ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «УРАЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»

На правах рукописи

Бондарчук Дмитрий Вадимович

АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА НА ОСНОВЕ МЕТОДА КАТЕГОРИАЛЬНЫХ ВЕКТОРОВ

Специальность 05.13.17—
«Теоретические основы информатики»

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
ТИМОФЕЕВА Галина Адольфовна
доктор физико-математических наук, профессор

Екатеринбург — 2016

Оглавление

Стр.

Введение

Актуальность темы. В последнее десятилетие интеллектуальный анализ текстовых данных получил широкое распространение в связи потребностью многих отраслей экономики и науки в систематизации и автоматической категоризации больших объемов таких данных. Одним из самых перспективных подходов к решению задач автоматического поиска является подход, основанный на машинном обучении.

Для классификации (автоматического распределения текстовых документов по рубрикам) в последнее время все чаще используется векторная модель представления знаний, а так же методы, основанные на латентно-семантическом анализе. Это современный инструмент анализа текстов, определяющий значимость термов и отсеивающий малозначимые.

На данный момент методы, основанные на латентно-семантическом анализе, успешно применяются в информационном поиске, категоризации и кластеризации документов, обнаружении спама. На их основе были достигнуты значительные успехи в выявлении трендов в научных публикациях и новостных потоках, в разработке рекомендательных систем, в решении других задач интеллектуального анализа текстов.

Однако, несмотря на значительный успех, алгоритмы, основанные на латентно-семантическом анализе, не лишены недостатков. Одним из них является использование модели «мешка слов», в которой каждый документ представляется в виде множества не связанных между собой слов. Данная модель не учитывает порядок слов и основывается на гипотезе независимости появлений слов друг от друга в текстах. Это предположение оправдано с точки зрения вычислительной эффективности, но оно далеко от реальности. Так же, можно сказать, что многие существующие модели не учитывают взаимодействие элементов информации между собой и отношение пользователя к знанию, вследствие чего снижается релевантность поиска.

Таким образом, **актуальной** является задача улучшения качества интеллектуального анализа текстов за счет учета семантической и лексикографической взаимосвязи термов и решения проблемы лексической многозначности и

разработки методов, обеспечивающих непустой результат для любой обучающей выборки.

Степень разработанности темы. В настоящее время исследованию интеллектуального анализа текстов и развитию методов автоматической классификации и кластеризации посвящен ряд работ, подавляющее большинство из которых основано на векторной модели представления знаний, а так же на использовании семантических сетей. Источниками при проведении диссертационного исследования послужили труды отечественных и зарубежных ученых по основам интеллектуального анализа данных: труды T. Landauer, S. Deerwester, S. Streeter, А.Д. Хомоненко, И.С. Некрестьянова и А.Н. Соловьева по методу латентно-семантического анализа и методу представления знаний с помощью терм-документной матрицы, труды M. Minsky и К.В. Воронцова по вероятностным алгоритмам, труды G. Salton, С.В. Моченова, А.М. Бледнова и Ю.А. Луговских по векторной модели представления знаний и труды G. Miller, C. Fellbaum, Н.В. Лукашевич, Б.В. Доброва по семантическим БД, труды С.О. Кузнецова, Д.А. Ильвовского, А.В. Бузмакова, Д.В. Гринченкова, Б.Ю. Лемешко, С.Н. Постовалова по обработке текстовых данных на основе решеток замкнутых описаний и таксономий.

На сегодняшний день область научных исследований, связанная с применением машинного обучения в задачах информационного поиска, продолжает активно развиваться. Алгоритмы классификации текстов, основанные на традиционных методах недостаточно учитывают семантическую и лексикографическую взаимосвязи термов и не обеспечивают непустой результат для неравномерных выборок.

Цель и задачи исследования. *Целью* данной работы являлась разработка алгоритма интеллектуального анализа текстов, гарантирующего, что пользователь на любой свой запрос получит непустую выборку, отсортированную по степени «полезности».

