из $\tilde{V}$, можно воспользоваться косинусной мерой сходства двух векторов, которая задается как косинус угла между ними.

$$\text{cosine\_similarity}(\mu, a) = \cos(\mu, a) = \frac{\mu \cdot a}{\|\mu\| \|a\|}$$

Чем больше значение косинусной меры, тем ближе вектора. Значит, для нахождения ближайшего класса к центроиду необходимо найти косинусную меру для всех векторов-описаний из множества $\tilde{V}$ и выбрать класс, соответствующий максимальному значению меры.

Использование косинусной меры весьма популярно при оценке близости документов из-за разреженности векторов, которыми эти документы представляются.

### 2.3.2. Дивергенция Дженсена–Шеннона

В результате применения метода LDA документы и описания классов представляются в виде вектора распределения тем в документе. Сравнивая полученные распределения, можно определить принадлежность новости к одному из классов. Для этого будем использовать дивергенцию Дженсена–Шеннона. Идея заключается в том, что расстояние между двумя распределениями не может сильно отличаться от среднеарифметического значения расстояний до их среднего распределения. Так, дивергенция Дженсена–Шеннона между двумя вероятностными распределениями $P$ и $Q$ выглядит следующим образом [14]:

$$JSD(P\|Q) = \frac{D(P\|A) + D(Q\|A)}{2},$$

где $A = \frac{P+Q}{2}$, $D$ — дивергенция Куллбэка-Лейблера, определяемая формулой:

$$D(P\|Q) = \sum_i P(i) \log_2 \frac{P(i)}{Q(i)}.$$

Для каждого документа можно найти дивергенцию Дженсена-Шеннона между распределением, соответствующим документу коллекции, и распределениями для описаний классов $\tilde{V}$. Минимальному значению дивергенции будет соответствовать наиболее подходящий класс для рассматриваемого документа.

# 3 Реализация и эксперимент

В этой главе описывается реализация автоматической классификации на основе метода кластеризации k-means и метода тематического моделирования LDA. Также рассматривается вопрос выбора инструментов разработки. В заключении описывается эксперимент над построенными классификаторами и интерпретируются результаты.

## 3.1. Реализация автоматического классификатора

Входными данными для рассматриваемых алгоритмов являются коллекция новостных документов и иерархическая структура классов, каждый класс которой имеет необходимый атрибут — название или описание. Общая часть для всех алгоритмов выглядит так:

1. Извлечение данных из корпуса новостей (id, заголовок, текст новости, тема новости), а также данных об элементах таксономии (код темы, имя, описание, ссылки на предков и потомков)

2. Построение графа таксономии на основе извлеченных данных с добавлением «фиктивного» узла.

3. Расширение описания классов при помощи поисковой машины, используя в качестве запроса название класса.

4. Предварительная обработка текстов: токенизация, удаление стоп-слов, стемминг.

5. Векторизация текстов новостей и описаний классов в едином векторном пространстве с присвоением весов tf-idf каждому терму каждого документа.

Следующий большой этап — отображение документов в построенный граф — ведется на основе двух различных методов. Ниже рассмотрим каждый из них по отдельности.

Алгоритмы рекурсивно обходят граф $G$, начиная с «фиктивного» узла $V_0$ и спускаясь вниз по потомкам. Входными данными также является число $k$ — количество кластеров (для k-means) или количество тем (для LDA). Множество $D$ по-прежнему определяет коллекцию документов, а параметр $tolerance$ задает порог присвоения документа классу. Обозначим $def(v)$ — вектор-описание узла $v$.

### 3.1.1. Алгоритм классификации на основе k-means

Для каждой вершины графа $G$, имеющего потомков, необходимо разбить на кластеры документы, присвоенные соответствующему вершине классу. Затем проводится классификация k-means, разбивающая множество документов на $k$ кластеров. Каждому кластеру ставится в соответствие один или несколько классов из множества потомков класса $V_0$. Подходящий класс выбирается путем нахождения косинусного сходства для центроида кластера и для каждого из потомков класса $V_0$. Если величина сходства больше некоторого задаваемого порога, то присваиваем всем документам рассматриваемого кластера метку соответствующего класса. Если же не нашлось классов достаточно близких к кластеру, то кластеру ставится в соответствие класс, имеющий наибольшее косинусное сходство. После того, как всем документам расставлены классы, алгоритм повторяется, углубляясь по графу. При этом по мере повышения глубины вложенности, документов для классификации становится в несколько раз меньше, поэтому вслед за углублением понижаем значение k. В качестве вектора, представляющего класс, берется векторное представление описания класса с tf-idf

весами.

