САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра Системного Программирования

Бусыгина Мария Александровна

Определение тематики сообществ в социальных сетях

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор А.Н. Терехов

Научный руководитель:
к. ф.-м. н., доцент Д.Ю. Бугайченко

Рецензент:
аспирант А.А. Дзюба

Санкт-Петербург
2014

SAINT-PETERSBURG STATE UNIVERSITY
Mathematics & Mechanics Faculty

Chair of Software Engineering

Busygina Mariya

Identifying communities domains in social networks

Graduation Thesis

Admitted for defence.
Head of the chair:
Professor A.N. Terekhov

Scientific supervisor:
PhD D.Y. Bugaychenko

Reviewer:
Graduate Assistant A.A. Dzyuba

Saint-Petersburg
2014

Оглавление

Введение

В современном мире количество информации постоянно растет. Значительная ее часть представляет собой неструктурированные текстовые данные, например, различные web-ресурсы, социальные сети, блоги, форумы, новостные сайты и пр. Человеку сложно самостоятельно обрабатывать их. Более того, ручной анализ неэффективен для больших объемов текста, т.к. он ограничен скоростью, погрешностями и ошибками, обусловленными человеческим фактором. Следовательно, требуются методы, способные автоматически обрабатывать такие данные. Методы могут основываться на извлечении определенной информации, например, списках тем, содержащихся в текстах [21].

В данном случае, тема определяется как дискретное вероятностное распределение в пространстве слов заданного словаря. Текст может состоять из огромного числа слов, однако эти слова могут порождаться небольшим числом тем, как смесью распределений [7]. Таким образом, с помощью тем можно представить текст в пространстве тем вместо пространства слов, что позволяет более эффективно обрабатывать их.

Наиболее популярные методы определения тем основаны на анализе семантических моделей текста. Модели семантики текста описывают текстовые единицы и их взаимосвязи между собой, основываясь на семантических значениях используемых текстовых единиц. В качестве текстовых единиц, в зависимости от модели, могут быть слова, фрагменты текста (например, предложения), документы коллекции и т.п [23]. В последнее время в задачах, связанных с определением тем активно используются тематические модели (topic models), которые являются разновидностью семантических моделей, что показано в работе [23]. В качестве текстовых единиц в тематических моделях рассматриваются слова (термы). Принцип работы заключается в том, что на основе статистических методов тематическая модель определяет, к каким темам относится каждый текст и какие слова образуют каждую из тем, представляющих собой список встречающихся рядом друг с другом слов, которые упорядоченны по степени принадлежности темы [23]. Процесс построения модели называется тематическим моделированием.

Тематическое моделирование применяется для решения широкого спектра задач, таких как [22]: тематический поиск документов и объектов, кластеризация, классификация, аннотирование текстовых документов, поиск экспертов, рецензентов, проектов, выявление трендов и фронта исследований, анализ и агрегирование новостных потоков, рекомендательные сервисы (коллаборативная фильтрация), аннотация генома и другие задачи биоинформатики и пр.

Как правило, коллекцию текстовых документов, которую используют в качестве исходных данных для построения тематической модели, принято называть корпу-

сом текстов (text corpora). Например, в различных исследованиях часто используют общедоступные корпусы «NIPS proceedings», «Cite seer», «TREC AP» и «Reuter's». В данной работе в качетве корпуса текстов рассмотрен набор постов, опубликованных в сообществах социальной сети «Одноклассники». Текстовое содержимое постов несколько отличается от содержимого других текстовых источников: статей, блогов [21]. Тексты постов включают в себя данные на нескольких языках, их длина сильно разнится, а также им свойственно большое количество авторского форматирования, клонов, рекламы и пр. В связи с этим при тематическом анализе данного текстового содержимого возникают трудности.

Работа имеет следующую структуру. В разделе 1 дан краткий обзор существующих методов тематического моделирования. Раздел 2 посвящен исследованию метода вероятностного латентно-семантического анализа. В разделе 3 дана оценка полученной модели.

1. Постановка задачи

Целью данной работы является определение тем сообществ социальных сетей с помощью методов вероятностного тематического моделирования на примере корпуса текстов, состоящего из постов, размещенных в сообществах социальной сети «Одноклассники». Для этого необходимо решить следующие задачи.

1. Исследовать существующие методы вероятностного тематического моделирования и методы оценки их качества.
2. Реализовать алгоритм построения тематической модели.
3. Оценить качество полученной тематической модели.

2. Обзор подходов

В данном разделе рассмотрены основные методы применяемые для построения тематических моделей.

2.1. Стандартные гипотезы тематического моделирования

Предполагается выполнение следующих гипотез [22].

- Порядок текстов в корпусе не важен — «мешок документов» (bag of documents).
- Порядок слов в тексте не важен — «мешок слов» (bag of words).
- Слово в разных формах — одно и тоже слово.
- Слова, которые часто встречаются во многих текстах не важны.

2.2. Векторное представление

Векторная модель (Vector Space Model, VSM) [19] представления текстов является одним из первых способов, применяемых для решения задач тематического моделирования. Изначально эта модель применялась в 1996-1997 годах в задачах определения тем (Topic Detection and Tracking) путем извлечения событий из потока информации [4, 24]. Представление корпуса в данном случае происходит с помощью векторов из одного общего для всей коллекции векторного пространства, в котором каждому слову сопоставляется вес в соответствии с выбранной весовой функцией [24]. Для полного определения векторной модели необходимо указать, каким именно образом будет определяться вес слова в документе. Для этого используются различные методы: статистический подход (булевский вес [19], tf-idf [17], логарифм вхождения слова в текст [25] и пр.), место появления слова, оформление слова и др [25]. При таком представлении текстов можно, например, находить расстояние между точками и решать задачу подобия документов — чем ближе расположены точки, тем больше похожи рассматриваемые тексты [24].