Для достижения поставленной цели были поставлены следующие *задачи*:

1. Разработка модели образа текстового документа и соответствующего метода отображения текста в семантическое пространство, обеспечивающих компактное представление документа в оперативной памяти.

2. Разработка алгоритма интеллектуального анализа текстов, гарантирующего непустой результат независимо от распределения обучающей выборки по категориям.
3. Разработка алгоритма перевзвешивания векторной модели представления знаний для учета семантической взаимосвязи между термами.
4. Проведение сравнительных экспериментов, оценивающих эффективность разработанных методов и подходов по сравнению с существующими.

Научная новизна работы заключается в разработке автором оригинального способа формирования семантического пространства, основанного на использовании матрицы корреспонденций термов (МКТ), которая подвергается ортогональному разложению, и метода перехода к категориальным векторам для компактного отображения документов с переопределением исходных весов термов с помощью вычисления семантической взаимосвязи между термами.

Теоретическая ценность работы состоит в том, что в ней проведен сравнительный анализ свойств сингулярного разложения терм-документной матрицы (ТДМ) и ортогонального разложения МКТ. Доказано, что термы, содержащиеся только в коротких документах, отбрасываются при использовании сингулярного разложения ТДМ, но учитываются при использовании предлагаемого подхода. Получены условия совпадения сингулярного разложения терм-документной матрицы, соответствующей всей коллекции, с разложением матрицы, содержащей только длинные документы. **Практическая ценность** работы заключается в том, что результаты работы являются основой для разработки поисковых систем, использующих интеллектуальный анализ текстовых данных. Предложенные в работе алгоритмы позволяют производить поиск, классификацию и формировать персональные рекомендации пользователю, а так же выдавать ему результат, упорядоченный по степени соответствия его запросу.

Методы исследования. Методологической основой исследования являются методы линейной алгебры, статистического и системного анализа, интеллектуального анализа текстов, семантического анализа.

Положения выносимые на защиту. На защиту выносятся следующие новые научные результаты:

1. Разработаны модель образа текстового документа и соответствующий метод отображения текста в семантическое пространство, обеспечива-

ющие компактное представление документа в оперативной памяти на основе матрицы корреспонденций термов, которая подвергается ортогональному разложению.

2. Разработан алгоритм интеллектуального анализа текстов, гарантирующий непустой результат независимо от распределения обучающей выборки по категориям на основе использования вычисления категориальных векторов для упорядочения результирующей выборки по степени релевантности запросу пользователя.
3. Предложен метод перевзвешивания термов векторной модели с помощью вычисления их семантической взаимосвязи друг с другом на основе авторской адаптации алгоритма Леска.
4. На основе разработанных методов и подходов реализован алгоритм подбора рекомендаций. Проведены вычислительные эксперименты, подтверждающие более высокую эффективность разработанного алгоритма по сравнению с существующими.

Степень достоверности результатов. Все утверждения, связанные со свойствами ортогонального разложения матицы корреспонденций термов, сформулированы в виде теорем и снабжены строгими доказательствами. Теоретические построения подтверждены тестами, проведенными в соответствии с общепринятыми методиками.

Апробация работы. Основные результаты работы докладывались на:

1. Научно-практической конференции «Дни науки ОТИ НИЯУ МИФИ-2012» (Озерск, ОТИ НИЯУ МИФИ, 2012).
2. Научно-практической конференции «Дни науки ОТИ НИЯУ МИФИ-2013» (Озерск, ОТИ НИЯУ МИФИ, 2013).
3. Научно-практической конференции «Математические методы решения исследовательских задач» (Екатеринбург, УрГУПС, 2013).
4. Научно-практической конференции «Актуальные проблемы автоматизации и управления» (Челябинск, ЮУрГУ, 2014).
5. Международной (46-ой Всероссийской) школе-конференции "Современные проблемы математики и ее приложений"(ИММ УрО РАН, Екатеринбург, 2015).

