МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
КАФЕДРА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

Дипломная работа

Определение тематической направленности текстового содержимого микроблогов

Выполнил:

Студент 528 группы

Гомзин Андрей Геннадьевич

Научные руководители:

ак. РАН, проф. Иванников Виктор Петрович

Коршунов Антон Викторович

Москва

2013

Содержание

Введение **3**

1 Постановка задачи **6**

2 Обзор существующих решений **7**

2.1 Вероятностные тематические модели 7

2.2 Методы оценки качества тематических моделей 12

2.3 Интерпретируемость результатов тематического моделирования 13

2.4 Особенности текстов микроблогов и методы их нормализации 15

3 Исследование и построение решения задачи **18**

3.1 Исходные данные 18

3.2 Нормализация твитов 18

3.3 Оценка интерпретируемости 20

3.4 Результаты экспериментов 22

4 Описание практической части **29**

4.1 Обоснование выбранного инструментария 29

4.2 Схема работы 29

Заключение **32**

Аннотация

В данной дипломной работе исследуются методы тематического моделирования текстов, а также оценки качества получаемых результатов. При этом, в качестве исходных данных используются тексты микроблогов, которые существенно отличаются от традиционных текстов книг, статей и пр. В работе предложены методы оценки качества результатов тематического моделирования, основанные на интерпретируемости получаемых тем. Интерпретируемость оценивается с использованием внешних баз знаний Wikipedia и Google, по ключевым словам тем. Представлены результаты экспериментального исследования этих методов – сравнение с ручными оценками. Произведено сравнение моделей Скрытое размещение Дирихле и Иерархический процесс Дирихле с различными значениями параметров на основе разработанного метода оценки интерпретируемости результатов тематического моделирования.

Введение

Часто при анализе текстовых документов, как ручном, так и автоматическом, не нужны полные тексты исследуемых документов. Достаточно лишь небольшого количества информации о них. Примером такой информации могут быть темы, затрагиваемые в каждом документе.

Для того, чтобы выделить из текста основные темы, человеку достаточно его прочитать. В условиях постоянно увеличивающегося количества информации, в частности, текстовой (так называемый, информационный бум), приходится анализировать данные такого объема, которые человек не в силах обработать. Поэтому необходимы методы, позволяющие автоматически извлекать темы из большого набора данных. Теоретически обоснованным и активно развивающимся направлением в анализе текстов на естественном языке, является тематическое моделирование коллекций текстовых документов.

Построение тематической модели может рассматриваться как задача одновременной кластеризации документов и слов по одному и тому же множеству кластеров, называемых темами. В терминах кластерного анализа тема – это результат би-кластеризации, то есть одновременной кластеризации и слов, и документов по их семантической близости. Как правило, выполняется нечёткая кластеризация, то есть документ может принадлежать нескольким темам в различной степени. Таким образом, сжатое семантическое описание слова или документа можно представить в виде распределения на множестве тем. Процесс нахождения этих распределений и называется тематическим моделированием.

В 2003 году Д.Блей предложил модель скрытого размещения Дирихле (Latent Dirichlet Allocation, LDA) [1]. Это одна из первых и широко используемых вероятностных тематических моделей. Основной идеей таких моделей является наличие генеративного процесса – процесса, порождающего документы с использованием предопределенных тем. Задача заключается в том, чтобы подобрать темы таким образом, чтобы вероятность сгенерировать данный набор документов была максимальной.

Особенностями этой и многих последующих моделей [2] являются:

- независимость слов в документе (т.е. последовательность слов не имеет значения)
- представление тем в виде вероятностных распределений над словами

- наличие априорных распределений для некоторых случайных величин модели

Тематические модели имеют большой спектр применения:

- кластеризация, классификация, ранжирование, аннотирование и суммаризация отчётов, научных публикаций, переписки, блогов, студенческих работ и т.д.;
- тематический поиск документов и связанных с ними объектов: рисунков, авторов, организаций, журналов, конференций;
- фильтрация спама;
- рубрикация коллекций изображений, видео, музыки;
- поиск генетических паттернов в различных популяциях и определение пропорции этих паттернов у конкретного индивидуума;
- коллаборативная фильтрация в сервисах рекомендаций;
- построение тематических профилей пользователей форумов, блогов и социальных сетей для поиска тематических сообществ и определения наиболее активных их участников;
- анализ новостных потоков и сообщений из социальных сетей для определения актуальных событий реального мира и реакции пользователей на них.

В вероятностных тематических моделях темы представляются в виде распределений над словами. Оценить качество полученных тем можно вручную: можно выбрать слова с наибольшей вероятностью и понять, что они вместе означают. При большом количестве тем требуется много времени, чтобы оценить, насколько понятными для человека они получились. В данной работе разрабатывается и исследуется алгоритм, позволяющий автоматически оценить интерпретируемость тем, получаемых в процессе тематического моделирования. Под интерпретируемостью здесь понимается мера семантической связности ключевых слов темы. При этом анализируются текстовые данные микроблогов, как наиболее активно развивающегося источника текстовых данных.

Текстовое содержимое микроблогов существенно отличается от содержимого других текстовых источников [3]: коллекций статей, традиционных блогов. Пользователи

микроблогов, как правило, не проверяют свои сообщения на предмет орфографических ошибок перед отправкой. Максимальное количество символов в одном сообщении ограничено. Сообщениям микроблогов также свойственно большое количество сокращений, эмотиконов, гиперссылок. В связи с этим при анализе тематической направленности текстового содержимого этих сообщений возникают трудности.

1 Постановка задачи

Целью данной работы является исследование и разработка методов тематического моделирования текстов микроблогов с высокой интерпретируемостью.

Для достижения поставленной цели необходимо:

1. Исследовать существующие методы тематического моделирования и способы оценки их качества
2. Разработать и реализовать алгоритм автоматической оценки интерпретируемости результатов тематического моделирования по ключевым словам тем
3. Выполнить экспериментальную оценку интерпретируемости методов тематического моделирования текстов микроблогов с использованием разработанного алгоритма

2 Обзор существующих решений

В данном разделе рассматриваются существующие вероятностные тематические модели и методы оценки их качества. Так как тематические модели в данной работе применяются к текстам микроблогов, в разделе также рассмотриваются некоторые особенности таких текстов.

2.1 Вероятностные тематические модели

Вероятностное тематическое моделирование — это набор алгоритмов, позволяющих анализировать слова в больших наборах документов и извлекать из них темы, связи между темами и изменение их во времени [2]. При этом документ рассматривается как набор слов, порядок которых не имеет значения. Для каждого документа определено распределение θ_d его слов по темам, т.е. вероятность θ_d^t для каждой темы встретить ее в данном документе, причём $\sum_t \theta_d^t = 1$.

Тема представляется в виде распределения φ_t слов из фиксированного словаря, т.е. каждое слово входит в тему с некоторой вероятностью φ_t^w, причём $\sum_w \varphi_t^w = 1$.

Вероятностные модели являются генеративными (порождающими), то есть их можно использовать для генерации документов. Описание модели, как правило, начинается со способа генерации документов — генеративного процесса. Однако основной целью тематического моделирования является не генерация, а извлечение тем из имеющегося набора документов. Это задача, обратная генерации: выяснить, с помощью каких скрытых структур вероятнее всего могли бы быть сгенерированы исходные документы.

Алгоритмы поиска наиболее правдоподобных скрытых параметров делятся на две категории: на основе сэмплирования и вариационные методы. Алгоритмы первой группы пытаются собрать конечную выборку переменных, на которой ищется максимум. Как правило, алгоритм принадлежит классу методов Монте-Карло для марковских цепей (Markov Chain Monte Carlo, MCMC). Примером такого алгоритма является сэмплирование по Гиббсу [4], которое состоит в том, чтобы на каждом шаге фиксировать все переменные, кроме одной, и выбирать оставшуюся переменную согласно распределению вероятности этой переменной при условии всех остальных (эта вероятность выводится в [4]).