2.3. Латентно-семантический анализ

Латентно-семантический анализ (LSA, Latent Semantic Analysis) — это теория и метод для извлечения контекстно-зависимых значений слов при помощи статистической обработки больших наборов текстовых данных [6]. LSA был запатентован в 1988 году. В области информационного поиска данный подход называют латентно-семантическим индексированием (LSI, Latent Semantic Indexing). LSA также работает

с векторным представлением типа «мешка слов» текстовых единиц. Текстовый корпус представляется в виде числовой матрицы — слово-текст, строки которой соответствуют словам, а столбцы текстовым единицам — текстам. Существуют различные схемы определения каждого элемента данной матрицы (см. пункт 2.2 Векторное представление).

Объединение слов в темы и представление текстовых единиц в пространстве тем осуществляется путем применения к данной матрице одного из матричных разложений. Наиболее популярными являются: сингулярное разложение [16] и факторизация неотрицательных матриц [15]. Например, согласно теореме о сингулярном разложении, любая вещественная прямоугольная матрица может быть разложена на произведение трех матриц: $A = USV^T$, где $A \in R^{n \times m}$, матрицы $U \in R^{n \times k}$ и $V \in R^{m \times k}$ – ортогональные, а $S \in R^{k \times k}$ – диагональная матрица, значения на диагонали которой называются сингулярными значениями матрицы A, V^T - транспонированная матрица.

Если в матрице S оставить только k наибольших сингулярных значений, а в матрицах U и V – только соответствующие этим значениям столбцы, то произведение получившихся матриц S, U и V будет наилучшим приближением исходной матрицы A к матрице A' ранга k: $A' \approx A = USV^T$. Так как в качестве матрицы A использовалась матрица слова-на-тексты, то матрица A', содержащая только k первых линейно независимых компонент A, отражает основную структуру различных зависимостей, присутствующих в исходной матрице. Структура зависимостей определяется весовыми функциями слов. Таким образом, каждое слово и текст представляются при помощи векторов в общем пространстве размерности k — пространстве гипотез. Сходство между любой комбинацией слов и/или текстов легко вычисляется при помощи скалярного произведения векторов. Как правило, выбор k зависит от поставленной задачи и подбирается эмпирически. Если выбранное значение k слишком велико, то метод теряет свою мощность и приближается по характеристикам к стандартным векторным методам. Слишком маленькое значение k не позволяет улавливать различия между похожими словами или текстами [13].

Существенным недостатком метода LSA является значительное снижение скорости вычисления при увеличении объема входных данных (например, при SVD-преобразовании). Как показано в [13], скорость вычисления соответствует порядку N^{2*k}, где $N = N_{text} + N_{word}$ — сумма количества текстов и слов, k — размерность пространства факторов. Также нельзя не отметить, что у результатов работы матричных разложений отсутствует явное лингвистическое обоснование, поэтому не всегда понятно, как их оценивать и интерпретировать.

2.4. Вероятностное моделирование

Вероятностное тематическое моделирование – это набор алгоритмов, позволяющих анализировать слова в текстовых корпусах и извлекать из них темы, связи между темами [24]. При этом выполняются следующие гипотезы [22]:

- каждое слово связано с некоторой темой $t \in T$;
- текстовый корпус рассматривается как множество троек (d, w, t), выбранных случайно и независимо из дискретного распределения $p(d, w, t)$, заданного на конечном множестве $D \times W \times T$; тексты $d \in D$ и слова $w \in W$ являются наблюдаемыми переменными, тема $t \in T$ — скрытой;
- гипотеза условной независимости: $p(w|d, t) = p(w|t)$;
- гипотеза разреженности: каждый текст и каждое слово w связаны с небольшим количеством тем, поэтому значительная часть вероятностей $p(t|d)$ и $p(w|t)$ должна обращаться в нуль.

Для каждого текста определено распределение $\theta_d = p(t|d)$ его слов по темам, т.е. вероятность для каждой темы встретить ее в данном тексте и $\sum_t \theta_{td} = 1$. Тема представляется в виде распределения $\phi_t = p(w|t)$ слов из фиксированного словаря, т.е. каждое слово входит в тему с некоторой вероятностью и $\sum_w \phi_{wt} = 1$.

Вероятностные модели являются генеративными (порождающими) [11], то есть их можно использовать для генерации текстов. Описание модели начинается со способа генерации текстов. Основной же целью тематического моделирования является не генерация, а извлечение тем из имеющегося корпуса. Это задача, обратная генерации: восстановить распределения соответствующие исходные данным.

2.4.1. Вероятностный латентно-семантический анализ

Вероятностный латентнтно-семантический анализ (индексирование) (PLSA, Probabilistic Latent Semantic Analysis) был предложен Томасом Хоффманом в работе [3]. В основе метода лежит аспектная модель (aspect model), которая связывает скрытые (латентные) переменные тем $t \in T = \{t_1, ..., t_n\}$ с каждой наблюдаемой переменной словом или темой. Задача состоит в выявление латентных переменных. Таким образом, каждый документ может относится к некоторым темам с некоторой вероятностью, что является отличительной особенностью этой модели по сравнению с походами, не основанными на вероятностном моделировании.

Фиксировав количество скрытых переменных, с помощью метода PLSA можно оценить следующие величины:

- $P(d_i)$ - вероятность того, что наблюдаемое слово будет находиться в случайно выбранном из корпуса тексте d_i;

- $P(t_k|d_i)$ - вероятность того, что наиболее тесно связанный с данным фактором t_k текст - это d_i;
- $P(w_j|t_k)$ - вероятность того, что для данного фактора t_k, наиболее тесно связанное с ним слово - это w_j.

Здесь $d \in D = \{d_1, ..., d_I\}$ — множество всех текстов, $w \in W = \{w_1, ..., w_n\}$ — множество всех различных слов, встретившихся в текстах, $t \in T = \{t_1, ..., t_K\}$ — множество латентных переменных.

Таким образом, можно определить следующую вероятностную модель порождения текстов.

1. Случайно выбрать текст d_i с вероятностью $P(d_i)$.
2. Случайно выбрать тему t_k с вероятностью $P(t_k|d_i)$.
3. Случайно выбрать слово w_j с вероятностью $P(w_j|t_k)$.