6. IX Международной научно-практическая конференция «Отечественная наука в эпоху изменений: постулаты прошлого и теории нового времени» (Национальная ассоциация ученых, Екатеринбург, 2015)
7. 41st International Conference «Applications of Mathematics in Engineering and Economics» (Sozopol, Bulgaria, 2015).
8. International Conference and PhD Summer School "Groups and Graphs, Algorithms and Authomata"(Екатеринбург, 2015)
9. Международной (47-ой Всероссийской) школе-конференции "Современные проблемы математики и ее приложений"(ИММ УрО РАН, Екатеринбург, 2016).

Публикации по теме диссертации

Основные результаты по теме диссертации изложены в следующих печатных работах:

Статьи в журналах из перечня ВАК

1. Бондарчук Д. В. Статистический способ определения семантической близости термов // Системы управления и информационные технологии. – 2015. – Т. 61, № 3. – С. 55–57.

2. Бондарчук Д. В. Алгоритм построения семантического ядра для текстового классификатора // В мире научных открытий. – 2015. – Т. 68, № 8.2. – С. 713–724.

3. Бондарчук Д. В., Тимофеева Г. А. Выделение семантического ядра на основе матрицы корреспонденций термов // Системы управления и информационные технологии. – 2015. – Т. 61, № 3.1. – С. 134–139.

4. Бондарчук Д. В., Тимофеева Г. А. Применение машинного обучения для формирования персональных рекомендаций в сфере трудоустройства // Экономика и менеджмент систем управления. – 2015. – Т. 18, № 4.2. – С. 215—221.

5. Бондарчук Д. В., Тимофеева Г. А. Математические основы метода категориальных векторов в интеллектуальном анализе данных // Вестник Уральского государственного университета путей сообщения. – 2015. – 4(28). – С. 4–8.

Статьи в изданиях, индексируемых в Scopus и Web of Science

6. Bondarchuk D. V., Timofeeva G. A. Vector space model based on semantic relatedness // AIP Conference Proceedings, Vol. 1690, Proceedings of 41st International Conference "Applications of Mathematics in Engineering and Economics"(AMEE'15). – 2015. – Pp. 1–5.

7. Bondarchuk D.V., Martynenko A.V. Spectral properties of a matrix of correspondences between terms // CEUR Workshop Proceedings, Vol. 1662, Proceedings of 47th International Youth School-Conference "Modern Problems in Mathematics and its Applications"(MPMA 2016). – 2016. – Pp. 186–190.

Статьи в изданиях, индексируемых в РИНЦ

8. Бондарчук Д. В. Использование латентно-семантического анализа в задачах классификации текстов по эмоциональной окраске // Бюллетень результатов научных исследований. – 2012. – 2(3). – С. 146–151.

9. Бондарчук Д. В. Выбор оптимального метода интеллектуального анализа данных для подбора вакансий // Информационные технологии моделирования и управления. – 2013. – 6(84). – С. 504–513.

10. Бондарчук Д. В. Интеллектуальный метод подбора персональных рекомендаций, гарантирующий получение непустого результата // Информационные технологии моделирования и управления. – 2015. – Т. 2(92).–С. 130–138.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и двух приложений. Полный объем диссертации составляет 141 страницу с 17 рисунками и 26 таблицами. Список литературы содержит 124 наименования.

Содержание работы. Во **введении** обоснована актуальность темы диссертации, изложены цель и задачи исследования, научная новизна и практическая ценность полученных результатов.

В первой главе, «Основные методы интеллектуального анализа текстов», рассматриваются тенденции развития интеллектуального анализа текстов и дается обзор научных исследований в области современных методов. Особое внимание уделяется латентно-семантическому анализу и использованию семантических сетей.

Во второй главе, «Интеллектуальный метод подбора персональных рекомендаций гарантирующий получение непустого результата», предлагается новый метод интеллектуального анализа текстов, который на любой запрос пользователя, независимо от размера и равномерности обучающей выборки дает пользователю непустой ответ, отсортированный по степени релевантности запросу пользователя.