Величиной порога можно регулировать к скольким классам будет отнесен кластер. Так, если задать слишком высокое значение порога, например, равным единице, каждый кластер будет относиться только к одному классу. Напротив, если установить малое значение порога, например, нуль, каждый кластер будет отнесен ко всем классам на своем уровне.

**Алгоритм 1** Псевдо-код алгоритма на основе k-means

**Вход:** $G, V_0, k, D, tolerance$; // *tolerance — порог присвоения документам кластера класса*

**Выход:** $G$;

**если** $|D| > 0 | (V_0)| > 0$ **то**
  **если** $|D| < k$ или $k == 0$ **то**
    $k = |D|$;
  $KMeans(D, k)$; // *метод k-means на D документах*
  **для всех** $c \in C$
    **для всех** $v \in \{child(V_0)\}$
      $a :=$ cosine_similarity$(centroid(c), def(v))$;
        // $def(v)$ — *вектор-описание класса* $v$
      **если** $a \geqslant tolerance$ **то**
        присвоить всем $d \in documents(c)$ метку класса $v$;
          // $documents(c)$ — *документы, определенные в кластер* $c$
    **если** все $a < tolerance$ **то**
      присвоить всем $d \in documents(c)$ метку класса $v$, соответствующему argmax(cosine_similarity$(\mu, \{def(child_i(V_0))\})$;
  **для всех** $v \in \overset{i}{\{child(V_0)\}}$
    $V_0 := v$;
    $k := k/2$;
    $D := documents(v)$; // $documents(v)$ — *документы, присвоенные классу* $v$
    повторить алгоритм для обновленных $G, V_0, k, D, tolerance$
      // *повторяем сначала весь Алгоритм 1, пока не пройдем по всем ветвям графа* $G$

### 3.1.2. Алгоритм классификации на основе LDA

Алгоритм во многом схож с предыдущим. Входные данные остаются такими же. Класс представляется в виде вектора, $i$-ый элемент которого — число, с которым $i$-ый терм встретился в описании класса. Будем называть этот вектор также вектором-описанием класса. Для всех неконечных вершин графа выполняются следующие действия: коллекция документов и вектора-описания классов объединяются в единое множество. Методом LDA формируются $k$ тем, и каждый документ коллекции представляется в виде вектора вероятностей $\theta$ порождения документа $i$-ой темой, $i \in \{1, \ldots, k\}$. Далее множество векторов $\theta$ разделяется на два подмножества: $\Theta_1$ — распределения документов из коллекции $D$, и $\Theta_2$ — распределения для векторов-описаний дочерних классов класса $V_0$. Чтобы определить релевантные документу $d_i \in D$ классы, находится дивергенция Дженсена–Шеннона для вектора $\theta_i \in \Theta_1$ и всех векторов из множества $\Theta_2$. Если значение дивергенции меньше или равно некоторого порога $tolerance$, то документу присваивается соответствующий класс. Таким образом, документ может быть определен в несколько классов. Если же не нашлось значений дивергенции, меньших порога, то документу присваивается класс, чей вектор задал минимальное значение дивергенции.

В алгоритме на основе k-means приходилось по мере повышения вложенности уменьшать значение $k$, т.к. количество кластеров не может превышать количества документов, а их становится с каждым шагом меньше. В этом алгоритме нет необходимости уменьшать $k$, потому что здесь $k$ обозначает число тем и задает длину вектора $\theta$. А значит, можно, к примеру, 10 документов представить в виде смеси 20 тем. Поэтому в этом алгоритме $k$ является фиксированной величиной.