В итоге получаем пару (d_i, w_j),в то время как t_k «забывается». На основе вышеизложенного совместная вероятностная модель определяется следующим образом:

$$P(d_i, w_j) = P(d_i)P(w_j|d_i),\ P(w_j, d_i) = \sum_{k=1}^{K} P(w_j|t_k)P(t_k|d_i) \tag{1}$$

.

Также стоит отметить, что вероятностная модель появления пары «текст-слово» (d,w) может записана тремя эквивалентными способами:

$$P(d, w) = \sum_{t \in T} P(t)P(w|t)P(d|t) = \sum_{t \in T} P(d)P(t|d)P(w|t) \tag{2}$$

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

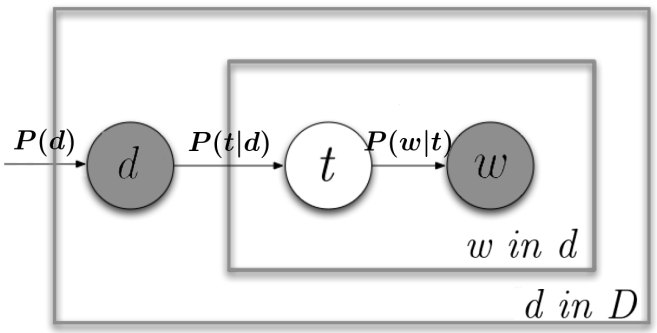

Рис. 1. Графическое представление модели PLSA

Графическое представление модели представлено на рисунке 2. Вершины графа соответствуют случайным переменным, а рёбра — непосредственным вероятностным взаимосвязям между случайными величинами. Вершины графа, соответствующие наблюдаемым величинам, значения которых известны закрашены. Вершины графа, соответствующие скрытым(латентным) величинам — их нужно найти — не закрашены. Направленное ребро из первой вершины во вторую обозначает условную зависимость второй величины от первой. Прямоугольник, включающий в себя некоторый подграф с указанным в правом нижнем углу числом N, обозначает совокупность N экземпляров данного подграфа.

По сравнению с LSA, PLSA имеет прочную основу в области статистики. В связи с этим он лучше подходит для практических применений [22]. Среди недостатков метода можно отметить, тот факт что число параметров растёт линейно по числу текстов в корпусе, что может приводить к переобучению модели [22]. Также при добавлении нового текста d в корпус распределение $p(t|d)$ невозможно вычислить по тем же формулам, что и для остальных текстов, не перестраивая всю модель заново [22].

2.4.2. Латентное размещение Дирихле

Латентное размещение Дирихле (LDA, Latent Dirichlet Allocation) — это порождающая модель, объясняющая результаты наблюдений с помощью неявных групп, что позволяет получить объяснение, почему некоторые части данных схожи. Например, если наблюдениями являются слова, собранные в тексты, утверждается, что каждый текст представляет собой смесь небольшого количества тем и что появление каждого слова связано с одной из тем документа. LDA впервые был представлен в качестве графической модели для обнаружения тем Дэвидом Блеем, Эндрю Нг и Майклом Джорданом в 2002 году [12].

В модели LDA каждый текст генерируется независимо, по следующей схеме [21]:

1. Случайно выбрать для текста его распределение по темам θ_d
2. Для каждого слова в текста:
   (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.

В рассматриваемом наборе текстов D каждый текст состоит из n_d слов. Наблюдаемыми переменными являются слова в текста - w_{dn}. Все остальные переменные — скрытые. Для каждого текста d переменная θ_d представляет собой распределение тем в данном тексте. В классической модели LDA количество тем фиксированно и изначально задается параметром T.

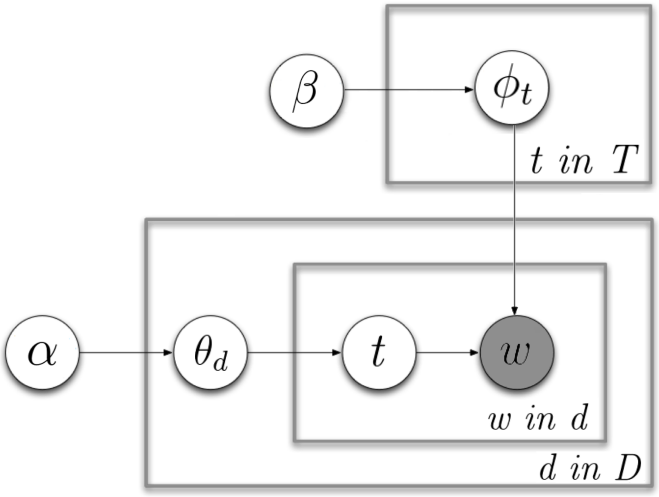

Рис. 2. Графическое представление модели LDA.

В модели LDA предполагается, что параметры θ_d и ϕ_w распределены следующим образом: $\theta \sim Dir(\alpha)$, $\phi \sim Dir(\beta)$, где α и β – задаваемые вектора-параметры (т.н. гиперпараметры) распределения Дирихле [21].

Основным недостатком распределения Дирихле является отсутствие убедительных лингвистических обоснований. С точки зрения текстов предположение о распределении Дирихле не является обоснованным.

Также существует множество других тематических моделей, основанных на вышеперечисленных. Выбор модели определяется в зависимости от условий поставленной задачи. Подробно обзор представлен в работе [24].

2.5. Качество тематической модели

Существует несколько способов оценки качества тематических моделей, которые представлены ниже.

2.5.1. Экспертная оценка

Самый тривиальный способ – это экспертная или ручная оценка. Эксперты решают, насколько темы, полученные с помощью конкретной тематической модели, соответствуют действительности, и, насколько подходит построенная модель для решения поставленных задач. Но данный способ имеет существенный недостаток – если объем данных слишком велик, то невозможно оценивать качество модели вручную.

2.5.2. Использование в работе приложений

Другой способ – попробовать 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 [21].