Методы второй группы — вариационные алгоритмы. В них сначала задается параметризованное семейство распределений над скрытыми переменными, а затем с помощью EM-алгоритма ищется распределение из этого семейства, наиболее близкое к исходному апостериорному распределению.

Скрытое размещение Дирихле

Скрытое размещение Дирихле (Latent Dirichlet Allocation, LDA) — генеративная графическая вероятностная модель, предложенная Дэвидом Блеем и соавторами в 2003 году [1]. Данная модель является классической моделью, на ее основе строятся другие модели.

В модели LDA каждый документ генерируется независимо, по следующей схеме:

1. Случайно выбрать для документа его распределение по темам θ_d
2. Для каждого слова в документе:
   a) Случайно выбрать тему из распределения θ_d, полученного на 1-м шаге
   b) Случайно выбрать слово из распределения слов в выбранной теме ϕ_t

На рисунке 1 показана схема (т. н. графическое представление) модели скрытого размещения Дирихле.

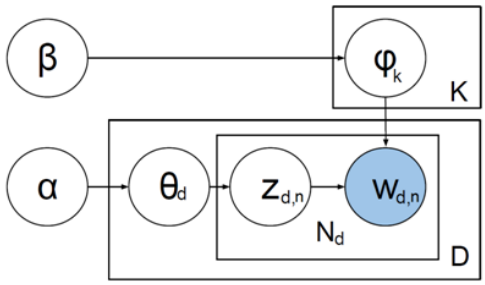

Рисунок 1: Модель «Скрытое размещение Дирихле»

Имеется набор из D документов. Каждый документ состоит из N_d слов, w_{dn} соответствует наблюдаемым переменным – словам в документе. Это единственные наблюдаемые переменные в модели, остальные переменные – скрытые. Переменная z_{dn} принимает значение темы, выбранной на шаге 2(a) для слова w_{dn}. Для каждого документа d переменная θ_d представляет собой распределение тем в этом документе. В

классической модели LDA количество тем фиксировано изначально и задаётся в явном виде параметром K.

В модели LDA предполагается, что параметры θ_d и φ_t распределены следующим образом: $\theta \sim Dir(\alpha)$, $\varphi \sim Dir(\beta)$, где α и β – задаваемые вектора-параметры (т.н. гиперпараметры) распределения Дирихле.

Модель LDA применима лишь к задачам, для которых верны следующие предположения об исходных данных [2]:

1. Последовательность слов в документе не имеет значения
2. Последовательность документов не имеет значения
3. Количество тем известно и не меняется

Если же какие-либо из данных условий не удовлетворяют поставленной задаче, то требуются более сложные модели. Например, при генерации документов, понятных человеку, последовательность слов в генерируемом документе имеет большое значение. Второе предположение может быть неверным при анализе тем в документах из большого временного промежутка. Например, тема, описывающая какую-нибудь научную область, может иметь разное распределение и состав в разные промежутки времени: с течением времени из-за смены приоритетов и терминологии какие-то термины начинают встречаться чаще, а какие-то реже. Третье предположение работает только в том случае, если в задаче априорно известно количество тем в документах, что на практике выполняется редко.

Иерархическое скрытое размещение Дирихле

Модель иерархического скрытого размещения Дирихле (Hierarchical Latent Dirichlet Allocation, hLDA), описанная в работе [5], основана на вложенном процессе китайского ресторана (Nested Chinese Restaurant Process, nCRP).

Рассмотрим стандартный процесс китайского ресторана (Chinese Restaurant Process, CRP)

Пусть некоторый китайский ресторан имеет счетное количество столов. В него по очереди заходят M клиентов. Первый клиент садится за первый стол. Очередной клиент с номером m выбирает стол согласно распределению:

$$
\begin{aligned}
p(\text{стол} = i|\text{клиенты } \overline{1, m-1}) &= \tfrac{m_i}{\gamma+m-1} \\
p(\text{новый стол}|\text{клиенты } \overline{1, m-1}) &= \tfrac{\gamma}{\gamma+m-1}
\end{aligned}
\tag{1}
$$

Здесь m_i – количество посетителей, сидящих за столом i,γ – так называемый концентрационный параметр процесса.

Процесс китайского ресторана можно расширить до вложенного процесса китайского ресторана [5]. Пусть в городе имеется счетное число ресторанов. Каждый стол в ресторане содержит ссылку на другой ресторан. Пусть имеется один корневой ресторан, и в каждый ресторан ведет только одна ссылка. Таким образом, получается древовидная структура ресторанов. Клиент прибывает в город на L дней. В первый вечер посещает корневой ресторан, выбирая стол согласно (1). На следующий день он идет в ресторан, определенный выбранным в корневом ресторане столом, снова выбирает стол согласно (1) и так далее. Каждый день клиент посещает один из ресторанов. Таким образом, он посетит L ресторанов. После того, как город посетят M клиентов, коллекция их путей по ресторанам будет представлять конечное поддерево глубины L бесконечного дерева ресторанов.

Полученное дерево может быть использовано для моделирования иерархии тем. В модели иерархического скрытого размещения Дирихле [5] каждому ресторану из процесса китайского ресторана соответствует тема. Генеративный процесс следующий:

1. Пусть c_1 — корневой ресторан
2. Для каждого уровня дерева $l \in \{2, ..., L\}$:
   a) Выбрать стол в ресторане c_{l-1} согласно CRP
   b) Установить c_l — ресторан, на который ссылается выбранный стол
3. Случайно выбрать для документа распределение по L темам $\theta_d \sim Dir(\alpha)$
4. Для каждого слова в документе:
   a) Случайно выбрать $z \in \{1, ..., L\}$ согласно распределению θ_d
   b) Случайно выбрать слово из распределения слов в теме, соответствующему ресторану c_z

Схема модели hLDA изображена на рисунке 2. Здесь T – дерево ресторанов, получаемое с помощью вложенного процесса китайского ресторана, c_1,c_2,. . . ,c_L – путь по ресторанам, причем значение c_i зависит от c_1,c_2,. . . ,c_{i-1}. Описанная модификация расширяет модель LDA, добавляя возможность существования неограниченного количества тем. Однако количество тем, описывающих один документ, по-прежнему постоянно и равно L.

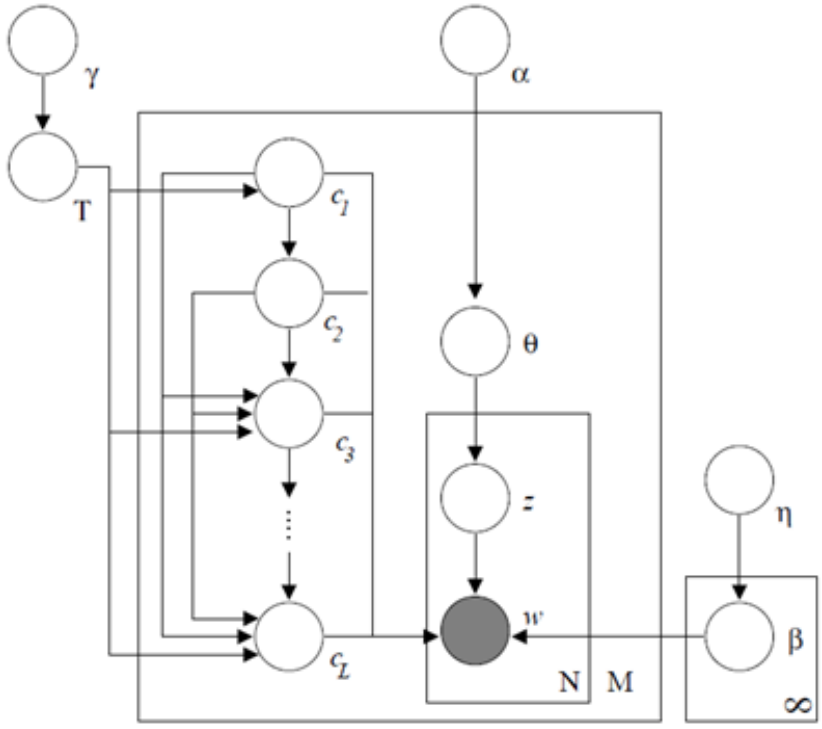

Рисунок 2: Модель «Иерархическое скрытое размещение Дирихле»

Иерархический процесс Дирихле

Иерархический процесс Дирихле (Hierarchical Dirichlet Process, HDP) является Байесовской непараметрической моделью [6] [7], которая может быть использована для тематического моделирования с потенциально бесконечным числом тем.

