Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

На правах рукописи

ЧЕРНЯК
ЕКАТЕРИНА ЛЕОНИДОВНА

РАЗРАБОТКА ВЫЧИСЛИТЕЛЬНЫХ МЕТОДОВ АНАЛИЗА ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ АННОТИРОВАННЫХ СУФФИКСНЫХ ДЕРЕВЬЕВ

Специальность 05.13.18 —
«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук
Б. Г. Миркин

Москва – 2016

Оглавление

Стр.

Введение

Актуальность темы. Проникновение вычислительной техники во все сферы производственной, социальной и политической систем привело к необходимости разработки методов автоматического семантического анализа текстовых документов, размещенных в индивидуальных компьютерах и в интернете. Часть связанных с этим задач хорошо осознана и получает решение в научной и технической литературе. Это, прежде всего, задачи поиска и извлечения информации, категоризации текстов, извлечения ключевых словосочетаний, извлечение фактов и др. Большинство методов решения таких задач основано на предварительной «ручной» разметке текстов (выделение ключевых слов и других данных для обучения). Однако, в связи с наступлением эры глобализации, существует явная потребность в разработке методов, не требующих предварительной разметки текстов. Кроме того, создание корректных и эффективных морфологических и синтаксических парсеров – это трудоемкая задача, решенная не для всех языков. Это делает актуальной задачу разработки методов анализа текстов, не требующих их предварительной разметки.

В большинстве практических задач анализа коллекций текстовых документов, включая задачу информационного поиска, предполагается вычисление оценок релевантности «строка–текст». В качестве текстов, разумеется, выступают те или иные документы, а в качестве строк – ключевые слова и словосочетания, заданные извне или извлеченные из текстовых документов по определённым принципам, или произвольные элементы текста, состоящие из фиксированного количества букв или слов. Мера релевантности должна удовлетворять следующим естественным свойствам:

1. Интуитивная простота (понятные единицы и границы измерения);
2. Независимость от длины текста;
3. Независимость от лексической вариативности текста;
4. Возможность эффективной вычислительной реализации.

Большинство известных мер релевантности основаны на использовании в качестве элементарной единицы текста слова (или его нормальной формы – леммы, или его (псевдо)основы – стема). К этому классу моделей релевантности относятся векторная модель релевантности [1], вероятностная модель релевантности [2] языковая модель релевантности на словах или символьных n-граммах

[3], модель суффиксного дерева [4]. Эти модели предполагают представление текста в виде неупорядоченного набора слов – «мешка» слов, а также предполагают учет морфологии и синтаксиса языка для идентификации и унификации слов. Существенным недостатком этих моделей можно считать невозможность учесть нечеткие (то есть, с различием на несколько символов) совпадения между строками и текстами. До некоторой степени этот недостаток помогают преодолеть языковая модель релевантности на символьных n–граммах [3] и модель суффиксного дерева [5]. Однако же, языковая модель релевантности на символьных n–граммах часто бывает неэффективной с вычислительной точки зрения, поскольку возникающая в ней проблема нулевых вероятностей зачастую решается с помощью вычислительно неэффективных алгоритмов сглаживания, а модель суффиксного дерева, предложенная в [5], по определению не удовлетворяет требованиям 3 и 4, сформулированным выше.

Для решения обозначенных выше задач – необходимости предобработки и нечеткости меры релевантности – и с учетом требований 1-4 необходима новая модель совокупности «строка – текст», а также структура данных, позволяющая вычислять нечеткие оценки релевантности.

В данном исследовании предлагается и верифицируется теоретико-множественная модель совокупности «строка – текст», а адекватной структурой данных для вычисления параметров оценки является аннотированное суффиксное дерево.

В теоретико-множественной модели совокупности «строка – текст» текст представляется в виде множества коротких строк, например, последовательных пар или троек слов, а строка S, состоящая из n символов, $S = s_1 s_2 \dots s_n$ – множеством всех подстрок $s_i \dots s_j$, где $i >= 1$, $j <= n$, $i <= j$. Для каждой пары строка – текст несложно найти все возможные общие подстроки, иначе говоря, совпадения. Максимальным совпадением назовем такое совпадение, при добавлении символа в начало или в конец которого, перестает быть совпадением. Допустим, существует совпадение строки с текстом $s_i \dots s_j$. Определим его вероятность, как условную частоту последнего символа s_j : $P(s_i \dots s_j) = P(s_j | s_i \dots s_{j-1})$ (УВС). Вероятностью максимального совпадения тогда является средняя сумма совпадений, в него входящих (СУВС), а полной релевантностью строки тексту – сумма вероятностей максимальных совпадений данному тексту (СУВСС). Для эффективной реализации вычисления оценок релевантности следует использовать аппарат аннотированного суффиксно-

го дерева – структуры данных, которая позволяет вычислять все частоты всех подстрок.

Объект исследования – вычислительные задачи анализа текстовых документов, написанных на естественном языке.

Предмет исследования – вычислительное моделирование текстов как строк символов и задачи их анализа, решаемые путем наложения разных строк друг на друга.

Цель данного диссертационного исследования – разработка оригинальных моделей, методов, алгоритмов и программных комплексов, предназначенных для решения некоторых задач анализа текстовых документов на естественном языке на уровне последовательностей символов.

К **задачам** исследования относятся:

1. Разработка модели представления коллекции текстовых документов строками и ассоциированной с ней функции релевантности;
2. Верификация разработанной модели на реальных задачах анализа коллекций текстовых документов:
    a) Рубрикация текстовых документов в соответствии с заданной системой рубрик;
    b) Пополнение таксономии с использованием внешней коллекции текстов;
    c) Фильтрация коллекции текстовых документов от обсценной лексики.
3. Реализация разработанных моделей и методов в виде комплекса программ.

К **методам**, использованным в исследовании, относятся:

1. Метод Укконена для построения аннотированного суффиксного дерева за линейное время;
2. Метод вычисления релевантности строки тексту с помощью наложения строки на аннотированное суффиксное дерево его представляющее;
3. Методы вычисления релевантности строки тексту, основанные на представлении текстов векторными пространствами и вероятностными моделями.

Научная новизна. В диссертации получен ряд новых научных результатов, которые **выносятся на защиту**:

1. Разработана теоретико-множественная модель совокупности «строка-текст» с методом оценки релевантности строк тексту, основанном на аннотированных суффиксных деревьев. Предложен новый метод вычисления оценок релевантности строки тексту СУВСС, апробированный в работе;
2. Предложен метод рубрикации научных статей с использованием критерия релевантности СУВСС, более точного, чем популярные методы, традиционно используемые в международных публикациях;
3. Разработан метод использования справочных материалов интернета, с учетом наличия в них шумовой компоненты, для пополнения предметных таксономий. Методика апробирована в задачах пополнения таксономий чистой и прикладной математики с использованием русскоязычной Википедии;
4. Показана эффективность использование критерия релевантности СУВСС в классе задач поиска по однословному ключу, в котором полнота важнее, чем точность;
5. Разработаны комплексы программ, реализующие предложенную теоретико-множественную модель совокупности «строка – текст» с использованием критерия релевантности СУВСС, применительно к решению задач в пунктах 2, 3 и 4.

Теоретическая значимость работы заключается в разработке принципиально новых моделей и методов: теоретико-множественной модели совокупности «строка – текст», модели нормированного аннотированного суффиксного дерева с критерием релевантности СУВСС, а также метода построения таблиц релевантности «строка – текст» (РСТ) для применения в конкретных задачах.

Практическая ценность подтверждена экспериментами по сравнительной оценке использования мер релевантности для рубрикации научных статей, результатами расчетов по пополнению таксономий с использованием материалов интернета и результатами решения задач поиска, ориентированных на его полноту. Все разработанные методы реализованы в виде программных комплексов, предназначенных для решения исследовательских и прикладных задач. Разработанные методы и алгоритмы были успешно применены в реальных проектах компании ООО «ФОРС-Центр разработки» (метод фильтрации обсценной лексики использован для анализа и определения тональности текстов в социальных сетях в системе FORSMedia) и «ЕС-Лизинг» (метод рубрикации

использован для категоризации проектной документации) и проектах, выполнявшихся по грантам ВШЭ в 2010 – 2015 гг., а также в преподавательской деятельности Департамента анализа данных и искусственного интеллекта Факультета компьютерных наук НИУ ВШЭ.

Достоверность полученных результатов подтверждена строгостью использованных математических моделей и методов, экспериментами по сравнению результатов применения разработанных традиционных методов на конкретных задачах, а также алгоритмической эффективностью программных реализаций.

Апробация результатов работы. Основные результаты работы обсуждались и докладывались на следующих научных конференциях и семинарах:

- 1-ой, 2-ой всероссийских научных конференция “Анализ изображений, сетей и текстов” (АИСТ-2012, АИСТ-2013), Екатеринбург, Россия; темы докладов – “Автоматизация использования таксономий для аннотирования текстовых документов”, “Использование ресурсов интернета для построения таксономии”
- 1-ом семинаре по кластерам, деревьям и порядкам (COT-2013), Москва, Россия; тема доклада – “An AST method for scoring string-to-text similiarity in semantic text analysis”
- 8-ой международной конференции “Диалог” (Диалог-2013), Бекасово, Россия; тема доклада – “Computational refining of Russian-language taxonomy using Wikipedia”
- 3-ей международной научной конференции “Анализ изображений, сетей и текстов” (АИСТ-2014), Екатеринбург, Россия; тема доклада – “Conceptual maps: construction over a text collection and analysis”
- 2-ой международной конференции “Информационные технологии и количественный менеджмент” (ITQM-2014), Москва, Россия; тема доклада – “A method for refining a taxonomy by using annotated suffix trees and Wikipedia recourses”
- 3-ей всероссийской конференции “Искусственный интеллект и естественный язык” (AINL-2014), Москва, Россия; тема доклада – “Создание и визуализация газетного интернет-корпуса”
- 8-ой международной конференции “Веб-поиск и майнинг данных” (WSDM-2015), Шанхай, КНР тема доклада – “An approach to the problem of annotation of research publication”;

– 2-ом международном семинаре по майнингу данных и автоматической обработке текстов (DMNLP-2015) тема доклада – “Some thoughts on using annotated suffix trees for NLP tasks”

Публикация результатов. Основные результаты работы изложены в 13 научных статьях. 7 статей опубликованы в рецензируемых сборниках трудов международных и всероссийских конференций, 3 статьи опубликованы в журналах из списка ВАК.

Основные результаты работы

1. Экспериментально показана целесообразность использования теоретико-множественной модели совокупности «строка-текст» и нормированного аннотированного суффиксного дерева (АСД) в качестве численного метода оценки параметров модели и ассоциированной с ним меры релевантности для решения задач анализа коллекций текстовых документов;
2. В рамках теоретико-множественной модели совокупности «строка-текст» предложена и обоснована естественная мера релевантности СУВСС, вычисляемая на основе нормированного АСД;
3. Показана эффективность использования меры релевантности, основанной на АСД, в задаче рубрикации коллекций текстовых документов без учителя – использование данной примеры приводит к лучшему ранжированию;
4. Предложен и применен к двум таксономиям прикладной математики метод пополнения таксономии, использующий Википедию в качестве внешнего источника;
5. Показана эффективность использования меры релевантности, основанной на АСД, в задаче фильтрации обсценной лексики – использование данной меры приводит к лучшим показателям полноты и вычислительной сложности по времени;
6. Предложена адаптация алгоритма Укконена для построения АСД;
7. Разработаны программные комплексы для извлечения данных из Википедии, для построения АСД, вычисления оценок релевантности и построения таблиц релевантности «строка – текст».

Во введении раскрывается актуальность темы диссертации, формулируются проблемы и задачи исследования, предмет исследования, определяются цели работы, описываются методы исследования, излагаются основные научные

результаты, обосновывается теоретическая и практическая значимость работы, даётся общая характеристика исследования.

В первой главе приводится обзор четырех подходов к машинному представлению коллекций текстовых документов: векторная модель представления коллекций текстовых документов, языковая модель представления коллекций текстовых документов, представление коллекции текстовых документов на основе модели скрытых тем, представление коллекции текстовых документов на основе модели суффиксных деревьев. Рассматриваются задачи обработки и анализа коллекций текстовых документов, в которых применяются те или иные модели представления, возможные преимущества и ограничения. Приводятся основные определения, связанные с предварительной обработкой текстовых документов, моделями представления текстовых документов, различными задачами обработки и анализа коллекций текстовых документов.

Во второй главе рассматривается проблематика определения релевантности строки текстовому документу, принадлежащему некоторой коллекции. Утверждается, что построение функции релевантности тесно связано с выбранным формальным представлением коллекции текстовых документов. В связи с этим рассматриваются различные функции релевантности, порождаемые различными формальными представлениями коллекций текстовых документов. Вводится понятие нормированного аннотированного суффиксного дерева и связанной с ним естественно интерпретируемой функции релевантности СУВСС. Описывается метод построения таблиц релевантности «строка – текст» (РСТ), используемых в дальнейшем для анализа коллекций текстовых документов, а также оптимальные по памяти и времени алгоритмы построения нормированного аннотированного суффиксного дерева.

В третьей главе рассматривается задача рубрикации аннотаций научных публикаций. Проводится сравнение методов рубрикации аннотаций научных публикаций с использованием различных функций релевантности, в том числе, с использованием предложенной в диссертационном исследовании меры релевантности СУВСС.

В четвертой главе рассматривается задача пополнения научной таксономии. Формулируется задача пополнения научной таксономии и предлагается вычислительный метод для ее решения. Метод применен к таксономиям двух областей чистой и прикладной математики.

В пятой главе проводится аналогия между задачей поиска по однословному ключу и фильтрации обсценной лексики. Утверждается, что несмотря на то, что для решения этих задач могут быть использованы одинаковые методы, оптимизируются разные критерии качества, которые влияют на выбор конкретного метода. Описывается эксперимент по разработке фильтра на основе СУВСС и демонстрируется его эффективность с точки зрения оптимизируемого критерия – полноты, а так же с точки зрения временной сложности.

В шестой главе приводится описание программных комплексов, реализующих разработанные в исследовании модели и методы, а также решающие некоторые вспомогательные задачи сбора и обработки данных. Библиотека EAST реализует предложенный алгоритм построения нормированного аннотированного суффиксного дерева за линейное время и с линейными затратами по памяти, а также выполняет предварительную обработку текстов. Утилита WikiDP позволяет извлекать из Википедии данные различных типов, такие как дерево категорий с корнем в заданном узле и принадлежащие к этом дереву статьи.

Объем и структура работы. Диссертация состоит из введения, шести глав и заключения. Полный объём диссертации составляет 124 страницы, включая 15 рисунков и 32 таблицы. Список литературы содержит 105 наименований.

Автор диссертационного исследования благодарит научного руководителя – Бориса Григорьевича Миркина – за 7 лет плодотворного сотрудничества и все уроки и советы, существенно повлиявшие на формирование автора как исследователя, и поддержку во всех научных начинаниях, руководителя Департамента анализа данных и искусственного интеллекта ФКН НИУ ВШЭ и Международной лаборатории интеллектуальных систем и структурного анализа НИУ ВШЭ – Сергея Олеговича Кузнецова – за создание азартного исследовательского духа на департаменте и в лаборатории, своих коллег – Михаила Дубова и Дмитрия Ильвовского – за бесконечные часы плодотворных обсуждений и совместной работы, студентов ФКН НИУ ВШЭ – Максима Яковлева, Анну Шишкову, Георгия Котова – за участие в проектах, связанных с развитием тематики диссертационного исследования, своих друзей – Ольгу Чугунову, Дину Шагалову и Марию Смирнову - за поддержку на каждом этапе учебы в аспирантуры и подготовки диссертации, а Ольгу – и за разметку данных.

Глава 1. Способы представления текстов для машинной обработки

1.1 Введение

Формальное представление текста – это математическая структура, построенная по неструктурированному тексту [6; 7]. Формальным представлением текста может быть алгебраическая структура, теоретико-множественная или графовая структура, комбинация распределений вероятностей слов. Чаще всего говорят о формальном представлении большого числа – коллекции / корпуса – текстов, поскольку представление одного текста с помощью математической конструкции не представляет особого интереса. Напротив, представление каждого текста из коллекции с помощью одной и той же конструкции делает возможным использование математических методов для обработки, анализа, сравнения, определения сходства между текстами, классификации, кластеризации, генерации текстов и так далее. В этой главе будут рассмотрены четыре основных класса представлений текстов: векторная модель, языковая модель, модели скрытых тем и модели суффиксных деревьев. Исторически первая векторная модель представления текста имеет наибольшее количество применений, однако некоторые ее недостатки (например, не учитывается порядок слов) делает не возможным ее использование в тех задачах, в которых необходимо сгенерировать фрагмент текста или оценить вероятность его появления. В таком случае используются генеративные модели представления текста, такие как языковая модель и некоторые модели скрытых тем, основанные на скрытом размещении Дирихле. И векторная модель, и языковая модель, и модель скрытых тем основаны на общей идее: текст является набором так называемых термов – слов в исходном виде или их значимых фрагментов, например, основ. Отсюда следует общий недостатков всех перечисленных моделей: при обработке и анализе текстов учитывается только четкое совпадение между термами. Модель суффиксных деревьев – менее популярная в силу невысокой вычислительной эффективности – до определенной степени позволяет учитывать нечеткие совпадения, что делает возможным ее использование в задачах интерпретации текстов.

1.2 Векторная модель представления текстов

Векторная модель – это одна из наиболее популярных моделей представления текста [6]. В основе этой модели лежит так называемый мешок слов – принцип максимального упрощения структуры текста [8]. Согласно этому принципу, текст является множеством или мультимножеством входящих в него слов. Очевидно, что использование этого принципа ведет к потере порядка слов, а следовательно, и коротких, и длинных, в том числе, анафорических и кореферентных связей [7]. В векторной модели текст представляется вектором в пространстве слов (или каких-нибудь других элементов текста, так называемых, термов), причём каждому терму соответствует своя координата векторного пространства. В качестве значения вектора используется частота терма в тексте. Если в общем пространстве термов представляют два или более текстов – так называемую коллекцию текстов – часто используют $tf - idf$ кодировку значений вектора, равную количеству вхождений терма в данный текст, делённому на логарифм относительного количества текстов, содержащих это слово [1]:

$$tf - idf = tf_{w,d} \times \log \frac{|D|}{|d' \in D | w \in d'|}.$$

В этой формуле первый сомножитель $tf_{w,d}$ – это локальный вес, то есть, частота терма w в тексте d, а второй сомножитель idf – это глобальный вес, показывающий логарифм от величины, обратной доле текстов d', содержащих терм w среди общего числа текстов $|D|$. $tf - idf$ кодировка снижает вес часто встречающихся во всех текстах коллекции термов и повышает вес термов, характерных для данного текста. Иногда формулу $tf - idf$ весов меняют, сохраняя при этом общий смысл: первый множитель – локальный вес – отвечает за выбор частотных слов в данном тексте, второй множитель – глобальный вес – за отсеивание слов, одинаково частотных во всей коллекции. Таким образом, общая схема взвешивания устроена так: $w_{ij} = l_{ij} \times g_i$ [1]. Некоторые другие возможные локальные веса представлены в работах [9]:

- **Бинарный вес**: $l_{ij} = 0$, если терм i не встречается в тексте j, 1, в обратном случае
- **Частота**: $l_{ij} = tf_{ij}$
- **Логарифмический вес**: $l_{ij} = \log(tf_{ij} + 1)$
- **Скорректированный Гауссов вес**: $l_{ij} = \frac{tf_{ij}}{2 \max_i(tf_{ij})} + 0.5$

Некоторые глобальные веса:

- **Бинарный вес**: $g_i = 1$
- **Гауссов вес**: $g_i = \frac{1}{\sum_i tf_{ij}}^2$
- $gf - idf$ **вес**: $g_i = \frac{gf_i}{df_i}$, где gf_i – сколько раз i–тый терм встретился во всей коллекции, а df_i – число текстов, в которых встретился i–тый терм
- idf **вес**: $g_i = \log \frac{N}{1+df_i}$, где N – количество термов во всей коллекции (иначе – объем словаря)
- **Энтропия**: $g_i = 1 - \sum_j \frac{p_{ij} \log p_{ij}}{\log N}$, где $p_{ij} = \frac{tf_{ij}}{gf_i}$.

В статье [10] следующая схема взвешивания $w_{ij} = \log(1 + tf_{ij}) \times \log \frac{N+1}{n(t_i)+1}$ получила название $tf - icf$ (term frequency – inverse corpus frequency).

Основным достоинством векторной модели является ее простота и тот факт, что векторное представление текстов делает возможным использование линейно-алгебраических операций для определения сходства между текстами и ранжирования текстов по соответствию запросу [11]. Для этих целей используется косинусная мера релевантности, которая будет описана более подробно ниже. В общих чертах косинусная мера релевантности определяется как нормированное скалярное произведение [1]. Другим очевидным достоинством векторной модели является простота ее построения по заданному корпусу текстов [12]. Во многих современных библиотеках автоматической обработки текстов, таких как `gensim` [13] и `NLTK` [14] реализованы индексаторы коллекций текстов на основе векторной модели – функции, задающие как координаты векторного пространства (т.е. выделяющие термы), так и соответствующие каждому тексту.

Однако, за внешней простотой векторной модели кроются некоторые существенные недостатки. Прежде всего, главная предпосылка векторный модели, а именно понятие мешка слов, с статистической точки зрения означает гипотезу о независимости слов, что в корне не верно с точки зрения лингвистики и анализа естественного языка [12]. Использование нормированного скалярного произведения в качестве меры сходства приводит к тому, что более длинные тексты всегда имеют низкую степень сходства с остальными текстами из-за нормировки длинной текста [15]. Главным же недостатком векторной модели является отсутствие учета синонимии между словами [15]: в векторной модели словам «Голландия» и «Нидерланды» будут соответствовать разные координаты, поэтому синонимичность этих слов никак не будет отражена.

Тем не менее, векторная модель широко используется во многих задачах автоматической обработки текстов: категоризации, классификации и кластеризации текстов, а также в задаче поиска по запросу, исторически первой задаче, для решения которой была использована векторная модель [1]. Задача категоризации текстов заключается в распределении текстов по заранее заданному множеству категорий. Как правило, задача категоризации текстов решается с помощью методов машинного обучения. Исчерпывающий обзор подходов к решению этой задачи приведен в [16]. В этом обзоре показано, в том числе, что представление коллекции текстов в виде общей матрицы терм–текст делает возможным использование любого метода машинного обучения. В статьях [17] и [18]. впервые возникает задача классификации текстов по тональности, которая заключается в том, чтобы определить имеет текст положительную или отрицательную окраску. Чаще всего, речь идет об отзывах на какие-либо товары, фильмы, музыкальные альбомы, продукты и т.д. [21]. В таком случае возникает потребность понять, остался ли пользователь доволен или нет. Эта задача тоже решается с помощью методов машинного обучения. На вход методу машинного обучения поступает стандартная матрица терм–текст, при этом, текстам из обучающей выборки приписан либо положительный, либо отрицательный класс. В более поздних работах по классификации текстов по тональности используются вспомогательные ресурсы, такие как WordNet [19]. Согласно [15] методы кластеризации текстов востребованы, в основном, в поисковых системах для улучшения результатов поиска или сжатого представления найденных по запросу текстов. Обзор методов кластеризации текстов [20] показывает, что чаще всего используются либо методы иерархического кластерного анализа, либо метод k-Means и его модификации. На вход этим методам подается стандартная матрица терм–текст, по которой и находятся кластеры.

Благодаря своей популярности векторная модель получила несколько направлений развития. К ним относятся обобщенная векторная модель (generalized vector space model, GVSM) [21], векторные модели семантики [22—24] и, в некотором смысле, вероятностная модель релевантности [2] и модели скрытых тем [25; 26], которые заслуживают отдельного рассмотрения. Обобщенная векторная модель [21] позволяет уйти от принципа попарной независимости термов и учесть попарную корреляцию между векторами, соответствующими термам, в новом пространстве большей размерности. В исходной формулировке пространство векторной модели имеет размерность n равную числу термов

в данной коллекции. В обобщенной векторной модели рассматривается пространство размерности 2^n. В этом пространстве базис задается векторами m_k каждый из которых соответствует конъюнктивному одночлену :

$$f(t_1^{\delta_1}) \cap f(t_2^{\delta_2}) \cap \cdots \cap f(t_n^{\delta_n}),$$

где

$$f(t_i^{\delta_i}) = \begin{cases} f(t_i) = D_{t_i}, \delta_i = 1 \\ f(\bar{t_i}) = \bar{D}_{t_i}, \delta_i = 0 \end{cases}$$

D_{t_i} – множество текстов, содержащих терм t_i, а δ_i определяет отрицание переменной t. В [21] предложен вычислительный алгоритм построения 2^n таких конъюктивных одночленов по матрице смежности термов. Таким образом, в обобщенной векторной модели текст представляется вектором в пространстве, образованном 2^n базисными векторами, имеющими смысл связей между термами.

Векторные модели семантики основаны на гипотезе, сформулированной в [8]. Согласно этой гипотезе, слова, встречающиеся в одинаковом контексте, имеют одинаковый смысл. В основе большинства векторных моделей семантики лежит матрица терм-терм, построенная по аналогии с традиционной матрицей терм-текст [12]. В [23], например, матрица терм-терм строится исключительно для существительных, встречающихся рядом друг с другом в окне $\pm$ 2 слова. Для построения такой матрицы необходимо, во-первых, извлечь все существительные из текста в том порядке, в котором они встречаются в тексте, во-вторых, расположить их в этом же порядке по строкам и столбцам матрицы, в третьих, убрать все стоп-слова из текста (под стоп-словами понимаются предлоги, артикли и местоимения). К значению в клетке матрицы добавляется единица, если слов по строке встречается среди двух слов справа или слева от слова по столбцу в тексте, очищенном от стоп-слов. Далее такая матрица используется для поиска синонимов: согласно косинусной мере, которая будет описана ниже, определяются пары близких друг другу векторов – столбцов матрицы, соответствующих разным словам. В [23] этот алгоритм был протестирован на материалах экзамена TOEFL. В результате применения этого алгоритма синонимов было найдено порядка 90% пар синонимичных пар слов из заданий на поиск синонимов. Анализу таких матриц посвящены работы [24] и [22]. В пер-

вой работе используются ансамбли классификаторов, а во второй – собственный метод кластеризации, названый «Комитеты кластеризации» (“Clustering By Committee”) для выделения групп синонимов. Таким образом, в этом случае векторная модель используется для представления смысла слова вектором других слов. Некоторым упрощением векторной модели является бинарная модель независимости (“Binary Independent Model”), разработанная авторами векторной модели [27]. Ее основное отличие от исходной модели заключается в том, что значения матрицы терм-текст являются бинарными и показывают, встречается ли терм в тексте или нет. Это отличие оказывается существенным и позволяет использовать Байесовский принцип для определения релевантности строки тексту (которая будет описана ниже в Главе 2).

На векторной модели представления текста основана и вероятностная модель релевантности, предложенная в [2]. Эта модель релевантности используется, в основном, в задаче поиска по запросу. Согласно этой модели слова в тексте не независимы, а распределены по смеси Пуассоновских распределений. Тем не менее, и запрос, и текст, следуя векторной модели, представляется вектором частот в пространстве слов.

1.3 Языковая модель представления текста

Языковая модель (language model) [3] позволяет оценить вероятность появления последовательности слов в тексте. В отличии от векторной и вероятностных моделей, языковая модель является генеративной [6; 7], то есть, позволяет генерировать текст. В этой модели текст представляется с помощью цепей Маркова, где каждому узлу соответствует одно слово, а на ребрах – вероятности того, что одно слово встретится после другого. Модель считается генеративной, поскольку позволяет сгенерировать искусственный текст. Обратимся к двум наиболее востребованным видам языковых моделей: модели униграм (одиночных слов) и модели биграм (последовательных пар слов).