2.5.3. Перплексия

Перплексия (perplexity) – критерий, используемый для оценки моделей языка в компьютерной лингвистике [22]. Это мера соответствия модели $p(w|d)$ терминам w, наблюдаемым в документе $d \in D$, определяемая через логарифм правдоподобия [22]:

$$Perplexity = exp(-\frac{1}{n} \sum_{d \in D} \sum_{w \in d} n_{dw} ln(p(w|d))). \tag{3}$$

Чем меньше эта величина, тем лучше модель предсказывает появление слов w в документах d корпуса D [22].

2.6. Оценка параметров модели PLSA

В данном разделе подробнее рассмотрен метод построения тематической модели с использованием PLSA.

2.6.1. Метод максимума правдоподобия

Для идентификации параметров тематической модели по коллекции документов применяется принцип максимума правдоподобия (Maximum Likelihood Estimation), который приводит к задаче максимизации функционала. Функции $P(t)$, $P(d|t)$ и $P(w|t)$ определяются путем максимизации функции правдоподобия:

$$L = \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) logP(d_i, w_j) = \tag{4}$$

$$= \sum_{i=1}^{N} n(d_i) \left[logP(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} log \sum_{k=1}^{K} P(w_i|t_k) P(t_k|d_i) \right]$$

2.6.2. Формирование начальных приближений

Существуют различные способы для формирования начальных приближений распределения слов по темам - ϕ_{wt} и тем по документам - θ_{td}. Они подробно рассмотрены в работе [22]. Выбор начальных приближений может существенно влиять на скорость сходимости EM-алгоритма (см. пункт 2.6.3 EM-алгоритм).

Самая распространенная рекомендация - задание приближений нормированными случайными векторами из равномерного распределения. Также можно использовать частотные оценки и частичное обучение, если заранее известны некоторые связи слов и/или текстов с темами [22].

2.6.3. EM-алгоритм

Задача максимизации правдоподобия не имеет простого аналитического решения и решается численно[22]. Для определения оптимальных параметров модели используется стандартная процедура оценки максимального правдоподобия — EM-алгоритм (Dempster, Laird, Rubin, 1977). Это итерационный процесс, состоящий из двух шагов: E (Expectation) и M (Maximization). На E-шаге с помощью формулы Байеса вычисляются условные вероятности $P(t_k|d_i, w_j)$ для всех тем для каждого термина в каждом документе:

$$P(t_k|d_i, w_j) = \frac{P(w_j|t_k)P(t_k|d_i)}{\sum_{l=1}^{K} P(w_j|t_l)P(t_l|d_i)}. \tag{5}$$

На M-шаге по условным вероятностям вычисляются новые приближения:

$$P(w_j|t_k) = \frac{\sum_{i=1}^{N} n(d_i,w_j)P(t_k|d_i,w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i,w_m)P(t_k|d_i,w_m)}, \tag{6}$$

$$P(t_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i,w_j)P(t_k|d_i,w_j)}{n(d_i)}. \tag{7}$$

Подробный вывод формул, используемых в EM-алгоритме, представлен в работе [3].

3. Решение

В данном разделе описаны детали подготовки данных, реализации алгоритма и оценки качества работы алгоритма. Для решения этих задач было реализовано Java-приложение. Также были использованы следующие библиотеки.

- Tika [2] — кроссплатформенный набор инструментов, написанный на Java для предварительной обработки и анализа текстовой информации, например, выделения метаданных, извлечения контента из разнообразных форматов файлов, автоматического определения языка текста и т.д. В работе использовалась библиотека для определения языка текстов (см. пункт 3.2 Определение языка).
- Snowball [9] — реализация алгоритма стемминга, разработанного Мартином Портером в 1979 году. На данный момент проект «Snowball» включает в себя стеммеры для распространённых индоевропейских языков, в том числе для русского.

3.1. Исходные данные

В качестве источника данных был использован набор данных [8], состоящий из постов, опубликованных в сообществах социальной сеть «Одноклассники». Он включает в себя 775927 постов на 5 языках: русский, английский, армянский, грузинский, азербайджанский. Набор данных хранился в формате .csv файлов. Пример представлен на рисунке 4.

Эти файлы имели следующие поля, разделенные пробелами:

- group_id — анонимизированный идентификатор группы, в которой размещен пост;
- post_id — анонимизированный идентификатор поста;
- timestamp — время публикации поста, представляющее собой количество миллисекунд, прошедшее с полночи 1-го января 1970 года (UTC);
- content — содержание поста

group_id post_id timestamp content

```
11      600     1361991683016   выпускники Александровской сш 1983 года выпуска в этом году и
сполняется 30 лет (страшно подумать) как мы закончили школу. Если есть предложения по встрече
 - пишите.
11      602     1318537259683   Всех земляков с Покровой!Чтоб солнце ярко светило,вино лилось
 рекой,песни пелись!
```

Рис. 3. Примеры исходных данных: посты из сообществ (рисунок взят из [8]*).*

3.2. Определение языка

В тестовом наборе данных присутствуют тексты на различных языках. Для определения языка текста была использована библиотека Apache Tika [2]. Алгоритм метода заключается в нахождении частот N-грамм для всех тренировочных документов, для которых известен язык, а также для каждого документа, язык которого необходимо определить. После этого среди всех тестовых документов ищется тот, для которого расстояние от его N-граммной статистики до статистики тестируемого документа минимально. После этого языком тестируемого документа считается язык найденного тренировочного документа.

В текущей версии Tika поддерживается 27 языков. Для русского, английского были использованы готовые N-gram профайлы. Для остальных языков N-gram профайлы были построены. По умолчанию в задачах определения языка используется значение N=3. Для выделения N-gram требовался предварительно обработанный (без форматирования) связный текст на исходном языке. Далее использовались стандартные методы класса org.apache.tika.language.LanguageProfilerBuilder.

3.3. Нормализация постов и словаря

Изначально тематические модели разрабатывались для применения к текстам из книг и статей. Посты из социальных сетей отличаются от таких текстов:

- содержат много «лишнего»: авторское форматирование текста, ошибки, клоны, реклама и проч.;
- на нескольких языках.