Случайный процесс $G \sim DP(\alpha, H)$ называется процессом Дирихле с базовым распределением H и параметром концентрации α, если для произвольного конечного разбиения A_1,A_2,. . . ,A_r вероятностного пространства над H выполняется:

$$(G(A_1), G(A_2), ..., G(A_r)) \sim Dir(\alpha H(A_1), \alpha H(A_2), ..., \alpha H(A_r)) \quad (2)$$

Здесь $G(A_i)$ и $H(A_i)$ — маргинальные вероятности G и H над A_i.

Иерархический процесс Дирихле можно понимать как процесс Дирихле над процессом Дирихле. Иначе говоря, при генерации случайного элемента из иерархического процесса Дирихле сначала выбирается, из какого процесса Дирихле верхнего уровня следует генерировать элемент, после чего к выбранному процессу Дирихле применяется стандартная схема генерации элемента. Он моделирует документы, в которых имеются некоторые темы, общие для всего набора данных, а также более специфичные темы.

Другие вероятностные модели

В настоящее время существует множество разнообразных тематических моделей. Каждая модель подходит для своего класса исходных данных.

Для решения задач, где нужно не только найти темы, но и проследить их динамику, используются темпоральные тематические модели [8][9].

Во многих случаях помимо текстов документов известна и информация о них. Эту информацию можно включить в модель. Например, в работе [10] рассматривается модификация LDA, которая учитывает метки (или тэги), которые пользователи назначают текстовым объектам.

Более подробно эти, а также другие методы тематического моделирования, рассмотрены в [11]

2.2 Методы оценки качества тематических моделей

Методов оценки качества тематических моделей существует немного.

Самый тривиальный способ – ручная оценка. Запускается вывод тем из набора данных, затем человек решает, удовлетворяют ли его полученные темы, подходит ли данная модель для данной задачи.

Второй способ – использовать тематические модели в приложениях и вместо оценки качества модели оценивать качество этих приложений. Например, в задаче классификации можно перейти из векторного представления документов в пространстве слов в

векторное представление в пространстве тем (уменьшив размерность векторов), а затем произвести оценку качества полученного алгоритма классификации документов.

Самым распространённым способом оценивания качества вероятностных тематических моделей является расчёт перплексии [12] на тестовом наборе данных D_{test} из M документов:

$$\text{Перплексия}(D_{test}) = \exp\{-\frac{(\sum_{d=1}^{M} \log(p(w_d|\text{модель}))}{(\sum_{d=1}^{M} N_d)}\} \tag{3}$$

Здесь N_d – количество слов в документе w_d, $p(w_d|\text{модель})$ – правдоподобие документа w_d при полученной модели.

Меньшее значение перплексии означает, что модель лучше описывает (обобщает) тестовые данные. Кроме того, минимизируя значение этого критерия, можно экспериментально подобрать оптимальное число различных тем в коллекции документов. Альтернативным подходом является оценивание вероятности второй части документа при условии наличия первой [13]. Для этого каждый документ разделяют на 2 части: первую часть считают обучающими данными, а с помощью второй тестируют качество модели.

2.3 Интерпретируемость результатов тематического моделирования

Приложения, которые предполагают непосредственное взаимодействие пользователя с результатами тематического моделирования, должны также учитывать их интерпретируемость. В исследовании [14] было показано, что модели с наименьшей перплексией обычно хуже интерпретируются обычными людьми. Однако предложенный авторами метод оценивания интерпретируемости моделей предполагает активное участие пользователей и не применим в общем случае.

В работе [15] согласованность темы вычисляется следующим образом:

$$C(t; V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})} \tag{4}$$

$V^{(t)} = (v_1, ..., v_M)$ – список M слов темы t с наибольшей вероятностью. $D(v)$ – количество документов, в которых хотя бы один раз встречается слово v, $D(v, w)$ – количество

документов, в которых хотя бы по одному разу встречаются слова v и w. В числителе 1 прибавляется для сглаживания (числитель не должен быть равен 0).

Данная оценка зависит от исходного набора данных, поэтому она не дает *объективной* оценки интерпретируемости результатов тематического моделирования.

Авторами [16] были предложены методы автоматического оценивания связности найденных тем с помощью внешних баз знаний (*WordNet*, *Wikipedia*, *Google*). В этой работе тема представляется в виде 10 слов, имеющих наибольшую вероятность в теме: $w = (w_1, ..., w_{10})$.

Методы, использующие *WordNet* и *Wikipedia* используют следующие оценки интерпретируемости:

$$\text{Mean-D-Score(w)} = mean\{D(w_i, w_j)|i, j \in 1...10, i < j\} \tag{5}$$

$$\text{Median-D-Score(w)} = median\{D(w_i, w_j)|i, j \in 1...10, i < j\} \tag{6}$$

Здесь $mean$ – среднее, $median$ – медиана. $D(w_i, w_j)$ – это мера близости слов w_i и w_j в используемой базе.

WordNet – это база знаний, в которой слова организованы в иерархию. Многие описанные в [16] меры близости $D(w_i, w_j)$ в *WordNet* основаны на подсчете расстояния в графе (например, длина кратчайшего пути от одной вершины до другой). Также имеются методы, которые вместо близости в графе вычисляют близость значений данных слов (краткое толкование слов имеется в базе *WordNet*).

Методы оценки интерпретируемости с помощью *Wikipedia* используют информацию о ссылках в статье на другие статьи, текстовое содержание статей. Также в этой работе [16] описывается метод PMI (поточечная взаимная информация, Pointwise Mutual Information), использующий весь корпус текстов *Wikipedia*:

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{7}$$

Оценка интерпретируемости темы по (5) и (6) основана на предположении о том, что из попарной близости входящих в тему M слов следует согласованность всех M слов.

При оценке согласованности темы с помощью *Google* вместо подсчета попарной близости используются другие методы. Из всех 10 слов темы составляется запрос в поисковую данную систему, замет анализируются результаты.

В работе [16] рассматривается два подхода.

Первый подход – вычислять, насколько заголовки возвращаемых страниц релевантны данной теме:

$$I(\overrightarrow{w}) = \mathbf{1}[w_i = v_j] \tag{8}$$

Здесь $i = 1, ..., 10$, $j = 1, ..., |V|$. v_j - уникальные слова, встречающиеся в заголовках первых 100 результатов поиска по словам $\overrightarrow{w} = (w_1, w_2, ..., w_{10})$. $\mathbf{1}$ – функция-индикатор для подсчета количества совпадений.

Второй подход – вычислять логарифм количества найденных страниц:

$$I(w) = \log_{10}(\#\text{результатов выдачи по запросу } w) \tag{9}$$

2.4 Особенности текстов микроблогов и методы их нормализации

Тематические модели разрабатывались для применения к текстам книг, статей. Тексты микроблогов отличаются от таких текстов. Поэтому необходимо рассмотреть особенности текстов микроблогов и способы предобработки микроблогов.

Микроблогинг – это разновидность блогинга, позволяющая пользователям писать короткие заметки и публиковать их. Как правило, эти заметки отличаются от обычных текстов, таких как новости, книги, обычные блоги. На рисунке 3 показан набор таких сообщений из сервиса микроблогинга Twitter.