В качестве модели представления рассматривается векторная модель, в которой каждый текстовый документ из коллекции представляется, как вектор в векторном пространстве. Алгоритм позволяет получить выборку, отсортированную по степени «полезности» конечному пользователю. Предлагаемый способ хорош в первую очередь тем, что в случае, когда данные распределены между категориями неравномерно, пользователь получит непустой результат.

Третья глава, «Векторная модель представления знаний использующая семантическую близость термов», посвящена применению семантической близости термов при обучении классификатора, а именно перевзвешиванию весов термов векторной модели представления знаний. Для вычисления семантической близости термов используется авторская адаптация расширенного алгоритма Леска.

Векторная модель с учетом семантической близости термов решает проблему неоднозначности синонимов. Чтобы учесть семантическую связь между терминами, вес термина в документе будем рассчитывать несколько иначе, чем в классической векторной модели представления знаний. Настройка весов термов производится с помощью вычисления семантической близости связанных термов.

Предлагается способ вычисления семантической близости, основанный на предположении, что семантически близкие термы употребляются в одинаковых

или схожих контекстах. В главе предлагается способ вычисления семантической близости между двумя словами или фразами, основанный на статистическом подходе. Главная идея состоит в том, что связность между словами удобнее представлять в виде *контекстного множества*, т.е. множества слов, связанных с заданным термином.

В четвертой главе, «Вычислительные эксперименты», описываются эксперименты по исследованию эффективности разработанных в диссертации моделей, методов и алгоритмов.

Для оценки эффективности векторной модели представления знаний учитывающую семантическую близость термов использовались известные меры оценки качества классификаторов *F-measure* и *purity*.

В заключении в краткой форме излагаются итоги выполненного диссертационного исследования, представляются отличия диссертационной работы от ранее выполненных родственных работ других авторов, даются рекомендации по использованию полученных результатов и рассматриваются перспективы дальнейшего развития темы.

Глава 1. Основные методы интеллектуального анализа текстов

Интеллектуальный анализ данных в последние годы получил широкое распространение в связи с увеличением количества документов, хранящихся в электронном виде, и возникшей необходимостью их упорядочения. Наиболее перспективным подходом к решению задач данного класса является применение технологий, основанных на машинном обучении.

В настоящее время существует множество методов интеллектуального анализа текстов. Большинство этих методов основано на одном из 3-х основных подходов: вероятностном подходе [17, 72, 75], искусственных нейронных сетях [57, 71, 79, 81], деревьях решений[106, 112]. Главными требованиями к методу извлечения знаний являются эффективность и масштабируемость, поскольку в большинстве случаев они применяются для анализа больших объемов данных. Кроме того следует помнить, что данные зачастую зашумлены, что в свою очередь может создать дополнительные проблемы для анализа.

1.1 Модели представления знаний

1.1.1 Векторная модель представления знаний

Одной из наиболее значимых проблем в области компьютерных алгоритмов является проблема извлечения "смысла"из текста на естественном языке и представление его в удобном для обработки компьютером виде. Наиболее известным и простым способом представления знаний является векторная модель.

Впервые векторное представление знаний было представлено Джерардом Салтоном. Данный способ представления знаний был разработан для системы поиска информации SMART [76]. Формальное представление и алгоритм формирования данной модели был впервые опубликован в статье [104]. Вектора и ранее использовались в системах интеллектуального анализа текстов, новшество данной модели заключалось в том, что в качестве компонент векторов

впервые стали использоваться частоты вхождений термов в документ из коллекции. *Термом* будем считать значимое слово предметной области, обработанное с помощью стеммера Портера. Автоматическое извлечение термов из текстовой коллекции с помощью методов машинного обучения подробно рассмотрено в [7].