После определения языка обработка данных состоит из нескольких этапов:

1. *Удаление идентификаторов и временных отметок (timestamps).* Каждый пост содержит: идентификатор, идентификатор группы, в которой он размещен, timestamp и текст. Вся информация, кроме текста и идентификатора поста удалялась.
2. *Приведение к нижнему регистру.*
3. *Выделение слов.*
4. *Удаление эмотиконов* [14] («:-)», « :-P », «>:-D» и т. д.).
5. *Удаление гиперссылок.*

6. *Удаление стоп-слов.* Общие стоп-слова — слова, не несущие какой-либо самостоятельной смысловой нагрузки и присутствующие во многих текстах корпуса. К стоп-словам можно отнести предлоги, суффиксы, причастия, междометия, цифры, частицы и т. п. Общие стоп-слова есть во многих постах. Их исключение из последующего рассмотрения не влияет на длину словаря, но может приводить к заметному сокращению длины некоторых текстов [22]. Для нормализации работе были использованы стандартные списки стоп-слов из библиотеки Apache Solr [1].

7. *Удаление слов, длина которых меньше трех символов.* Предполагалось, что слова длиной меньше трех символов не несут смысловой нагрузки, и их можно удалить.

8. *Стемминг* — процесс нахождения основы слова (неизменяемой части, которая необязательно совпадает с морфологическим корнем) для заданного исходного слова. Для нормализации текста был выбран стеммер из библиотеки Snowball [9], так как имеются готовые реализации для большого количества языков. В работе были использованы стеммеры для русского и армянского языков. Данный алгоритм не использует баз основ слов, а лишь, применяя последовательно ряд правил, отсекает окончания и суффиксы, основываясь на особенностях языка, в связи с чем работает быстро, но не всегда безошибочно.

9. *Получение обработанного текста из нормализованных слов.*

В таблице 1 представлены примеры нормализации постов. Также на диаграмме 1 представлена UML-диаграмма классов, соответствующая алгоритму обработки данных.

Исходный текст поста	Обработанный текст поста
1 63 1212913891440 Давайте собирём всех Наташек! Приглашайте их сюда!! Вместе мы сила!!!	собир наташ приглаша вмест сил
2818 377665 1369122156909 только в сибири такая красота :) Images[499534147409]	сибир красот
2920 386634 1355071331719 УРА! ПОЛУЧИЛА ЕЩЁ ОДНУ ЧАСТЬ ЗАКАЗ ОКТЯБРЬ 2, НАША ДОЛГОЖДАННАЯ ПОСЫЛОЧКА СЕГОДНЯ У МЕНЯ И УЖЕ РАЗОБРАНА! ТАК ЧТО ЗАБИРАЕМ СВОИ ЗАКАЗЫ,ХРАНИТЬ ЭТО ВСЁ НЕГДЕ! :)	ур получ част заказ октябр долгожда посылочк разобра забира заказ

Таблица 1. Пример предварительной обработки текстовых данных

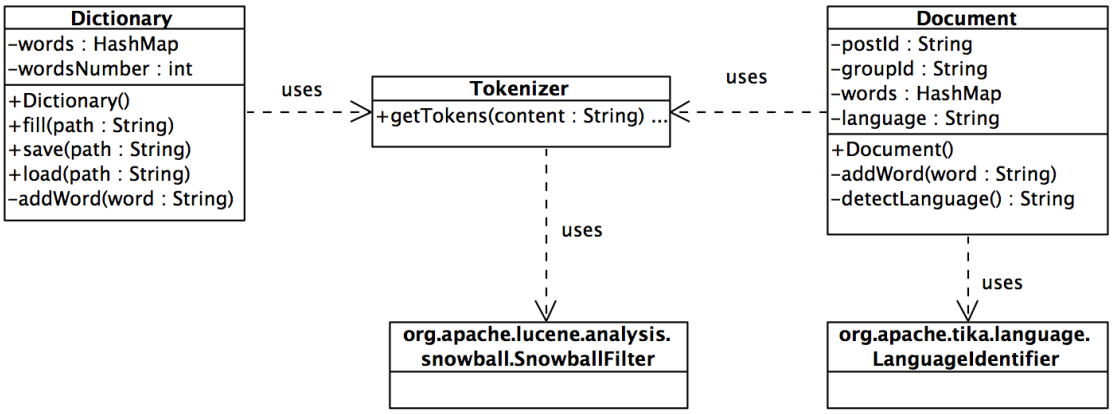

Диаграмма 1. Реализация алгоритма обработки данных

3.4. Реализация EM-алгоритма

Начальная инициализация распределений ϕ_{wt} (слов по темам) и θ_{td} (тем по документам) была задана случайными нормированными векторами из равномерного распределения.

Так как набор данных был довольно большим, то тексты были поделены на пакеты D_1, D_2, D_3,... Размер одного пакета составлял 40000 текстов. Очередность рассмотрения пакетов выбиралась случайно. Каждый пакет рассматривался по 4 раза. Для достижения сходимости выполнялось 60 итераций.

Изначально был реализован просто алгоритм PLSA с использованием формул (см. формулы 5, 6, 7), а затем были добавлены шумовая и фоновые компоненты [22] с целью повышения качества интерпретируемости тем:

$$P(w|d) = \frac{\sum_{t=1}^{T} \phi_{wt}\theta_{td} + \gamma\pi_{dw} + \epsilon\pi_{dw}}{1+\epsilon+\gamma}. \tag{8}$$

В формуле были использованы следующие обозначения:

- $\pi_{dw} = p_{noise}(w|d)$ — неизвестное распределение слов в тексте (шум — это слова, относящиеся к темам, мало представленным в коллекции текстов);
- $\pi_d = p_{background}(w)$ — неизвестное распределение слов в коллекции текстов (фон — слова, имеющие значимые вероятности, но не определяющие темы);
- γ и ϵ — неотрицательные параметры, задающие вероятности шумовой и фоновой компонент, $\gamma = 0.1$, $\epsilon = 0.02$.