Как и в предыдущем методе, здесь можно регулировать распределе-

ние документов по нескольким классам при помощи значения порога, но строго наоборот: при пороге равном нулю документ будет отнесен ко всем классам уровня, а при пороге равном единице документу будет присвоен всего один класс.

**Алгоритм 2** Псевдо-код алгоритма на основе LDA

**Вход:** $G, V_0, k, D, tolerance$; // *tolerance — порог присвоения документа классу*

**Выход:** $G$;

**если** $|D| > 0 | (V_0)| > 0$ **то**
  **если** $|D| < k$ или $k == 0$ **то**
    $k = |D|$;
  $LDA(D \bigcup A, freq((V_0))$; // *метод LDA на D документах*
  **для всех** $x \in \Theta_1$
    **для всех** $y \in \Theta_2$
      $a := JSD(x, y)$; // *пусть документ d соответствует своему распределению x, класс v — распределению y*
      **если** $a \geqslant tolerance$ **то**
        присвоить $d$ метку класса $v$;
    **если** все $a > tolerance$ **то**
      присвоить $d$ метку класса $v$, соответствующему $\underset{i}{\operatorname{argmin}}(JSD(x, \theta_i \in \Theta_2)$;
  **для всех** $v \in \{child(V_0)\}$
    $V_0 := v$;
    $D := documents(v)$; // *documents(v) — документы, присвоенные классу v*
    повторить алгоритм для обновленных $G, V_0, k, D, tolerance$
    // *повторяем сначала весь Алгоритм 2, пока не пройдем по всем ветвям графа G*

## 3.2. Эксперимент

Для проведения эксперимента использовалась коллекция новостей Reuters-21578 [15]. Она находится в открытом доступе в формате sgm. Количество новостей в коллекции – 21578. Каждой новости соответствуют следующие атрибуты: дата создания, темы, места, личности, организаторы, компании, название и текст новости. Не все атрибуты являются обязательными, поэтому было решено не включать в выборку новости, не содержащие в себе основной текст. Таких новостей, состоящих только лишь из заголовка, оказалось 2535. Также в коллекции оказалось 550 очень коротких новостей, состоящих преимущественно из сокращений. Подобные новости не были включены в итоговую коллекцию, потому что для них сложно определить категорию даже эксперту (Рис. 2).

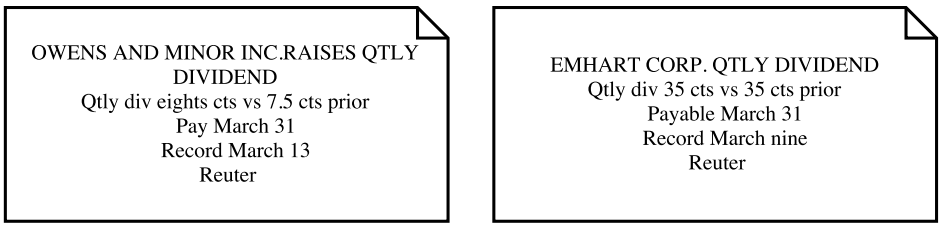



Рис. 2: Пример исключенных из коллекции документов.

Полученная мощность коллекции – 18493. Среднее количество слов в новости до обработки – 137. Размер словаря, составленного из текстов новостей и расширенных описаний классов, после предварительной обработки составил 46.593 терма. В словарь не включались термы короче трех символов.

Для проведения эксперимента использовался метод экспертной оценки, описанный в [15]. В поставленной задаче необходимо оценить однородность документов внутри класса, а также соответствие этих документов метке присвоенного класса (т.е. как хорошо согласуются описание класса

с документами, присвоенными классу). Для этих целей были разработаны тесты двух типов.

**Тест первого типа.** Направлен на определение однородности документов внутри класса. Тест составляется следующим образом: случайным образом выбирается класс, соответствующий концевому узлу графа $G$. Выбираются первые четыре документа, принадлежащие этому классу. Далее случайным образом выбирается другой класс, тоже не имеющий потомков и лежащий в другой крупной ветке дерева классов. Из этого класса берется первый документ. Все пять документов сортируются в случайном порядке.