Идея использования модели состоит в том, чтобы представить каждый текстовый документ из коллекции, как точку в пространстве (или вектор в векторном пространстве). Документы (точки), которые в пространстве располагаются близко друг к другу, считаются семантически схожими. Запрос пользователя, либо характеристики пользователя (в случае системы формирования персональных рекомендаций) представляется в виде точки в том же самом пространстве, что и все документы.

Векторная модель — это представление коллекции документов в информационном поиске векторами из одного общего для всей коллекции векторного пространства [112].

В простейшем случае векторная модель предполагает сопоставление каждому документу частотного спектра слов и соответственно вектора в лексическом пространстве. В процессе поиска частотный портрет запроса рассматривается как вектор в том же пространстве и по степени близости (расстоянию или углу между векторами) определяются наиболее релевантные документы. В более продвинутых векторных моделях [110] размерность пространства сокращается отбрасыванием наиболее распространенных или редко встречающихся слов, увеличивая тем самым процент значимости основных слов.

Располагая таким представлением для всех документов, можно, например, находить расстояние между точками пространства и тем самым решать задачу подобия документов - чем ближе расположены точки, тем больше похожи соответствующие документы [88]. В случае поиска документа по запросу, запрос тоже представляется как вектор того же пространства - и можно вычислять соответствие документов запросу [90].

На основе векторного представления могут быть решены некоторые проблемы обработки текстовой информации, в частности:

- сокращение объема исходной информации для выполнения процедур анализа текста и формирования систем и баз знаний;
- синтез текста с использованием информации, извлекаемой из баз знаний.

1.1.2 Терм-документная матрица

Терм-документная матрица представляет собой математическую матрицу, описывающую частоту терминов, которые встречаются в коллекции документов. В терм-документной матрице строки соответствуют документам в коллекции, а столбцы соответствуют терминам [103]. Существуют различные схемы для определения значения каждого элемента матрицы. Одной из таких является схема $tf.idf$ [27, 110]. Они полезны в области обработки естественного языка, особенно в методах латентно-семантического анализа.

tf.idf — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Предполагается, что вес слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции [27].

В дополнение к стандартной частоте терма tf (term frequency), используемой в классических векторых алгоритмах [102—104], вводится специальная мера *idf* (inverce document frequency), указывающую на количество документов, в которых встречается каждое слово из словаря. Этот показатель позволяет сгладить результат в случае частого употребления терма в различных документах.

1.1.3 Наивная байесовская модель

Главное предназначение вероятностной модели — определение вероятностей наступления некоторых событий. Поэтому в основе вероятностных моделей лежит теория вероятности и использование ее базовых элементов, таких как теорема Байеса [25]. Основой для вероятностного метода обучения классификатора является наивная байесовская модель. Пусть документы разбиты на несколько классов c_1,...,c_k, C — общее множество классов. Суть ее заключается в том, что. вероятность того, что документ d попадет в класс c, записывается как $P(c|d)$:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{1.1}$$

где $P(d|c)$ — вероятность встретить документ d среди всех документов класса c, $P(c)$ — безусловная вероятность встретить документ класса c в корпусе документов, $P(d)$ — безусловная вероятность документа d в корпусе документов. Чтобы оценить условную вероятность $P(d|c) = P(t_1, t_2, \ldots, t_n|c)$, где t_k — терм из документа d, n - общее количество термов в документе (включая повторения), необходимо ввести упрощающие предположения об условной независимости термов и о независимости позиций термов. Другими словами, мы пренебрегаем, во-первых, тем фактом, что в тексте на естественном языке появление одного слова часто тесно связано с появлением других слов (вероятнее, что слово интеграл встретится в одном тексте со словом уравнение, чем со словом бактерия), и, во-вторых, что вероятность встретить одно и то же слово различна для разных позиций в тексте. Именно из-за этих упрощений рассматриваемая модель естественного языка называется наивной (тем не менее она является достаточно эффективной в задаче классификации [17]).

Таким образом, вероятностные модели предоставляют удобные средства прогнозирования наступления различных событий.