Таким образом, появления слова в тексте распределялись не только между темами, но между шумом и фоном. Смысл робастной модели состоит в том, что в случаях,

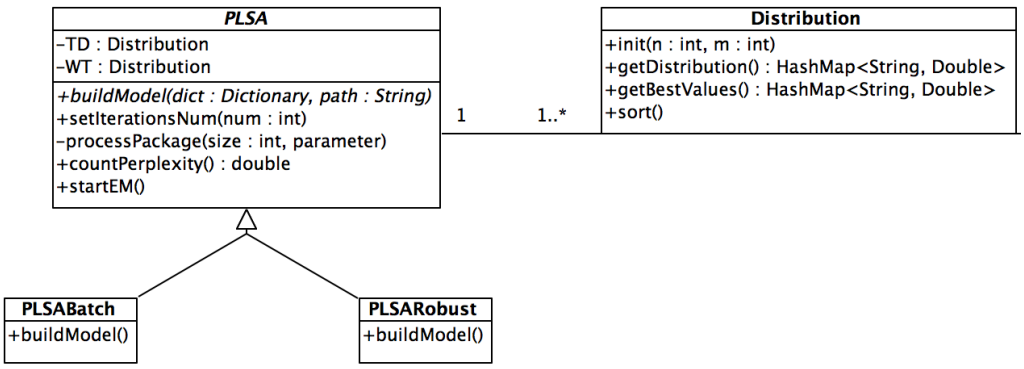

Диаграмма 2. Реализация алгоритма PLSA и PLSARobust

когда тематическая компонента плохо объясняет частоту появления слова, она может быть объяснена с помощью шумовой или фоновой компонент.

Формулы для E-шага:

$$P(t_k|d_i, w_j) = \frac{P(w_j|t_k)P(t_k|d_i)}{\sum_{l=1}^{|T|} P(w_j|t_l)P(t_l|d_i)+\gamma P_{noise}(w_j|d_i)+\epsilon P_{background}(w_j)}. \tag{9}$$

$$P_{noise}(w_j|d_i) = \frac{\gamma P_{noise}(w_j|d_i)}{\sum_{l=1}^{|T|} P(w_j|t_l)P(t_l|d_i)+\gamma P_{noise}(w_j|d_i)+\epsilon P_{background}(w_j)}. \tag{10}$$

$$P_{background}(w_j) = \frac{\epsilon P_{background}(w_j)}{\sum_{l=1}^{|T|} P(w_j|t_l)P(t_l|d_i)+\gamma P_{noise}(w_j|d_i)+\epsilon P_{background}(w_j)}. \tag{11}$$

На M-шаге значения параметров $P(w_j|t_k)$ и $P(t_k|d_i)$ вычисляются по формулам для PLSA (см. формулы 5, 6, 7), а значения параметров для шумовой и фоновой компонент по формулам [22]:

$$P_{noise}(w_j|d_i) = \frac{n(d_i,w_j)P_{noise}(w_j|d_i)}{\sum_{m=1}^{M} P_{noise}(w_m|d_i)}. \tag{12}$$

$$P_{background}(w_j) = \frac{\sum_{i=1}^{|D|} n(d_i,w_j)P_{background}(w_j)}{\sum_{m=1}^{|W|}\sum_{i=1}^{|D|} n(d_i,w_m)P_{background}(w_m)}. \tag{13}$$

4. Эксперименты

В данном разделе дана сравнительная оценка качества построенной тематической модели с использованием рассмотренных в разделе 2.6 методов.

4.1. Перплексия

Для выбора параметра T, отвечающего за количество тем, использовалась перплексия (см. пункт 2.5.3 Перплексия). Для того, чтобы выбрать оптимальное значение параметра T были взяты значения: 50, 100, 150, 200, 250. Далее строилась тематическая модель с заданным значением параметра T и вычислялось значение перплексии. Наименьшее значение данной величины достигалось при значении параметра T=270: рисунок 4. График PLSA(1) соответствует обычному PLSA, график PLSA(2) — робастному PLSA, график LDA — LDA (была использована готовая реализация [5]) со значением гиперпараметров $\alpha = 0,4$ и $\beta = 0,02$.

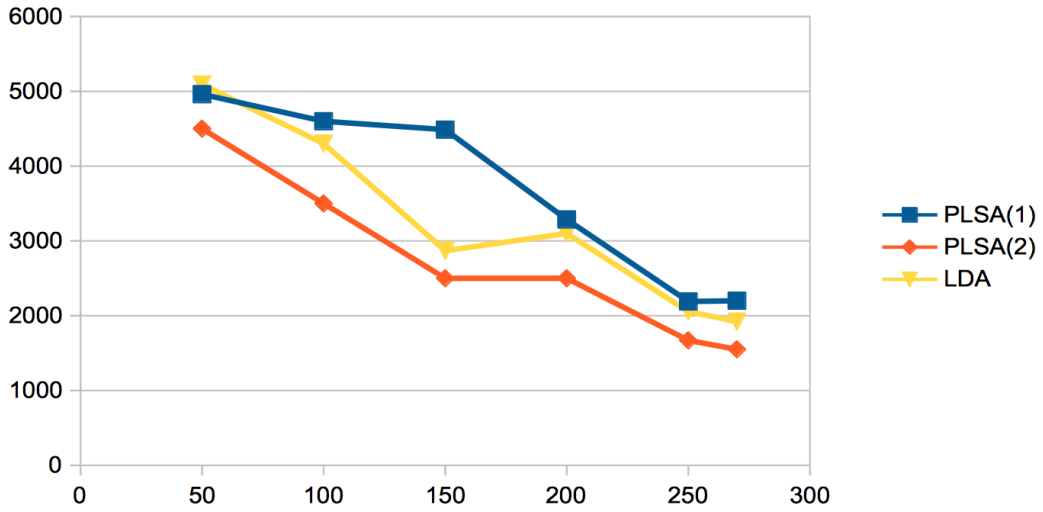

Рисунок 4. График перплексии

4.2. Темы сообществ

При определении тем сообществ одно сообщество рассматривалось как текст (без разделения на посты), а все сообщества как текстовый корпус. Всего было рассмотрено более 4000 сообществ. Если получившийся текст соответствующий одному сообществу получался слишком большим, то бралась лишь его часть, ограниченная заданным взможным максимальным размером текста. Пример полученных тем сообщества приведен ниже на рисунке 5. Тема по умолчанию состояла из 35 слов: после того, как распределение слов по темам было вычислено, для каждой темы значения соответствующие словам были отсортированы и выбраны максимальные 35.