При использовании языковых моделей нет необходимости в формальном преставлении всего текста. Говорят о вероятности текста, или о вероятности появления его фрагмента – последовательности слов. Так же как и векторная модель, модель униграм основана на предположении о независимости появления

слова в тексте от предыдущего слова. Согласно модели униграм, текст $t_{1,n}$ – это последовательный набор слов из n слов t_i, $i = 1,n$, причем вероятность всего текста равна произведению

$$P(t_{1,n}) = P(t_1,t_2,\ldots,t_n) = \prod_i P(t_i),$$

то есть, произведению вероятностей появления каждого слова по отдельности. В модели биграм вероятность появления слова зависит от вероятности появления предшествующего слова: $P(t_i|t_1,t_2,\ldots t_{i-1}) \approx P(t_i|t_{i-1})$. Таким образом в модели биграм учитывается локальный контекст слова. Тогда вероятность всего текста:

$$P(t_{1,n}) = P(t_1,t_2,\ldots,t_n) = P(t_1) \times \prod_{i=2} P(t_i|t_{i-1}).$$

Следуя принципу максимального правдоподобия, такая вероятность может быть оценена как $P(t_i|t_{i-1}) = \frac{n(t_{i-1},t_i)}{n(t_{i-1})}$, где $n(t_{i-1},t_i)$ – частота пары слов t_{i-1},t_i в тексте, а $n(t_{i-1})$ – частота слова t_{i-1}.

Аналогичным образом можно сформулировать языковую модель на буквенных последовательностях: вместо вероятности одного слова вычисляем вероятность одной буквы. [3]

В последнее время, языковые модели вновь стали популярны и востребованы среди исследователей благодаря обощению на непрерывный случай и появлению эффективных методов глубинного обучения для оценки параметров таких моделей [28; 29].

Языковые модели используются в тех случаях, когда важно сохранить короткие семантические связи: в задачах машинного перевода [30], распознавания речи [31], исправлении опечаток [32]. Данная работа посвящена задачам другого рода, поэтому мы не будем в дальнейшем заострять внимание на языковых моделях.

1.4 Представление текста на основе моделей скрытых тем

Тематические модели – это класс моделей, объеденный общим предположением о существовании скрытых (латентных) тем. Допустим, есть коллекция

текстов. В этих текстах отражено некоторое количество тем. Темы представляются набором слов, а текст – набором тем. Каждый текст характеризуется вектором, составленным из оценок степени принадлежности текста к различным темам. Каждая тема представляет собой вектор, состоящий из оценок степени принадлежности слова к данной теме. Одна из первых тематических моделей – это латентно-семантический анализ (или латентно-семантическая индексация) [25]. Главное новшество латентно-семантического анализа заключается не столько в математических построениях, сколько в интерпретации получаемых результатов. Латентно-семантический анализ основан на следующем принципе: слова, похожие по смыслу, встречаются в похожих контекстах. Похожесть контекстов может быть установлена с помощью сингулярного разложения матриц. Пусть X – матрица слово (или любая его модификация, терм) – текст. Строки в этой матрице соответствует термам, столбцы – текстам. Значения матрицы показывают, как часто встречается терм в тексте. Заметим, что иногда в матрицу записывают не частоты, а $tf - idf$ веса термов. В этом случае справедливо так называемое сингулярное разложение, представляющее матрицу X как произведение трех матриц:

$$X_{t\times d} = U_{t\times n}\Sigma_{n\times n}V^T_{n\times d},$$

где t –число термов, d – число текстов, $n = \min(t,d)$, $\texttt{rank}(A) = r$, $\Sigma = \texttt{diag}(\sigma_1, \ldots, \sigma_n)$, матрицы U, V ортогональны. Матрицы U представляет термы, матрица V представляет тексты, диагональная матрица Σ – сингулярные значения. Сингулярные числа подчиняются следующем принципу: $\sigma_i > 0$, если $1 \leq i \leq r$, $\sigma_i = 0$, если $i >\geq r + 1$. Сингулярное разложение матрицы позволяет получить приближенное представление ранга k исходной матрицы:

$$\hat{X_{t\times d}} = U_{t\times k}\Sigma_{k\times k}V^T_{k\times d}.$$

В новой матрице $\hat{X}$ слова и тексты представлены векторами размерности k значительно меньшей, чем исходная. Этот факт позволяет интерпретировать столбцы матрицы как группы близких по смыслу слов, т.е. скрытые темы, а строки матрицы – как представления текстов в новом семантическом пространстве [25]. Модель латентного семантического анализа относят к классу векторных моделей, однако другие модели скрытых тем имеют более сложную структуру и уходят от векторного представления текстов [6; 7].

К таким моделям относится модель вероятностного латентного семантического анализа [33]. Она основана на предположении о том, что каждый текст является смесью так называемых тем, причем каждая тема задается собственным распределением слов. В основе модели вероятностного латентного семантического анализа лежит вероятностная модель коллекции текстов:

$$P(d,w) = \sum_{t \in T} P(t)P(d|t)P(w|d,t) \tag{1.1}$$

Здесь $d \in D$ – текст из коллекции текстов, состоящий из слов $w \in V$, T – множество скрытых тем. Для численного решения уравнения используется EM-алгоритм, на каждом шаге которого оцениваются параметры модели $P(t)$, $P(d|t)$, $P(w|d,t)$.

Модель вероятностного латентного семантического анализа получила широкое распространение. Она используется в тех случаях, в которых требуется оценить скрытые переменные, связующие две явные. Например, в задаче коллаборативной фильтрации в качестве скрытое переменной может выступать переменная, соответствующая классу пользователей, а через нее связаны пользовательские сообщества и модели поведения пользователей [34]. Аналогично, в задаче персонификации поиска в Интернете скрытые переменные, связывающие данные о пользователях и их запросы в поисковой системе, строятся на основе истории поведении пользователя [35].

Латентное размещение Дирихле является генеративной моделью, так же, как и языковая модель. Так же, как и вероятностный латентный семантический анализ, латентное размещение Дирихле основано на уравнении 1.1. Каждый текст представляется смесью тем, причем вероятности тем распределены по закону Дирихле. Каждая тема состоит набора слов (термов) и вероятностей, что данное слово относится к этой теме. Вероятность слова принадлежать к теме описывается так же законом Дирихле. Генерация корпуса текстов D, состоящего из M текстов, длинной N_i каждый, устроена так:

1. Пусть распределение тем в тексте i – это распределение Дирихле с параметром α $\texttt{Dir}(\alpha)$: $\theta_i \sim \texttt{Dir}(\alpha)$, $1 \leq i \leq M$.
2. Пусть распределение слов w в теме i – это распределение Дирихле с параметром – это распределение Дирихле с параметром β $\texttt{Dir}(\beta)$: $\phi_k \sim \texttt{Dir}(\beta)$, $1 \leq k \leq K$, K – заданное число тем.
3. Для каждой позиции слова i, j, $1 \leq i \leq M$, $1 \leq j \leq M$:

a) Выбрать тему $z_{ij} \sim \texttt{Multinomial}(\theta_i)$.

b) Выбрать слово $w_{ij} \sim \texttt{Multinomial}(\phi_{z_{ij}})$.

Здесь `Multinomial` – мультиномиальное распределение с одним исходом.

Вероятностные тематические модели получили широкое распространение и используются в задачах поиска по запросу [36; 37], классификации текстов [38; 39], автоматического реферирования текстов [40; 41], фильтрации спама [42; 43], а так же в других областях, не связанных с автоматической обработкой текстов, таких как коллаборативная фильтрация [34; 44], анализ изображений [45; 46].

В задаче поиска по запросу латентный семантический анализ может быть использован для снижения размерности. Допустим, исходная матрица терм-текст X имела размерность $t \times d$, а запрос был представлен вектором q из t компонент. После использования сингулярного матрицы X, вектор запроса может быть преобразован как $\hat{q} = q^T U_k \Sigma_k^{-1}$, после чего используется косинусная мера близости (которая будет описана ниже) для поиска ближайших $\hat{q}$ столбцов матрицы X_k, , соответствующих текстам. Такой поиск по запросу дает результаты точнее, чем поиск по лексическому совпадению, поскольку учитывает скрытые отношения между термами и текстами. В [37] формулируется вероятностная генеративная модель, позволяющая по аналогии с моделью языка оценить вероятность генерации одного текста и вероятность появления запроса в тексте.

В задаче классификации текстов латентно-семантический анализ так же может быть использован для снижения размерности. В [38] для классификации текстов на два класса используется метод ближайшего соседа и машины опорных векторов. Утверждается, что использование латентно-семантического анализа для аппроксимации исходной матрицы терм-текст матрицей меньшего ранга позволяет значительно сократить объем вычислений при незначительной (порядка 2-3%) потере в аккуратности. Адаптация метода латентного размещения Дирихле на случай заранее известных тем, предложенная в [39] носит название labeled LDA. Предполагается, что количество тем, существующих в зафиксированной коллекции текстов, известно заранее, при этом, известно, к какой теме или каким темам относится каждый текст. Примером такой коллекции может служить коллекция сообщений в блогах, помеченных различными тегами-метками. Предложенная в [39] генеративная модель такой коллекции текстов и основанный на ней классификатор превосходит машины опорных векторов, которые обычно используются для подобных задач классификации.

Сравнение векторной модели и модели скрытых тем на основе латентно-семантического анализа в задаче автоматического реферирования текстов проводится в [40]. Рефератом текста считается набор из фиксированного числа предложений из текста, наиболее полно отражающий его содержаний. Предложен следующий алгоритм суммаризации текста:

1. Разбить исходный текст на множество предложений кандидатов S.
2. В пространстве всех слов для каждого предложения составить свой вектор A_i и общий вектор D для всего текста (следуя принципам векторной модели).
3. Найти близость каждого вектора A_i вектору D по косинусной мере близости, которая будет описана ниже.
4. Выбрать предложение S_k соответствующее вектору A_k, наиболее близкому вектору D. S_k будет входит в реферат $S_k \in S$. Если достигнуто искомое число предложений в реферате, алгоритм останавливается. Иначе переходит на шаг 5.
5. Исключить из рассмотрения все термы, входящие в S Составить представления предложений и текста в новом пространстве термов. Перейти на шаг 3.

Для использования латентно-семантического анализа предложена следующая модификация этого алгоритма:

1. Разбить исходный текст на множество предложений кандидатов S, а предложения – на множество термов.
2. Создать матрицу терм – предложение
3. Выполнить сингулярное разложение $A = U\Sigma V^T$, столбцы правой сингулярной матрицы V^T отвечают предложениям: $\psi_i = [v_{i1}, \ldots, v_{ir}]^T$ – вектор-столбец, соответствующий предложению i.
4. Выбрать k-тый столбец правой матрицы сингулярной матрицы V^T.
5. Выбрать предложение, соответствующее максимальному значению выбранного k-того столбца правой матрицы сингулярной матрицы V^T. Согласно гипотезе авторов статей, это предложение будет соответствовать k-той скрытой теме, т.е. его необходимо включить в реферат исходного текста $S_k \in S$.
6. Если достигнуто искомое число предложений в реферате, алгоритм останавливается. Иначе переходит на шаг 4.

Показано, что вторая версия алгоритма незначительно превосходит первую.

В статье [41] предложено использовать латентное размещение Дирихле для автоматического реферирования текста. Согласно предложенному алгоритму, для автоматического построения реферата необходимо:

- Найти скрытые темы в тексте, используя латентное размещение Дирихле.
- Оценить вероятность порождения каждого предложения каждой темой.
- Выбрать наиболее вероятное предложение из каждой темы. Если предложение уже входит в состав реферата, выбрать второе по вероятности.

Существующие методы фильтрации спама позволяют достичь высокой точности при сравнительно невысокой полноте [42]. В этой же статье [42] показано, что использование скрытых тем, полученных с помощью латентно-семантического анализа в качестве признаков для обучения трех разных классификаторов и ансамбля классификаторов, позволяет сохранить точность на высоком уровне и повысить полноту. Однако, автор отмечает важный недостаток предложенного метода, который затрудняет его использование в системах фильтрации спама: латентно-семантический анализ не является интерактивным методом, то есть, при появлении нового текста в коллекции необходимо заново формировать матрицу терм – текст и заново вычислять сингулярные матрицы и матрицу сингулярных значений. В [43] предложен метод разделения коллекции текстов на две части в соответствии с предположением о том, что одна часть коллекции является спамом, а вторая – нет. Авторы использовали размеченную на спам и не-спам коллекцию текстов UK2007-WEBSPAM. На обеих частях коллекции было использовано латентное размещение Дирихле для поиска скрытых тем. Распределения тем получаются разные, несмотря на то, что слова, формирующие темы присутствуют в обеих частях коллекции. Использование найденных скрытых тем в качестве признаков для классификации по признаку спам/не-спам позволяет получить результаты на 10% превосходящие по F-мере другие известные методы, примененные к этой же коллекции текстов.

1.5 Теоретико-множественная модель представления текстов

В простейшей формулировке теоретико-множественная модель представления текстов предполагает следующее: каждый текст представляется неупорядоченным набором термов (то есть, слов или любых других его элементов – лемм, стемов, символьных n-грамм) [6; 7]. Естественным применением такой теоретико-множественной модели можно считать вычисления сходства двух текстов. Пусть дано два текста и каждый текст представлен множеством термов. Тогда сходство между двумя текстами можно оценить с использованием любого теоретико-множественной меры близости. Как правило, любой коэффициент тем или иным образом учитывает количество совпадающих термов (мощность пересечения двух множеств термов), так же как и мощности каждого множества по отдельности или мощность объединения множеств [47].

Приведем несколько примеров теоретико-множественных мер близости. Обозначим множество термов, на которые разбиваются тексты через A и B. Будем оценивать по сходство двух множеств: $\texttt{sim}(A, B)$. Тогда:

- Расстояние городских кварталов (или манхэттенское расстояние) [48] предполагает, что каждый терм – это одна из координат в многомерном пространстве размерности N, где N – общее число термов в обоих множествах. $\texttt{sim}(A, B) = \sum_{i=1}^{N} |a_i - b_i|$, a_i,b_i – частоты соответствующих i-той координате термов;
- Коэффициент Дайса [49]: $\texttt{sim}(A, B) = \frac{2|A \cap B|}{|A|+|B|}$;
- Коэффицинт Жаккара [50]: $\texttt{sim}(A, B) = \frac{|A \cap B|}{|A \cup B|}$;
- Количество совпавших элементов – это абсолютное количество совпавших термов в множествах A,B;
- Коэффициент Симпсона [51]: $\texttt{sim}(A, B) = \frac{|A \cap B|}{\min[|A|,|B|]}$;
- Коэффициент Отиаи [52]: $\texttt{sim}(A, B) = \frac{|A \cap B|}{\sqrt{|A| \cdot |B|}}$.

Однако такая теоретико-множественная модель является тривиальной и не представляет особого практического и исследовательского интереса. В данном диссертационном исследовании предполагается использовать теоретико-множественный аппарат для построения другой модели представления текстов.

В предлагаемой модели текст представляется в виде фрагментов произвольной длины и их частот. Поскольку использование всех возможных фрагментов вряд ли имеет смысл и невероятно неэффективно с вычислительной

точки зрения, мы предлагаем ограничить объём учитываемых фрагментов следующим образом. Для начала разобьем весь текст на строки: одиночные слова или последовательные пары или тройки слов. Найдем все возможные подстроки строк и определим их частоты в исходном тексте. Это и будет итоговой теоретико-множественной моделью представления текста.

Вычисление частоты каждой подстроки данной совокупности строк требует как минимум квадратичного от числа подстрок времени. Для произвольного достаточно большого текста такой перебор будет малоэффективным. Следовательно, возникает необходимость в использование эффективной структуры данных для поиска всех возможных фрагментов строк и их частот. Такой структурой является аннотированное суффиксное дерево. Прежде чем определять данную структуру и ее свойства, совершим исторический экскурс в область суффиксных деревьев вообще.

Суффиксные деревья были предложены в статье [53] в качестве средства для поиска нечетких совпадениях между строками. Суффиксные деревья получили широкое распространение в биоинформатике, где они используются, в основном, для поиска закономерностей в ДНК или в белках, которые записаны длинной последовательностью символов [54]. Иногда суффиксные деревья используются для индексации текстов и организации словарей [55] и реже – в качестве моделей представления текста. Впервые суффиксные деревья в качестве модели представления текстовой коллекции были предложены в [4], где суффиксное дерево, построенное по коллекции текстов послужило основой для поиска кластеров – т.е. групп – близких по смыслу текстов. Структура суффиксного дерева, использованного в [4] несколько отличалась от классического определения суффиксного дерева, и предполагала определенный способ аннотирования суффиксного дерева порядковыми номерами текстов в коллекции, в результате чего стало возможным использование суффиксного дерева для поиска кластеров в коллекции текстов. Другие работы, в которых суффиксные деревья использованы в качестве модели представления коллекции текстов, так же предполагают тот или иной способ аннотирования дерева номерами текстов или частотами [56—58].

Согласно [59], суффиксное дерево для m-символьной строки S представляет собой ориентированное дерево с корнем, имеющее ровно m листьев, занумерованных от 1 до m. Каждая внутренняя вершина, отличная от корня, имеет не меньше двух детей, а каждая строка помечена непустой подстрой строки S. Ни-

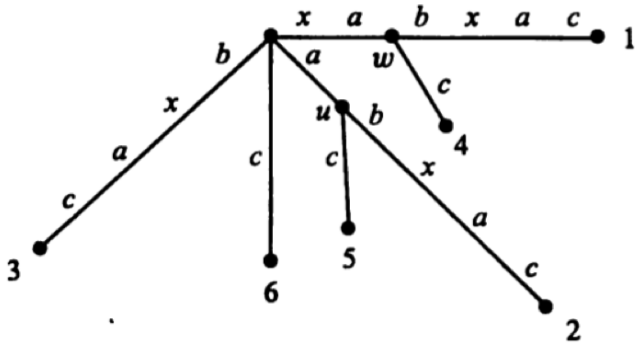

Рисунок 1.1 — Суффиксное дерево для строки $S =$ `xabxac` [59]

какие две дуги, выходящие из одной и той же вершины, не могут иметь пометок, начинающихся с одного и того же символа. Главная особенность суффиксного дерева заключается в том, что для каждого листа i конкатенация меток дуг на пути от корня к листу i составляет / произносит / кодирует / прочитывает суффикс строки S, который начинается в позиции i, то есть, $S[i : m]$.

Там же приводится пример суффиксного дерева для строки $S =$ ‘‘`xabxac`’’ (Рис. 1.1).

Путь до листа 1 прочитывает первый суффикс строки $S[1 : 6] =$ ‘‘`xabxac`’’, совпадающий с исходной строкой, путь до листа 2 – второй суффикс строки $S[2 : 6] =$ ‘‘`abxac`’’, путь до листа 3 – третий суффикс строки $S[3 : 6] =$ ‘‘`bxac`’’, путь до листа 4 – второй суффикс строки $S[4 : 6] =$ ‘‘`xac`’’, путь до листа 5 – второй суффикс строки $S[5 : 6] =$ ‘‘`ac`’’. Путь до листа 6 прочитывает последний шестой суффикс $S[6 : 6] =$ ‘‘`c`’’. В таком дереве всего две внутренние вершины: они помечены буквами “u”, “v”. Заметим, что с точки зрения частотного анализа, наличие внутренних вершин в таком суффиксом дереве означает, что путь от корня до внутренней вершины прочитывает повторяющийся фрагмент строки. Так, фрагмент ‘‘`xa`’’ встречается в строке $S =$ ‘‘`xabxac`’’ дважды: $S[1 : 2] = S[4 : 5]$‘‘`xa`’’, аналогично меньший фрагмент ‘‘`a`’’ встречается в строке $S =$ ‘‘`xabxac`’’ дважды: $S[2] = S[5] =$ ‘‘`a`’’. Тем не менее, такое представление суффиксного не дерева не позволяет учитывать сколько именно раз встречается во входной строке повторяющийся фрагмент.

Представленное на Рис. 1.2 суффиксное дерево построено по двум строкам $S_1 =$ `xabxac`, $S_2 =$ `babxba`. Каждому листу в нем приписана своя метка – одна или две пары чисел: первое число в паре означает номер строки, второе – номер суффикса. Символ “$” использован в качестве терминального и добавлен к концу каждой строки. Использование терминального символа позволяет

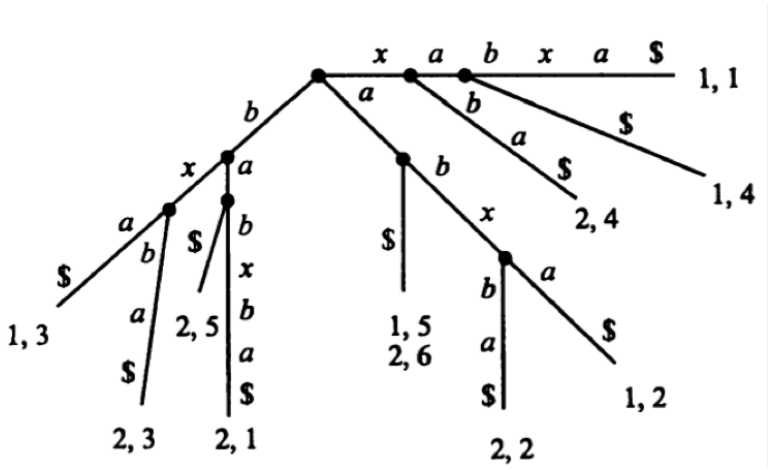

Рисунок 1.2 — Суффиксное дерево для двух строк $S_1 = \texttt{xabxac}$, $S_2 = \texttt{babxba}$ [59]

различать листы и промежуточные вершины Не трудно убедиться, что пути от корня до внутренних вершин кодируют фрагменты, повторяющиеся во входных строках. Так, например, фрагмент ‘‘a’’ встречается во входных строках четыре раза: дважды в качестве последнего суффикса строк – $S_1[5]$, $S_2[6]$ и дважды в качестве префикса суффиксов $S_1[2:5]$, $S_2[2:6]$. Заметим, что количество повторений фрагмента ‘‘a’’ можно установить по количеству меток у листьев, которые покрывает соответствующая внутренняя вершина.

Естественным развитием модели суффиксного дерева представляется модель аннотированного суффиксного дерева (АСД), в которой частоты всех фрагментов присутствуют явно [5]. В этой работе аннотированное суффиксное дерево определяется как суффискное дерево, в котором:

– Символы стоят не на ребрах, а в узлах;
– Каждому узлу соответствует один символ;
– Каждый узел помечен частотой фрагмента, который прочитывает путь от корня до этого узла;
– Опущены терминальные символы и метки листьев, представляющие номер суффикса и входной строки.

По аналогии с [59] построим АСД для строки $S =$ ‘‘xabxac’’ Рис. 1.3.

У этой строки шесть суффиксов: $S[1:6] =$ ‘‘xabxac’’ – этому суффиксу соответствует цепочка узлов, помеченная буквой A, $S[2:6] =$ ‘‘abxac’’ – B, $S[3:6] =$ ‘‘bxac’’ – C, $S[4:6] =$ ‘‘xac’’ – D, $S[3:6] =$ ‘‘ac’’ – E, $S[6:6] =$ ‘‘c’’ – F. Фрагменты “xa”, “a” входной строки $S =$ ‘‘xabxac’’ встречаются дважды: $S[1:2] = S[4:5] =$ ‘‘xa’’, поэтому частоты соответствующим их

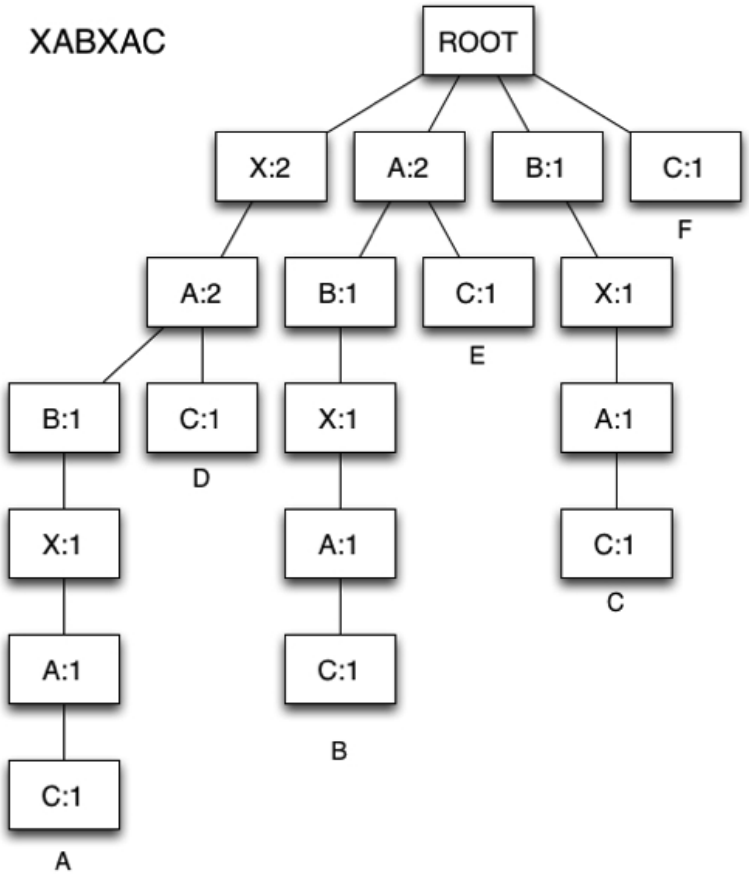

Рисунок 1.3 — Аннотированное суффиксное дерево для строки $S =$ ‘‘xabxac’’

узлов составляют 2, а частоты остальных узлов равны 1, поскольку они представляют уникальные фрагменты строки. Продолжая аналогию с [59] построим обобщенное суффиксное дерево для двух строк $S_1 =$ ‘‘xabxac’’ , $S_2 =$ ‘‘babxac’’.

За исключением первого суффиксы этих строк совпадают: $S_1 =$ ‘‘xabxac’’ $\neq S_2 =$ ‘‘babxac’’. Эти несовпадающие суффиксы представлены цепочками узлов A и G. Частоты почти всех узлов в этих строках составляют 1. От 1 отличаются только частоты первых узлов этих цепочек “xa”, “b”, поскольку они встречаются по 3 раза во входных строках. Остальные суффиксы представлены такими же узлами, как на Рис. 1.4, но с удвоенными частотами.

В [59] показаны два важных свойства АСД:

Свойство 1. Частота любого узла равна сумме частот его узлов-детей, так как родительский узел соответствует префиксу нескольких суффиксов и его частота складывается из частот этих суффиксов. Отсюда же следует другое свойство АСД.

Свойство 2. Частота родительского узла равна сумме частот листьев, которые он покрывает.