Тексты таких сообщений обладают следующими особенностями:

- **Сообщения короткие**. Обычно одно сообщение несет в себе одну законченную мысль и состоит из одного-двух предложений. Кроме того, сервисы микроблогов ограничивают максимальную длину сообщений (например, в Twitter максимальная длина – 140 символов)

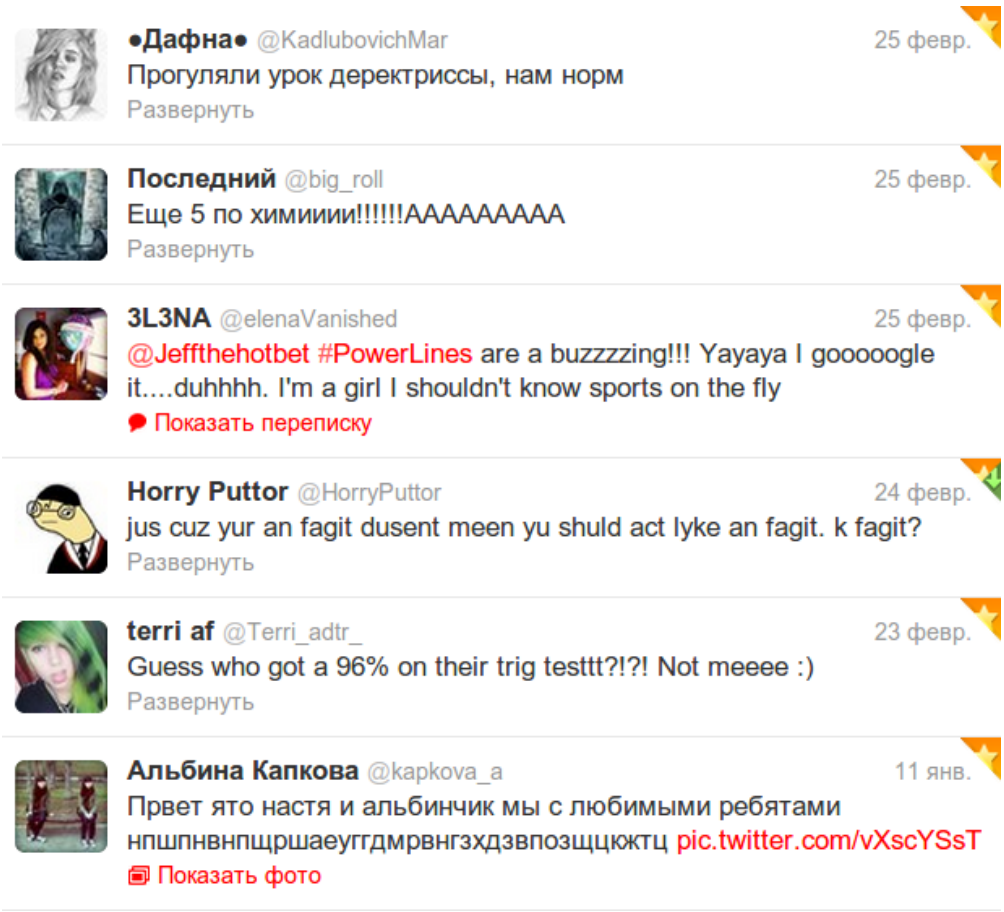

Рисунок 3: Сообщения в сервисе микроблогов – Twitter

- **Сообщения содержат орфографические ошибки**. Так же, как в чатах, пользователи не проверяют свои сообщения на наличие орфографических ошибок, отправляют их сразу после набора последнего символа. Часть ошибок делаются преднамеренно: если длина сообщения больше допустимой, пользователь сокращает слова.

- **Эмоции в сообщениях**. Сообщения в микроблогах представляют собой новости, мнения, которые имеют некоторую эмоциональную окраску. Пользователи передают свои эмоции с помощью эмотиконов (сокращения и значки для обозначения эмоций, напрмер, «:-)»), повторяющихся символов и знаков препинания (например, «AAAA», «!!!!»)

- **Особенности сервиса**. В сервисах микроблогов есть свои особенности и символы, имеющие специальные значения. Например, в Twitter есть символ @, после

которого следует имя пользователя Twitter. Также, в Twitter есть специальный символ #, который означает хэштеги – специальные ключевые слова, указываемые пользователями.

Из-за перечисленных особенностей применять стандартные методы анализа текстов к данным из микроблогов нельзя. В этом случае перед применением алгоритмов производится *нормализация* текстов микроблогов, т.е. приведение текстов к виду, близкому к традиционным текстам.

Выделяется 5 типов орфографических ошибок [17]:

- Письмо и звук (Letter&Sound): b4 → before
- Письмо (Letter): shuld → should
- Звук (Sound): 4 → for, 2 → to
- Сленг (Slang): omg → Oh my god
- Другие (Other): sucha → such a

Орфографические ошибки исправляются с помощью поиска "неправильных"слов (т.е. тех, которых нет в словаре) и заменой их на наиболее подходящее "правильные"слова из словаря [3]. Другая группа методов основана на использовании статистического машинного перевода [18]: перевод с языка микроблогов на естественный. Для этого нужно иметь обученный переводчик.

Для поиска и удаления эмотиконов используются специальные списки популярных "смайлов"[1] и регулярные выражения.

[1] `http://en.wikipedia.org/wiki/List_of_emoticons`

3 Исследование и построение решения задачи

В этом разделе описывается практическая часть дипломной работы.

Сначала будут описаны исходные данные. Затем будет рассказано о разработанном методе нормализации текстов. После этого будут описаны разработанные методы оценки интерпретируемости тем, получаемых в результате тематического моделирования.

3.1 Исходные данные

В качестве источника данных в данной работе используется социальная сеть Twitter, являющаяся самым популярным на сегодняшний день сервисом микроблогинга.

С помощью Streaming Twitter API [1] был скачан поток твитов в период с 6 Марта 2013 11:51:10 +0000 по 06 марта 2013 20:43:03 +0000. Каждый твит представляет собой JSON-объект, в котором имеется его текст, а также метаинформация, в том числе и расположение в тексте гиперссылок, хэштегов (#hashtag), упоминаний пользователей (@username).

Полученные сообщения были отфильтрованы: остались только твиты на английском языке. Язык твита определялся по тексту с помощью свободной java-библиотеки langdetect [2]. Окончательный размер набора данных - 490415 твитов.

3.2 Нормализация твитов

После скачивания потока твитов, все тексты твитов были нормализованы. Для этого был разработан и реализован алгоритм, приводящий твиты к виду, близкому к текстам традиционных источников.

Нормализация проходит в несколько этапов:

1. *Удаление упоминаний пользователей* (@username). Используются метаданные твита. Каждое сообщение, получаемое с помощью Twitter API, представляет собой JSON объект с различными полями. В частности, там присутствет текст и информация о том, что в некоторох подстроках текста твита содержатся упоминания пользователей, гиперссылки, хэштеги.

[1] `http://dev.twitter.com/docs/streaming-apis`

[2] `http://code.google.com/p/language-detection/`

2. *Удаление гиперссылок* (http://example.com). Используются метаданные твита.

3. *Удаление эмотиконов* («:)», «xD», «o_O», «:-(», ...). Используется список часто встречаемых "смайликов"[1].

4. *Удаление пунктуации, неанглийских символов и повторяющихся пробелов*. На данном этапе удаляется пунктуация, так она не учитывается в рассматриваемых далее тематических моделях. При этом рассматриваются и те случаи, в которых знаки препинания не отделены пробелами справа: символ заменяется на пробел. Затем все повторяющиеся пробелы заменяются одним. В конце этого этапа получается набор слов из английских букв и цифр, каждая пара слов разделена одним пробелом.

5. *Приведение к нижнему регистру.*

6. *Токенизация*: выделение слов по пробелам для дальнейшей работы со списком слов. Здесь используется самый простой способ токенизации, так как пунктуация уже была удалена на предыдущих этапах.

7. *Удаление стоп-слов* (about, been, does, when, ...). Используется список стоп-слов длиной 537 слов[2].