**Тест второго типа.** Предназначен для определения соответствия описания класса содержанию документов. Аналогично тесту первого типа выбираются три документа из произвольного концевого класса. К ним добавляется один документ из другого произвольного класса с теми же характеристиками, что и в тесте первого типа. Далее к этим документам добавляется пятый — описание класса.

Затем из полученных документов извлекаются 10 слов с наибольшим весом tf-idf. Слова формируются без стемминга, чтобы экспертам было удобнее их читать. В итоге получается группа из пяти наборов ключевых слов. Экспертам ставится задача найти лишний набор слов. Если большинство экспертов правильно угадывает лишний набор, то класс считается удачно укомплектованным. Если в тесте второго типа эксперты считают лишним описание класса, то это сигнализирует о некачественном присвоении документам класса.

Так как коллекция Reuters-21578 является англоязычной, то для удобства экспертов наборы были продублированы на русском языке. Перевод осуществлялся при помощи онлайн-переводчика Google Translator. Было предложено пять тестов первого типа, и пять — второго. Сами тесты бы-

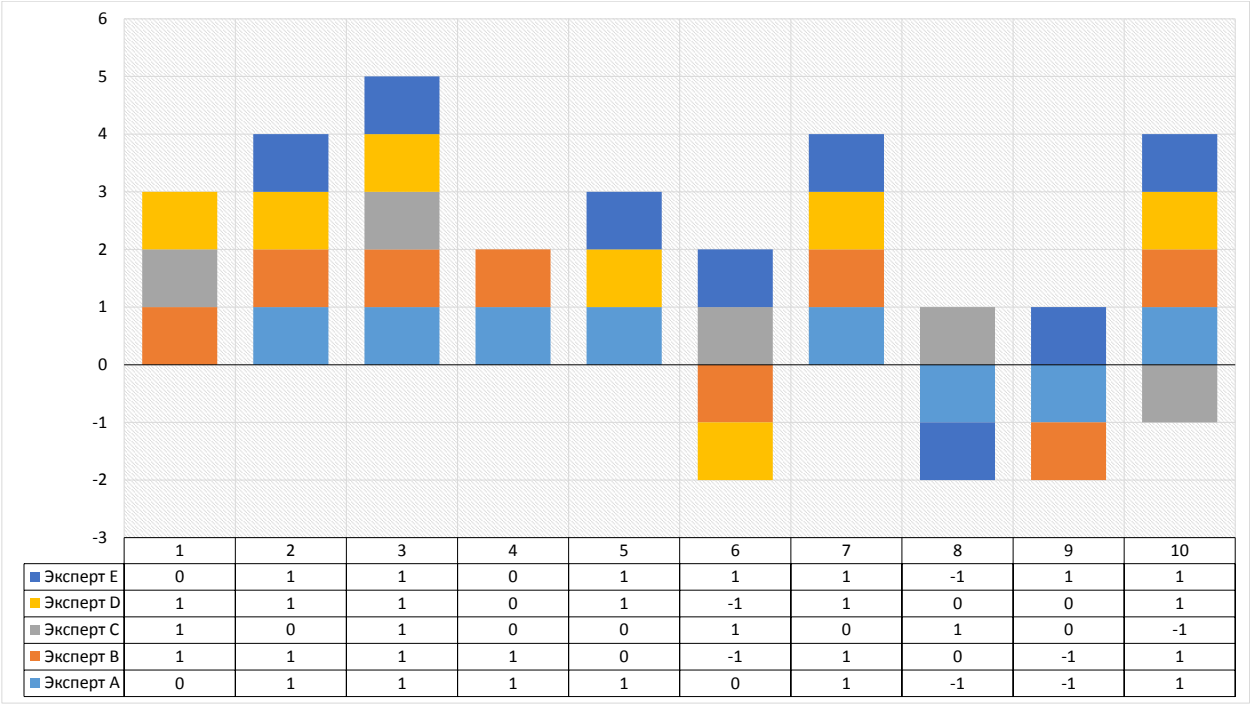
ли оформлены при помощи Google Forms и были размещены в открытом доступе, чтобы любой желающий мог их пройти. Некоторые эксперты пожелали сохранить анонимность.