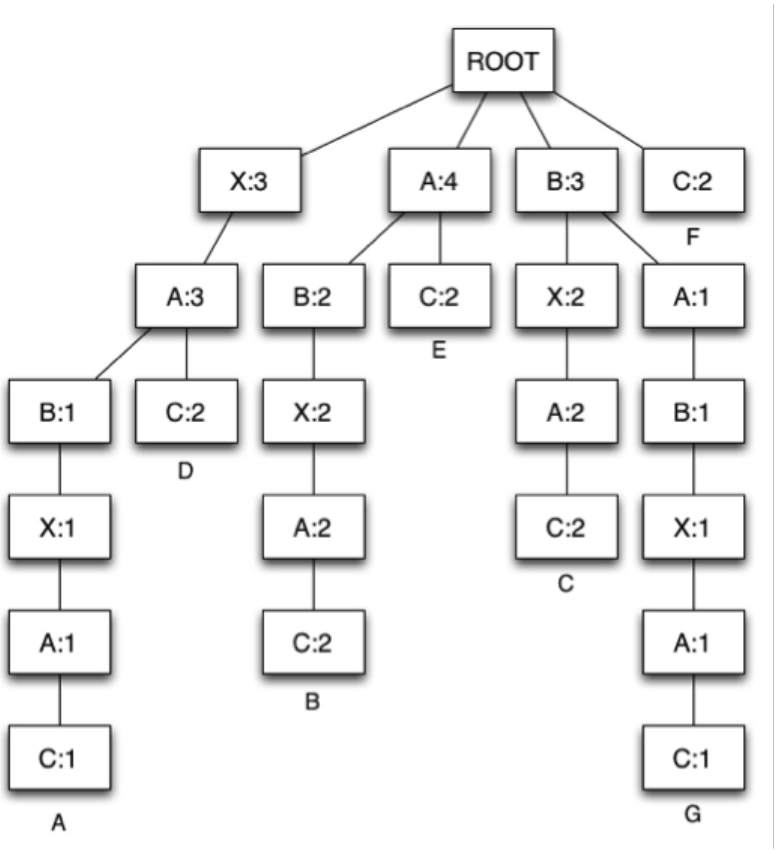

Рисунок 1.4 — Обобщенное аннотированное суффкисное дерево для строк $S_1 =$ ‘‘xabxac’’, $S_2 =$ ‘‘babxac’’

Рассмотрим алгоритм суффиксного дерева и его адаптацию на случай аннотированного суффиксного дерева.

1.5.1 Наивный алгоритм построения суффиксного дерева

Пусть на вход поступает строка S из m символов , по которой требуется построить суффиксное дерево. Следуя [59], поместим сначала в дерево простую дугу для первого суффикса $S[1:m]\$$, затем последовательно добавим в растущее дерево суффиксы $S[i:m]\$$, $2 \leq i \leq m$.

Algorithm 1 Наивный алгоритм построения аннотированного суффиксного дерева

Вход: строка S

Выход: АСД для строки S

Инициализация: создаем пустую структуру, в которой будет храниться АСД. Обозначим ее `ast`

for $i \in \{2,M\}$ **do**

Для суффикса $S[i : m]$ найти в `ast` совпадение $\texttt{match}_i$ – самый длинный путь от корня, метки в котором совпадают с префиксом $S[i : m]$. Такой путь будет единственным, так как никакие две дуги, выходящие из одной вершины, не могут иметь одинаковых меток. Пусть совпадение найдено. Тогда

if $|\texttt{match}_i| = |S[i : m]|$ – длина совпадения совпадает с длиной текущего суффикса, то есть, текущий суффикс целиком присутствует в дереве **then**

Алгоритм останавливается в листе w с меткой $S[m]$, к которой ведет путь $\texttt{match}_i$, частоты всех узлов в найденном совпадении увеличиваются на 1.

else

Длина совпадения меньше длины суффикса: $|\texttt{match}_i| < |S[i : m]|$. Тогда увеличим на 1 частоту всех узлов в совпадении, а к последнему совпавшему узлу $node_i = S[|\texttt{match}_i|]$ добавим цепочку узлов $S[|\texttt{match}_i| : m]$ с частотами 1. Если совпадения не найдено $k = 0$, поступаем аналогичным образом, считая $node_i$ корнем дерева и создавая новую цепочку от корня `ast`.

end if

end for

1.5.2 Построение аннотированного суффиксного дерева на основе разбиения текста на фрагменты

Для построения аннотированного суффиксного дерева разобьём текст на строки: несколько (например, три) последовательно идущих слова в тексте. Таким образом получается построить АСД заведомо ограниченное по глубине.

Предварительна подготовка текста заключается в формировании строк длины, начинающихся с 1, 2, 3 и т.д. слова текста. Первая строка для построения АСД состоит из 1, 2 и 3 слова в тексте, вторая – 2, 3 и 4, и так далее. Обозначим такие строки через $f_1, \ldots, f_N$. Длина всех фрагментов не превышает l, где l – максимальная длина строки в символах, поэтому не возникает необходимости добавлять терминальный символ к концам фрагментов.

Как пример, построим АСД для строки $s =$ ‘‘mining’’. Для суффиксов первых трех суффиксов $s =$ ‘‘mining’’, $s =$ ‘‘ining’’, $s =$ ‘‘ning’’ и последнего суффикса $s =$ ‘‘g’’ совпадений не будет найдено. Поэтому в дерево будут добавлены соответствующие цепочки с частотами равными 1. При добавлении четвертого суффикса $s[4:] =$ ‘‘ing’’ будет найдено непустое совпадение ‘‘in’’, поэтому частота узлов из совпадения будет увеличена на 1, а у узла с меткой ‘‘n’’ будет создан новый потомок с меткой ‘‘g’’ и частотой 1. Аналогично, при добавлении суффикса $s[5:] =$ ‘‘ng’’ будет найдено совпадение из одного узла с меткой ‘‘n’’ Следуя алгоритму, частота узла будет увеличена на 1 и у него будет создан новый потомок с меткой ‘‘g’’ и частотой 1.

Если к уже построенному для строки $s =$ ‘‘mining’’ АСД требуется добавить строку $t =$ ‘‘dining’’, то для первого суффикса $t =$ ‘‘dining’’ не будет найдено совпадений, поэтому будет создана новая цепочка узлов с частотами 1. Для всех остальных суффиксов строки t будут найдены совпадения, полностью покрывающие суффиксы, поэтому у всех узлов в дереве частоты будут увеличены вдвое, но новых узлов создано не будет. Результирующее АСД представлено на 1.5.

Опишем наивный алгоритм построения АСД на псевдокоде и оценим его сложность по времени и по памяти, следуя [60].

Анализ сложности по времени и памяти данного алгоритма достаточно прост. Перебирая строки, мы посимвольно просматриваем все ее суффиксы, затрачивая на i-ю строку длины n_i количество операций, пропорциональное

Algorithm 2 Построение АСД для коллекции строк $f_1, \ldots, f_N$

Вход: Коллекция строк $f_1, \ldots, f_N$

Выход: АСД для коллекции строк $f_1, \ldots, f_N$

Инициализация: создаем пустую структуру, в которой будет храниться АСД. Обозначим ее `ast`. Далее итеративно будем добавлять в `ast` фрагменты входной коллекции.

for $i \in \{1,N\}$ **do**

for $j \in \{1,l\}, l = |f_i|$ **do**

Для каждого суффикса $f_i[j:]$ ищем в `ast` совпадение – путь от корня, совпадающий с максимальным префиксом суффикса $f_i[j:]$. Пусть $\texttt{match}_{ij}$ – совпадение $f_i[j:]$, $|\texttt{match}_{ij}| = k$.

if $k = l$ **then**

Частоты всех узлов в найденном совпадении $|\texttt{match}_{ij}|$ увеличиваются на 1.

else

$k < l$. Требуется создать новые узлы для фрагмента строки $f_i[k+1:l]$. Для этого создаем у последнего узла в найденном совпадении нового потомка, помечаем его символом $f_i[k+1]$ и приписываем ему частоту 1. Таким же образом последовательно создаем узлы для всех оставшихся символов в фрагменте. В результате будет создана новая цепочка узлов, кодирующая текущий суффикс. Если совпадения не найдено $k = 0$, поступаем аналогичным образом, создавая новую цепочку от корня `ast`.

end if

end for

end for

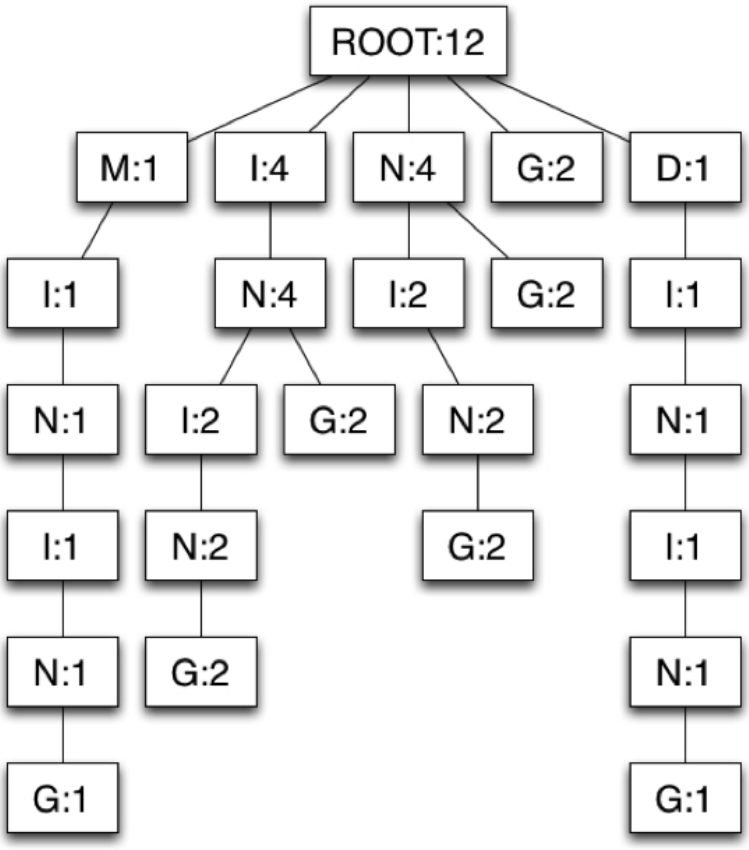

Рисунок 1.5 — Аннотированное суффиксное дерево для строки $S =$ ‘‘`mining`’’

Algorithm 3 NaiveConstruction(C)

Вход: Коллекция фрагментов $C = f_1, \ldots, f_m$

Выход: АСД для C

for $i \leftarrow 1$ to m **do**
  for $i \leftarrow 1$ to $n_i = |s_i|$ **do**
    1. **do** $k \leftarrow$ длина совпадения $s_i[j:]$ с АСД
    for узел u из $s_i[j : (j + k - 1)]$ **do**
      2. **do** k присвоить $f(u) \leftarrow f(u) + 1$
    end for
    for $l \leftarrow j + k$ to n_i **do**
      3. вставить узел v
      4. присвоить $f(v) \leftarrow 1$
    end for
  end for
end for

$1 + 2 + \cdots + n_i = \Theta(n_i^2)$. Общее время работы алгоритма для коллекции из строк, таким образом, может быть оценено как $\Theta(n_1^2 + \cdots + n_1^m)$, или, если использовать более грубую оценку, $O(mn_{max}^2)$. Отметим, что определенное способом выше АСД невозможно построить с использованием меньшего числа операций, так как само оно занимает в памяти место, квадратично зависящее от длины закодированных в нем строк.

1.6 Выводы по главе

В первой главе представлены и описаны четыре основных класса моделей представления коллекций текстов: векторная модель, языковая модель, модель на основе скрытых тем и теоретико-множественная модель. Под формальным представлением коллекций текстов понимаются следующие структуры: векторное пространство термов, Марковские цепи, системы распределений вероятностей и суффиксные деревья – частный вид помеченных графов. Каждая из моделей представлений текстов удобна для использования в определенном классе задач автоматической обработки текстов. Проведен обзор задач автоматической обработки текстов и показано, какая модель представления текста используется для решения той или иной задачи. Сформулирована собственная теоретико-множественная модель представления текста и предложено использовать аннотированное суффкисное дерево для быстрого вычисления частот в теоретико-множественной модели. Введено понятие аннотированного суффиксного дерева как структуры данных, используемой для хранения и вычисления всех фрагментов и их частот одного текстового документа или входной коллекции текстовых документов. В дальнейшем будет показано, как эта модель может быть адаптирована к решению конкретных задач анализа текстов.

Глава 2. Оценивание релевантности строки тексту с использованием метода аннотированного суффиксного дерева (АСД)

2.1 Проблема оценивания релевантности строки тексту и основные подходы к ее решению

Проблема оценивания релевантности строки тексту формулируется следующем образом. Пусть дана некоторая коллекция текстов и некоторая строка. Под строкой понимается, как правило, одно слово или согласованное словосочетание. Требуется определить, насколько релевантна строка каждому тексту из коллекции, то есть, встречается она или ее фрагменты в тексты, и, если да, насколько строка соответствует содержанию текста. Другими словами, необходимо ранжировать тексты по степени релевантности им данной строки. Численные оценки релевантности тому или иному тексту сами по себе не имеют смысла и интересны исключительно с сравнительной точки зрения: какому тексту более релевантна строка. Понятие релевантности имеет двойственный характер. С одной стороны, чаще всего термин релевантность возникает в области информационного поиска (поиска по запросу). В задаче поиска по запросу требуется показать пользователю релевантные его запросу (строке) документы (тексты). Говорят, что релевантные документы – это такие тексты, которые удовлетворяют информационные нужды пользователя [61; 62]. С другой стороны, в формальных моделях релевантность строки тексту определяется по близости между формальными представлениями строки и текста или по вероятности появления строки в тексте. Как правило, в таких моделях отсутствует пользователь и его информационные нужды. Они появляются позже, в качестве надстройки над математическими моделями и учитывают не непосредственные характеристики строки или текста, а их контекст, время, место появления и другие свойства [63]. В этой работе мы будем обращаться исключительно к математическим мерам релевантности. Перечислим основные математические модели и меры релевантности.

2.1.1 Теоретико-множественные меры релевантности

В качестве примитивной меры релеватности можно использовать теоретико-множественные меры близости между множествами термов, на которые разбиваются строка и текст. В [47] приведен исчерпывающий обзор теоретико-множественных мер близости. Каждая мера близости тем или иным образом учитывает количество совпадающих термов (мощность пересечения двух множеств термов), так же как и мощности каждого множества по отдельности или мощность объединения множеств.

Перечеслим данные коэффициенты. Обозначим множество термов, на которые разбивается строка, через S, а множество термов, на которые разбивается текст, через T. Будем оценивать по мере близости строки и текста релевантность строки тексту: $\texttt{relevance}(string,text) = sim(S,T)$. Тогда:

- Расстояние городских кварталов (или манхэттенское расстояние) [48] предполагает, что каждые терм – это одна из координат в многомерном пространстве размерности N, где N – общее число термов в строке и в тексте. $sim(S,T) = \sum_{i=1}^{N} |s_i - t_i|$, s_i,t_i – частоты соответствующих i-той координате термов;
- Коэффициент Дайса [49]: $sim(S,T) = \frac{2|S \cap T|}{|S|+|T|}$;
- Коэффицинт Жаккара [50]: $sim(S,T) = \frac{|S \cap T|}{|S \cup T|}$;
- Количество совпавших элементов – это абсолютное количество совпавших термов в множествах S,T;
- Коэффициент Симпсона [51]: $sim(S,T) = \frac{|S \cap T|}{\min[|S|,|T|]}$;
- Коэффициент Отиаи [52]: $sim(S,T) = \frac{|S \cap T|}{\sqrt{|S| \cdot |T|}}$.

Заметим, что перечисленные выше меры близости являются симметричными. При необходимости, например, в задаче поиска по запросу могут быть использованы несимметричные меры близости – т.н. меры включения

2.1.2 Релевантность в векторной модели

Релевантность строки тексту определяется в векторной модели так: текст и строка представляются векторами в общем пространстве слов, а релевантность определяется по сходству двух построенных векторов.

Пусть дана строка $string$ и коллекция текстов, в которой N текстов. Требуется определить релевантность $\texttt{relevance}(string,text)$ строки одному из текстов $text$ из коллекции. Для начала зададим координаты векторного пространства: V – словарь, содержащий все слова коллекции текстов, $t \in V$ – слова. Каждому слову t соответствует своя координата. Тогда тексту соответствует вектор $w = (w_1, \ldots, w_{|V|})$ Компоненты вектора – это либо частоты слов, либо $tf - idf$ веса, которые вычисляются по формуле

$$w_{ij} = tf \times idf = tf_{ij} \times \log \frac{N}{n(t_i) + 1},$$

где tf_{ij} – частота слова t_i в тексте j, $n(t_i)$ – число текстов, содержащих слово t_i , N – – количество текстов. Каждому тексту соответствует вектор в пространстве слов. Размерность этого вектора совпадает с количеством различных слов во всех текстах из коллекции. Составим аналогичный вектор для строки с использованием $tf - idf$ весов для слов из строки. Релевантность строки тексту определяется через косинусную меру близости между соответствующими векторами:

$$\texttt{relevance}(string,text) = \texttt{cos}(string,text) = \frac{string \times text}{||string|| \times ||text||} \qquad (2.1)$$

2.1.3 Релевантность в бинарной модели независимости

Релевантность строки тексту в бинарной модели независимости определяется по следующему Байесовскому правилу:

$$\texttt{relevance}(string,text) = P(R|string,text) = \frac{P(string|R,text) \times P(R|text)}{P(string|text)}, \qquad (2.2)$$

где R – бинарная переменная, принимающая два значения, 1, если строка релевантна тексту и 0 в обратном случае. Следовательно, $P(string|R = 1,text)$ и $P(string|R = 0,text)$ – вероятности того, что строка релевантна тексту и того, что строка нерелеватна тексту, соответственно. Заметим, что $P(string|R = 1,text) + P(string|R = 0,text) = 1$.

2.1.4 Релевантность в вероятностной модели

Вероятностная модель представления текста используется, в основном, в задачах извлечения/поиска информации и сформулирована в терминах задачи поиска по запросу. Она основана на теоретическом принципе ранжирования вероятностей (“Probability Ranking Principle”, PRP) [2]: наиболее эффективная поисковая машина выдает тексты пользователю в соответствии с убыванием вероятности релевантности его запросу. Здесь под релевантностью понимается соответствие содержания текста запросу (более широкое понимание понятия релевантности будет изложено ниже). Предполагается, что релевантность rel – это случайная величина, которая принимает бинарные значения: rel – если запрос релевантен тексту, $\bar{rel}$ – в обратном случае.

Она построена в предположениях теоретической модели, согласно которой каждый текстовый текст представляется как смесь двух Пуассоновских распределений [2]. Одно из них отвечает за распределение обычных слов, другое – за распределение «элитных» слов, то есть, тех, на которых лежит основная смысловая нагрузка в разрезе рассматриваемой тематики. Обычно тематика задаётся тем запросом на извлечение информации, относительно которого и оценивается релевантность. Релевантность строки тексту в этой модели определяется по вероятности того, что слова, принадлежащие строке, окажутся элитными в тексте.

Следуя векторной модели представления текстов вероятностная модель релевантности предполагает, что строка и текст – два набора слов. Ставшая очень популярной в последнее время мера релевантности BM25 придаёт больший вес значимым словам и меньший – незначимым:

$$\texttt{relevance}(string,text) = \sum_{i=1}^{N} \texttt{IDF}(t_i)\frac{(k_1+1)tf_i}{tf_i + k_1(1-b+b\frac{|N|}{avgdl})}, \qquad (2.3)$$

где $avgdl$ – среднее количество слов в тексте, а b, k_1 – константы, равные, как правило 1.5 и 0.75, соответственно, согласно citerobertson2009probabilistic.

В качестве нормализующего сомножителя используется функция IDF, имеющая следующий вид: $\texttt{IDF}(t_i) = \log\frac{N-n(t_i)+0.5}{n(t_i)}$, где $n(t_i)$ – число текстов, содержащих слово t_i. Функции IDF имеет смысл обратной частоты: чем больше текстов содержат данное слово, тем менее он значим.

2.1.5 Релевантность в тематических моделях

Релевантность строки тексту в модели латентно-семантического анализа определяется следующим образом. Пусть для строки определен вектор частот слов $string$ в исходном пространстве слов. Представим в его новом пространстве меньшей размерности: $\hat{string} = \Sigma_k^{-1}U_k^T string$. Релевантность строки тексту определяется по косинусной мере близости между преобразованным вектором, соответствующем строке, и вектором, соответствующем тексту, т.е. столбцу в матрице $\hat{X}$.

В генеративных моделях представления текста, таких как языковая модель или модель латентного размещения Дирихле, релевантность строки тексту составляет вероятность порождения текстом строки, то есть,

$$P(string|text) = \prod_{t\in string} P(t|text),$$

где t – слова из которых состоит строка $string$.

В [64] предложены следующие оценки вероятностей

$$P(w|D) = \frac{N_d}{N_d+\mu}P_{ML}(w|D) + (1-\frac{N_d}{N_d+\mu})P_{ML}(w|coll),$$

где μ – параметр распределения Дирихле (в [64] предложено использовать $\mu = 1000$), N_d – количество текстов в коллекции, $P_{ML}(w|D)$, $P_{ML}(w|coll)$ – оценки по принципу максимального правдоподобия вероятностей слова w в тексте D и коллекции $coll$, соответственно.

Альтернативная схема вычисления оценок вероятностей предложена в [37]:

$$P(w|D) = \lambda(\frac{N_d}{N_d+\mu}P_{ML}(w|D) + (1-\frac{N_d}{N_d+\mu})P_{ML}(w|coll)) + (1-\lambda)P_{lda}(w|D) \quad (2.4)$$

которая отличается от предыдущей схемы наличием последнего члена формулы $(1-\lambda)P_{lda}(w|D)$. Здесь λ – это нормировочный показатель, а $P_{lda}(w|D)$ – оценка вероятности слова w в тексте по модели ЛРД, которая находится по стандартному алгоритму ЛРД, примененному к проиндексированной коллекции текстов.

2.1.6 Релевантность в теоретико-множественной модели представления текстов

В предложенной в данном диссертационном исследовании теоретико-множественной модели представления текстов каждый текст представляется набором всех фрагментов строк и их частотами. В качестве строк выступают одно-, двух- или трехсловные последовательности. Для определения релевантности строки тексту в данной модели введем понятие совпадения. Совпадение – это такая подстрока входной строки, которая встречается и в множестве фрагментов текста. Максимальным совпадением назовем такое совпадение, которое при добавлении символа в начало или в конец, перестает быть совпадением.

Допустим, что существует совпадение строки с текстом $s_i \dots s_j$. Определим его вероятность, как условную частоту последнего символа в совпадении s_j : $P(s_i \dots s_j) = P(s_j|s_i \dots s_{j-1})$ (УВС). Вероятностью максимального совпадения тогда является средняя сумма совпадений, в него входящих (СУВС): $P_{max}(s_i \dots s_j) = \frac{\sum_{k=1}^{j} P(s_k \dots s_j)}{(j-i)}$. Полной релевантностью строки тексту является средняя сумма вероятностей максимальных совпадений данному тексту (что эквивалентно средней условной вероятности символа в совпадении, СУВСС): $\texttt{relevance}(string,text) = \frac{\sum_{i=1}^{|n|} P_{max}(s_i \dots s_n)}{n}$, где n – количество символов в строке $string$.

Для эффективной реализации вычисления оценок релевантности следует использовать аппарат аннотированного суффиксного дерева. Оценивание релевантности строки тексту с использованием АСД предполагает построение АСД для текста и последующее наложение строки на АСД [5].

Метод AST оценивания релевантности строки тексту

Каждый текст представляется собственным АСД, с которым сличается строка для вычисления оценок релевантности. Оценка релевантности $\texttt{relevance}(string,text)$ строки $string$ тексту $text$ вычисляется следующим образом:

1. Выделяются все суффиксы строки $string$
2. Для каждого суффикса вычисляется оценка его совпадения `match` с АСД:

$$\texttt{score}(\texttt{match}(suffix,ast)) = \sum_{node \in match} \phi(\frac{f(\texttt{node})}{f(\texttt{node}_{parent})}), \tag{2.5}$$

   где совпадение – это путь от корня дерева, кодирующий совпадающий с ним префикс суффикса или суффикс целиком, $f(\texttt{node})$ – частота, приписанная узлу `node` АСД из совпадения, $f(\texttt{node}_{parent})$ – частота, приписанная родителю данного узла
3. Оценка релевантности вычисляется как сумма всех оценок:

$$\texttt{relevance}(string,text) = \texttt{SCORE}(string,text) = \sum_{suffix} \texttt{score}(\texttt{match}(suffix,ast)) \tag{2.6}$$

В формуле 2.5 ϕ – это шкалирующая функция, переводящая оценку совпадения в уровень релевантности. Рассмотрим три вида шкалирующей функции ϕ, рекомендованных в [5] на основе экспериментов по категоризации электронной почты:

- $\phi(x) = 1$ – константа (обозначение – constant);
- $\phi(x) = x$ – линейная (обозначение – linear);
- $\phi(x) = \log \frac{x}{1-x}$ – логистическая (обозначение – logit);

– $\phi(x) = \sqrt{x}$ – корень квадратный (обозначение – root);
– $\phi(x) = x^2$ – квадратичная функция (обозначение – square);
– $\phi(x) = \log(x)$ – логарифмическая функция (обозначение – log);
– $\phi(x) = \frac{1}{1+e^{-x}}$ – сигмоида (обозначение – sigmoid).

Из этих трёх только линейная, ничего не меняющая функция, имеет очевидный операциональный смысл – средней условной вероятности символа в совпадении (СУВСС); две нелинейные шкалы из [5] могут быть использованы для контроля.

2.2 Метод nAST-k оценивания релевантности строки тексту с использованием нормированного АСД

Метод nAST-k используется для оценивания релеватности строки (или коллекции строк) тексту (коллекции текстов). Метод nAST-k имеет несколько радиикальных отличий от метода аннотированного суффиксного дерева, описанного в [5]. Во-первых, используется другой способ подготовки текстов: текст представляется набором строк нефиксированной длины, а не набором фрагментов. Во-вторых, используется нормированная оценка релевантности. В-третьих, метод nAST-k предусматривает параметризацию АСД, в том числе, процедуру очистки АСД от шума. В-четвертых, для АСД построения используется алгоритм, имеющий линейную сложность по времени.

2.2.1 Структура метода

Метод оценивания релевантности строки тексту с использованием нормированного АСД заключается в

– подготовке текстов к обработке путем разбиения на последовательные фрагменты
– определении и вычислении параметров АСД
– вычислении нормированной оценки релевантности

2.2.2 Подготовка текстов к обработке

Подготовка текстов к обработке проводится согласно стандартной схеме, представленной в [6]: удаление xml- и html-разметки, если она присутствует в тексте, токенизация, удаление знаков препинания и прочих символов, включая цифры и псевдографику, приведение всех слов к нижнему регистру. Под токенизацией мы понимаем процедуру последовательного разбиения текста на предложения и на слова.

Обработанный текст представляет собой последовательность строк. Под строкой мы понимаем несколько последовательно идущих слов из одного предложения. В [65] экспериментально показано, что глубина дерева, построенного по строкам из 2-4 слов, вполне достаточна для задач анализа текстов. Таким образом, текст после обработки состоит из строк из 2-4 слов, соединенных через пробел. Строки строятся следующим образом: первая строка начинается с первого слова в тексте и заканчивается 2-4 словом в тексте, вторая начинается со второго и заканчивается соответственно на 3-5 слове. Например, если первая строка обработанного текста такова: “слово1 слово2 слово3”, то вторая строка текста будет такой: “слово2 слово3 слово4”. При этом учитываются границы предложений: в одну строку не должны попадать слова из разных предложений.

2.2.3 Параметризация АСД

Рассмотрим три параметра АСД: глубину, уровень очистки от шума и размах. Эти параметры представляют некоторые свойства АСД, изменяя которые можно повысить эффективность метода оценивания релевантности строки тексту.

Глубина АСД

Глубина АСД – это количество узлов в максимальной по длине цепочке. Очевидно, что глубина АСД определяется количеством символов в самой длинной цепочке. Очевидно, что глубина АСД не превосходит длину строк, которые на него накладываются (с поправкой на 1-10 символов). При разбиении текста на строки следует учитывать, что от длины строк зависит глубина АСД и выбирать длину строки, т.е. количество слов в строке разумным способом. Например, если заранее известно, что большая часть входных строк состоят из трех слов, то и текст следует разбивать на строки из трех слов, как это сделано в [65]. Таким образом становится возможным ограничение на объем АСД, а следовательно на объем используемой памяти.