8. *Удаление слов длины меньше 3*. Предполагается, что слова длины меньше 3 не несут смысла, и их можно удалить при тематическом моделировании

9. *Исправление орфографических ошибок* При этом #хэштеги и именованные сущности игнорируются. Хэштеги – это слова, начинающиеся на #. Таким способом пользователи помечают и группируют сообщения группируют сообщения. Именованные сущности извлекаются с использованием списка именованных сущностей, полученных в отделе Информационных систем ИСП РАН с использованием Википедии [3]: именованные сущности извлекаются из заголовков статей, с использованием структуры категорий Википедии. На данном этапе используются два метода:

[1] `http://en.wikipedia.org/wiki/List_of_emoticons`
[2] `http://members.unine.ch/jacques.savoy/clef/englishST.txt`
[3] `http://en.wikipedia.org/`

(a) Замена неправильных слов правильными с использованием словаря *неправильное слово* → *правильное слово* (ffor → for, amazzing → amazing, kidn → kidding). Данный словарь был предоставлен автором статьи ([19]).

(b) Замена неправильных слов с использованием сторонней билиотеки Java Suggester [1] и юниграммную модель языка, полученную из набора данных блогов, собравшегося в отделе Информационных систем ИСП РАН. Сначала для слова, которого нет в словаре, Java Suggester предлагает несколько кандидатов. Затем из кандидатов выбирается наиболее частое слово согласно юниграммной модели.

10. *Детокенизация*. Получение из набора слов строки со словами, разделенными проблемами.

    Примеры работы алгоритма нормализации:

    - I swear geminis are the most dangerous → swear geminis dangerous
    - eric went crazy todae . :pppp → eric crazy today
    - Frustated . → frustrated

3.3 Оценка интерпретируемости

За основу разрабатываемых методов были взяты методы, основанные на использовании внешних баз знаний: Wikipedia и Google.

В главе 2.3 описаны алгоритмы, которым на вход подается 10 слов. В качестве этих слов берутся 10 слов темы, имеющих наибольшую вероятность. Здесь предполагается, что эти слова являются ключевыми в теме. В данной работе будет предложены и исследованы другие способы выбора ключевых слов.

Извлечение ключевых слов из темы

Тема – это распределение над словами. Из этого распределения нужно получить множество слов, которые считаются ключевыми. В данной работе предполагается, что клю-

[1] `http://softcorporation.com/products/suggester/`

чевые слова – это первые k слов, имеющих наибольшую вероятность в теме. Основная проблема – понять, сколько слов выбрать, т.е. найти k.

Далее предполагается, что слова в теме отсортированы по убыванию их вероятностей: $p(w_1|\text{тема}) \leq p(w_2|\text{тема}) \leq ...$

Предлагается три метода оценки k:

- $k = const$. Заранее выбирается константа, и каждая тема содержит одинаковое количество ключевых слов. Данный подход использовался в [16] ($k = 10$).

- Во втором подходе предполагается, что распределение темы должно влиять на количество ключевых слов: чем больше суммарная вероятность первых слов, тем меньше должно быть k. Здесь сначала выбирается константа T, а затем для каждой темы выбирается k по следующему правилу:

$$\sum_{i=1}^{k} p(w_i|\text{тема}) \leq T \tag{10}$$

$$\sum_{i=1}^{k+1} p(w_i|\text{тема}) > T \tag{11}$$

  Здесь $p(w_i|\text{тема})$ – вероятность i-го слова в данной теме. Смысл данного метода в том, что если в распределении несколько слов имеют существенно б*о*льшую вероятность, чем, остальные, то выбирается это небольшое количество слов. Если распределение ближе к равномерному, то ключевых слов будет больше.

- При работе с разными тематическими моделями и разными значениями параметров этих моделей, получаются разные распределения тем. Поэтому для каждой модели приходится вручную подбирать число T. В связи с этим предлагается способ выбора T. Рассматривается H слов для каждой темы, полученной с помощью одной модели, и выбирается T следующим образом:

$$T = \text{медиана}_t\{\sum_{i=1}^{H} p(w_i|\text{тема}_t)\} \tag{12}$$

  Таким образом, в половине тем количество ключевых слов меньше H, а в другой половине – больше H. В данном методе задается медианное количество слов, которые будут считаться ключевыми, однако в зависимости от распределение количество ключевых слов для каждой темы получается разным.

В первом методе задается параметр k, во втором – T, в третьем – H.

Оценка на основе Wikipedia

Пусть имеется k ключевых слов: $w_1, w_2, ..., w_k$. Оценка интерпретируемости I вычисляется как среднее значение PMI (7) по всем парам ключевых слов:

$$I(w_1, w_2, ..., w_k) = \frac{\sum_{1 \le i < j \le k} PMI(w_i, w_j)}{|i, j : 1 \le i < j \le k|}$$

Здесь используются вероятности встретить слово в случайной статье Википедии, а также вероятности встретить два слова в одной статье Википедии. Для того, чтобы избежать нулевых вероятностей, применяется сглаживание Лапласа:

$$PMI(w_1, w_2) = \log \frac{N^2(\#(w_1, w_2) + \delta)}{\#(w_1)\#(w_2)(N + |V|)} \tag{13}$$

Здесь $\delta > 0$ – параметр сглаживания, в данной работе он равен 1. $|V|$ – количество различных слов в документах.

Данный метод использовался в [16].

Оценка на основе Google

Пусть имеется k ключевых слов: $w_1, w_2, ..., w_k$. Оценка интерпретируемости I вычисляется следующим образом:

$$I(w_1, w_2, ..., w_k) = \log(|\text{результаты выдачи по запросу } w_1\ w_2\ ...\ w_k|)$$

При этом, для каждой темы происходит автоматическое скачивание страницы результатов выдачи по запросу в поисковую систему Google и извлечение из нее количества результатов поиска.

Данный метод использовался в [16].

3.4 Результаты экспериментов

В данном разделе описываются проведенные эксперименты и их результаты.

Перед проведением экспериментов с тематическими моделями скачанный поток твитов был нормализован (см. главу 3.2).

С использованием нормализованных текстов сравниваются различные методы оценки интерпретируемости между собой. Полученные значения сравниваются с оценками экспертов.

Затем на основе этих экспериментов выбирается один метод, который далее используется для сравнения различных тематических моделей с разными параметрами.

Сравнение методов оценки интерпретируемости

На нормализованных текстах твитов было произведено тематическое моделирование с использованием моделей "Скрытое размещение Дирихле"(LDA) и "Иерархический процесс Дирихле"(HDP). Использовались готовые реализации на языке C [1].

После этого полученные темы были предложены 5 экспертам для оценки интерпретируемости. Каждая тема представлялась экспертам как 10 слов, имеющих наибольшую вероятность в данной теме. Задача экспертов – для каждой темы поставить целое число от 1 до 3. 3 означает, что слова хорошо связны, из них легко составить короткое предложение, описывающее тему. 1 означает, что слова плохо связаны между собой и вряд ли образует тему, сложно придумать короткое предложение, содержащее данные слова. Дополнительно были предложены заранее подготовленные наборы ключевых слов: один набор представляет собой множество хорошо связанных слов, для него ожидалась оценка 3, другой – случайные несвязные слова, ожидалась оценка 1. Эти темы добавлены для того, чтобы убедиться, что эксперт правильно понял задачу и ответственно отнесся к ее выполнению.