1.1.4 Семантическая сеть

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (ребра) задают отношения между ними. Объектами могут быть понятия, события, свойства, процессы [50, 55, 84, 100].

Особенность дуг между узлами заключается в том, что они имеют некоторую смысловую нагрузку, выраженную в именовании связи. Наиболее общими и часто встречающимися являются связи, обозначающие "часть-целое"(part-of), конкретный объект — класс объектов (is-a), подкласс-класс (a-kind-of, ako).

Для всех семантических сетей справедливо разделение по арности и количеству типов отношений [63].

По количеству типов отношений, сети могут быть однородными и неоднородными.

Однородные сети обладают только одним типом отношений (стрелок), например, таковой является вышеупомянутая классификация биологических видов [70, 73].

В неоднородных сетях количество типов отношений больше двух. Классические иллюстрации данной модели представления знаний представляют именно такие сети. Неоднородные сети представляют больший интерес для практических целей, но и большую сложность для исследования. Неоднородные сети можно представлять как переплетение древовидных многослойных структур [122].

По арности:

– типичными являются сети с бинарными отношениями (связывающими ровно два понятия).

  Бинарные отношения очень просты и удобно изображаются на графе в виде стрелки между двух концептов. Кроме того, они играют исключительную роль в математике [100, 107]. На практике, однако, могут понадобиться отношения, связывающие более двух объектов

– N-арные.

  При этом возникает сложность — как изобразить подобную связь на графе, чтобы не запутаться. Концептуальные графы (см. ниже) снимают это затруднение, представляя каждое отношение в виде отдельного узла [113].

По размеру [96]:

– Для решения конкретных задач, например, тех которые решают системы искусственного интеллекта.
– Семантическая сеть отраслевого масштаба должна служить базой для создания конкретных систем, не претендуя на всеобщее значение.
– Помимо концептуальных графов существуют и другие модификации семантических сетей, это является еще одной основой для классификации (по реализации).

1.2 Методы интеллектуального анализа текстов

Самый старый способ анализа данных — ручной анализ, выполняемый без использования средств вычислительной техники. Этот метод трудоемкий и неприемлем в случаях, когда необходимо анализировать с высокой скоростью значительное количество информации.

Другой подход заключается в написании правил и регулярных выражений, по которым можно отнести анализируемую информацию к той или иной категории. Например, одно из таких правил может выглядеть следующим образом: «если текст содержит слова производная и уравнение, то отнести его к категории математика». Специалист, знакомый с предметной областью и обладающий навыком написания регулярных выражений, может составить ряд правил, которые затем автоматически применяются к поступающим документам для их классификации [4]. Этот подход лучше предыдущего, поскольку процесс классификации автоматизируется и, следовательно, количество обрабатываемой информации практически не ограничено. Однако создание и поддержание правил в актуальном состоянии требует постоянных усилий специалиста.

При машинном анализе информации набор правил и общий критерий принятия решения текстового классификатора, вычисляется автоматически, обучая классификатор стандартными общепринятыми словами, фразами или количественной оценкой. Безусловно, при таком подходе необходима ручная разметка, какая-то первоначальная упорядоченность информации. Термин разметка означает присвоения документу (или отдельной информации) класса, ранга или важности. Разметка более простая задача, чем написание правил. Кроме того, разметка может быть произведена в обычном режиме использования системы. Например, в программе электронной почты может существовать возможность помечать письма как спам [42], тем самым формируя обучающее множество для классификатора - фильтра нежелательных сообщений. Таким образом, классификация текстов, основанная на машинном обучении, является примером обучения с учителем, где в роли учителя выступает человек, задающий набор классов и размечающий обучающее множество [72].

Рассмотрим несколько подходов реализации метода обучения классификатора.