Уровень очистки АСД от шума

В больших АСД на первых уровнях располагается довольно большое число узлов с относительно высокими частотами, которые дают примерно одинаково большой вклад в оценку любой строки, накладываемой на АСД. Первый уровень характеризует частоты отдельных символов, второй – частоты упорядоченных пар символов, третий – частоты упорядоченных троек и т.д. Очевидно, что такие короткие элементы текста не могут нести особой семантической направленности. Поэтому возникает гипотеза, что вклады узлов начальных уровней АСД в оценки релевантности носят характер шума, и оценка релевантности станет более адекватной, если ее очистить от вклада узлов начальных уровней. Для проверки этой гипотезы мы обнуляли частоты узлов на первом, втором и т.д. уровнях.

Обозначим очистку АСД от шума через $\phi.X$, где ϕ – вид шкалирующей функции, а X – уровень в АСД, до которого обнулялись частоты.

Размах АСД

Под размахом АСД понимается среднее количество потомков у одного узла. Чем больше цепочек в АСД, не имеющих разветвлений, тем меньше размах АСД. Определим размах следующим образом:

$$\texttt{range}(ast) = \frac{\sum_{node \in Parent} |node|}{|Parent|},$$

где $Parent$ – множество узлов, у которых есть потомки (т.е. которые не являются листьями). Так, например, размах АСД, показанного на Рис. 1.5 составляет 1.18.

АСД, построенные по разным коллекциям строк, могут отличаться между собой по размаху. Чем больше и разнообразнее коллекция строк, тем больше размах соответствующего АСД.

2.2.4 Нормирование оценки релевантности

Часто возникает потребность сравнить оценки сходства строк с двумя или более АСД. Получаемые оценки могут сильно зависеть от размаха АСД. Чем больше узлов в АСД, тем больше разброс оценок, получаемых при сличении строк с этим деревом. Для того, чтобы сделать оценки по разным деревьям сравнимыми между собой, в формуле 2.5 введем нормирование на длину суффикса, а **??** – модифицируем так, чтобы нормировать результаты по длине строки.

$$\texttt{score}(\texttt{match}(suffix,ast)) = \frac{\sum_{node \in match} \phi(\frac{f(\texttt{node})}{f(\texttt{node}_{parent})})}{|suffix|}, \tag{2.7}$$

$$\texttt{relevance}(string,text) = \texttt{SCORE}(string,text) = \frac{\sum_{suffix} \texttt{score}(\texttt{match}(suffix,ast))}{|string|} \tag{2.8}$$

В случае линейной ϕ в формуле 2.7 имеет смысл условной вероятности, приходящейся на одну букву суффикса в совпадении `match`. Это делает оценки сравнимыми как по текстам, так и по словосочетаниям. Приведем пример

вычисления оценки релевантности. Оценка релевантности строки “dine” при использовании линейной шкалирующей функции АСД, изображенному на Рис 1.5 равна, согласно вышеприведённому определению:

$$\texttt{relevance}(string,text) = \frac{0 + \texttt{score}(\text{“}ine''{,}ast)/3 + \texttt{score}(\text{“}ne''{,}ast)/2 + 0}{4} =$$

$$= \frac{\phi(\frac{4}{9}) + \phi(\frac{1}{6})}{4} = \frac{4/9 + 1/6}{4} = 0.152$$

2.2.5 Распространение линейных алгоритмов построения суффиксных деревьев на случай АСД

Определенное выше аннотированное суффиксное дерево, строго говоря, не является суффиксным деревом, поскольку не обладает одним из основных свойств суффиксных деревьев, приведенных в [59]. В суффиксном дереве у каждой внутренней вершины, отличной от корня, должно быть не менее двух детей, а в АСД в большом количестве присутствуют цепочки узлов, т.е. узлы с единственным потомком. В [60] показано преобразование АСД, необходимые для выполнения этого условия, а именно схлопывание вершин. Схлопывание вершин заключается в объединение каждой цепи узлов с единственным потомком в одну вершину и переносе пометки на входящее в нее ребро. Метка ребра получается конкатенацией символов, которыми были помечены узлы в цепи. Частота самой вершины остается неизменной, так как у всех вершин в цепи она была одинаковой. Преобразовав таким образом исходное дерево, получим структуру данных, изображенную на Рис. 2.1 (АСД построено для строки ‘mining’).

Реализованное таким образом АСД требует $\Theta(mn^2_{max})$ памяти из-за необходимости хранить все метки ребер в явном виде. Если использовать прием сжатия дуговых меток, описанный в [60] получится снизить объем используемой деревом памяти до линейного относительно длин строк в коллекции. Прием сжатия дуговых меток заключается в том, чтобы хранить в каждом ребре только индексы начала и конца соответствующей подстроки, а не всю подстроку в явном виде. Окончательный вид АСД после всех описанных оптимизаций представлен на Рис. 2.2.

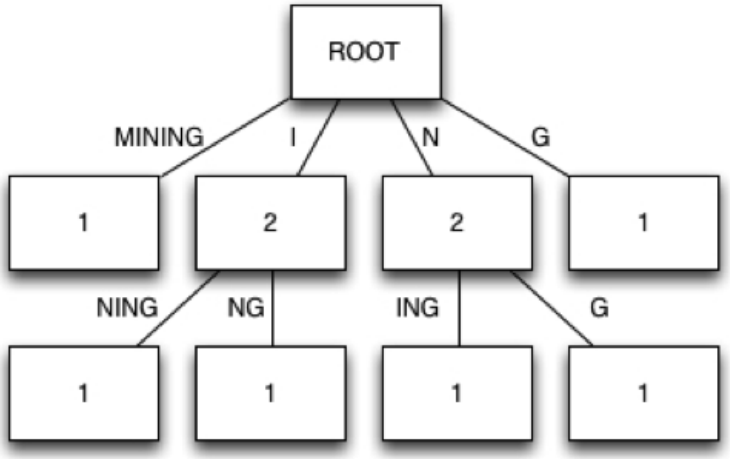

Рисунок 2.1 — Оптимизация представления АСД. Схлопывание вершин для $S =$ ‘‘mining’’

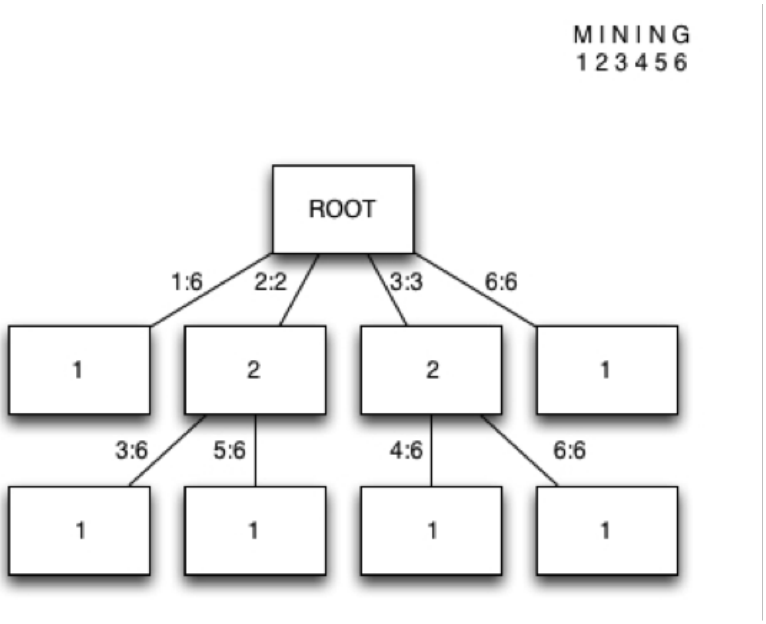

Рисунок 2.2 — Оптимизация представления АСД. Сжатие меток для $S =$ ‘‘mining’’

Значительное снижение объема используемой для хранения АСД памяти позволяет использовать асимптотически менее трудоемкие алгоритмы для построения АСД, чем наивный алгоритм, который был описан выше. Существует целый ряд линейных по времени алгоритмов построения обычных (неаннотированных) суффиксных деревьев. В [60] приведен обзор основных алгоритмов: алгоритмов П. Вайнера (1973), Э. МакКрейга (1976) и Э. Стонена (1995). Использование этих алгоритмов для построение АСД становится возможным благодаря Свойству 1 АСД: сначала построим обычное суффиксное дерево, затем, во время обхода снизу, аннотируем его частотами. После несложной обработки входных строк частоты листьев будут равны 1, а затем частоты узлов на уровнях выше получаются как сумма частот потомков. Предобработка строк заключается в добавлении уникального терминального символа в конец каждой строки. Будем обозначать терминальные символы через $\$_i$. Таким образом, каждая подстрока, соответствующая одному из путей от корня до листа, встречается в исходном наборе строк только один раз. Приведем псевдокод такого алгоритма построения АСД и покажем, что построение АСД осуществляется за линейное время.

Algorithm 4 LinearConstruction(C)

Вход: Коллекция фрагментов $C = s_1, \ldots, s_m$

Выход: Обобщенное АСД для C

1. Построить $C' = s_1\$_1, \ldots, s_m\$_m$
2. Построить обобщенное суффиксное дерево T для коллекции C', используя алгоритм с линейной сложностью.

for $l \in \texttt{leaves}(T)$ **do**

3. **do** присвоить $f(l) \leftarrow 1$

end for

4. Выполнить обход дерева T снизу вверх; в каждом внутреннем узле v присвоить $f(v) \leftarrow \sum_{u \in T: parent(u)=v} f(u)$

Обход дерева требует времени, пропорционального числу вершин. Таким образом, все шаги алгоритма выполняются за линейное время, и общая оценка его трудоемкости составляет $\Theta(n_1 + \cdots + n_m)$ или $O(mn_{max})$.

Заметим, что при представлении АСД, показанном на Рис. 2.2, необходимо видоизменить формулы 2.5 или 2.6:

$$\texttt{score}(\texttt{match}(suffix,ast)) = \sum_{\texttt{node}\in\texttt{match}} \phi(\frac{f(\texttt{node})}{\texttt{node}_{parent}}) + |\texttt{match}| - k \qquad (2.9)$$

или

$$\texttt{score}(\texttt{match}(suffix,ast)) = \frac{\sum_{\texttt{node}\in\texttt{match}} \phi(\frac{f(\texttt{node})}{\texttt{node}_{parent}}) + |\texttt{match}| - k}{|match|}, \qquad (2.10)$$

где $|match|$ – длина фактического совпадения в символах, k – количество узлов в совпадении. Заметим, что при таком преобразования смысл обеих формул останется без изменения. В [60] приведен эксперимент, в ходе которого сравнивается эффективность обоих алгоритмов на стандартных тестовых коллекциях текстов. Показано, что, действительно, линейный алгоритм работает быстрее, чем наивный. Сложность линейного алгоритма по памяти не отличается от сложности наивного алгоритма, поскольку тем или иным образом приходится хранить в памяти одинаковое количество символов и их частот.

2.2.6 Построение таблицы релевантности «Строка – Текст»

С помощью метода nAST-k построим таблицу релевантности «строка – текст» (PCT таблица), в которой строки соответствуют отдельным входным строкам, столбцы – отдельным текстам, а элементы – оценки релевантности строк соответствующим текстам. Основное отличие PCT таблицы от традиционного построения матрица терм-текст, многократно использованного в учебниках [7] и [15], заключается в том, что PCT таблица строится для зафиксированного множества входных строк, а матрица терм–текст строится по термам, извлеченным из текстов. Более того, элементами матрицы терм–текст, как правило, являются частоты, в то время как элементами PCT таблицы являются оценки релевантности. Таким образом, PCT таблицу, построенную для фиксированного множества строк-словосочетаний и фиксированной коллекции текстов, можно считать моделью данной коллекции текстов: каждый текст представляется вектором оценок релевантностей в пространстве строк–словосочетаний. Таблица 1 представляет фрагмент PCT таблицы из [65].

Таблица 1 — Фрагмент PCT таблицы [65]. Столбцы соответствуют публикациям, строки-словосочетаниям, а элементы – оценкам релевантности

	Доклад Всемирного Банка об экономике России	Международные стандарты финансовой отчетности	Если генеральный директор иностранец
Изменение организационно-правовой формы	0.3145	0.3616	0.3644
Изменение уровня концентрации собственности	0.5016	0.3148	0.2706
Повышение эффективности управления затратами	0.4433	0.2351	0.2445
Смена генерального директора	0.2264	0.2351	0.5947

Для построения PCT таблицы с использованием метода nAST-k необходимо проделать следующие шаги:

1. Зафиксировать коллекцию текстов. Для того, что бы последующий анализ данной коллекции имел смысл, требуется сформировать однородную коллекцию текстов одинаковой стилистической и жанровой специфики, принадлежащих к общей предметной области. Например, в [65] предметом анализа была коллекция новостных сообщений о бизнес-процессах в пост-кризисной России в 2009-2010 годах, а в [66] – коллекция аннотаций научных статей по анализу и майнингу данных.
2. Зафиксировать множество входных строк, описывающих основные события, явления и термины в той же предметной области. В [65] словосочетания были сформированы с помощью экспертов и описывали основные события в сфере бизнеса, в [66] в качестве входных строк использовались темы таксономии ACM CCS.
3. Для каждого текста построить собственное АСД, согласно методу описанному выше: каждый текст разбивают на строки из 2-4 слов, все множество строк подают на вход алгоритму построения АСД. В [65] были использованы строки из трех слов, поскольку большая часть строк состояла из трех слов, так что глубина АСД получается близкой к длине словосочетаний, на него накладываемых.
4. На каждое АСД последовательно наложить все строки и получить оценки релевантности. Оценки релевантности сохранить в таблицу, которая и будет искомой PCT таблицей.

Заметим, что, во-первых, строго говоря, PCT таблица не должна храниться в памяти компьютера как таблица. Представление PCT таблица в виде разреженной матрицы [67] вполне допустимо и оправдано с технической точки зрения: значения оценок релевантности часто не превосходят 0. Во-вторых, для построения PCT таблицы может быть использована любая другая мера релевантности. Однако, СУВСС (мера релевантности, основанная на АСД с линейной шкалирующей функцией), обладает некоторыми преимуществами. Она учитывает все нечеткие совпадения строки с текстом и дает им количественную оценку. Другие меры релевантности, в том числе, описанные выше, учитывают исключительно четкие совпадения между отдельными словами, составляющими словосочетание, и не могут дать оценку целой строке. Заметим, так же, что в отличии от остальных мер релеватности, СУВСС не содержит ни прямой, ни обратной документной частоты (IDF), не связана с размером коллекции и при вычислении релевантности строки тексту не использует оценки, получаемые для других текстов.

2.3 Выводы по главе

Эта глава посвящена проблеме оценивания релевантности строки тексту. В первой части главы приведен обзор основных мер релевантности строки тексту и их теоретических обоснований: меры релевантности в векторной, вероятностной, языковой и тематических моделях. Все перечисленные методы обладают несколькими общими свойствами: например, они учитывают только четкие совпадения строки с текстом. Использование теоретико-множественной модели преставления текстов и подхода, основанного на аннотированных суффиксных деревьях [5] (АСД), преодолевает эту проблему и позволяет учитывать и нечеткие совпадения строки с текстом. Нами предложена такая оценка релевантности, которая имеет чёткий операциональный смысл – суммарной условной вероятности символа в совпавшем фрагменте (СУВСС). Во второй части главы представлен метод оценивания релеантности строки тексту nAST-k, являющийся модификацией метода СУВСС. Этот метод учитывает такие параметры АСД, как глубина и разброс, что позволяет нормировать оценки. Рассмотрены два алгоритма построения АСД: наивный алгоритм, имеющий квадратичную

оценку сложности по времени, и линейный алгоритм, имеющий соответственно линейную оценку сложности по времени. Оба алгоритма не отличаются по сложности по памяти. Показано, что меру релевантности строки тексту, получаемую по методу nAST-k можно использовать для построения таблиц релевантности «строка – текст».

Глава 3. Задача рубрикации научных статей темами из заданного списка

Задача рубрикации научных статей относится к задачам категоризации текстов [16]. Общая постановка задачи категоризации текстов такова: для заданной коллекции текстов и заданного множества категорий, представленных текстовыми метками, требуется каждому тексту приписать релевантные ему категории. При этом число категорий заведомо не меньше двух. Задача рубрикации научных статей заключается в категоризации статей в системе рубрик, заданных классификатором или таксономией соответствующей области знания или технологии. Под таксономией понимается дерево тем: чем выше тема в дереве, тем более общей она является. В таком дереве родитель и потомки находятся в отношении «целое – часть» или «быть более общим». Например, англоязычные статьи в из области информатики и вычислительной техники могут индексироваться темами так называемой Computing Classification System – таксономии, разработанной международной Ассоциацией вычислительной техники, (Association for Computing Machinery (ACM)), русскоязычные публикации – рубриками государственного рубрикатора научно-технической информации. ACM CCS представляет собой иерархическую систему, в которой каждая тема является частью более общей темы и сама, в свою очередь, делится на более конкретные темы. Например, согласно ACM CCS, “майнинг данных” [data mining] – это часть “приложений информационных систем” [information system application], в свою очередь, содержащая такие темы как “кластерный анализ” [cluster analysis] и “ассоциативные правила” [associative rules]. Существует два основных подхода к решению задачи категоризации текстов: первый основан на использовании методов с учителем, второй – без учителя [16]. В работе [68] приводятся обзор и результаты экспериментального сравнения методов обучения с учителем для задачи рубрикации текстов, в которых категории образуют иерархическую систему, а в работе [69] подобный метод предлагается применительно к таксономии ACM CCS. Один из способов решения задачи категоризации в режиме без учителя основан на вычислении оценок релевантности категорий текстам и построении РСТ таблицы категория– текст. Из построенной РСТ таблицы выделяют для каждого текста категории, получившие наивысшие оценки релевантности. Выше мы перечислили несколько основных мер релевантности

строки тексту и подробно описали отдельно стоящую меру релевантности строки тексту, основанную на АСД. Теперь мы экспериментально сравним три из них: косинусную меру релевантности (мера релевантности в векторной модели), меру релевантности в вероятностной модели и меру релевантности, основанную на АСД. В качестве входных данных мы используем аннотации статей, опубликованных некоторыми журналами, издаваемыми вышеупомянутой Ассоциацией вычислительной техники ACM. Авторы статей в этих журналах сами выполняли рубрикацию своих статей с помощью тем таксономии ACM CCS. Мы постарались включить в эксперимент все наиболее популярные способы предобработки текстов. Для оценки результатов рубрикации мы используем два популярных способа оценки, которые по-разному обобщают оценки точности и полноты, используемые для оценки результатов в традиционных задачах классификации, а также предложили ещё одну, в некоторых отношениях более адекватную меру.

3.1 Метод рубрикации AnnAST

Метод рубрикации AnnAST получает на вход систему рубрик и коллекцию текстов. Рубрикация текста заключается в приписывании ему наиболее подходящих рубрик. Такие рубрики определяются по оценкам релевантности: требуется найти рубрики с наибольшими оценками релевантности тексту. Ограничения метода AnnAST заключаются в том, что каждая рубрика должна быть задана одной уникальной строкой.

3.2 Экспериментальная верификация метода AnnAST

3.2.1 Постановка эксперимента

Для того, чтобы поставить вычислительный эксперимент по сравнению относительных преимуществ использования различных мер релевантности в

проблеме рубрикации научных публикаций, надо определить три основных составляющих такого эксперимента:

- набор данных, на которых производится сравнение;
- набор мер релевантности, участвующих в сравнении;
- способ оценки качества результатов, получаемых при использовании того или иного метода.

Эти составляющие описаны в нижеследующих разделах. В качестве дополнительно параметра для экспериментирования мы рассматривали различные способы представления текстов.

Выбор данных

Данные взяты из электронной библиотеки ACM Digital Library. В этой библиотеке хранятся архивы журналов ACM. В свободном доступе находятся аннотации большей части научных статей и вспомогательные сведения, такие как ключевые слова и таксономические темы таксономии ACM CCS, приписанные авторами к научным статьям для рубрикации статей в библиотеке, т.н. авторские темы. Задача заключается в том, чтобы подобрать к каждой научной статье несколько наиболее релевантных таксономических тем. Эксперимент проводился для коллекции данных, состоящей из трех частей: аннотаций научных статей, таксономии ACM CCS 2012, а также приписанных статьям их авторами тем из этой таксономии (см. Рис. 3.1). Эти части кратко представлены ниже.

- Аннотации всех научных статей, опубликованных за период с начала 2007 года по первый квартал 2013 года включительно, во всех журналах, размещённых на портале ACM. Общее число аннотаций в данной коллекции – 5079;
- Таксономия ACM CCS 2012, состоящая из 2074 таксономических тем. В таксономии ACM CCS 2012 6 уровней. На первом уровне располагается 13 основных разделов, на втором уровне – 90, на третьем – 547, на четвертом уровне находится большая часть листьев таксономии – 1074 тем. Некоторые из листьев появляются при дальнейшем дроблении тем – на пятом и шестом уровнях; но их сравнительно немного – 326 и 24.

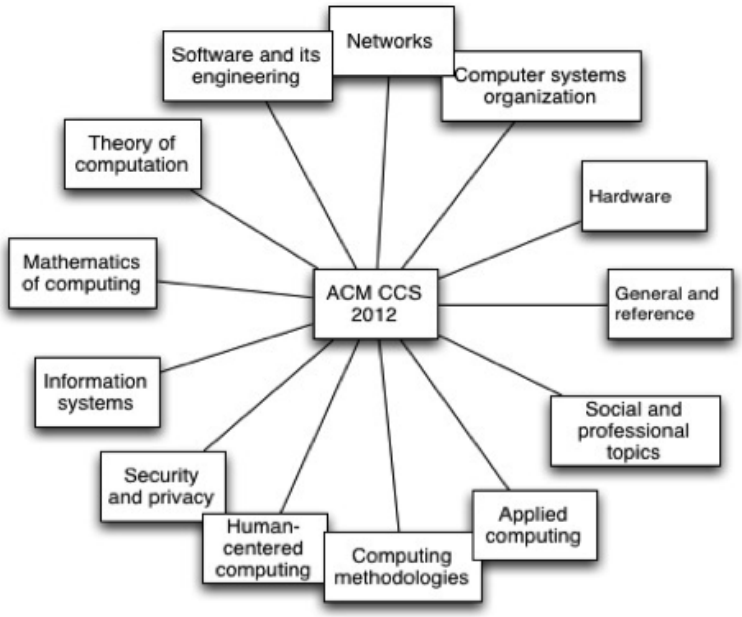

Рисунок 3.1 — Первый уровень таксономии ACM CCS 2012

– Авторские темы, приписанные аннотациям – это, как правило, 2-3 таксономические темы низших уровней таксономии, а также все темы, лежащие на пути от корня до них в дереве таксономии ACM CCS. Пример аннотации из рассматриваемого множества приведён в Таблице .

Отметим, что авторы статьи, представленной в Таблице 2, в своей рубрикации предпочли оттенить взаимодействие человека и компьютера как её основной сюжет. На наш взгляд, это не очень согласуется с содержанием аннотации. Согласно аннотации, статья представляет собой скорее упражнение в применении вероятностной модели кластер-анализа для выявления сообществ на основе информации с сайтов, пользователи которых задают вопросы и получают ответы от других пользователей. Термины “cluster” и “clustering” 6 раз участвуют в различных подразделениях таксономии ACM CCS, но никак не отражены в авторской рубрикации. Подобные нюансы интерпретации должны учитываться при оценке систем автоматической рубрикации.

Выбор мер релевантности

В качестве мер оценки релевантности таксономической темы и аннотации научной статьи, берутся популярные меры, рассмотренные выше:

Таблица 2 — Пример аннотации, участвующей в эксперименте. Аннотация выбрана случайным образом.

Discovering Knowledge-Sharing Communities in Question-Answering Forums
Mohamed Bouguessa, Shengrui Wang, Benoit Dumoulin
ACM Transactions on Knowledge Discovery from Data (TKDD),V. 5, no.1, December 2010
In this article, we define a knowledge-sharing community in a question-answering forum as a set of askers and authoritative users such that, within each community, askers exhibit more homogeneous behavior in terms of their interactions with authoritative users than elsewhere. A procedure for discovering members of such a community is devised. As a case study, we focus on Yahoo!Answers, a large and diverse online question-answering service. Our contribution is twofold. First, we propose a method for automatic identification of authoritative actors in Yahoo!Answers. To this end, we estimate and then model the authority scores of participants as a mixture of gamma distributions. The number of components in the mixture is determined using the Bayesian Information Criterion (BIC), while the parameters of each component are estimated using the Expectation-Maximization (EM) algorithm. This method allows us to automatically discriminate between authoritative and nonauthoritative users. Second, we represent the forum environment as a type of transactional data such that each transaction summarizes the interaction of an asker with a specific set of authoritative users. Then, to group askers on the basis of their interactions with authoritative users, we propose a parameter-free transaction data clustering algorithm which is based on a novel criterion function. The identified clusters correspond to the communities that we aim to discover. To evaluate the suitability of our clustering algorithm, we conduct a series of experiments on both synthetic data and public real-life data. Finally, we put our approach to work using data from Yahoo!Answers which represent users activities over one full year.
Таксономические темы ACM CCS, приписанные автором (авторские темы)
Human-centered computing → Human computer interaction (HCI) → Interaction paradigms → Web-based interaction Information systems → Information systems applications → Data mining

– косинусная мера близости на векторах tf-idf в векторной модели релевантности;
– косинусная мера близости на векторах tf-idf в векторной модели релевантности со снижением размерности по методу латентного семантического анализа;
– мера релевантности, основанная на генеративной модели релевантности латентного размещения Дирихле;
– мера релевантности BM25 в вероятностной модели релевантности;
– коэффициент Жаккара на множестве буквенных n-грамм;
– мера средней условной вероятности символа в совпадении, основанная на АСД, с семью шкалирующими функциями, СУВСС. Эти меры перечислены в Таблице 3.

Оценка качества результатов

Зная авторские таксономические темы, можно оценить, насколько корректными получились упорядоченные списки таксономических тем, получаемые в результате применения той или иной меры релевантности. Мы использовали для оценки результатов две популярные характеристики точности: `MAP` (Mean Average Precision) и `nDCG` (normalized discounted cumulative gain) [15]. Они часто используются в тех вычислительных экспериментах и разработках, в которых возникает задача оценки качества ранжирований, например, при разработке рекомендательных систем [70] или систем извлечения новостей [71]. Другое приложение этим мерам находится в работах по обучению ранжированию (learning to rank) [72; 73]. В этих работах `MAP` и `nDCG` используются в качестве оптимизируемого критерия в ходе обучения. Меры `MAP` и `nDCG` применимы и для задачи рубрикации, так как результаты использования той или иной меры релевантности тоже ранжированы по значению этой меры. Для вычисления этих характеристик может использоваться следующая общая схема отбора таксономических тем:

1. Строится РСТ таблица таксономическая тема – аннотация;
2. Таксономические темы ранжируются по убыванию их оценок релевантности каждой аннотации;

Таблица 3 — Обозначения рассматриваемых мер релевантности

Обозначение	Мера релевантности
cosine	Косинусная мера релевантности
LSI.N	Косинусная мера релевантности со снижением до N размерностей методом LSI
LDA.N	Мера релевантности, основанная на ЛРД с N темами
okapibm25	Мера релевантности BM25
Jaccard	коэффициент Жаккара на множестве буквенных n-грамм
AST.constant.X	мера СУВСС с константной шкалирующей функцией и очисткой шума от уровня X
AST.linear.X	мера СУВСС с линейной шкалирующей функцией и очисткой шума от уровня X
AST.square.X	мера СУВСС с шкалирующей квадратичной функцией и очисткой шума от уровня X
AST.root.X	мера СУВСС с линейной шкалирующей функцией корень квадратный и очисткой шума от уровня X
AST.log.X	мера СУВСС с логарифмической шкалирующей функцией и очисткой шума от уровня X
AST.logit.X	мера СУВСС с логистической шкалирующей функцией и очисткой шума от уровня X
AST.sigmoid.X	мера СУВСС с шкалирующей функцией сигмоид и очисткой шума от уровня X

3. Отбираются первые k (топ k) таксономические темы, отсекая все остальные;
4. Вычисляется оценка получившегося ранжирования;

Мера MAP может быть представлена следующим образом:

$$\texttt{AveP} = \frac{\sum_k^n P(k)\texttt{rel}(k)}{\texttt{relevant topics}|},$$

$$\texttt{MAP} = \frac{\sum_{a \in abstracts} \texttt{AveP}(a)}{|\texttt{abstracts}|},$$

где $P(k)$ – точность на уровне k в упорядоченном по убыванию меры релевантности списке таксономических тем, rel – бинарный показатель, принимающий значение 1, если k-тая таксономическая тема в списке является авторской, и 0 в обратном случае, |relevant topics| – число авторских таксономических тем, n – количество рассматриваемых таксономических единиц из топа списка. Здесь AveP – средняя точность – рассчитывается для каждого текста рассматриваемого множества. Мера MAP имеет смысл средней точности, нормализованной по всем аннотациям.