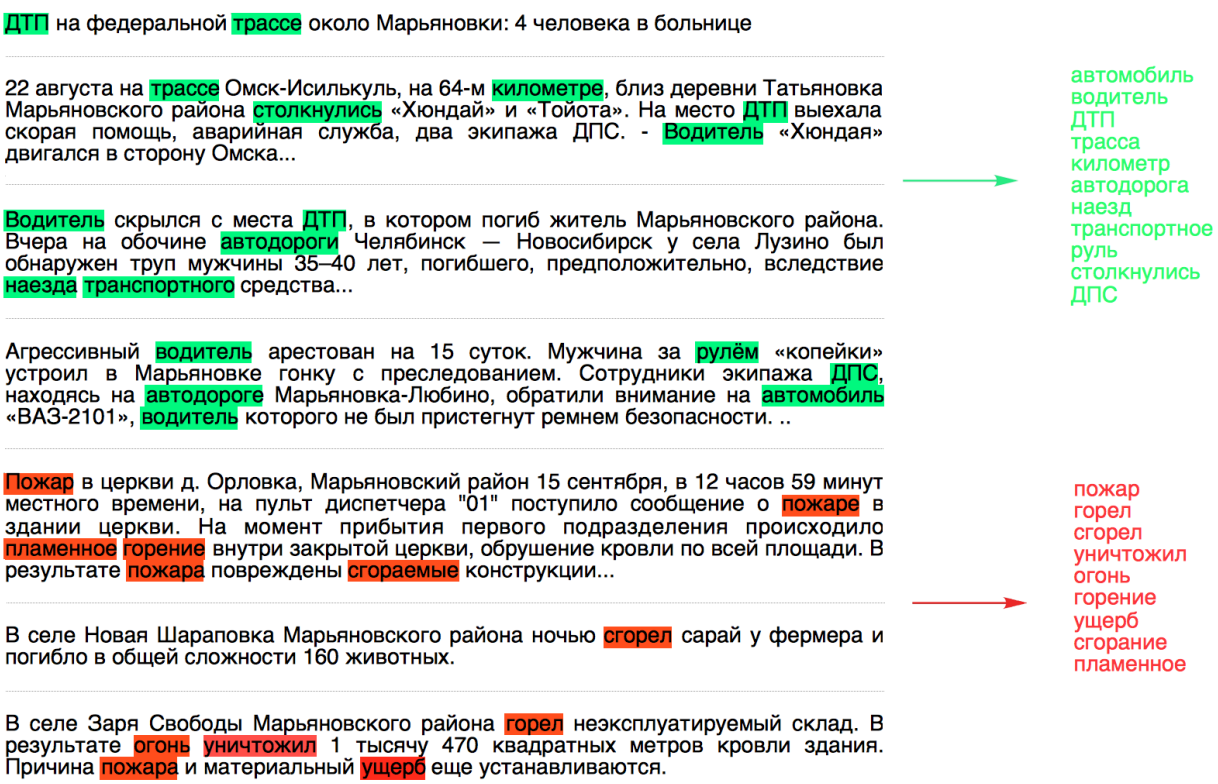

Рисунок 5. Пример тем сообщества

4.3. Прогноз количества «Классов!»

Оценка качества построенной тематической модели проводилась с помощью использования ее для решения задачи предсказания количества «Классов!» [8]. «Класс!» (лайк) — понятие в социальных сервисах, распространившееся вместе с социальными сетями, означающее условное выражение одобрения материалу, пользователю, фотографии, выражающиеся нажатием одной кнопки (см. рисунок 6).

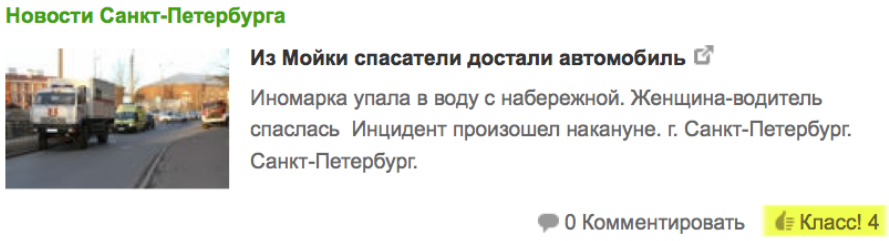

Рис. 6. Кнопка «Класс!»

В качестве входных данных использовались посты, опубликованные в группах «Одноклассников», а также «Классы!», набранные этими постами. Набор данных состоял их тренировочного и тестового множеств. Тренировочное множество включало в себя 519871 пост, а тестовое из — 256056 постов. Задача состояла в том, чтобы предсказать количество классов, которое наберут посты из тестового множества.

Данные о постах хранились в двух файлах идентичного формата: train_content.csv и test_content.csv. Эти файлы имели формат, представленный на рисунке 4.

Данные о «Классах!» тренировочного множества хранились в файле train_likes.csv. Эти файлы имели следующие поля, разделенные пробелами:

- user_id — анонимизированный идентификатор пользователя, который поставил «Класс!»
- post_id — анонимизированный идентификатор поста
- timestamp — время «Класс!», представляющее собой количество миллисекунд, прошедшее с полночи 1-го января 1970 года (UTC).