### Эксперимент для алгоритма, основанного на k-means

Для начального значения количества кластеров экспериментальным путем было выбрано $k = 60$. Порог включения документов в класс был выбран немного выше, чем средняя величина максимального косинусного сходства на каждом шаге рекурсивного процесса. Таким образом, $tolerance = 0,2$. В результате эксперимента из 919 концевых узлов рассматриваемой в работе структуры классов 93 оказались покрыты новостями, остальные 826 — пустые. Получается, 10% покрытия. Всего коллекции документов было присвоено 30401, следовательно, в среднем пришлось по 1,64 класса на документ.

Тесты были пройдены пятью экспертами: 4 студентами старшего курса и 1 анонимным пользователем сети Интернет.

В таблице на Рис. 3 единица ставится, если ответ эксперта совпадает с правильным, 0 — в противном случае и -1, когда в качестве лишнего набора было выбрано описание класса.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Эксперт E | 0 | 1 | 1 | 0 | 1 | 1 | 1 | -1 | 1 | 1 |
| Эксперт D | 1 | 1 | 1 | 0 | 1 | -1 | 1 | 0 | 0 | 1 |
| Эксперт C | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | -1 |
| Эксперт B | 1 | 1 | 1 | 1 | 0 | -1 | 1 | 0 | -1 | 1 |
| Эксперт A | 0 | 1 | 1 | 1 | 1 | 0 | 1 | -1 | -1 | 1 |

Рис. 3: Результаты эксперимента для алгоритма, основанного на k-means

В результате эксперимента 4 из 5 тестов первого типа можно назвать удачными: большинство экспертов правильно указали набор из другого класса. Более того, в 3 тесте все эксперты верно выбрали лишний элемент. 80% успешных тестов говорит об однородности документов внутри класса. Во второй группе тестов (под номерами 6–10) всего 2 теста дали положительный результат. В 6-ом, 8-ом и 9-ом тестах можно склоняться к тому, что классы документам были присвоены неверно.

### Эксперимент для алгоритма, основанного на LDA

Количество тем для построения тематической модели было выбрано равным 50. Порог присвоения класса документу был вычислен как минимум дивергенций Дженсена–Шеннона на каждой итерации алгоритма. Получилось, $tolerance = 0,566$. Документы были распределены между 650 концевыми классами. Процент покрытия — 70%. Документам в среднем было присвоено по 2,5 класса.

В эксперименте принимали участие 5 экспертов: 3 студента старшего курса и 2 анонимных пользователя сети Интернет.


| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Эксперт E | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Эксперт D | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | -1 | 0 |
| Эксперт C | 1 | 0 | 0 | 1 | 0 | 1 | -1 | 1 | 1 | 0 |
| Эксперт B | 0 | 0 | 1 | 0 | 1 | 0 | 0 | -1 | 1 | 0 |
| Эксперт A | 1 | 0 | 0 | 0 | 1 | 1 | 0 | -1 | 1 | 0 |

Рис. 4: Результаты эксперимента для алгоритма, основанного на LDA

Как видно из диаграммы на Рис. 4, из первых пяти групп всего две набрали большинство правильных ответов. Значит разбиение документов на однородные внутри классы нельзя назвать удовлетворительным. В последних пяти группах тоже только два из пяти классов можно назвать соответствующими документам, которые они содержат. В последнем классе нельзя сделать однозначный вывод, в чем причина неправильных ответов.

## 3.3. Выводы из эксперимента

Результаты эксперимента позволили сделать следующие выводы. По результатам тестов для алгоритма, основанного на LDA, можно сказать, что классы были сформированы весьма удачно. Необходимо напомнить, что непустой класс может включать в себя один или несколько кластеров,

полученных методом k-means. Из этого можно сделать вывод, что кластеры хорошо группируются, при этом группы плохо соответствуют меткам класса. Схожая ситуация наблюдается и во второй группе тестов, при этом классы содержат неоднородные элементы.