Мера nDCG – это отношение оценки полученного ранжирования к оценке идеального случая: $\texttt{nDCG}_k = \frac{\texttt{DCG}_k}{\texttt{IDCG}_k}$, где $\texttt{DCG}_k = rel(1) + \sum_{i=2}^{k} \frac{\texttt{rel}(i)}{\log_2 i}$ – количество авторских таксономических тем среди топ k таксономических тем, нормированное на их место в ранжировании, $\texttt{IDCG}_k = \texttt{rel}(1) + \sum_{i}^{|\texttt{relevant topics}|} \frac{1}{\log_2 i}$ – значение DCG у идеального ранжирования. Как предложено в [73] меру nDCG что мы и делаем, чтобы получить общее значение nDCG для всех рассматриваемых научных статей.

Помимо мер MAP и nDCG, мы использовали собственные меры оценки полученных результатов, имеющие простой операциональный смысл – доля публикаций, у которых авторские темы попали в топ k k ранжированных таксономических тем и доля авторских тем, попавших в топ k k ранжированных таксономических тем. Будем обозначать эти меру через I_k (Intersection at k) и J_k.

Данная мера удобна тем, что позволяет легко

– отделить “хорошие” публикации – те, для которых удалось восстановить все или почти все авторские темы – от “трудных”, для которых авторские темы находятся в конце соответствующего ранжирования;

Таблица 4 — Способы представления текста как “мешка” термов

Обозначение	Описание
words	Все вхождения слов в неизменённом виде
stems	Стемы (основы) всех слов. Для выделения стемов использован стеммер Портера [74] из библиотеки NLTK [75]
lemmas	Леммы (нормальные формы) всех слов. Для выделения лемм использован лемматизатор на основе словарей WordNet [76] из библиотеки NLTK [75]
3gram	Буквенные 3-граммы
4gram	Буквенные 4-граммы
5gram	Буквенные 5-граммы
6gram	Буквенные 6-граммы

– определить оптимальный порог отсечения k, т. е. число таксономических тем, которые могут быть использованы для рубрикации конкретной публикации.

В принципе, меры `MAP` и `nDCG` тоже позволяют устанавливать пороговые значения, но они имеют значительно менее интуитивных характер, чем пороговые значения, которые определяются мерой I_k.

Способы представления текстов

Использование векторной и вероятностных моделей предполагает представление текста в виде неупорядоченного набора термов. Под термами понимаются либо слова в исходном виде, либо некоторые значимые фрагменты слов, как правило, основы, часто называемые стемами, либо же словарные формы (леммы) слов [16]. Кроме этих традиционных способов представления текста, мы рассматривали представление текста в виде множеств буквенных n–грамм. Полный список рассмотреных способов представления текстов и их обозначения приведены в Таблице 4.

3.2.2 Схема эксперимента

Эксперимент осуществляется в несколько этапов:

1. Предобработка текстов: представление текстов в виде мешков слов 7 способами;
2. Использование мер релевантности cosine, LSI.15, LSI.50. LSI.100, LSI.150, LDA.15, LDA.50, LDA.100, LDA.150, BM25, Jaccard;
3. Использование 7 шкалирующих функций для АСД;
4. Отсутствие очистки АСД от шума, очистка АСД от шума с 1, 2 уровня;
5. Построение PCT таблиц таксономическая тема – аннотация. В общей сложности, было построено 84 PCT таблиц с учетом всех возможных параметров.

3.2.3 Результаты эксперимента

При использовании косинусной меры релевантности для аннотирования научных публикаций и с учетом всех способов представления текстов, оказалось, что представление текстов в виде множества n–грам существенно эффективнее, то есть, точнее, по всем рассмотреным мерам качества, чем представление текстов в виде множеств слов в исходной форме, стемов слов или их лемм. При этом, оптимальное n на основе данного эксперимента выбрать не получается: результаты полученные при $n = 5$ и $n = 6$ несравнимы (так, например, $\texttt{MAP}_5$ при $n = 5$ превосходит $\texttt{MAP}_5$ при $n = 6$, однако $\texttt{MAP}_{15}$ при $n = 6$ превосходит $\texttt{MAP}_{15}$ при $n = 5$), но на 1-2% точнее чем результаты, полученные при $n = 3$ и $n = 4$. Отметим также, что при использовании лемматизации точность полученных результатов превосходит результаты, получаемые после стемминга.

Использование ЛСИ для снижения числа размерностей в векторной модели подверждает полученные резульаты: в частности, представление текста в виде множества 5–грам или 6–грам позволяет достичь большой точности, чем представление текста в виде множества 3–грам или 4–грам, а использование лемматизации – чем использование стемминга. В целом, представление текста в виде множества n–грам приводит к лучшим результатам, чем представление

в виде множества стемов или лемм. Однако, использование ЛСИ для снижения размерности не позволяет улучшить показатели точности по сравнению с оригинальной векторной моделью: лучшие результаты, получаемые при $N = 150$ хуже результатов оригинальной модели в среднем на 10%.

Аналогично и использование ЛРД с различным числом скрытых тем не позволяет хоть каких-либо улучшений, но и опровергает превосходство лемматизации перед стеммингом. В прочем, полученные при использовании ЛРД показатели точности настолько малы, что не позволяют сделать основательных выводов.

Вероятностная мера релевантности Okapi BM 25 позволяет получить результаты не превосходящие косинусную модель релеватности, однако подверждающие превосходство представления текстов в виде n-грамм над лемматизацией и стеммингом и лемматизации над стеммингом. При использовании этой меры релевантности оптимальным является $n = 6$. Использование теоретико-множественной меры Жаккара в качестве меры релевантности подветрждает эти наблюдения.

Использование мер релеватности, основанных на АСД с различными шкалирующими функциями и очисткой от шума на разных уровнях, позволяет существенным образом превзойти результаты, полученные при использовании косинусной меры релевантности (на 5-6%). При этом, полученные результаты очевидным образом демонстрируют нецелесобразность использования логарифмической и логистической функции в качестве шкалирующей функции. Точность всех остальных шкалирующих функций сопоставима, из них лучшей можно назвать шкалирующую функцию сигмоид, а худшей – квадратичную. Так же, отметим, что, в принципе, подвержается гипотеза об эффективности очистки от шма на 1 уровне, то есть, на уровне одиночных букв, и не оправдывает себя очистка от шума на втором уровне – уровне пар букв.

Несмотря на то, что в целом показатели точности невелики и не превосходят 30%, можно подвести общие итоги: при использовании таких четких мер релевантности, как алгебраические (косинусная мера), вероятностные (Okapi BM 25), теоретико-множественные (мера близости Жаккара) меры, представление текста в виде множества n-грамм существенно эффективнее, чем представление текста в виде множества лемм или стемов. Оптимальным при этом является $n = 5{,}6$. Лемматизация по сравнению со стеммингом позволяет достичь лучших результатов. Использование нечетких мер релевантности, основанных

Таблица 5 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью косинусной меры релевантности

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.1775	0.2073	0.2242	0.0478	0.1073	0.1770	0.0570	0.0930	0.1269	0.2598	0.3783	0.4641
stems	0.1874	0.2206	0.2368	0.0482	0.1146	0.1806	0.0589	0.0990	0.1312	0.2587	0.3826	0.4772
lemmas	0.1970	0.2302	0.2464	0.0478	0.1141	0.1806	0.0599	0.1000	0.1324	0.2750	0.3967	0.4859
3gram	0.2040	0.2340	0.2478	0.0511	0.1117	0.1673	0.0639	0.1004	0.1276	0.2804	0.3870	0.4707
4gram	0.2202	0.2569	0.2733	0.0516	0.1242	0.1914	0.0663	0.1103	0.1430	0.3000	0.4337	0.5185
5gram	0.2297	0.2663	0.2822	0.0551	0.1282	0.1928	0.0699	0.1141	0.1456	0.3109	0.4489	0.5261
6gram	0.2307	0.2659	0.2788	0.0549	0.1256	0.1791	0.0697	0.1123	0.1384	0.3207	0.4359	0.5033

Таблица 6 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью косинусной меры релевантности и ЛСИ для снижения размерности векторной модели до $N = 15$ размерностей

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0251	0.0312	0.0356	0.0081	0.0198	0.0383	0.0087	0.0159	0.0248	0.0533	0.0891	0.1293
stems	0.0379	0.0468	0.0533	0.0120	0.0299	0.0572	0.0128	0.0236	0.0368	0.0739	0.1207	0.1707
lemmas	0.0364	0.0449	0.0501	0.0123	0.0289	0.0499	0.0125	0.0226	0.0329	0.0717	0.1109	0.1620
3gram	0.0281	0.0347	0.0382	0.0058	0.0196	0.0345	0.0082	0.0164	0.0236	0.0522	0.0946	0.1293
4gram	0.0316	0.0400	0.0441	0.0072	0.0252	0.0421	0.0094	0.0200	0.0283	0.0587	0.1109	0.1467
5gram	0.0406	0.0477	0.0536	0.0110	0.0247	0.0489	0.0128	0.0212	0.0331	0.0804	0.1272	0.1880
6gram	0.0441	0.0554	0.0600	0.0104	0.0331	0.0517	0.0135	0.0272	0.0363	0.0848	0.1500	0.1924

Таблица 7 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью косинусной меры релевантности и ЛСИ для снижения размерности векторной модели до $N = 50$ размерностей

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0299	0.0365	0.0431	0.0092	0.0233	0.0509	0.0104	0.0188	0.0322	0.0554	0.1011	0.1685
stems	0.0469	0.0555	0.0608	0.0123	0.0301	0.0533	0.0147	0.0253	0.0365	0.0826	0.1293	0.1783
lemmas	0.0329	0.0418	0.0488	0.0104	0.0286	0.0580	0.0115	0.0224	0.0367	0.0663	0.1239	0.1957
3gram	0.0395	0.0486	0.0528	0.0072	0.0253	0.0421	0.0106	0.0216	0.0298	0.0598	0.1174	0.1565
4gram	0.0682	0.0790	0.0847	0.0137	0.0357	0.0602	0.0188	0.0320	0.0438	0.1065	0.1641	0.2087
5gram	0.0951	0.1121	0.1195	0.0235	0.0571	0.0884	0.0295	0.0498	0.0649	0.1609	0.2489	0.3011
6gram	0.1130	0.1338	0.1422	0.0320	0.0734	0.1082	0.0375	0.0625	0.0795	0.1946	0.2902	0.3391

Таблица 8 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью косинусной меры релевантности и ЛСИ для снижения размерности векторной модели до $N = 100$ размерностей

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0267	0.0330	0.0398	0.0066	0.0190	0.0477	0.0084	0.0159	0.0298	0.0457	0.0837	0.1554
stems	0.0501	0.0598	0.0658	0.0151	0.0351	0.0603	0.0171	0.0291	0.0413	0.0891	0.1467	0.2022
lemmas	0.0409	0.0514	0.0596	0.0113	0.0328	0.0665	0.0132	0.0260	0.0425	0.0728	0.1391	0.2098
3gram	0.0590	0.0686	0.0741	0.0158	0.0356	0.0584	0.0188	0.0307	0.0418	0.1054	0.1630	0.2033
4gram	0.0968	0.1116	0.1183	0.0222	0.0521	0.0795	0.0288	0.0468	0.0601	0.1609	0.2337	0.2815
5gram	0.1349	0.1527	0.1604	0.0340	0.0705	0.1022	0.0420	0.0639	0.0793	0.2228	0.2967	0.3478
6gram	0.1503	0.1708	0.1799	0.0350	0.0757	0.1130	0.0450	0.0697	0.0879	0.2359	0.3174	0.3685

Таблица 9 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью косинусной меры релевантности и ЛСИ для снижения размерности векторной модели до $N = 150$ размерностей

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0303	0.0377	0.0426	0.0070	0.0225	0.0431	0.0091	0.0183	0.0283	0.0478	0.0935	0.1500
stems	0.0590	0.0714	0.0780	0.0186	0.0432	0.0708	0.0206	0.0355	0.0488	0.1054	0.1674	0.2141
lemmas	0.0451	0.0562	0.0619	0.0130	0.0364	0.0604	0.0149	0.0288	0.0404	0.0783	0.1457	0.1989
3gram	0.0797	0.0922	0.0985	0.0197	0.0438	0.0708	0.0247	0.0394	0.0524	0.1304	0.1935	0.2424
4gram	0.1278	0.1412	0.1494	0.0337	0.0599	0.0937	0.0406	0.0565	0.0730	0.2109	0.2630	0.3120
5gram	0.1511	0.1713	0.1802	0.0389	0.0782	0.1162	0.0473	0.0712	0.0896	0.2413	0.3250	0.3652
6gram	0.1575	0.1826	0.1909	0.0335	0.0815	0.1147	0.0449	0.0743	0.0906	0.2326	0.3304	0.3761

Таблица 10 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью меры релевантности, основанной на ЛРД, при числе скрытых тем $N = 15$

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0118	0.0132	0.0140	0.0043	0.0072	0.0104	0.0045	0.0062	0.0077	0.0217	0.0315	0.0413
stems	0.0153	0.0191	0.0219	0.0051	0.0137	0.0258	0.0055	0.0104	0.0163	0.0348	0.0609	0.0957
lemmas	0.0104	0.0126	0.0139	0.0052	0.0102	0.0159	0.0051	0.0080	0.0108	0.0326	0.0500	0.0663
3gram	0.0029	0.0048	0.0056	0.0015	0.0055	0.0090	0.0014	0.0038	0.0055	0.0087	0.0228	0.0337
4gram	0.0066	0.0083	0.0094	0.0024	0.0057	0.0103	0.0024	0.0045	0.0067	0.0141	0.0261	0.0402
5gram	0.0163	0.0180	0.0192	0.0033	0.0072	0.0126	0.0046	0.0069	0.0094	0.0293	0.0413	0.0554
6gram	0.0168	0.0179	0.0187	0.0016	0.0039	0.0070	0.0036	0.0050	0.0065	0.0228	0.0293	0.0391

Таблица 11 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью меры релевантности, основанной на ЛРД, при числе скрытых тем $N = 50$

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0172	0.0213	0.0231	0.0043	0.0129	0.0129	0.0053	0.0104	0.0139	0.0315	0.0620	0.0761
stems	0.0220	0.0264	0.0275	0.0055	0.0143	0.0194	0.0069	0.0122	0.0146	0.0348	0.0576	0.0707
lemmas	0.0099	0.0143	0.0165	0.0028	0.0116	0.0209	0.0033	0.0086	0.0130	0.0185	0.0467	0.0685
3gram	0.0111	0.0115	0.0123	0.0017	0.0026	0.0062	0.0027	0.0033	0.0050	0.0174	0.0207	0.0315
4gram	0.0079	0.0114	0.0134	0.0018	0.0094	0.0183	0.0024	0.0069	0.0111	0.0141	0.0348	0.0587
5gram	0.0071	0.0112	0.0127	0.0020	0.0094	0.0158	0.0024	0.0070	0.0101	0.0152	0.0424	0.0576
6gram	0.0136	0.0192	0.0220	0.0048	0.0155	0.0270	0.0051	0.0116	0.0173	0.0304	0.0663	0.0989

Таблица 12 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью меры релевантности, основанной на ЛРД, при числе скрытых тем $N = 100$

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0130	0.0158	0.0173	0.0050	0.0116	0.0179	0.0051	0.0089	0.0120	0.0293	0.0522	0.0685
stems	0.0112	0.0143	0.0168	0.0022	0.0080	0.0183	0.0031	0.0067	0.0116	0.0174	0.0370	0.0620
lemmas	0.0056	0.0084	0.0097	0.0027	0.0080	0.0133	0.0026	0.0058	0.0084	0.0163	0.0348	0.0467
3gram	0.0128	0.0164	0.0192	0.0039	0.0113	0.0230	0.0043	0.0087	0.0144	0.0272	0.0500	0.0696
4gram	0.0117	0.0152	0.0171	0.0043	0.0110	0.0183	0.0045	0.0086	0.0122	0.0261	0.0435	0.0620
5gram	0.0126	0.0159	0.0175	0.0034	0.0102	0.0165	0.0039	0.0080	0.0111	0.0250	0.0467	0.0641
6gram	0.0204	0.0266	0.0289	0.0045	0.0175	0.0270	0.0058	0.0135	0.0182	0.0359	0.0793	0.1000

Таблица 13 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью меры релевантности, основанной на ЛРД, при числе скрытых тем $N = 150$

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0143	0.0224	0.0251	0.0057	0.0218	0.0327	0.0058	0.0156	0.0209	0.0293	0.0804	0.0989
stems	0.0200	0.0256	0.0286	0.0045	0.0147	0.0275	0.0060	0.0123	0.0185	0.0337	0.0630	0.0957
lemmas	0.0143	0.0214	0.0260	0.0047	0.0188	0.0368	0.0053	0.0139	0.0228	0.0283	0.0772	0.1109
3gram	0.0079	0.0099	0.0115	0.0017	0.0057	0.0119	0.0024	0.0048	0.0079	0.0152	0.0293	0.0446
4gram	0.0082	0.0115	0.0149	0.0035	0.0096	0.0237	0.0034	0.0072	0.0140	0.0174	0.0402	0.0772
5gram	0.0141	0.0190	0.0207	0.0048	0.0147	0.0223	0.0051	0.0111	0.0147	0.0283	0.0565	0.0783
6gram	0.0200	0.0250	0.0268	0.0057	0.0156	0.0229	0.0067	0.0127	0.0163	0.0402	0.0728	0.0935

Таблица 14 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью вероятностной меры релевантности BM25.

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.0294	0.0372	0.0423	0.0062	0.0222	0.0439	0.0084	0.0180	0.0284	0.0489	0.0946	0.1467
stems	0.0602	0.0724	0.0789	0.0185	0.0442	0.0709	0.0207	0.0360	0.0490	0.1065	0.1696	0.2304
lemmas	0.0455	0.0556	0.0629	0.0127	0.0340	0.0643	0.0149	0.0276	0.0423	0.0783	0.1413	0.2087
3gram	0.0816	0.0943	0.1006	0.0188	0.0436	0.0697	0.0243	0.0394	0.0521	0.1304	0.1967	0.2402
4gram	0.1247	0.1407	0.1489	0.0309	0.0613	0.0961	0.0382	0.0569	0.0736	0.2043	0.2728	0.3141
5gram	0.1505	0.1718	0.1806	0.0373	0.0795	0.1162	0.0464	0.0719	0.0897	0.2402	0.3272	0.3707
6gram	0.1643	0.1866	0.1953	0.0342	0.0779	0.1142	0.0466	0.0731	0.0908	0.2380	0.3293	0.3783

Таблица 15 — Оценка полученных при использовании различных способов предобработки текстов результатов с помощью теоретико-множественного коэффицента Жаккара

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
words	0.1794	0.2115	0.2264	0.0414	0.1059	0.1681	0.0531	0.0920	0.1221	0.2696	0.3924	0.4772
stems	0.1805	0.2113	0.2280	0.0409	0.1020	0.1705	0.0533	0.0903	0.1237	0.2630	0.3978	0.4880
lemmas	0.2023	0.2375	0.2535	0.0415	0.1118	0.1776	0.0567	0.0992	0.1312	0.2707	0.4272	0.4989
3gram	0.0798	0.0841	0.0115	0.0307	0.0483	0.0483	0.0180	0.0296	0.0382	0.1978	0.2989	0.3913
4gram	0.1182	0.1404	0.1531	0.0310	0.0748	0.1273	0.0372	0.0637	0.0892	0.1978	0.2989	0.3913
5gram	0.1693	0.2000	0.2133	0.0464	0.1077	0.1612	0.0548	0.0918	0.1180	0.3120	0.4120	0.5130
6gram	0.2116	0.2408	0.2548	0.0512	0.1084	0.1666	0.0644	0.0992	0.1274	0.3120	0.4120	0.5130

Таблица 16 — Оценка полученных результатов при использовании меры релевантности, основанной на АСД, при использовании различных видов шкалирующих функций и очистки от шума на различных уровнях

	$\texttt{MAP}_5$	$\texttt{MAP}_{10}$	$\texttt{MAP}_{15}$	$\texttt{nDCG}_5$	$\texttt{nDCG}_{10}$	$\texttt{nDCG}_{15}$	I_5	I_{10}	I_{15}	J_5	J_{10}	J_{15}
constant.0	0.2874	0.3247	0.3395	0.0574	0.1307	0.1930	0.0795	0.1240	0.1541	0.3576	0.4946	0.5739
constant.1	0.2869	0.3218	0.3373	0.0584	0.1273	0.1919	0.0800	0.1218	0.1531	0.3674	0.4859	0.5663
constant.2	0.2814	0.3154	0.3291	0.0544	0.1204	0.1768	0.0764	0.1166	0.1440	0.3533	0.4707	0.5402
linear.0	0.2734	0.3071	0.3221	0.0508	0.1162	0.1786	0.0735	0.1134	0.1437	0.3489	0.4598	0.5337
linear.1	0.2742	0.3075	0.3233	0.0500	0.1142	0.1793	0.0731	0.1123	0.1440	0.3500	0.4609	0.5413
linear.2	0.2696	0.2997	0.3127	0.0530	0.1135	0.1678	0.0740	0.1105	0.1368	0.3489	0.4500	0.5185
square.0	0.2570	0.2887	0.3036	0.0473	0.1092	0.1709	0.0687	0.1064	0.1363	0.3326	0.4446	0.5217
square.1	0.2584	0.2897	0.3043	0.0481	0.1095	0.1716	0.0695	0.1069	0.1368	0.3348	0.4467	0.5207
square.2	0.2610	0.2886	0.3016	0.0515	0.1077	0.1612	0.0718	0.1055	0.1315	0.3391	0.4391	0.5065
root.0	0.2854	0.3226	0.3369	0.0534	0.1268	0.1870	0.0772	0.1218	0.1509	0.3565	0.4848	0.5598
root.1	0.2826	0.3170	0.3324	0.0548	0.1221	0.1868	0.0774	0.1183	0.1497	0.3609	0.4761	0.5554
root.2	0.2747	0.3056	0.3186	0.0560	0.1183	0.1727	0.0766	0.1141	0.1404	0.3576	0.4598	0.5272
log.0	0.0161	0.0263	0.0317	0.0079	0.0292	0.0514	0.0074	0.0200	0.0308	0.0391	0.1065	0.1554
log.1	0.0011	0.0011	0.0012	0.0000	0.0000	0.0007	0.0002	0.0002	0.0005	0.0011	0.0011	0.0033
log.2	0.0025	0.0033	0.0035	0.0007	0.0021	0.0028	0.0009	0.0017	0.0021	0.0054	0.0109	0.0130
logit.0	0.0497	0.0630	0.0703	0.0161	0.0428	0.0737	0.0176	0.0337	0.0486	0.0913	0.1674	0.2359
logit.1	0.0255	0.0288	0.0303	0.0052	0.0121	0.0189	0.0072	0.0113	0.0146	0.0413	0.0598	0.0717
logit.2	0.0475	0.0542	0.0560	0.0062	0.0199	0.0278	0.0113	0.0195	0.0233	0.0663	0.0989	0.1141
sigmoid.0	0.2904	0.3258	0.3400	0.0576	0.1264	0.1848	0.0800	0.1219	0.1504	0.3609	0.4924	0.5533
sigmoid.1	0.2873	0.3207	0.3359	0.0591	0.1257	0.1874	0.0802	0.1204	0.1505	0.3663	0.4783	0.5576
sigmoid.2	0.2793	0.3123	0.3258	0.0549	0.1190	0.1750	0.0764	0.1154	0.1427	0.3511	0.4696	0.5391

на аннотированных суффиксных деревьях существенным образом увеличивает точность, при этом, вид шкалирующей функции (квадратичная, корень квадратный, линейная или сигмоид) не играет особой роли (за исключением логарифмической или логистической), и оправдывает себя очистка от шума на первом уровне.

Глава 4. Пополнение научной таксономии с использованием справочных материалов интернета

Таксономия – один из наиболее популярных и удобных инструментов для представления, хранения и использования знания из некоторой предметной области [77; 78]. Автоматизация построения таксономий является важной задачей как обработки текстов на естественном языке [79], так и информационного поиска [80]. Основной подход к автоматическом построению таксономий основан на извлечении ключевых слов и словосочетаний из больших коллекций текстов и семантических отношений между ними. По извлеченным ключевым словам и словосочетанием и семантическим отношениям восстанавливают таксономию. У этого подхода есть несколько очевидных недостатков:

– Не каждая предметная область может быть представлена достаточно большой коллекцией текстов;
– Современные методы извлечения семантических отношений далеки от совершенства, поэтому построенные таким образом таксономии могут быть не полными [81].

Первый недостаток можно компенсировать, используя различные ресурсы Интернета, в том числе, Интернет энциклопедию Википедия [82]. В обзоре [82] перечислены основные подходы к построению онтологий и таксономий на основе Википедии. Во-первых, существуют большие онтологии общего назначения, такие как DBPedia [83], организованные по правилам Семантического Веба. Во-вторых, существует множество меньших таксономий конкретных предметных областей. Создание таких онтологий требует больше предварительной работы с данными Википедии и более точного извлечения из Википедии объектов и понятий, принадлежащих к данной предметной области. К этому направлению относится и описанная ниже работа по построению таксономии математических понятий.

4.1 Метод пополнения таксономии ReTAST-w

Метод, названный в [84] ReTAST-w, состоит из двух шагов. На первом шаге задается основа таксономии, два или три уровня, в ручную, основываясь на формальных текстах и определениях. Второй шаг заключается в пошаговом пополнении основы таксономии фрагментами дерева категорий и статей русскоязычной Википедии, предварительно очищенными от шума. Для соотнесения категорий, названий статей, таксономических тем и статей и очистки дерева категорий от шума использована мера релевантности, основанная на АСД и аппарат РСТ таблиц. Основная идея метода пополнения таксономии заключается в следующем. После того, как из Википедии извлечены все необходимые данные, для каждой темы из основы таксономии мы ищем релевантные ей категории и статьи и пополняем тему найденными релевантными категориями и статьями. Метод проиллюстрирован двумя экспериментами: построением таксономии теории вероятностей и математической статистики и таксономии численных методов.

В качестве источников тем для основы таксономии мы использовали номенклатурные материалы ВАК, в которых представлены верхние уровни классификации современных наук и паспорта научных специальностей ВАК из которых можно извлечь 2-3 дополнительных уровня. Однако, для построения полной и сбалансированной таксономии этих материалов недостаточно: для описания математических понятий требуется еще 2-3 уровня в таксономии. Отсюда возникает потребность в использовании Википедии. Таким образом, возникает задача пополнения таксономии. Основу таксономии, извлеченную из материалов ВАК, требуется достроить до полноценной таксономии, используя данные извлеченные из Википедии. Дополнительное требование к таксономии: следуя золотому стандарту таксономии ACM CCS 2012, каждой листовой теме в таксономии приписать множество уточнений – словосочетаний, объясняющих ее содержание.