Для этих же тем были применены методы автоматической оценки интерпретируемости:

- База знаний Google. $k = const = 10$.
- База знаний Google. $T = 0.3$ (LDA), $T = 0.03$ (HDP), k вычисляется для каждой темы согласно (10), (11)
- База знаний Google. $H = 10$, T вычисляется согласно (12), k вычисляется для каждой темы согласно (10), (11)

[1] `http://www.cs.princeton.edu/~blei/topicmodeling.html`

- База знаний Wikipedia. $k = const = 10$.
- База знаний Wikipedia. $T = 0.3$ (LDA), $T = 0.03$ (HDP), k вычисляется для каждой темы согласно (10), (11)
- База знаний Wikipedia. $H = 10$, T вычисляется согласно (12), k вычисляется для каждой темы согласно (10), (11)

Далее вычисляется корреляция между оценками экспертов и автоматическими оценками. Значения корреляции отображены в таблицах 1 и 2

Таблица 1: Корреляция между оценками экспертов и методами автоматической оценки интерпретируемости. Модель: Скрытое Размещение Дирихле (LDA)

выбор ключевых слов	*параметры*	*Wikipedia*	*Google*
константа	$k = 10$	-0.03	0.12
из распределения	$k \leftarrow T = 0.3$	0.26	**0.48**
по совокупности распределений	$k \leftarrow T \leftarrow H = 10$	0.23	0.30

Таблица 2: Корреляция между оценками экпсертов и методами автоматической оценки интерпретируемости. Модель: Иерархический процесс Дирихле (HDP)

выбор ключевых слов	*параметры*	*Wikipedia*	*Google*
константа	$k = 10$	0.30	0.51
из распределения	$k \leftarrow T = 0.03$	-0.01	**0.53**
по совокупности распределений	$k \leftarrow T \leftarrow H = 10$	0.17	0.48

Согласно полученным результатам, методы, использующие Google показывают наибольшую корреляцию с оценками экспертов. Поэтому для дальнейших экспериментов будет использоваться метод, основанный на Google. Несмотря на то, что при заданном T корреляция выше, чем при заданном H, далее будет использоваться метод, где T вычисляется из H согласно (12). Этот выбор связан с тем, что при разных параметрах моделей получаются темы с различными распределениями слов, поэтому один раз задается H вместо многократного ручного подбора T.

Стоит заметить, что в статье [16] также сравнивались методы, использующие Wikipedia и Google. Число ключевых слов задается постоянным для всех тем: $k = 10$. Эксперименты проводятся на текстах книг и новостных статей. Наибольшую корреляцию с оценками экспертов показывает метод на основе Google, так же, как и в данной работе.

Из таблиц 1 и 2 видно, что предложенные методы (из распределения и по совокупности распределений) показывают б*о*льшую корреляцию с мнениями экспертов, чем метод (константа), используемый в [16].

Сравнение тематических моделей

Для сравнения тематических моделей был выбран метод оценки интерпретируемости, использующий базу знаний Google. При этом, количество используемых ключевых слов в темах выбирается по совокупности распределений (с заданием параметра H).

На основе выбранного метода оценки интерпретируемости было проведено сравнение двух моделей: скрытое размещение Дирихле (LDA) и иерархический процесс Дирихле (HDP). Выбор данных моделей обусловлен тем, что на данный момент они наиболее известные и широко используемые. LDA – классическая модель, в которой число тем задается в виде входного параметра. HDP – одна из самых распространенных непараметрических моделей, в которой количество тем не задается явно, а выводится при ее работе.

С моделью LDA были проведены две серии экспериментов: оценка интерпретируемости получаемых тем с различными количествами тем (рисунок 4), а затем с различными значениями параметра α (рисунок 5). При этом во второй серии экспериментов число тем равно 5, так как на нем достигается максимум в первой серии.

С моделью HDP были также проведены две серии экспериментов: оценка интерпретируемости получаемых тем с различным значением параметра γ процесса Дирихле (рисунок 6), а затем с различными значениями параметра α (рисунок 7). При этом во второй серии экспериментов $\gamma = 0.05$, так как на нем достигается максимум в первой серии.

На представленных графиках видно, что при анализе текстов микроблога Twitter темы, получаемые с помощью модели LDA имеют б*о*льшую оценку интерпретируемости, чем темы, получаемые с помощью HDP, при условии правильного выбора параметра

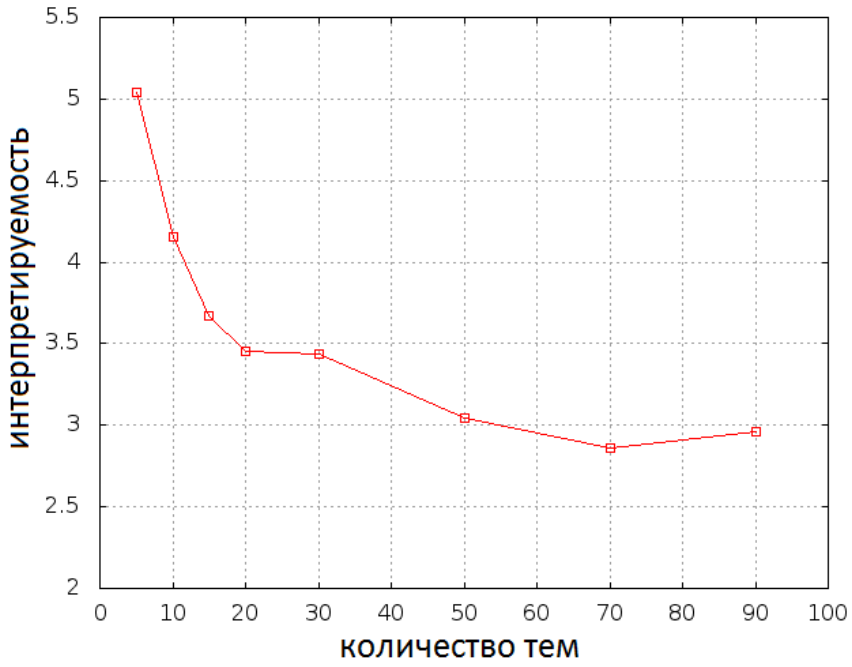

Рисунок 4: Скрытое размещение Дирихле (LDA). Зависимость интерпретируемости от числа тем (T)

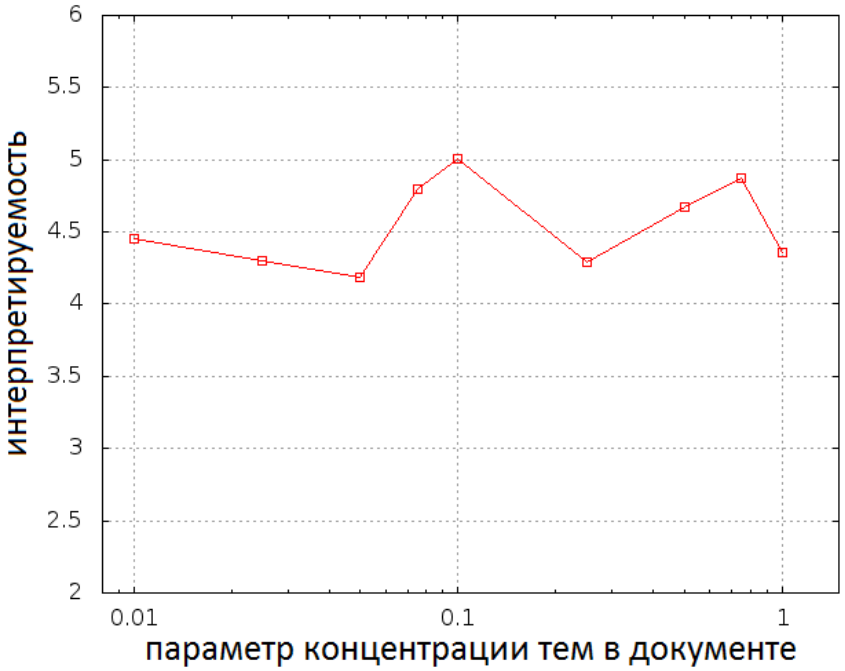

Рисунок 5: Скрытое размещение Дирихле (LDA). Зависимость интерпретируемости от параметра концентрации тем в документе (α). $T = 5$