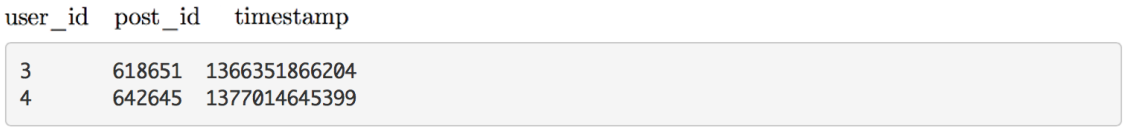

Рис. 7. Примеры исходных данных: «Классы!» (рисунок взят из [8]*).*

4.4. Оценка прогноза

Оценка прогноза производилась с использованием метрики R^2, умноженной на 1000 для удобства отображения:

$$score(f, p) = (1 - \frac{Var(f|p)}{Var(f)}) * 1000. \quad (14)$$

В данной формуле были использованы следующие обозначения:

- f — фактическое значение количества «Классов!»
- p — прогноз количества «Классов!»
- $Var(x)$ — выборочная дисперсия [18] величины x

Таким образом, максимальный балл, который может набрать прогноз, составлял 1000.

Для прогноза количества «Классов!» использовался метод ближайших соседей или knn-алгоритм (kNN, k-nearest neighbor algorithm) из библиотеки Weka [10]. [20]. Каждый текст d, как правило, был связан с небольшим числом тем, поэтому значительная часть вероятностей $P(t|d)$ вырождалась в ноль. В качестве признаков (features) использовались: временная отметка поста, идентификатор группы и темы поста. В последнем случае вместо тем использовались другие признаки, такие как: количество

слов, количество символов, количество знаков. Результаты представлены в таблице 2. Наибольшее значение метрики R^2 достигалось при использовании тем, полученных с помощью робастного вероятностного латентно-семантического анализа.

features	R^2
PLSA(1)	233,649
PLSA(2)	259,630
LDA (David Blei)	220,348
количество слов, количество символов, количество знаков, препинания, среднее количество символов в слове	153,334

Таблица 2. Прогнозы количества «Классов!»

Заключение

В рамках данной работы рассмотрены методы тематического моделирования на основе вероятностного латентно-семантического анализа. В частности были выполнены следующие задачи:

- Проанализированы основные методы тематического моделирования.
- Реализованы алгоритмы построения тематической модели для больших объемов данных с помощью вероятностного латентно-семантического анлиза и робастного вероятностного латентно-семантического анализа (PLSA, Probabilistic Latent Semantic Indexing).
- Проведен сравнительный анализ качества тематических моделей.

Было показано, что информация о текстовых данных (темы), извлеченная с помощью методов вероятностного тематического моделирования, может применяться с целью улучшения качества работы других приложений. Исходя из специфики текстового корпуса нельзя не отметить важность предварительной обработки текстовых данных и подбора параметров модели. От этих шагов непосредственно зависит качество результатов. Среди рассмотренных методов вероятностного моделирования наилучший результат показал робастный вероятностный латентно-семантический анализ.

Список литературы

[1] Apache Solr. — URL: http://lucene.apache.org/solr/.

[2] Apache Tika. — URL: http://tika.apache.org/.

[3] Hofmann Thomas. Probabilistic latent semantic indexing // In Proc. of the SIGIR'99. — P. 50–57.

[4] James Allan Jaime Carbonell George Doddington Jonathan Yamron Yiming Yang. Topic Detection and Tracking Pilot Study. Final Report // Proceedings of the Broadcast News Transcription and Understanding Workshop (Supported by DARPA). — 1998.

[5] LDA Implemetations. — URL: http://www.cs.princeton.edu/~blei/topicmodeling.html.

[6] Landauer T. K. Foltz P. Laham D. chapter An Introduction to Latent Semantic Analysis, DiscourseProcesses // DiscourseProcesses, volume 25, chapter "An Introduction to Latent Semantic Analysis". — 1998. — P. 259–284.

[7] Muhammad Ali Daud Juanzi Li Lizhu Zhou Faqir. Knowledge discovery through directed probabilistic topic models: a survey.- перевод на русский К.В. Воронцов, А.В. Темлянцев и др. // In Proceedings of Frontiers of Computer Science in China. — 2010. — P. 280–301.

[8] SNA Hackaton. — URL: http://sh2014.org/index/.

[9] Snowball. — URL: http://snowball.tartarus.org/.

[10] Weka Project. — URL: http://www.cs.waikato.ac.nz/ml/weka/.

[11] Wikipedia. Generative Model // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Generative_model.

[12] Wikipedia. Latent Dirichlet Allocation // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation.

[13] Wikipedia. Latent Semantic Analysis // Википедия, свободная энциклопедия. — URL: http://ru.wikipedia.org/wiki/Latent_semantic_analysis.

[14] Wikipedia. List of emoticons // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/List_of_emoticons.

[15] Wikipedia. Non-negative matrix factorization // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Singular_Value_Decomposition.

[16] Wikipedia. Singular value decomposition // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Non-negative_matrix_factorization.

[17] Wikipedia. Tf-idf // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Tf-idf.

[18] Wikipedia. Variance // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Variance.

[19] Wikipedia. Vector Space Model // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/Vector_space_model.

[20] Wikipedia. kNN // Википедия, свободная энциклопедия. — URL: http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[21] А.Г. Гомзин. Определение тематической направленности текстового содержимого микроблогов // Дипломная работа, ВМК МГУ. — 2013.

[22] Воронцов К.В. Лекции по вероятностными тематическим моделям. — 2013. — URL: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf.

[23] Д.В. Машечкин И.В. Петровский М.И. Царёв. Методы вычисления релевантности фрагментов текста на основе тематических моделей в задаче автоматического аннотирования // Вычислительные методы и программирование, Т.14. — 2013. — P. 91–102.

[24] Коршунов А. Гомзин А. Тематическое моделирование текстов на естественном языке // Труды Института Системного Программирования Ран, Выпуск том 23. — 2012. — P. 216–243.

[25] М.В. Губин. Модели и методы представления текстового документа в системах информационного поиска. — 2005. — P. 20–22.