Задача пополнения таксономии достаточно широко освещена в литературе. Во всех работах, посвященных пополнению таксономий, возникает общий вопроса: что должно служить источником новых тем. Иногда, например в [85], предлагается использовать результаты поиска вида “A состоит из ... ”, “A – это ...”, где A – тема таксономии, которую следует достроить. Из результа-

тов такого поиска достаточно просто извлечь подтемы темы A. Другой способ предложен в [86]: если таксономия описана посредством формального языка типа OWL, ее несложно пополнить темами другой таксономии, тоже описанной на OWL. Следовательно, источником тем для пополнения могут служить не только коллекции текстов, но и другие таксономии или онтологии. Несколько компромиссным решением является использование Википедии в качестве источника новых тем [85; 87—89], поскольку Википедия содержит как неструктурированные данные, так и структурированные, причем и те, и другие подчиняются общей организации. В [90] перечислены существенные преимущества Википедии в качестве источника тем для построения и для пополнения таксономий:

– Википедия постоянно обновляется, поэтому таксономии, построенные на основе Википедии легко обновлять;
– Википедия мультиязычна, поэтому любой метод, разработанный для одного языка, может быть перенесен на другой язык.

В работах [85; 87—89] представлены разные подходы к построению [87; 88] или пополнения [85; 89]. В [85] в качестве источника тем использованы инфобоксы, в [87] – и тексты статей, и названия категорий, в [88] – только тексты статей, а в [89] – только названия категорий. Мы использовали структуру дерева категорий, названия категорий, названия статей и тексты статей в качестве источника новых тем и уточнений, и ограничивались только категориями “Теория вероятностей и математическая статистика” и “Численные методы”.

Метод пополнения таксономии ReTAST-w состоит из двух частей. На первом, неавтоматическом этапе метода ReTAST-w требуется определить предметную область таксономии, зафиксировать ее основу и выбрать соответствующую категорию в Википедии, данные из которой будут использованы для пополнения таксономии. Далее мы будем использовать дерево категорий Википедии для наращивания дерева таксономии: к каждой теме исходной таксономии мы будем достраивать категории. При этом, будем проверять, стоит ли оставлять в достроенной категории подкатегории, или подкатегории стоит так же достраивать к одной из тем исходной таксономии. Поскольку почти в каждой категории есть статьи, названия статей мы будем использовать в качестве листьев в новой таксономии. Из текстов статей мы будем извлекать уточнения листьев листьев – ключевые слова и словосочетания, которые описывают содержание листа. Таким образом, достроенная таксономия будем удовлетворять золотому стандарту таксономии ACM: каждый раздел в дереве будет примерно одной и

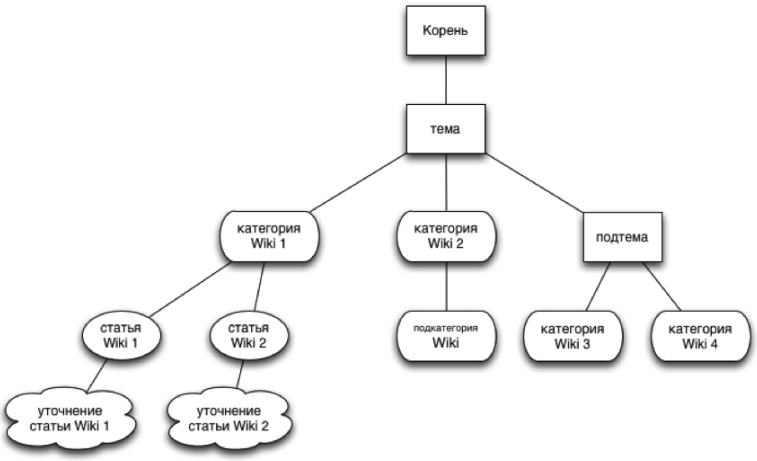

Рисунок 4.1 — Схема пополнения таксономии. В прямоугольниках находятся темы основы таксономии, в скругленный прямоугольниках – достроенные категории и подкатегории Википедии. Листья достроенной таксономии – названия статей Википедии – помещены в овалы. В облачках находятся уточнения листьев.

той же глубины, у каждого узла будет примерно одно и то же количество потомком, а листья в таксономии будут снабжены уточнениями, играющими роль подразделений понятий, представленных листьями.

Структура любой Википедии, в том числе, и русскоязычной, довольно зашумлена. Строго говоря, дерево категорий является не деревом, а графом, поскольку содержит циклы. Иногда между категорий и ее подкатегорией нет никакой логической и смысловой связи, не говоря уж о связях с категориями на два или три уровня ниже. Например, категория “Убитые случайно” лежит в категории “Случайность”. Объяснение этому феномену дано в [91]: авторы Википедии считают, что каждую статью и подкатегорию нужно помещать в как можно большее число категорий для упрощения навигации. Таким образом, данные, извлеченные из Википедии необходимо предварительно очистить от шума перед тем, как использовать их для пополнения таксономии. Требуется удалить цикли из дерева категорий, если они в нем есть, и оставить в дереве только такие подкатегории и статьи, которые имеют логическую и смысловую связь с родительсикми категориями. Нам, в некотором смысле повезло, и деревья категорий “Теория вероятностей и математическая статистика” и “Численные методы” содержали только один цикл (“Машинное обучение” – “Оптимизация” – “Поисковая система” – “Машинное обучение”), который легко разрешить вручную путем удаления связи между “Поисковой системой” и

“Машинным обучением”, поэтому нам осталась только очистка этих деревьев категорий от шума.

Основные шаги автоматического этапа метода ReTAST-w таковы:

1. Извлечение дерева категорий и статей из Википедии
2. Очистка дерева категорий от иррелевантных статей
3. Очистка дерева категорий от ирререлевантных подкатегорий
4. Достраивание категорий Википедии к темам таксономии
5. Формирование промежуточных уровней таксономии
6. Использование названий статей Википедии в качестве листьев в таксономии
7. Излечение ключевых слов и словосочетаний из статей Википедии и использование их в качестве уточнений листьев.

4.2 Экспериментальная верификация метода ReTAST-w

4.2.1 Постановка эксперимента

Как уже было сказано выше, метод ReTAST-w должен состоять из двух этапов. Первый заключается в ручном построении основы таксономии. В качестве основы таксономии мы предлагаем использовать темы, извлечённые из классификации наук и паспортов специальностей ВАК. На втором этапе основа таксономии пополняется темами, извлеченными из дерева категорий и статей русскоязычной Википедии.

С помощью ReTAST-w мы построили две таксономии: таксономию теории вероятностей и математической статистики (ТВиМС) и таксономию численных методов (ЧМ).

Кроме того, была проведена экспертная оценка качества достроенной таксономии и проведено сравнение с косинусной мерой релевантности, использованной по аналогии в задаче достраивания таксономии.

Таблица 17 — Основа таксономии теории вероятностей и математической статистики (ТВиМС), извлеченная из материалов ВАК

ТВиМС	Теория вероятностей и математическая статистика
ТВиМС.01	Теория вероятностей
ТВиМС.01.01	Оптимизационные и алгоритмические вероятностные задачи
ТВиМС.01.02	Комбинаторные и геометрические вероятностные задачи
ТВиМС.01.03	Распределения вероятностей и предельные теоремы
ТВиМС.01.04	Случайные процессы и поля
ТВиМС.01.05	Модели и характеристики случайных явлений
ТВиМС.02	Математическая статистика
ТВиМС.02.01	Методы статистического анализа и вывода
ТВиМС.02.02	Статистические параметры и их оценивание по выборке
ТВиМС.02.03	Статистические критерии и проверка статистических гипотез
ТВиМС.02.04	Временные ряды и случайные процессы
ТВиМС.02.05	Машинное обучение
ТВиМС.02.06	Многомерная статистика и анализ данных

4.2.2 Выбор данных

По материалам ВАК мы задали основы двух таксономий: таксономия ТВиМС (Таблица 17) и таксономия ЧМ (Таблица 18).

Из одноименных категорий Википедии “Теория вероятностей и математическая статистика” и “Численные методы” мы извлекали данные двух типов: деревья категорий, корни которых находится в соответствующих категориях и все статьи, принадлежащие данным категориям и их подкатегориям. В Таблице 19 представлено общее число статей и категорий, извлеченных из Википедии.

Таблица 18 — Основа таксономии численных методов (ЧМ), извлеченная из материалов ВАК

ЧМ	Вычислительная математика
ЧМ.01	Алгоритмы численного решения задач
ЧМ.04	Реализация численных методов в решении прикладных задач
ЧМ.03	Программные комплексы, связанные с численными методами
ЧМ.02	Теория численных методов
ЧМ.02.02	Свойства алгоритмов
ЧМ.02.03	Эффективность алгоритмов
ЧМ.02.01	Обоснование алгоритмов

Таблица 19 — Число статей и категорий в категориях ТВиМС и ЧМ

Предметная область	Число статей	Число категорий
ТВиМС	928	54
ЧМ	1340	91

4.2.3 Пошаговое описание метода ReTAST-w

Извлечение дерева категорий и статей из Википедии

Для извлечения дерева категорий и статей из Википедии была использована программа WikiDP. Она начинала обход дерева категорий в категории “Теория вероятностей и математическая статистика” и обходила дерево категорий по подкатегориям по принципу обхода дерева в глубину. Программа сохраняла все подкатегории и статьи, попавшиеся ей на пути. Аналогичным образом, программа обошла дерево категорий с корнем в “Численных методах”. Общее количество извлеченных категорий и статей представлено выше в Таблице 19.

Таблица 20 — Примеры иррелевантных статей согласно условию A

Предметная область	Оценка релевантности	Родительская категория	Статья
ТВиМС	0.0174	Теория вероятностей	Полная группа событий
ТВиМС	0.0048	Теория вероятностей	Тематическое моделирование
ЧМ	0.0108	Численное интегрирование	Интегрирование Верле

Очистка дерева категорий от иррелевантных статей

Мы считали статью иррелевантной (т.е. шумовой), если для нее выполнялось одно из двух условий:

A Оценка релевантности по СУВСС названия статьи тексту статьи ниже заданного порога;

B Оценка релевантности по СУВСС родительской категории тексту статьи была ниже заданного порога.

Условие A помогает избавиться от так называемых заглушек – пустых или коротких незаконченных статей и статей-шаблонов. Согласно условию B, мы удаляли те статьи, которые не имеют предположительно смысловой связи с родительской категорией. Для оценивания релевантности мы использовали, разумеется, меру релевантности, основанную на АСД. В качестве порога на оценку релевантности мы снова выбрали 0.2 как треть от максимального эмпирически получаемого значения меры релевантности.

На первый взгляд, все оценки приведенные в Таблице 20 могут показаться ошибочными. Тем не менее, они все правомерны. Статья “Полная группа событий” является заглушкой, поэтому не может быть использована для пополнения таксономии. “Тематическое моделирование” предполагает использования аппа-

Таблица 21 — Примеры иррелевантных статей согласно условию B

Предметная область	Оценка релевантности	Родительская категория	Статья
ТВиМС	0.1020	Теория вероятностей	Поиск наилучшей проекции
ТВиМС	0.0156	Байесовская статистика	Перл, Джуда
ЧМ	0.1948	Регресионный анализ	ROC-кривая
ЧМ	0.1944	Численное интегрирование	БШСН формализм

рата теории вероятностей, но, относится скорее к “Автоматической обработке текстов” или “Информационному поиску”, чем к “Теории вероятностей”. Аналогично, “Интегрирование Верле” скорее принадлежит к “Численному решению дифференциальных уравнений”, чем к “Численному интегрированию”.

Схожие сомнения может вызвать и Таблица 21. На самом деле, “БШСН формализм” является частью “Общей теории относительности”, а не “Численного интегрирования”, тем более, что по размеру (2 абзаца) эта статья больше напоминает заглушку, чем полноценную статью. “ROC-кривая” – способ оценки качества классификаторов – это понятие из области “Машинного обучения”, а не из области “Регрессионного анализа”. “Перл, Джуда” вовсе не понятия, а имя одного известного ученого. “Поиск наилучшей проекции” в самом деле принадлежит “Математической статистике”, но скорее, в качестве непрямого потомка. Правильней было бы поместить это понятие в категорию “Многомерная статистика” (однако, такой категории в русскоязычной Википедии нет).

Очистка дерева категорий от ирререлевантных подкатегорий

Таблица 22 — Примеры иррелевантных подкатегорий

Предметная область	Оценка релевантности	Родительская категория	Подкатегория
ТВиМС	0.1923	Статистика	Статистика по странам
ТВиМС	0.1515	Машинное обучение	Теория оптимизации
ТВиМС	0.0142	Статистика	Мета-анализ
ЧМ	0.0632	Алгоритмы	Вычислительная теория групп
ЧМ	0.0287	Численные методы	Численные методы механики сплошных сред

Мы считали подкатегорию иррелеватной родительской категории, если оценка релевантности названия родительской категории всем статьям подкатегории, объединённым в один текст, ниже заданного порога. Мы снова использовали СУВСС – меру релевантности, основанную на АСД, и в качестве порога на оценку снова выбрали 0.2. Такой подход к определению иррелевантных категорий не применим в том случае, если в подкатегории нет статей.

Рассмотрим Таблицу 22. Она действительно выявляет некоторые слабости связи категория – подкатегория в русскоязычной Википедии. Так, например, понятие “Теория оптимизации” должно было бы быть “сестрой”, а не потомком “Машинного обучения”. Примеры из области численных методов (ЧМ) показывают, как понятия, принадлежащие к частной теории ошибочно становятся составляющими более общей. Примеры из категории “Статистика” выявляют двойственность это категории: с одной стороны, в нее попадают статьи и подкатегории связанные с “Математической статистикой”, с другой стороны, статьи и категории, связанные с использованием статистики в общественных науках.

Достраивание категорий Википедии к темам таксономии

После очистки дерева категорий от иррелевантных статей и категорий мы достраивали категории статей к темам таксономии. Для этого мы оценивали релевантность таксономических тем категориям, представленным всеми статьями, объеденными в один текст. Мы достраивали категорию в качестве потомка к той теме таксономии, оценка релевантности которой оказалась максимальной. Таблицы и демонстрируют два примера достраивания категорий Википедии к темам таксономии. В первом случае рассматривается достраивание категории “Байесовская статистика” к темам таксономии ТВиМС, во втором – категории “Методы решения СЛАУ” к темам таксономии ЧМ. Все темы таксономий в таблицах приведены в порядке возрастания оценки релевантности, так что последней оказывается та тема, к которой достраивается категория (“Теория вероятностей” и “Алгоритмы численного решения задач”).

Формирование промежуточных уровней таксономии

На промежуточном уровне в таксономии остаются те подкатегории, оценка релеватности по СУВСС которым названия их родительских категорий выше, чем оценка релевантности по СУВСС им таксономических тем.

Согласно Таблице 25, из 6 подкатегорий оставшихся в категории “Случайные процессы” после процедуры очистки дерева категорий, 3 подкатегории (“Марковские процессы”, “Мартингалы”, “Метод Монте-Карло”) более релевантны родительской категории, чем темам таксономии, а три (“Стохастические модели”, “Шум”, “Теория массового обслуживания” – теме таксономии “Случайные процессы”. Заметим, что, во-первых, в русскоязычной Википедии отсутствуют статьи, посвященные случайным полям, а во-вторых, что все подкатегории ка-

Таблица 23 — Оценки релевантности категории “Байесовская статистика” темам таксономии ТВиМС

Оценка релевантности	Теория вероятностей и математическая статистика
0.0190	Временные ряды и случайные процессы
0.0789	Случайные процессы и поля
0.1212	Оптимизационные и алгоритмические вероятностные задачи
0.1504	Модели и характеристики случайных явлений
0.1957	Распределения вероятностей и предельные теоремы
0.2003	Комбинаторные и геометрические вероятностные задачи
0.2012	Статистические критерии и проверка статистических гипотез
0.2452	Статистические параметры и их оценивание по выборке
0.2870	Методы статистического анализа и вывода
0.3201	Математическая статистика
0.3450	Многомерная статистика и анализ данных
0.4210	Машинное обучение
0.5323	**Теория вероятностей**

тегории “Случайные процессы” получают максимальную оценку релевантности одной и той же теме таксономии – “Случайные процессы и поля”.

Полученный согласно Таблице 25 фрагмент таксономии ТВиМС представлен на Рис. .

Таблица 24 — Оценки релевантности категории “Методы решения СЛАУ” темам таксономии ЧМ

Оценка релевантности	Вычислительная математика
0.1631	Эффективность алгоритмов
0.1803	Теория численных методов
0.2071	Программные комплексы, связанные с численными методами
0.2138	Обоснование алгоритмов
0.2761	Свойства алгоритмов
0.3865	Реализация численных методов в решении прикладных задач
0.5134	Численные методы
0.6210	**Алгоритмы численного решения задач**

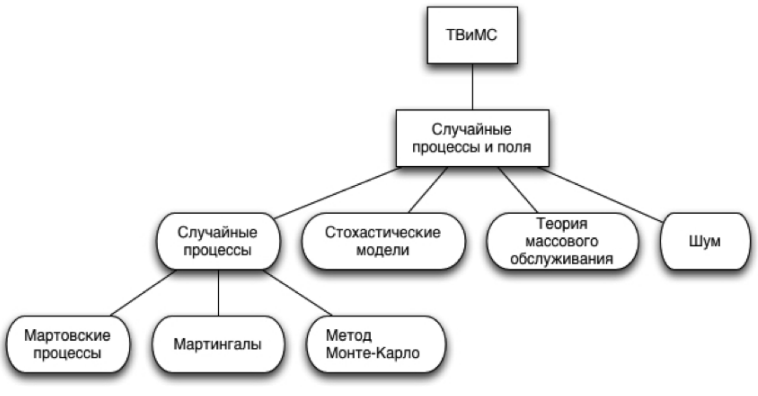

Рисунок 4.2 — Фрагмент таксономии ТВиМС: промежуточные уровни раздела “Случайные процессы и поля”

Таблица 25 — Примеры подкатегорий, формирующих промежуточные уровни в таксономии

Предметная область	Оценка релевантности теме таксономии	Тема таксономии	Оценка релевантности родительской категории	Подкатегория
ТВиМС	0.4961	Случайные процессы и поля	0.4842	Стохастические модели
ТВиМС	0.4914	Случайные процессы и поля	0.3825	Шум
ТВиМС	0.4671	Случайные процессы и поля	0.4813	Марковские процессы
ТВиМС	0.4423	Случайные процессы и поля	0.3814	Теория массового обслуживания
ТВиМС	0.4267	Случайные процессы и поля	0.4372	Метод Монте-Карло
ТВиМС	0.3752	Случайные процессы и поля	0.3982	Мартингалы

Использование названий статей Википедии в качестве листьев в таксономии

Если после процедуры очистки дерева категорий от шума в категории остались статьи, то мы назначали их листьями в достроенной таксономии, вне зависимости от того, на какой уровень была достроена категория.

В Таблице 26 приведены релевантные (выделенные жирным начертанием) и иррелевантные статьи категории “Метод Монте-Карло”. 6 релевантных статей

Таблица 26 — Релевантные и иррелевантные статьи в категории “Метод Монте-Карло”

Предметная область	Оценка релевантности	Категория	Статья
ТВиМС	0.4529	Метод Монте-Карло	**Метод Монте-Карло**
ТВиМС	0.3974	Метод Монте-Карло	**Метод Монте-Карло для переноса фотонов**
ТВиМС	0.3864	Метод Монте-Карло	**Семплирование**
ТВиМС	0.3193	Метод Монте-Карло	**Алгоритм имитации отжига**
ТВиМС	0.2974	Метод Монте-Карло	**Семплирование по Гиббсу**
ТВиМС	0.2423	Метод Монте-Карло	**Выборка по значимости**
ТВиМС	0.1973	Метод Монте-Карло	Выборка с отклонением
ТВиМС	0.1537	Метод Монте-Карло	Выборка по уровням
ТВиМС	0.1294	Метод Монте-Карло	Тасование Фишера-Йетса
ТВиМС	0.0475	Метод Монте-Карло	Дифференциальная эволюция

остаются детьми темы “Метод Монте-Карло” становятся листьями в таксономии ТВиМС.

Таблица 27 — Ключевые слова и словосочетания, извлеченные из статьи “Семплирование по Гиббсу”

Ключевое слово / словосочетание	Частота
Случайная величина	13
Алгоритм	12
Совместное распределение	7
Плотность вероятности	6
Условная вероятность	4
Отклонение	4

Извлечение ключевых слов и словосочетаний из статей

Под уточнением листа мы понимаем набор из ключевых слов и словосочетаний, объясняющих содержание листа. Поскольку в качестве листьев в достроенной таксономии выступают названия статей, из текстов статей непосредственно можно извлечь уточнения. Мы не использовали сложных методов извлечениях ключевых слов и словосочетаний, а извлекали наиболее частотные существительные и словосочетания и считали их уточнениями. Словосочетания мы определяли по грамматическим шаблонам вида существительное + существительное или прилагательное + прилагательное, следуя [92]. Части речи определялись с использованием морфологического анализатора PyMorphy2. Ключевыми мы считали те слова и словосочетания, частота которых превосходит 4 (порог установлен эмпирически).

Таблица 27 представляет ключевые слова и словосочетания, извлеченные по этой схеме из статьи “Семплирование по Гиббсу”.

4.2.4 Схема эксперимента

Эксперимент по пополнению каждой таксономии осуществляется в несколько этапов:

1. Для построения четырех PCT таблиц с использованием СУВСС:
   - название статьи – текст статьи;
   - название родительской категории – текст статьи;
   - название родительской категории – объединенный текст подкатегории;
   - тема таксономии – объединенный текст подкатегории.

   a) Предобработка текстов: удаление Вики-разметки из статей Википедии разбиение текстов на строки из трех слов
   b) Использование меры релевантности, основанной на АСД с линейной шкалирующей функцией для оценивания релевантности названий статей, категорий и таксономических тем статьям и объединенным текстам подкатегории.
   c) Формирование четырех PCT таблиц.
2. Для пополнения таксономии по построенным PCT таблицам:
   a) Иррелевантные статьи определяются поPCTтаблицам название статьи – текст статьи и название категории – текст статьи;
   b) Иррелевантные категории определяются поPCTтаблице название родительской категории – объединенный текст подкатегории;
   c) Достраивание категорий Википедии к темам таксономии осуществляется по таблице тема таксономии – объединенный текст подкатегории;
   d) Промежуточные уровни таксономии определяются по PCT таблицам тема таксономии – объединенный текст подкатегории и название родительской категории – объединенный текст подкатегории;
   e) Уточнения листьев извлекаются из текстов статей.

4.2.5 Экспертное оценивание

Как уже было сказано выше, адекватных русскоязычных таксономий прикладной или чистой математики не существует, что делает сравненения с эталоном невозможны, поэтому для оценки качества построенных таксономий были привлечены двое экспертов.

Для экспертов был подготовлен опросник, состоящих из двух частей, в соответствии с двумя логическими этапами достраивания таксономии. В первой части экспертам был задан вопрос, направленный на проверку точности очистки дерева категорий от шума, во второй части – на точность определения родителя для категории Википедии (тема из таксономии или надкатегория по дереву категорий).

Экспертная оценка проводилась только для достроенной таксономии теории вероятностей и математической статистики.

Приведем инструкцию для экспертов по заполнению опросника.

Просим Вас помочь оценить качество машинной классификации. В первой части анкеты (Лист "Часть 1") Вам будет предъявлено некоторое множество понятий математической статистики и теории вероятностей (выделены жирным шрифтом), каждому из которых сопоставлено некоторое количество понятий-детей, автоматически выбранных из русской Википедии в соответствии с методом, описанным в статье " Chernyak E. L., Mirkin B. G. Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources // Annals of Data Science. 2015. Vol. 2. No. 1. P. 61-82". Предполагается, что каждое из понятий-детей выражает часть понятия родителя. Приведем пример.

Ваша задача - для каждого из понятий-детей указать, в соответствии с Вашим представлением о предмете, является ли оно действительно частью понятия-родителя. Ваш ответ - "да", "нет" или оставить поле ответа пустым, если вы не знаете, как ответить.

На второй странице эксперимента (Лист "Часть 2") тот же формат выражает обратное отношение, при котором понятия-дети – это кандидаты на роль более общего понятия, чем понятие-родитель. Пример:

Нужно выбрать одно и только одно понятие - родитель, частью которого является данное понятие - дитя. В данном случае в качестве такового

Таблица 28 — Пример вопроса из Части 1

Метод Монте-Карло	
	CompHEP
	FLUKA
	Geant4
	MCNP
	Monte Carlo Universal
	PYTHIA
	Алгоритм имитации отжига
	Алгоритм Метрополиса — Гастингса
	Андросенко, Пётр Александрович
	Бюффон, Жорж-Луи Леклерк де
	Выборка по значимости
	Выборка по уровням

следует выбрать Алгоритмы и методы оптимизации. Для обозначения своего выбора поставьте символ "X" ("икс") в левый столбец рядом с выбранным понятием-родителем. Хотя, конечно, Выпуклый анализ, вообще говоря, далеко выходит за рамки задач оптимизации, но применительно к тематике теории вероятностей и математической статистики, о которой только и идёт речь, предлагаемое соответствие можно признать корректным.

Результаты экспертного оценивания могут быть использованы в качестве эталонной таксономии теории вероятностей и математической статистики. С эталонной таксономией можно сранить полученную достроенноую таксономию той же предметной области, и оценить таким образом ее качество. Для проверки целесобразности достраивания таксономии с использованием меры релевантности СУВСС, а не какой-либо другой, можно повторить процедуру достраивания таксономии с любой другой мерой релеватности (например, с использованием косинусной меры релеватности), и так же сравнить достроенную таксономию с эталоном, полученным в результате опроса экспертов.

Таблица 29 — Пример вопроса из Части 2

Выпуклый анализ	
X	*Алгоритмы и методы оптимизации*
	Временные ряды и случайные процессы
	Комбинаторные и геометрические вероятностные задачи
	Методы статистического анализа и вывода
	Математическая статистика
	Многомерная статистика и анализ данных

Непосредственно для оценивания качества достроенной таксономии можно использовать следующие показатели:

– Качество очистки от шума:
  – tp – число истинно-положительных статей и категорий, то есть, число статей, являющихся шумовыми с точки зрения экспертов и признанными шумовыми по АСД мере;
  – tn – число истинно-отрицательных статей и категорий, то есть, число статей, являющихся нешумовыми с точки зрения экспертов и признанными нешумовыми по АСД мере;
  – fp – число ложно-положительных статей и категорий, то есть, число статей, являющихся нешумовыми с точки зрения экспертов согласно и признанными нешумовыми по АСД мере;
  – fn – число ложно-отрицательных статей и категорий, то есть, число статей, являющихся шумовыми с точки зрения экспертов согласно и признанными шумовыми по АСД мере;
  – аккуратность $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$ – доля истинно-положительных и истинно-отрицательных статей и категорий среди об-

щего числа статей и категорий – аггрегированная мера качества очистки от шума;

- Качество достраивания категорий Википедии к темам таксономии и формирования промежуточных уровней:
  - tp – число истинно-положительных пар категория – родитель, где родитель, назначенный экспертами совпадает с родителем, выбранным по АСД мере;
  - tn – число истинно-отрицательных пар категория – родитель, где родитель, не назначенный экспертами, не выбран по АСД мере;
  - fp – число ложно-положительных пар категория – родитель, где родитель, не назначенный экспертами, выбран по АСД мере;
  - fn – число ложно-отрицательных пар категория – родитель, где родитель, назначенный экспертами, не выбран по АСД мере;
  - аккуратность $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$ – доля истинно-положительных и истинно-отрицательных статей и пар категория – родитель среди всех возможных пар категория – родитель – аггрегированная мера качества достраивания категорий Википедии к темам таксономии и формирования промежуточных уровней.