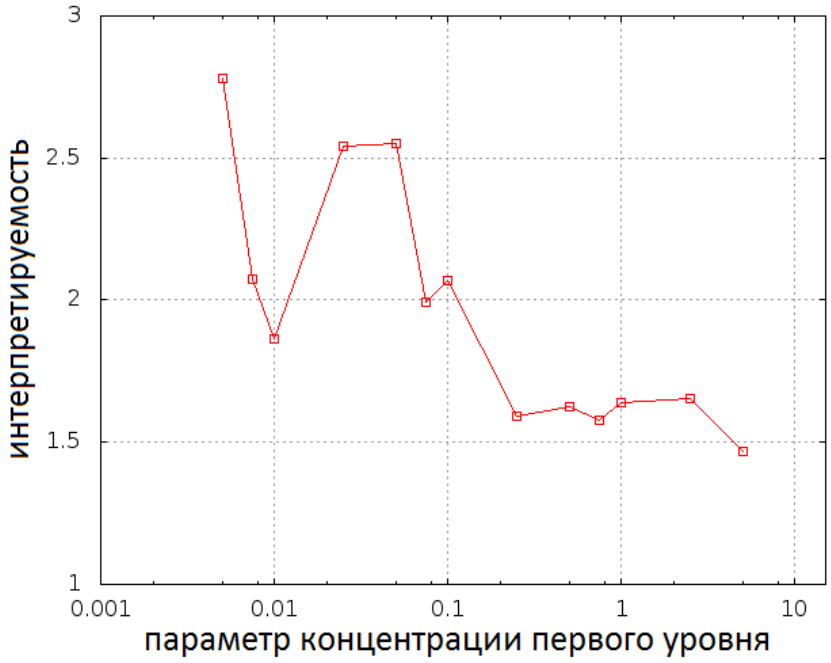

Рисунок 6: Иерархический процесс Дирихле (HDP). Зависимость интерпретируемости от параметра концентрации первого уровня (γ)

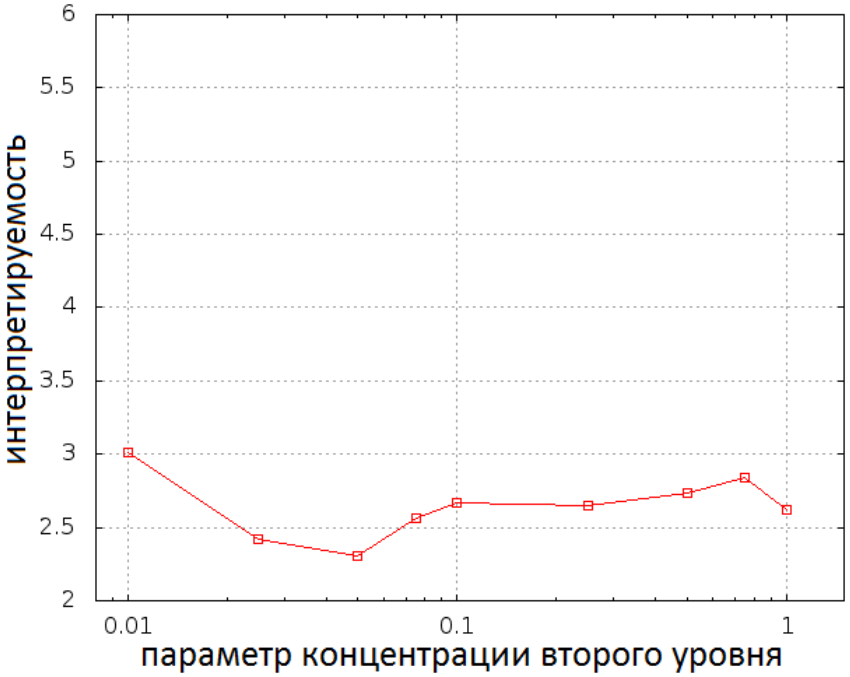

Рисунок 7: Иерархический процесс Дирихле (HDP). Зависимость интерпретируемости от параметра концентрации второго уровня (α). $\gamma = 0.05$

LDA – количества тем.

Из рисунках 6 и 7 видно, что наиболее интерпретируемые темы получаются при $\gamma = 0.05$. От параметра концентрации второго уровня интерпретируемость зависит несущественно.

4 Описание практической части

4.1 Обоснование выбранного инструментария

В качестве языка программирования, на котором выполнялась реализация представленных в данной работе методов, был выбран язык Java. Это объектно-ориентированный язык, который хорошо подходит для прикладных задач. Кроме того, программы, написанные и скомпилированные на Java можно запускать на любой операционной системе, где поддерживается запуск виртуальной машины Java.

При проверке орфографии на этапе предобработки данных использовалась библиотека BasicSuggester[1]. Для оценки интерпретируемости с помощью Google использовалась разрабатываемая в ИСП РАН утилита для скачивания веб-страниц из сети Интернет. Обе библиотеки реализованы на Java, что также является доводов в пользу данного языка программирования.

Использовались готовые реализации тематических моделей на языке C[2]. Выбор этих реализаций обусловлен тем, что они принадлежат авторам исследуемых методов тематического моделирования. Кроме того, язык C хорошо подходит для таких задач, где производится большое количество вычислений.

4.2 Схема работы

Исходные данные – поток твитов на английском языке. При этом, нормализация и оценка интерпретируемости были реализованы на языке Java. Для тематического моделирования использовались готовые реализации на языке C.

Общая схема работы представлена на рисунке 8.

Сначала выполняется нормализация твитов. Дальнейшая работа будет выполняться только с нормализованными твитами.

Общая архитектура представлена на рисунке 9.

На нормализованных твитах запускаются готовые реализации тематических моделей. Используемые реализации имеют свой формат водных и выходных данных. Поэтому была реализована утилита для преобразования исходного набора текстов в требуе-