Одной из причин не очень удачного соответствия групп документов описаниям класса являются обстоятельства их происхождения. Напомним, что документами являются новости 1987 г., а расширенное описание классов формировалось из современной поисковой выдачи. Также на результат повлиял тот факт, что коллекция является узкоспециализированной в основном на финансовых новостях. При этом таксономия IPTC охватывает все темы, которые могут освещать СМИ.

## 3.4. Дальнейшее направление исследования

Дальнейшие исследования могут быть направлены на модернизацию процесса описания классов. Также есть смысл в проведении более масштабного эксперимента на современной коллекции новостей с большим охватом сфер деятельности человека, при этом могут быть разработаны дополнительный тип теста на определение качества кластеризации. В дальнейших исследованиях могут быть реализованы алгоритмы, основанные на методах кластеризации, которые не рассматривались в этой работе.

# Заключение

В работе была поставлена задача разработать автоматический классификатор для новостных статей. Эта задача актуальна в связи с большим ростом данных и информационной потребностью интернет-пользователей. Для решения задачи были рассмотрены и реализованы метод кластеризации k-средних и латентное размещение Дирихле. Также были разработаны алгоритмы для построения отображения документов в таксономию медиа-тематик IPTC. Алгоритм, основанный на методе k-средних, показал лучшие результаты. Дальнейшие модификации этого метода могут привести к большей точности классификатора, а значит, могут быть включены в основной инструментарий новостных агентств и агрегаторов.

# Список литературы


[1] Малахов Д. П. Методы автоматической рубрикации текстовых документов предметной области // Научный семинар Института системного программирования РАН, 2015

[2] Bacan H., Pandzic I., Gulija D. Automated News Item Categorization // Proceedings of JSAI 2005 Workshop on Conversational Informatics, in conjunction with the 19th Annual Conference of The Japanese Society for Artificial Intelligence JSAI 2005 Kitakyushu, Japan: Kyoto University, 2005. P. 57–62.

[3] Janik M., Kochut K. J. Wikipedia in action: Ontological knowledge in text categorization //Semantic Computing, 2008 IEEE International Conference on. – IEEE, 2008. P. 268-275.

[4] Wermter S., Hung C. Selforganizing classification on the Reuters news corpus // The 19th International Conference on Computational Linguistics (COLING2002), Taipei, Taiwan, 2002. P. 1086-1092.

[5] Загоруйко Н.Г., Барахнин В.Б., Борисова И.А., Ткачев Д.А. Кластеризация текстовых документов из электронной базы публикаций алгоритмом FRiS-Tax // Вычислительные технологии. 2013. Т. 18. № 6. С. 62-74.

[6] Media Topics. https://iptc.org/standards/media-topics/

[7] The Porter Stemming Algorithm. http://tartarus.org/martin/PorterStemmer/

[8] Маннинг К. Д., Рагхаван П., Шютце Х. Введение в информационный поиск // М: Вильямс, 2011. С. 353-359.

[9] Воронцов К. В., Потапенко А. А. Регуляризация вероятностных тематических моделей для повышения интерпретируемости и определения числа тем // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2014 г.). — Вып. 13 (20). М: Изд-во РГГУ, 2014. С. 676–687.

[10] Wallach H., Mimno D., McCallum A. Rethinking LDA: Why priors matter // Proceedings of Advances in Neural Information Processing Systems, 2009.

[11] Daud A., Li J., Zhou L., Muhammad F. Knowledge discovery through directed probabilistic topic models: a survey // Proceedings of Frontiers of Computer Science in China. 2010, P. 280-301.

[12] Де Гроот М. Оптимальные статистические решения. М.: Мир, 1974. С. 56-58.

[13] Вероятностные тематические модели коллекций текстовых документов. http://www.machinelearning.ru/wiki/images/c/c2/Vorontsov-2apr2012.pdf

[14] Louis A., Nenkova A. Automatic Summary Evaluation without Human Models // Notebook Papers and Results, Text Analysis Conference (TAC-2008), Gaithersburg, Maryland (USA), 2008.

[15] Chang J., Boyd-Graber J., Gerrish S., Wang C., Blei D. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors // Advances in Neural Information Processing Systems 22, 2009. P. 288–296.