4.2.6 Результаты эксперимента

Полученная таксономия ТВиМС насчитывает 6 уровней, ее глубина изменяется нот 4 до 6. В ходе ее построения 20 категорий и 108 статей были признаны иррелевантными и убраны из дерева категорий Википедии. Таксономия ЧМ имеет похожую форму: в ней 8 уровней, глубина изменяется от 4 до 8. На этапе очистки 11 категорий и 30 статей были признаны иррелевантными. На Рис. 4.3 и Рис. 4.4 представлены фрагменты таксономии ТВиМС (с акцентом на листья и их уточнения) и ЧМ (с акцентом на промежуточные уровни таксономии).

В ходе пополнения обеих таксономий мы столкнулись с несколькими проблемами, выявляющими недостатки метода ReTAST-w. Во-первых, положение категории “Деревья принятия решений” в таксономии ТВиМС нас удивило.

Таблица 30 — Качество очистки от шума

		эксперты	
		шум	не шум
АСД	шум	731	67
	не шум	21	264

Таблица 31 — Качество достраивания категорий Википедии к темам таксономии и формирования промежуточных уровней

		эксперты	
		родитель	не родитель
АСД	родитель	403	51
	не родитель	95	78

Согласно нашему методы ReTAST-w категория “Деревья принятия решений” должна быть помещена под темой “Математическая статистика”. Однако, у темы “Математическая статистика” уже есть потомок “Машинное обучение”. Методу ReTAST-w не удается поймать связь между “Машинным обучением” и “Деревьями принятия решения” из-за невысокой релевантности строки “Машинное обучение” всем четырем статьям в категории “Деревья принятия решения”. Низкое значение релевантности объясняется тем, что ни в одной из четырех статей не упоминается машинное обучение. Во-вторых, категория “Преобразователи”, которая релевантна своей родительской категории “Эффективность алгоритмов”, имеет подкатегории “Пьезоэлектрики”, “Источники питания”, “Излучатели и приемники звука”. Эти три подкатегории релевантны категории “Преобразователи”, но никак не связаны с “Эффективностью алгоритмов”. Это объясняется тем, что слово “преобразователь” имеет двойственный смысл в русском языки. В-третьих, обе таксономии заполнены статьями, описывающих персоналии, например, вероятностниками или лекторами МФТИ, и категорями, содержащими данные статьи. Следовательно, требуется разработать дополнительные процедуры очистки, исключающие статьи, описывющие персоналии и категории, их содержащие.

В экспертной оценке построенной таксономии ТВиМС участвовали два эксперта. Полученные результаты представлены в Таблицах 30 и 31.

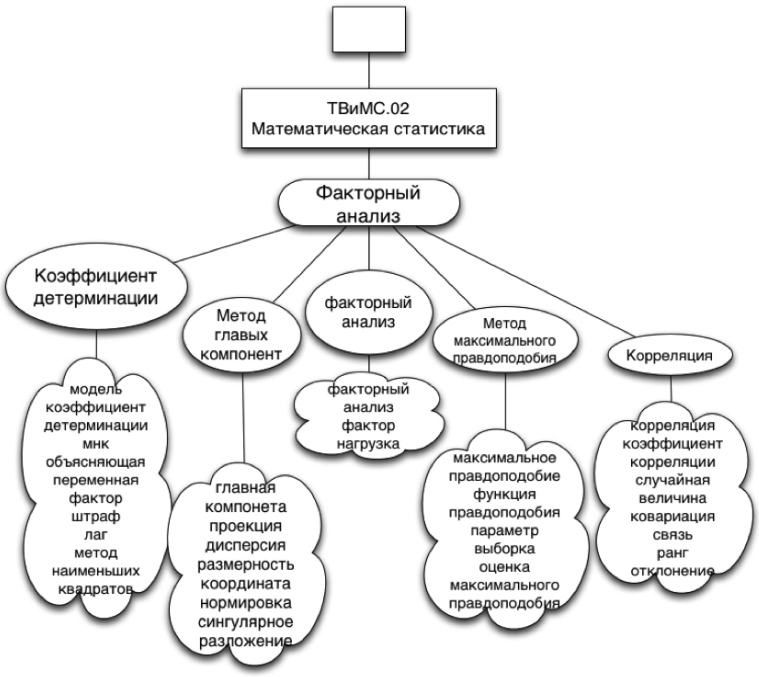

Рисунок 4.3 — Фрагмент достроенной таксономии ТВиМС. В прямоугольниках находятся темы основы таксономии, в скругленный прямоугольниках – достроенные категории и подкатегории Википедии. Листья достроенной таксономии – названия статей Википедии – помещены в овалы. В облачках находятся уточнения листьев.

Аккуратность *accuracy* очистки от шума составлят 0.91, достраивания категорий Википедии к темам таксономии и формирования промежуточных уровней – 0.76.

Достоверность проведенного экспертного оценивания определяется независимо для обеих частей исследования. Согласованность ответов экспертов на вопросы из Части 1 определяется с помощью коэффициента κ Коэна, на вопросы из Части 2 – долей несовпавших ответов. Коэффициент согласованности κ Коэна ответов на вопросы из Части 1 составляет 0.319, т.е., в принципе, ответы экспертов можно считать согласованными. Доля несовпавших ответов на вопросы из Части 2 составляет 12%.

Предложенный метод пополнения таксономии ReTAST-w позволяет построить качественную таксономию: доля полученных ошибок не велика, экспертные оценки, подтверждающие высосоке качество таксономии, – достаточно высоки и согласованы на приемлимом статистическом уровне.

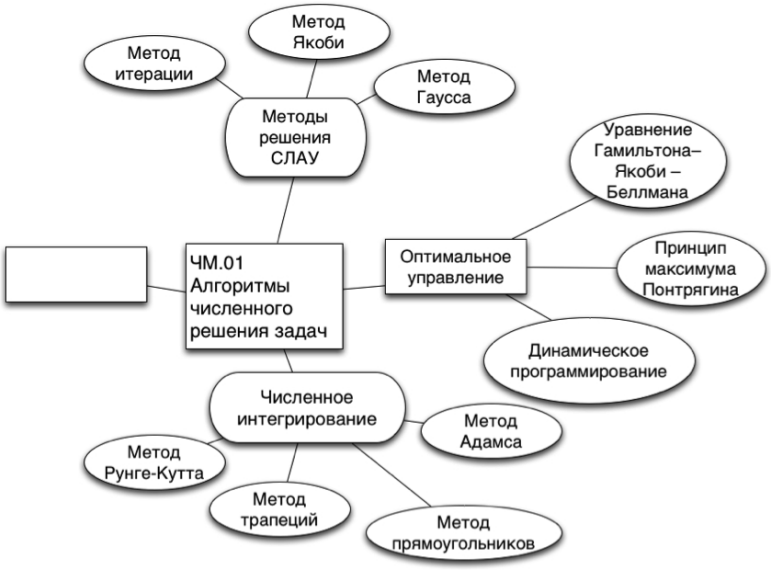

Рисунок 4.4 — Фрагмент достроенной таксономии ЧМ. В прямоугольниках находятся темы основы таксономии, в скругленный прямоугольниках – достроенные категории и подкатегории Википедии. Листья достроенной таксономии – названия статей Википедии – помещены в овалы.

Глава 5. Фильтрация обсценной лексики

Введение возрастной классификации информационной продукции в России делает задачу автоматической фильтрации контента и Веб-цензуру особенно актуальной. Под фильтрацией контента обычно понимают ограничение доступа к ресурсам нежелательной тематики и содержания [93]. Идея фильтрации контента заключается в автоматическом ограничении доступа к определенному и ресурсу и осуществляется за счет:

– использования предопределенных баз категорий ресурсов;
– категоризации данных в момент обращения пользователя;
– предоставления Веб-ресурсом своей категории.

Использование предопределённых баз категорий предполагает, что существуют списки адресов ресурсов и условий, при которых доступ к ним может быть ограничен. При обращении пользователя к какому-то ресурсу, проверяется, входит ли данный ресурс в базу запрещенных ресурсов и, если входит, возвращается страница ошибки. Примером реализации такого алгоритма фильтрации может быть недавняя блокировка РуТреккера российскими провайдерами.

Категоризация данных в момент обращения пользователя предполагает анализ контента, опубликованного на странице, к которой пользователь обращается. Могут выполняться проверки на наличие определенных слов, словосочетаний и фиксированных выражений, которые сигнализируют о наличии запрещённого контента [94; 95]. Рынок программного обеспечения подобного рода контент-фильтров хорошо развит, существуют десятки программ, которые могут быть использованы как на государственном, так и на частном уровне. К этому же направлению можно отнести и разработку спам-фильтров [96].

Предоставления Веб-ресурсом своей категории предполагает, что владельцы ресурса самостоятельно делают фильтры для ограничения доступа: например, проверяют, что все пользователи ресурса достигли определенного возраста или сообщают пользователю категорию ресурса, после чего пользователь (или определённые надстройки пользовательского браузера) принимают решение о переходе на запрашиваемую страницу или отказе от нее.

Если первый и третий способ фильтрации требуют обращения ко внешним источникам, то для второго способа достаточно анализа опубликованного на странице контента. Сконцентрируемся на этом направлении.

Сформулируем формальную постановку задачи. Пусть дана коллекция текстов D и список стоп-слов, являющихся маркерами запрещенного контекста, $stop$ – стоп-лист. Для каждого слова $t \in D$, входящего в коллекцию, требуется определить:

- входит ли само слово t непосредственно в список $stop$;
- и входят ли производные или однокоренные слова t или составляющие слова t в список $stop$.

Таким образом, задача фильтрации стоп-слов эквивалента задаче поиска по однословному ключу [15]. Однако в качестве оптимизируемого критерия следует использовать не точность, которая обычно используется в задачах поиска, а полноту: необходимо найти как можно больше вхождений стоп-слов в текст, при этом, допустимо ложное срабатывание фильтра. Другими словами, стоимость ошибки первого рода существенно ниже, чем ошибки второго рода, в отличии от задачи поиска. Другим немаловажным параметром фильтра является время его работы: если предполагается использование фильтра в реальном режиме времени, он должен работать быстро.

Эксперимент, описанные ниже, посвящен проверке применимости метода АСД в задаче фильтрации слов по стоп-листу. В качестве стоп-листа рассмотрен список обсценной, то есть, нецензурной и ненормативной лексики, а в качестве фильтров – различные меры релевантности, в том числе, СУВСС.

5.1 Метод фильтрации обсценной лексики fAST

Рассмотрим несколько вариантов фильтрации обсценной лексики:

- Поиск по совпадению: слово t входит в стоп-лист в неизмененной форме
- Поиск по лемме: нормальная форма слова t входит в стоп-лист
- Поиск по основе (стему): основа (стем) слова t входит в стоп-лист
- Поиск по составляющим: найдено такое стоп-слово s, что коэффициент Жаккара между множеством n-грамм, на которые разбивается слово

s и множеством n-грамм, на которые разбивается слово t превышает некий заранее заданный порог;

– Поиск по редакционному расстоянию: найдено такое стоп-слово s, что редакционное расстояние Левенштейна (то есть, число операций вставки, удаления и замены символа) [97] между ним и словом t ниже некого заранее заданного порога;
– Поиск с использованием СУВСС: оценка вхождения слова t в АСД, построенное по стоп-листу, превышает некий заранее заданный порог.

Обозначим метод фильтрации с использованием СУВСС через fAST и проведем сравнение этого метода с остальными.

5.2 Экспериментальная верификация метода фильтрации fAST

Для экспериментальной верификации метода fAST необходимы две составляющие:

– стоп-лист;
– коллекция текстов, содержащих обсценную лексику, и разметка (указания на обсценные слова).

В качестве стоп-листа был использован список слов, запрещенных к использованию для наименования ресурсов в доменной зоне “рф”. Стоп-лист содержит 4023 слова, например, таких как “говнецо”, “сиська”, “шалашовка”. Коллекция текстов была составлена и размечена автором исследования самостоятельно. Она состоит из научных статей об этимологии русского мата, текстов произведений Юза Алешковского, Игоря Губермана и Владимира Сорокина, песен групп Ленинград и Красная Плесень, стихотворений Сергея Есенина, Владимира Маяковского и Александра Пушкина, постов Артемия Лебедева в Живом Журнале (http://tema.livejournal.com/), статей, опубликованных на портале Луркмор (https://lurkmore.to/), а так же частушек, анекдотов и пословиц. Общий размер коллекции составляет 294916 словоупотреблений и 60868 словоформ.

5.2.1 Постановка эксперимента

Сравним все методы фильтрации между собой. Поскольку составленная коллекция размечена, то есть, про каждое слово, известно является ли оно обсценным или нет, вычислим следующие показатели качества:

- tp – число истинно-положительных слов, то есть, число слов, являющихся обсценными согласно разметке и признанными обсценными фильтром;
- tn – число истинно-отрицательных слов, то есть, число слов, неявляющихся обсценными согласно разметке и непризнанными обсценными фильтром;
- fp – число ложно-положительных слов, то есть, число слов, неявляющихся обсценными согласно разметке и признанными обсценными фильтром;
- fn – число ложно-положительных слов, то есть, число слов, являющихся обсценными согласно разметке и непризнанными обсценными фильтром;
- точность $precision = \frac{tp}{tp+fp}$ – доля обсценных слов, среди общего числа слов, признаными обсценными фильтром;
- полнота $recall = \frac{tp}{tp+fn}$ – доля обсценных слов, среди общего числа слов, являющихся обсценными согласно разметке;
- аккуратность $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$ – доля истинно-положительных и истинно-отрицательных слов среди общего числа слов – аггрегированная мера качества фильтра;
- F_2-мера $F_2 = 2 \cdot \frac{precision \cdot recall}{precision+recall}$

– среднее гармоническое точности и полноты – агрегированная мера качества фильтра.

5.2.2 Схема эксперимента

Эксперимент осуществлялся в несколько шагов:

1. Считывание и первичная обработка (удаление знаков пунктуации, токенизация, приведение к нижнему регистру) коллекции текстов, общее число словоупотреблений – 294916;
2. Составление частотного словаря по коллекции текстов, общее число словоформ – 60868;
3. Считывание и приведение к нижнему регистру стоп-листа;
4. Поиск совпадений между словоформами из частотного словаря и стоп-листом, вычисление показателей качества;
5. Лемматизация частотного словаря, составленного по коллекции, с помощью PyMorphy2 [98], поиск совпадений между нормальными формами и стоп-листом, вычисление показателей качества;
6. Лемматизация частотного словаря, составленного по коллекции, с помощью Mystem3 [99], поиск совпадений между нормальными формами и стоп-листом, вычисление показателей качества;
7. Стемминг стоп-листа частотного словаря, составленного по коллекции, с помощью алгоритма Портера [74] в реализации NLTK [75], поиск совпадающих стемов, вычисление показателей качества;
8. Разбиение слов из частотного словаря, составленного по коллекции, на n-граммы ($n = 3,4,5,6$), разбиение стоп-слов на n-граммы ($n = 3,4,5,6$), вычисление меры Жаккара по множествам n-грамм, определение обсценных слов по порогу, составляющему 0.8, вычисление показателей качества;
9. Вычисление редакционного расстояния Левенштейна между словам из частотного словаря, составленного по коллекции, и стоп-словами, определение обсценных слов по порогу, составляющему 5, 8, вычисление показателей качества;
10. Построение АСД по стоп-листу, вычисление оценок сходства слов из частотного словаря, составленного по коллекции, с АСД, определение обсценных слов по порогу, составляющему 0.2, вычисление показателей качества.

Таблица 32 — Сравнение фильтров обсценной лексики по точности, полноте, аккуратности и F_2-мере

	точность	полнота	аккуратность	F_2-мера
совпадения	**0.7288**	0.1363	0.9810	0.2297
лемматизация				
pymorphy2	0.6492	0.2466	0.9815	0.3574
mystem3	0.6807	0.3195	**0.9827**	0.4349
стемминг	0.6113	0.4028	0.9822	**0.4856**
АСД, порог = 0.2	0.1578	**0.6201**	0.9233	0.2516
коэффициент Жаккара				
3-граммы	0.6799	0.1633	0.9810	0.2634
4-граммы	0.7126	0.1475	0.9810	0.2430
5-граммы	0.7168	0.1284	0.9808	0.2179
6-граммы	0.6989	0.0975	0.9803	0.1711
расстояние Левенштейна				
$d < 8$	0.0234	**0.9127**	0.8086	0.0456
$d < 5$	0.0209	**0.9825**	0.9629	0.0409

5.2.3 Результаты эксперимента

Результаты эксперимента приведены в Таблице 32.

По точности лучшим методом фильтрации является поиск совпадения, что очевидно: если слово входит в стоп-лист, то оно является безусловно обсценным. Худшими фильтрами по точности оказываются фильтры, основанные на расстоянии Левенштейна: среди тех слов, которые эти фильтры определяют как обсценные, на самом деле обсценными являются 2%. Эти же фильтры являются лучшими по полноте: с их использованием получается обнаружить до 98% обсценных слов. Второе место по полноте занимает фильтр, основанный на АСД: этот фильтр обнаруживает порядка 60% обсценной лексики. Остальные фильтры существенно проигрывают по полноте, но выигрывают по точности. Среди двух использованных лемматизаторов, лучшие результаты достигаются при использовании Mystem. Стемминг позволяет достичь выигрыша порядка 10% по полноте сравнению с лемматизацией при относительно несущественном

падении точности. Вычисление меры Жаккара на n-граммах при сравнительно высокой точности приводит к низким значением полноты.

Важным параметром для сравнения фильтров является их вычислительная сложность. Приведем оценки вычислительной сложности каждого фильтра. Допустим, что n – это максимум из всех возможных длин слов, m – максимум из длины частотного слова и стоп-листа, $n \ll m$. Тогда:

- Сложность поиска по совпадению (лемме, стему) составляет $O(m)$ – слово (лемма, стем) проверяется на совпадение со словами (стемами) из стоп-листа;
- Сложность попарного вычисления коэффициента Жаккара на множествах n-грамм для слов из частотного словаря и стоп-листа и расстояния Левенштейна составляет $O(n^2 \cdot m^2)$, сложность проверки одного слова – $O(n^2 \cdot m)$;
- Сложность построения АСД с помощью алгоритма Укконена для стоплиста составляет $O(m \cdot n)$, сложность проверки одного слово – $O(n)$.

Таким образом, мы видим, что по общим мерам качества (аккуратности и F_2-мере), выигрывают фильтры, использующие поиск совпадения по леммам. Однако, с учетом приведенного выше утверждения о важности именно полноты, а не общего качества фильтра, целесообразным представляется использование фильтров, вычисляющих расстояние Левенштейна или СУВВС. Остановимся подробнее на сравнение этих двух фильтров. Фильтры, использующие расстояние Левенштейна, имеют показатели точности близкие к нулю, а полноты – к 1, за счет чего достигают высоких оценок аккуратности и мизерных оценок F_2-меры. Фильтры, использующие СУВВС, более сбалансированы по точности и полноте, имеют аккуратность, сопоставимую с другими рассматриваемыми фильтрами, но и невысокое значение F_2-меры, благодаря низкой точности. В итоге, с учетом того, что при уже построенном заранее по стоп-листу АСД, проверка одного слова имеет линейную сложность по времени, разумной представляется следующая схема фильтрации. Заранее строится АСД по стоплисту, которое занимает небольшое место и хранится в оперативной памяти. При необходимости проверки текста на наличие обсценной лексики находятся все словоупотребления, после чего для каждого находится оценка по СУВСС. Если найденная оценка превосходит некий порог, слово признается обсценным и срабатывают следующие процедуры фильтрации.

Глава 6. Комплексы программ

В [100; 101] описывается современный подход к разработке современных систем автоматической обработки текстов в форме pipeline. Примерами систем, разработанных в рамках этого подхода, могут выступить Stanford CoreNLP [102] Стэндфордского университета, свободная библиотека для машинного обучения SciKit-Learn [103], Google SyntaxNet [104], DKPro Технического университета Дармштадта [105]. Идея pipeline заключается в том, что одна система отвечает за все необходимые этап обработки текстов и, если такая необходимость возникает, обращается к внешним инструментам без участия пользователя. Стандартный pipeline включает в себя предобработку текстов (удаление лишних символов и разметки), токенизацию, морфологический анализ (приведение слов к нормальной форме и/или лемматизацию и/или стемминг) и синтаксический анализ (выделение именных и/или глагольных групп и/или построение деревьев зависимостей или составляющих). Пользователю при этом достаточно задать входные файлы и параметры конфигурации и не нужно вызывать отдельные модули последовательно.

В духе pipeline реализована описанная ниже библиотека EAST, выполняющая все необходимые шаги для построения АСД таблицы. Заметим, что pipeline библиотеки EAST существенно меньше, чем pipeline библиотеки SciKit-Learn или индустриальных CoreNLP или DKPro, хотя бы потому, что использование метода АСД не требует существенной обработки текстов. При необходимости подключить к библиотеке EAST возможности морфологического или синтаксического анализа не составит особого труда.

6.1 Программная реализация построения таблиц PCT и метода АСД

В статье [60] приведено описание реализации наивного и линейного алгоритмов построения АСД и таблиц pCT. Программная реализация обоих алгоритмов находится в свободном доступе по следующей ссылке: https://github.com/mikhaildubov/AST-text-analysis. Программа, разработанная совместно

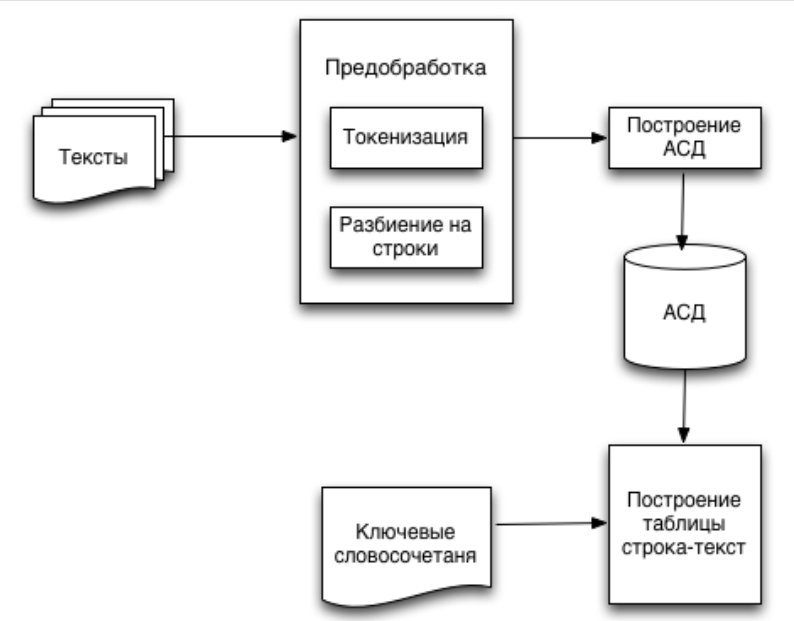

Рисунок 6.1 — Схема pipeline библиотеки EAST

с М. С. Дубовым, может работать в двух режимах: как из командной строки, так и в качестве библиотеки на языке Python 2.7.

На Рис. 6.1 представлена схема pipeline библиотеки EAST: на вход системе поступает коллекция текстовых документов и список ключевых слов и словосочетаний. Коллекция текстовых документов подвергается предобработке: токенизации и разбиению на строки, по которым в последствии строится аннотированное суффиксное дерево (АСД). После чего вычисляется сходство ключевых слов и словосочетаний с построенным АСД и на выход подается готовая таблица релевантности. Все последующие аналитические алгоритмы используют построенную таблицу либо для определения максимальных значений меры релевантности, либо для ранжирования по убыванию меры релевантности.

6.1.1 Использование программы EAST из командной строки

Для вызова из командной строки следует задать следующую последовательность команд:

$ east [-s] [-d] [-f <table_format>] [-a <ast_algorithm>] keyphrases table <keyphrases_file> <directory_with_txt_files>

На вход программе подается адрес текстового файла, в котором записаны словосочетания и адрес директории, в которой хранится коллекция текстов. Каждый текст должен быть сохранен в отдельном текстовом формате. Также программе на вход подается несколько параметров:

- **s**: возможность учета синонимов (в стадии разработки).
- **d**: расчет оценки релевантности без нормализация. По умолчанию используется оценка релевантности с нормализаций, если указан параметр d, то используется оценка релевантности без нормализации.
- **f**: формат выдачи PCT таблицы: либо XML, либо .csv файл.
- **a**: алгоритм построения АСД. По умолчанию используется линейный алгоритм построения АСД, однако, при желании, пользователь может использовать наивный алгоритм.

Если программа запушена из командной строки, то существа два варианта выдачи результатов: либо выдача результатов непосредственно в консоль, либо перенаправление выдачи в текстовый файл. Первый вариант предлагается исключительно для ознакомления с функциональностью программы. Для перенаправления выдачи следует использовать инструкцию > filename.txt.

6.1.2 Использование программы EAST как библиотеки языка Python 2.7

Для вызова из среду Python строки следует задать следующую последовательность команд. Приведенный ниже фрагмент кода показывает, как построить АСД для коллекции из двух строк и посчитать релевантность других двух строк построенному АСД.

```
From east.asts import base #импорт библиотеки
ast = base.AST.get_ast([‘‘XABXAC’’, ‘‘HI’’]) #построение АСД
print ast.score(‘‘ABCI’’) # 0.1875 (подсчет оценки релевантности)
print ast.score(‘‘NOPE’’) # 0 (подсчет оценки релевантности)
```

6.1.3 Структура программы EAST

Программа EAST написана языке Python 2.7 и состоит из 24 модулей, большая часть из которых носит вспомогательный характер: модули для инициализации, тестирования, хранения списка исключений, форматирования выдачи. Перечислим модули, в которых реализован метод АСД и метод построения PCT таблиц:

- east/asts/ast.py – в этом модуле хранится класс AST (Annotated Suffix Tree, АСД) и его методы: добавить узел в дерево, убрать узел из дерева, обойти дерево в глубину, обойти дерево в ширину и вычислить оценку релевантности строки дереву.
- east/asts/ast_naive.py – наивный алгоритм построения АСД.
- east/asts/ast_linear.py – линейный алгоритм построения АСД
- east/asts/utils.py – некоторые полезные утилиты: токенизатор, функция для разбиения текста на строки, функция, которая добавляет уникальные терминальные символы к строкам.

Общая схема работы программы: на вход программе поступает коллекция текстов и коллекция строк-словосочетаний. Программа по очереди обрабатывает все тексты. Каждый текст программа разбивает сначала на токены, потом токены объединяет в строки, добавляет к ним уникальные терминальные символы и получает набор строк для построения АСД по данному тексту. Затем программа инициализирует пустой объект класса AST, назовем его, например, `tree`, и вызывает один из алгоритмов построения АСД, передавая ему на вход коллекцию строк. Алгоритм строит АСД и сохраняет его в существующий объект класса АСД `tree`. После этого программа передает коллекцию словосочетаний-строк `tree`. У класса `tree` есть метод для вычисления оценок релевантности с коллекцией строк, который возвращает вектор оценок релевантности. Полученный вектор программа записывает как столбец в PCT таблицу. После того, как программа обработает все тексты, она окончательно формирует PCT таблицу и выдает ее в командную строку или записывает в выходной файл.