[1] `http://softcorporation.com/products/suggester/`

[2] `http://www.cs.princeton.edu/~blei/topicmodeling.html`

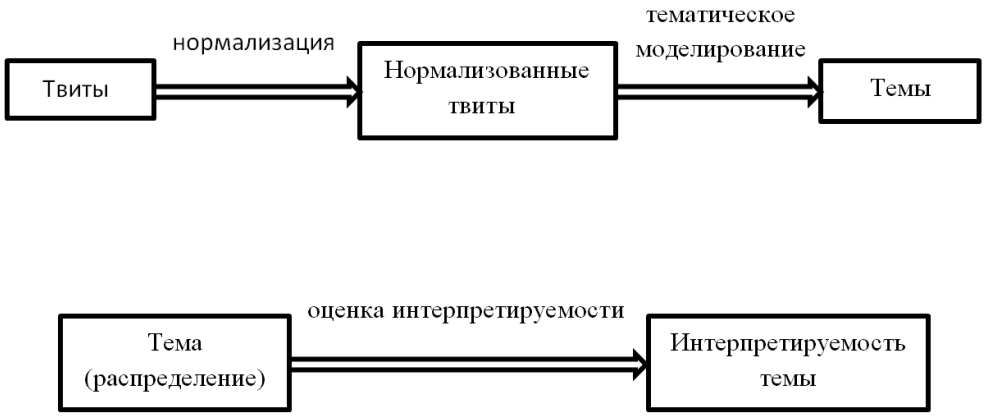

Рисунок 8: Общая схема работы

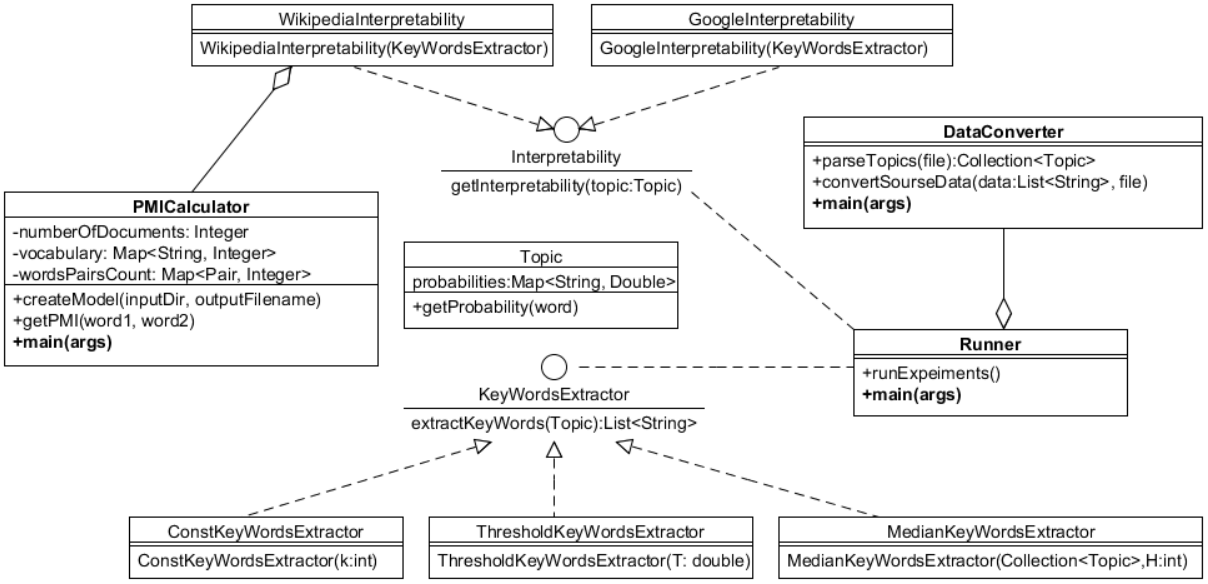

Рисунок 9: Диаграмма классов. Общая архитектура системы

мый формат. Также реализовано преобразование тем, получаемых в формате используемых реализаций во внутреннее представление, которое используется для вычисления их интерпретируемости. За это отвечает класс *DataConverter* (см. рисунок 9)

Для быстрого вычисления значений PMI для пар слов сначала была создана модель, которая была сохранена на диске и использовалась в дальнейшем. Модель содержит: общее количество документов Википедии; для каждого слова – количество документов, в которых это слово встретилось хотя бы один раз; для каждой пары слов – количество документов, в которых встретились оба слова пары. Значения PMI извлекались из 4

Подзадача	среднее время
Создание модели PMI из 4млн. документов	5 ч 27 мин
Сортировка одной темы (qsort)	4.7 с
Оценка интерпретируемости одной темы. Google	0.4 с
Оценка интерпретируемости одной темы. Wikipedia	<1 мс

Таблица 3: Время работы отдельных частей системы

млн. статей Википедии на английском языке. Так как в данной работе используются только значения PMI для ключевых слов получаемых тем, то слова, не входящие в первые 100 слов в любой полученной теме, игнорировались. Процесс создания и сохранения модели реализован в классе *PMICalulator* (см. рисунок 9). Также этот класс используется для вычисления PMI согласно (13).

Были реализованы три способа выбора количества ключевых слов: константное значение, из распределения, из совокупности распределений (классы *ConstKeyWordsExtractor*, *ThresholdKeyWordsExtractor* и *MedianKeyWordsExtractor* соответственно). Первому в конструктор передается целая константа, второму – вещественный порог, третьему – семейство тем, по которому оценивается порог, и целый параметр H. При выборе ключевых слов происходит сортировка слов по убыванию их вероятности в теме.

Для оценки интерпретируемости с помощью внешних баз знаний используются классы *WikipediaInterpretability* и *GoogleInterpretability* (см. рисунок 9). Класс *Wikipedia Interpretability* вычисляет среднее значение PMI по всем парам различных ключевых слов темы, использует *PMICalulator* для вычисления PMI для двух слов. Класс *Google Interpretability* составляет из ключевых слов запрос и запускает функцию из библиотеки, которая по запросу возвращает количество результатов выдачи по данному запросу в поисковой системе Google.

Класс *Runner* загружает темы, полученные в результате тематического моделирования, в память (используя класс *DataConverter*), выполняет оценку интерпретируемости набора тем с помощью представленных в работе методов. При этом создаются различные экземпляры классов, реализующих интерфейсы *Interpretability* и *KeyWordsExtractor*.

В таблице 3 представлены времена работы отдельных частей системы.

Заключение

В данной работе были исследованы методы определения тематической направленности текстового содержимого микроблогов и реализован алгоритм автоматической оценки интерпретируемости результатов тематического моделирования текстов микроблогов. Были выполнены следующие задачи:

1. Исследованы существующие методы тематического моделирования и способы оценки их качества
2. Разработаны и реализованы методы автоматической оценки интерпретируемости результатов тематического моделирования по ключевым словам тем
3. Выполнена экспериментальная оценка интерпретируемости методов тематического моделирования текстов микроблогов с использованием разработанных методов

В работе было показано, что методы автоматической оценки обладают положительной корреляцией с оценками экспертов, поэтому есть основания применять их для анализа тематических моделей. Также было показано, что непараметрические модели выдают менее интерпретируемые темы, чем параметрические, при условии выбора правильных параметров.

Дальнейшие направления развития:

- Улучшение процесса нормализации текстов микроблогов
- Поиск или разработка тематических моделей, ориентированных на короткие документы
- Исследование методов объединения нескольких сообщений микроблогов в один документ для лучших результатов тематического моделирования с использованием традиционных моделей
- Исследование и разработка новых подходов для оценки интерпретируемости результатов тематического моделирования

Список литературы

[1] Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation //the Journal of machine Learning research. – 2003. – Т. 3. – С. 993-1022.

[2] Blei D. M. Probabilistic topic models //Communications of the ACM. – 2012. – Т. 55. – №. 4. – С. 77-84.

[3] Han B., Baldwin T. Lexical normalisation of short text messages: Makn sens a #twitter //Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. – 2011. – Т. 1. – С. 368-378.

[4] Heinrich G. Parameter estimation for text analysis //Web: http://www. arbylon. net/publications/text-est. pdf. – 2005.

[5] Griffiths T., Jordan M., Tenenbaum J. Hierarchical topic models and the nested Chinese restaurant process //Advances in neural information processing systems. – 2004. – Т. 16. – С. 106-114.

[6] Teh Y. W. et al. Hierarchical dirichlet processes //Journal of the American Statistical Association. – 2006. – Т. 101. – №. 476.

[7] Wang C., Paisley J., Blei D. M. Online variational inference for the hierarchical Dirichlet process //Artificial Intelligence and Statistics. – 2011.

[8] Blei D. M., Lafferty J. D. Dynamic topic models //Proceedings of the 23rd international conference on Machine learning. – ACM, 2006. – С. 113-120.

[9] Wang X., McCallum A. Topics over time: a non-Markov continuous-time model of topical trends //Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2006. – С. 424-433.

[10] Ramage D. et al. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora //Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. – Association for Computational Linguistics, 2009. – С. 248-256.

[11] Коршунов А., Гомзин А. Тематическое моделирование текстов на естественном языке //Труды Института системного программирования РАН. – 2012.

[12] Jelinek F. et al. Perplexity—a measure of the difficulty of speech recognition tasks //The Journal of the Acoustical Society of America. – 1977. – Т. 62. – С. S63.

[13] Rosen-Zvi M. et al. The author-topic model for authors and documents //Proceedings of the 20th conference on Uncertainty in artificial intelligence. – AUAI Press, 2004. – С. 487-494.

[14] Boyd-Graber J. et al. Reading tea leaves: How humans interpret topic models //Proceedings of the 23rd Annual Conference on Neural Information Processing Systems. – 2009.

[15] Mimno D. et al. Optimizing semantic coherence in topic models //Proceedings of the Conference on Empirical Methods in Natural Language Processing. – Association for Computational Linguistics, 2011. – С. 262-272.

[16] Newman D. et al. Automatic evaluation of topic coherence //Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. – Association for Computational Linguistics, 2010. – С. 100-108.

[17] Bo Han, Timothy Baldwin Lexical Normalisation of Tweets [PDF] (`http://www.nicta.com.au/pub?doc=4751`)

[18] Kaufmann M., Kalita J. Syntactic normalization of Twitter messages //International Conference on Natural Language Processing, Kharagpur, India. – 2010.

[19] Han B., Cook P., Baldwin T. Automatically constructing a normalisation dictionary for microblogs //Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. – Association for Computational Linguistics, 2012. – С. 421-432.