6.2 Утилита WikiDP

Совместно с Н. Левицким и Е. Моренко мы разработали небольшую утилиту WikiDP (Wiki Download and Parse) для извлечения дерева категорий из Википедии. Она устроена так: пользователь вводит название категории из Википедии и получает на выходе дерево категорий, лежащее под этой категорией. В программе WikiDP мы учили все перечисленные трудности. Во-первых, при попадании в новую категорию мы проверяем, была ли уже она посещена, если да, то переходим к следующей категории, если нет, выполняем обход данной категории – так мы избегаем циклы. Во-вторых, мы предлагаем пользователю ввести ограничение на глубину дерева категорий, после чего программа WikiDP показывает пользователю все подкатегории на заданном расстоянии от исходной категории и пользователь может выбрать нужные ему подкатегории. Используя WikiDP мы смогли извлечь из Википедии необходимые статьи.

Пользовательский интерфейс WikiDP представлен на Рис. 6.2. Для работы с утилитой пользователь должен ввести в поле “Название категории” название категории Википедии, которая его интересует. При этом, пробелы следует заменить на знак нижнего подчеркивания “_” . После этого возможны три сценария работы с утилитой:

- Создать дерево категорий: сохранить в файл “data.txt” дерево категорий, корнем которой будет введенная категория;
- Скачать статьи по дереву: извлечь из Википедии все тексты статей, принадлежащих к введенной категории и всем ее подкатегориям. При этом, если дерево категорий содержит цикл, WikiDP не сможет завершить работу и войдет в бесконечный цикл;
- Скачать статьи данной категории: извлечь из Википедии все тексты статей, принадлежащих к введенной категории. Этот сценарий не содержит угрозы бесконечного цикла, поскольку не предполагает перехода в подкатегории.

Флажок “Названия статей” позволяет управлять пользователю содержанием дерева категорий: если он находится в положении “включено”, в дерево категории будут включены названия статей. В обратном случае, названия статей в дерево категорий включены не будут.

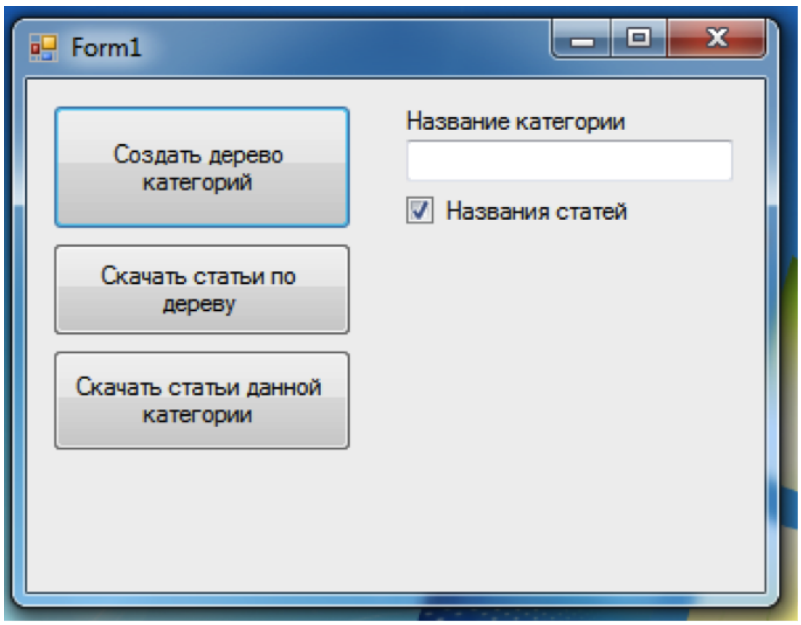

Рисунок 6.2 — Пользовательский интерфейс WikiDP

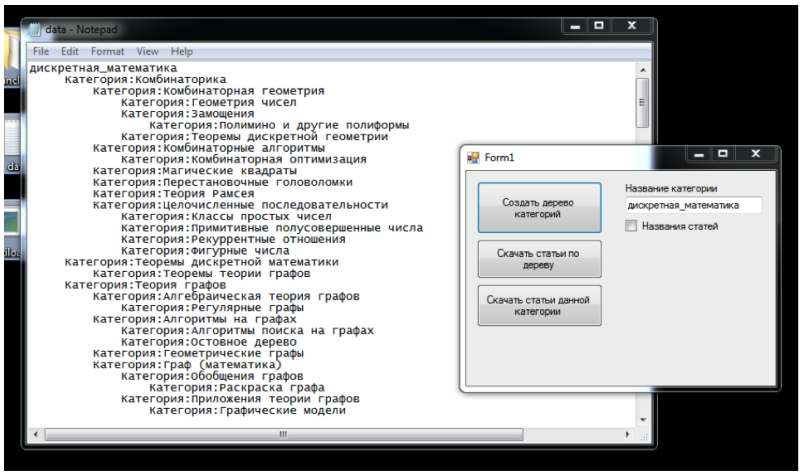

Рисунок 6.3 — Извлеченное дерево категорий категории “Дискретная математика” без названий статей

На Рис. 6.3 приведен пример извлеченного дерева категорий категории “Дискретная математика” без названий статей.

Заключение

В работе предложена теоретико-множественная модель представления коллекций текстовых документов. В отличии от классических моделей – векторной, вероятностной или языковой моделей – в предлагаемой теоретико-множественной модели коллекции текстовых документов текст рассматривается не как набор термов, а как последовательность символов. Представлением текста служат все символьное последовательности фиксированной длины и короче и их частоты. Утверждается, что только такое модельное представление текста позволяет создать меру релевантности «строка – текст», не зависящую от размера входной коллекции и учитывающую нечеткие (то есть, с различием на несколько символов) совпадения между строкой и текстом. Вводится понятие максимального совпадения – такого совпадения между строкой и текстом, которое при добавлении к нему символа слева и справа, перестает быть совпадением. Именно максимальные совпадения и их частоты служат основной для вычисления нечетких оценок релевантности.

Для вычисления частот теоретико-множественной модели предлагается использовать метод аннотированного суффиксного дерева, который позволяет за линейное от размера текста время найти всего его фрагменты заданной длины и короче, а также вычислить их частоты. В работе впервые построена модель нормированного аннотированного суффиксного дерева и введена ассоциированная с ней естественно интерпретируемая мера релевантности СУВСС, а также метод ее вычисления nAST-k. Предлагаемая мера релевантности СУВСС представляет собой среднюю условную частоту символа в максимальном совпадении и позволяет находить оценки релевантности строки тексту, которые

– не зависят от размера входного текста или коллекции текстов;

– учитывают нечеткие совпадения между входной строкой и текстом.

Мера релевантности СУВСС была использована для построения таблиц релевантности «строка – текст», которые в дальнейшем используются для решения конкретных практических задач.

Предложены и верифицированы методы для решения следующих задач:

1. Метод рубрикации научных статей AnnAST в соответствии с системой рубрик, заданной таксономией. Метод позволяет получить для каждой научной статьи некоторое фиксированное количество таксономических

тем, отражающих содержание научной статьи. Множество таксономических тем формируется в соответствии с оценками СУВСС. Показано, что использование СУВСС в задаче рубрикации статей более эффективно, чем использование косинусной меры релевантности и меры релевантности BM 25. Метод AnnAST применен к коллекции аннотаций научных статей журналов ACM по информатике.

2. Метод пополнения таксономии предметной области ReTAST-w. Метод состоит из двух основных этапов. На первом этапе эксперт задает основу таксономии, на втором этапе таксономия автоматически пополняется за счет ресурсов Википедии. Дерево категорий используется для достраивания промежуточных уровней, названия статей – в качестве листьев в новой таксономии, тексты статей – в качестве источника уточнений листьев. При этом, мера релевантности СУВСС используется на двух шагах метода: для очистки данных Википедии от шума и для определения связей между названиями категорий и темами таксономии. Метод применен для пополнения таксономий а) теории вероятностей и математической статистики, б) численных методов, при этом основы таксономии были заданы по паспортам ВАК соответствующих специальностей.
3. Метод фильтрации обсценной лексики fAST. Метод используется для очистки от обсценной лексики собственной коллекции текстов. Устанавливается аналогия между очисткой от обсценной лексики и поиском по однословному ключу с поправкой на оптимизируемый критерий. Демонстрируется эффективность метода fAST по сравнению со стандартными методами поиска по однословным ключам и редакционному расстоянию по полноте и временной эффективности.

В работе приведено описание двух программных комплексов. Программный комплекс WikiDP используется для скачивания статей и дерева категорий русскоязычной Википедии и работает в интерактивном режиме. Программный комплекс EAST полностью реализует pipeline для построения таблиц релевантности «строка – текст», начиная предобработки текстов и разбиения их на строки фиксированной длины и заканчивая вычислением СУВСС и построением искомых таблиц релевантности «строка – текст для входных списков строк и коллекции текстов».

Нельзя не сказать об ограничениях вычислительного плана, связанных с необходимостью создания и поддержки аннотированного суффиксного дерева для сколь-нибудь значительной коллекции тестов. В этом плане нельзя не упомянуть еще одну нашу инновацию – переход от рассмотрения текста как единой строки к рассмотрению его как совокупности коротких строк. В какой-то степени это снижает чувствительность метода, так как разрывает связи между словами, далеко находящимися друг от друга в тексте. Вместе с тем, это существенно понижает глубину получаемого дерева и, следовательно, вычислительную трудоемкость метода. Вероятно, дальнейшие успехи по ускорению работы метода могут получиться на путях его параллелизации и перехода к дистрибутивным вычислениям.

Список литературы

1. *Salton G.*, *Buckley D.* Term Weighting Approaches in Automatic Text Retrieval // Information Processing & Management. — 1988. — Т. 24, № 5. — С. 513—523.
2. *Robertson S.*, *Zaragoza H.* The Probabilistic Relevance Framework: BM25 and Beyond. — Now Publishers Inc., 2009.
3. *Ponte J. M.*, *Croft W. B.* A Language modeling Approach to Information Retrieval // Proc. Conference on Research and Development in Information Retrieval. — ACM. 1998. — С. 275—281.
4. *Zamir O.*, *Etzioni O.* Web Document Clustering: a Feasibility Demonstration // Proc. International Conference on Research and Development in Information Retrieval. — ACM. 1998. — С. 46—54.
5. *Pampapathi R.*, *Mirkin B.*, *Levene M.* A Suffix tree Approach to Anti-spam E-mail Filtering // Machine Learning. — 2006. — Т. 65, № 1. — С. 309—338.
6. *Manning C. D.*, *Schütze H.* Foundations of Statistical Natural Language Processing. Т. 999. — MIT Press, 1999.
7. *Martin D.*, *Jurafsky D.* Speech and Language Processing. An introduction to natural language processing, computational linguistics, and speech recognition. — 2000.
8. *Harris Z. S.* Distributional Structure // Word. — 1954. — Т. 10, № 2—3. — С. 146—162.
9. *Berry M. W.*, *Browne M.* Understanding Search Engines: Mathematical Modeling and Text Retrieval. Т. 17. — Siam, 2005.
10. TF-ICF: A New term Weighting Scheme for Clustering Dynamic Data Streams / J. W. Reed [и др.] // Proc. International Conference on Machine Learning and Applications. — IEEE. 2006. — С. 258—263.
11. *Raghavan V. V.*, *Wong S. K. M.* A Critical Analysis of Vector Space Model for Information Retrieval // Journal of the American Society for information Science. — 1986. — Т. 37, № 5. — С. 279.

12. *Turney P. D.*, *Pantel P.* From Frequency to Meaning: Vector Space Models of Semantics // Journal of Artificial Intelligence Research. — 2010. — Т. 37, № 1. — С. 141—188.

13. *Rehurek R.*, *Sojka P.* Software Framework for Topic Modelling with Large Corpora // Proc. Workshop on New Challenges for NLP Frameworks. — Citeseer. 2010.

14. *Bird S.*, *Klein E.*, *Loper E.* Natural Language Processing with Python. — O'Reilly Media, Inc., 2009.

15. *Manning C. D.*, *Raghavan P.*, *Schütze H.* Introduction to Information Retrieval. Т. 1. — Cambridge university press Cambridge, 2008.

16. *Sebastiani F.* Machine Learning in Automated Text Categorization // ACM computing surveys (CSUR). — 2002. — Т. 34, № 1. — С. 1—47.

17. *Turney P. D.* Thumbs Up or Thumbs Down?: Demantic Orientation Applied to Unsupervised Classification of Reviews // Proc. Annual Meeting on Association for Computational Linguistics. — Association for Computational Linguistics. 2002. — С. 417—424.

18. *Pang B.*, *Lee L.*, *Vaithyanathan S.* Thumbs up?: Sentiment Classification Using Machine Learning Techniques // Proce. Empirical Methods in Natural Language Processing. — Association for Computational Linguistics. 2002. — С. 79—86.

19. *Strapparava C.*, *Valitutti A.* WordNet Affect: an Affective Extension of WordNet. // Proc. Language Resources and Evaluation Conference. Т. 4. — 2004. — С. 1083—1086.

20. *Andrews N. O.*, *Fox E. A.* Recent Developments in Document Clustering. — 2007.

21. *Wong S. K. M.*, *Ziarko W.*, *Wong P. C. N.* Generalized Vector Spaces Model in Information Retrieval // Proc. Conference on Research and Development in Information Retrieval. — ACM. 1985. — С. 18—25.

22. *Pantel P.*, *Lin D.* Discovering Word Senses from Text // Proc. Conference on Knowledge Discovery and Data Mining. — ACM. 2002. — С. 613—619.

23. *Rapp R.* Word Sense Discovery Based on Sense Sescriptor Sissimilarity // Proc. Machine Translation Summit. — 2003. — С. 315—322.

24. Combining Independent Modules to Solve Multiple-choice Synonym and Analogy Problems / P. Turney [и др.]. — 2003.

25. Indexing by Latent Semantic Analysis / S. Deerwester [и др.] // Journal of the American Society for Information Science. — 1990. — Т. 41, № 6. — С. 391.

26. *Blei D. M.*, *Ng A. Y.*, *Jordan M. I.* Latent Dirichlet Allocation // Journal of Machine Learning Research. — 2003. — Т. 3. — С. 993—1022.

27. *Yu C. T.*, *Salton G.* Precision Weighting – an Effective Automatic Indexing Method // Journal of the ACM. — 1976. — Т. 23, № 1. — С. 76—88.

28. A Neural Probabilistic Language Model / Y. Bengio [и др.] // journal of Machine Learning Research. — 2003. — Т. 3, Feb. — С. 1137—1155.

29. *Mikolov T.*, *Dean J.* Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems. — 2013.

30. *Koehn P.*, *Och F. J.*, *Marcu D.* Statistical Phrase-based Translation // Proc. Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. — Association for Computational Linguistics. 2003. — С. 48—54.

31. *Katz S. M.* Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer // Transactions on Acoustics, Speech and Signal Processing. — 1987. — Т. 35, № 3. — С. 400—401.

32. *Brill E.*, *Moore R. C.* An Improved Error Model for Noisy Channel Spelling Correction // Proce. Annual Meeting on Association for Computational Linguistics. — Association for Computational Linguistics. 2000. — С. 286—293.

33. *Hofmann T.* Probabilistic Latent Semantic Indexing // Proc. International Conference on Research and Development in Information Retrieval. — ACM. 1999. — С. 50—57.

34. *Hofmann T.* Latent Semantic Models for Collaborative Filtering // Transactions on Information Systems. — 2004. — Т. 22, № 1. — С. 89—115.

35. Using Probabilistic Latent Semantic Analysis for Personalized Web Search / C. Lin [и др.] // Web Technologies Research and Development-APWeb 2005. — Springer, 2005. — С. 707—717.

36. *Berry M. W.*, *Dumais S. T.*, *O'Brien G. W.* Using Linear Algebra for Intelligent Information Retrieval // Society for Industrial and Applied Mathematics Review. — 1995. — Т. 37, № 4. — С. 573—595.

37. *Wei X.*, *Croft W. B.* LDA-based Document Models for Ad-Hoc Retrieval // Proc. International Conference on Research and Development in Information Retrieval. — ACM. 2006. — С. 178—185.

38. *Kim H.*, *Howland P.*, *Park H.* Dimension Reduction in Text Classification with Support Vector Machines // Journal of Machine Learning Research. — 2005. — С. 37—53.

39. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora / D. Ramage [и др.] // Proc. Conference on Empirical Methods in Natural Language Processing. — Association for Computational Linguistics. 2009. — С. 248—256.

40. *Gong Y.*, *Liu X.* Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis // Proc. International Conference on Research and Development in Information Retrieval. — ACM. 2001. — С. 19—25.

41. *Arora R.*, *Ravindran B.* Latent Dirichlet Allocation Based Multi-document Summarization // Proc. Workshop on Analytics for Noisy Unstructured Text Data. — ACM. 2008. — С. 91—97.

42. *Gee K. R.* Using Latent Semantic Indexing to Filter Spam // Proc. Symposium on Applied Computing. — ACM. 2003. — С. 460—464.

43. *Bıró I.and Szabó J.*, *Benczúr A. A.* Latent Dirichlet Allocation in Web Spam Filtering // Proc. International Workshop on Adversarial Information Retrieval on the Web. — ACM. 2008. — С. 29—32.

44. *Krestel R.*, *Fankhauser P.*, *Nejdl W.* Latent Dirichlet Allocation for Tag Recommendation // Proc. Conference on Recommender Systems. — ACM. 2009. — С. 61—68.

45. *Monay F.*, *Gatica-Perez D.* On Image Auto-annotation with Latent Space Models // Proc. International Conference on Multimedia. — ACM. 2003. — С. 275—278.

46. *Wang X.*, *Grimson E.* Spatial Latent Dirichlet Allocation // Advances in Neural Information Processing Systems. — 2008. — С. 1577—1584.

47. *Gomaa W. H.*, *Fahmy A. A.* A Survey of Text Similarity Approaches // International Journal of Computer Applications. — 2013. — Т. 68, № 13.

48. *Krause E. F.* Taxicab Geometry: an Adventure in Non-Euclidean Geometry. — Courier Corporation, 2012.

49. *Dice L. R.* Measures of the Amount of Ecologic Association between Species // Ecology. — 1945. — Т. 26, № 3. — С. 297—302.

50. *Jaccard P.* Etude Comparative de la Distribution Florale Dans Une Portion des Alpes et du Jura. — Impr. Corbaz, 1901.

51. *Sempson G. G.* Holarctic Mammalian Faunas and Continental Relationships During the Cenozoic // Geological Society of America Bulletin. — 1947. — Т. 58, № 7. — С. 613—688.

52. *Cheetham A. H.*, *Hazel J. E.* Binary (Presence – Absence) Similarity Coefficients // Journal of Paleontology. — 1969. — С. 1130—1136.

53. *Weiner P.* Linear Pattern Matching Algorithms // Proc. Annual Symposium on Switching and Automata Theory. — IEEE. 1973. — С. 1—11.

54. Suffix Trees for Very Large Genomic Sequences / M. Barsky [и др.] // Proc. Conference on Information and Knowledge Management. — ACM. 2009. — С. 1417—1420.

55. *Grossi R.*, *Vitter J. S.* Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching // Journal on Computing. — 2005. — Т. 35, № 2. — С. 378—407.

56. *Hu Z.*, *Zhang Y.*, *Zhou J. F.* Method for Extracting Name Entities and Jargon Terms using a Suffix Tree Data Structure. — Март 2007. — US Patent 7,197,449.

57. *Chim H.*, *Deng X.* A New Suffix Tree Similarity Measure for Document Clustering // Proc. International Conference on World Wide Web. — ACM. 2007. — С. 121—130.

58. Finding Surprising Patterns in Textual Data Streams / T. Snowsill [и др.] // Proc. Workshop on Cognitive Information Processing. — IEEE. 2010. — С. 405—410.

59. *Gusfield D.* Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. — Cambridge University Press, 1997.

60. *Дубов М. С.*, *Черняк Е. Л.* Аннотированные Суффиксные деревья: Особенности Реализации // Сборник Всеройссийской Конференции Молодых Ученых Анализ Изображений, Сетей и Текстов. — 2013.

61. *Hjørland B.* The Foundation of the Concept of Relevance // Journal of the American Society for Information Science and Technology. — 2010. — Т. 61, № 2. — С. 217—237.

62. *Сегалович И.* Как Работают Поисковые Системы // Мир Internet. — 2002. — Т. 10. — С. 24—32.

63. *Taylor A.* User Relevance Criteria Choices and the Information Search Process // Information Processing & Management. — 2012. — Т. 48, № 1. — С. 136—153.

64. *Zhai C.*, *Lafferty J.* A Study of Smoothing methods for Language Models Applied to Ad Hoc Information Retrieval // Proc. International Conference on Research and Development in Information Retrieval. — ACM. 2001. — С. 334—342.

65. *Миркин Б. Г.*, *Черняк Е. Л.*, *Чугунова О. Н.* Метод Аннотированного Суффиксного Дерева для Оценки Степени Вхождения Строк в Текстовые Документы // Бизнес-информатика. — 2012. — 3 (21).

66. *Миркин Б. Г.*, *Черняк Е. Л.* Использование Мер Релевантности Строка – Текст для Автоматизации Рубрикации Научных Статей // Бизнес-информатика. — 2014. — 2 (28).

67. *Pissanetzky S.* Sparse Matrix Technology. — Academic Press, 1984.

68. *Ceci M.*, *Malerba D.* Classifying Web Documents in a Hierarchy of Categories: a Comprehensive Study // Journal of Intelligent Information Systems. — 2007. — Т. 28, № 1. — С. 37—78.

69. *Santos A. P.*, *Rodrigues F.* Multi-label Hierarchical Text Classification Using the ACM taxonomy // 14th Portuguese Conference on Artificial Intelligence. — 2009. — С. 553—564.

70. *Cantador I.*, *Bellogın A.*, *Vallet D.* Content-based Recommendation in Social Tagging Systems // Proc. Conference on Recommender Systems. — ACM. 2010. — С. 237—240.

71. *Gupta A.*, *Kumaraguru P.* Credibility Ranking of Tweets During High Impact Events // Proc. Workshop on Privacy and Security in Online Social Media. — ACM. 2012. — С. 2.

72. Listwise Approach to Learning to Rank: Theory and Algorithm / F. Xia [и др.] // Proc. Conference on Machine learning. — ACM. 2008. — С. 1192—1199.

73. *Duh K.*, *Kirchhoff K.* Learning to Rank with Partially-labeled Data // Proc. Conference on Research and Development in Information Retrieval. — ACM. 2008. — С. 251—258.

74. *Porter M. F.* An Algorithm for Suffix Stripping // Program. — 1980. — Т. 14, № 3. — С. 130—137.

75. *Bird S.* NLTK: the Natural Language Toolkit // Proc. Interactive Presentation Sessions. — Association for Computational Linguistics. 2006. — С. 69—72.

76. *Miller G. A.* WordNet: a Lexical Database for English // Communications of the ACM. — 1995. — Т. 38, № 11. — С. 39—41.

77. *Robinson P. N.*, *Bauer S.* Introduction to Bio-ontologies. — CRC Press, 2011.

78. *Лукашевич Н. В.* Тезаурусы в Задачах Информационного Поиска // М.: Издательство МГУ, 2011. — М.: Издательство МГУ, 2011. 2010.

79. Clustering query refinements by user intent / E. Sadikov [и др.] // International Conference on World Wide Web. — ACM. 2010. — С. 841—850.

80. *White R. W.*, *Bennett P. N.*, *Dumais S. T.* Predicting Short-Interests Using Activity-based Search Context // Proc. International Conference on Information and Knowledge Management. — ACM. 2010. — С. 1009—1018.

81. Automatic Taxonomy Construction from Keywords / X. Liu [и др.] // Proc. International Conference on Knowledge Discovery and Data Mining. — ACM. 2012. — С. 1433—1441.

82. *Ding C.* A Survey of Ontology Construction and Information Extraction from Wikipedia. — 2010.

83. Dbpedia: a Nucleus for a Web of Open Data / S. Auer [и др.]. — Springer, 2007.

84. *Chernyak E.*, *Mirkin B.* A Method for Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources // Procedia Computer Science. — 2014. — Т. 31. — С. 193—200.

85. *Van Hage W. R.*, *Katrenko S.*, *Schreiber G.* A Method to Combine Linguistic Ontology-Mapping Techniques // The Semantic Web. — Springer, 2005. — С. 732—744.

86. YAGO2: a Spatially and Temporally Enhanced Knowledge Base From Wikipedia / J. Hoffart [и др.] // Artificial Intelligence. — 2013. — Т. 194. — С. 28—61.

87. Mining Concepts from Wikipedia for Ontology Construction / G. Cui [и др.] // Proc. International Joint Conference on Web Intelligence and Intelligent Agent Technology. — IEEE Computer Society. 2009. — С. 287—290.

88. *Ponzetto S. P.*, *Strube M.* Deriving a Large Scale Taxonomy from Wikipedia // Proc. National Conference on Artificial Intelligence. Т. 7. — 2007. — С. 1440—1445.

89. *Xavier C. C.*, *De Lima V. L. S.* A Semi-automatic Method for Domain Ontology Extraction from Portuguese Language Wikipedia's Categories // Advances in Artificial Intelligence. — Springer, 2010. — С. 11—20.

90. *Jiang S.*, *Bing L.*, *Zhang Y.* Towards an Enhanced and Adaptable Ontology by Distilling and Assembling Online Encyclopedias // Proc. International Conference on Information & Knowledge Management. — ACM. 2013. — С. 1703—1708.

91. *Kittur A.*, *Chi E. H.*, *Suh B.* What's in Wikipedia?: Mapping Topics and Conflict Using Socially Annotated Category Structure // Conference on human factors in computing systems. — ACM. 2009. — С. 1509—1512.

92. *Hulth A.* Improved Automatic Keyword Extraction Given More Linguistic Knowledge // Conference on Empirical Methods in Natural Language Processing. — Association for Computational Linguistics. 2003. — С. 216—223.

93. *Тихомиров И. А.*, *Соченков И. В.* Метод Динамической Контентной Фильтрации Сетевого Трафика на Основе Анализа Текстов на Естественном Языке // Вестник НГУ, Информационные технологии. — 2008. — Т. 6, № 2. — С. 94—100.

94. *Dong C.*, *Agarwal A.* WS 2 F: A Weakly Supervised Framework for Data Stream Filtering // Proc. International Conference on Big Data. — IEEE. 2014. — С. 50—57.

95. Text Normalization and Semantic Indexing to Enhance Instant Messaging and SMS Spam Filtering / T. A. Almeida [и др.] // Knowledge-Based Systems. — 2016.

96. Технология Фильтрации Содержания для Интернет / И. Ашманов [и др.] // Труды Международного Семинара «Диалог. — 2002.

97. *Левенштейн В. И.* Двоичные Коды с Исправлением Выпадений, Вставок и Замещений Символов // Доклады Академий Наук СССР. — 1965. — Т. 163, № 4. — С. 845—848.

98. *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages // Proc. Analysis of Images, Social Networks and Texts. — Springer, 2015. — С. 320—332.

99. *Segalovich I.* A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine // Proc. International Conference on Machine Learning; Models, Technologies and Applications. — Citeseer. 2003. — С. 273—280.

100. *Ильвовский Д. А.*, *Черняк Е. Л.* Системы Автоматической Обработки Текстов // Открытые системы. — 2014. — № 1. — С. 51—53.

101. *Мальковский М. Г.*, *Грацианова Т. Ю.*, *Полякова И. Н.* Прикладное Программное Обеспечение: Системы Автоматической Обработки Текстов // М.: МГУ. — 2000.

102. The Stanford CoreNLP Natural Language Processing Toolkit / C. D. Manning [и др.] // ACL (System Demonstrations). — 2014. — С. 55—60.

103. Scikit-learn: Machine Learning in Python / F. Pedregosa [и др.] // Journal of Machine Learning Research. — 2011. — Т. 12, Oct. — С. 2825—2830.

104. Globally Normalized Transition-based Neural Networks / D. Andor [и др.] // arXiv preprint arXiv:1603.06042. — 2016.

105. *Bär D.*, *Zesch T.*, *Gurevych I.* DKPro Similarity: An Open Source Framework for Text Similarity. // ACL (System Demonstrations). — 2013. — С. 121—126.

Список рисунков

Список таблиц