1.2.1 Байесовский классификатор

Поскольку цель классификации — найти самый подходящий класс для данного документа, то в наивной байесовской классификации задача состоит в нахождении наиболее вероятного класса c_m, который рассчитывается по формуле [75]:

$$c_m = \underset{c \in C}{\operatorname{argmax}} P\left(c|d\right) \tag{1.2}$$

где c — класс, d —- документ, $argmax$ —- элемент, на котором достигается максимум.

Вычислить значение этой вероятности напрямую невозможно, поскольку для этого нужно, чтобы обучающее множество содержало все (или почти все) возможные комбинации классов и документов. Однако, используя формулу Байеса, можно переписать выражение для $P(c|d)$, в виде:

$$c_m = \underset{c \in C}{\operatorname{argmax}} \frac{P\left(d|c\right) P\left(c\right)}{P\left(d\right)} = \underset{c \in C}{\operatorname{argmax}} P\left(d|c\right) P\left(c\right) \tag{1.3}$$

где знаменатель $P(d)$ опущен, так как не зависит от c и, следовательно, не влияет на нахождение максимума; $P(c)$ — вероятность того, что встретится класс c, независимо от рассматриваемого документа; $P(d|c)$ - вероятность встретить документ d среди документов класса c.

Используя обучающее множество, вероятность $P(c)$ можно оценить по формуле:

$$P(c) = \frac{N_c}{N} \tag{1.4}$$

где N_c — количество документов из обучающего множества в классе c, N — общее количество документов в обучающем множестве.

С учетом сделанных в разделе 1.1.3 предположений, используя правило умножения вероятностей независимых событий [25], можно записать следующую формулу:

$$P(d|c) = P(t_1, t_2, \ldots, t_n|c) = P(t_1|c)P(t_2|c)...P(t_n|c) = \prod_{k=1}^{n} P(t_k|c) \qquad (1.5)$$

Оценка вероятностей $P(t|c)$ с помощью обучающего множества будет рассчитываться по формуле:

$$P(t|c) = \frac{T_{ct}}{T_c} \qquad (1.6)$$

где T_c — общее количество термов в документах класса c; T_{ct} — количество вхождений терма t во всех документах класса c (и на любых позициях — здесь существенно используется второе упрощающее предположение, иначе пришлось бы вычислить эти вероятности для каждой позиции в документе, что невозможно сделать достаточно точно из-за разреженности обучающих данных — трудно ожидать, чтобы каждый терм встретился в каждой позиции достаточное количество раз). При подсчете учитываются все повторные вхождения.

После того, как классификатор «обучен», то есть найдены величины $P(t|c)$ и $P(c)$, можно отыскать класс документа с помощью соотношения:

$$c_m = \operatorname*{argmax}_{c \in C} P(d|c)P(c) = \operatorname*{argmax}_{c \in C} P(c) \prod_{k=1}^{n} P(t_k|c) \qquad (1.7)$$

Чтобы избежать в последней формуле переполнения снизу из-за большого числа малых сомножителей, на практике вместо произведения обычно используют сумму логарифмов. Логарифмирование не влияет на нахождение максимума, так как логарифм является монотонно возрастающей функцией. Поэтому в большинстве реализаций вместо критерий (1.7) в виде:

$$c_m = \operatorname*{argmax}_{c \in C} \left[\log P(c) + \sum_{k=1}^{n} \log P(t_k|c) \right] \qquad (1.8)$$

Эта формула имеет простую интерпретацию. Шансы классифицировать документ часто встречающимся классом выше, и слагаемое $\log P(c)$ вносит в общую сумму соответствующий вклад. Величины же $\log P(t|c)$ тем больше, чем важнее терм t для идентификации класса c, и, соответственно, тем весомее их вклад в общую сумму [106].

1.2.2 Латентное размещение Дирихле

Метод латентного размещения Дирихле *(Latent Dirichlet Allocation, ЛРД)* предложен Дэвидом Блеем, Эндрю Нг и Майклом Джорданом в 2003 году [59]. Латентное размещение Дирихле — это порождающая модель, объясняющая результаты наблюдений с помощью неявных групп тем, что позволяет получить объяснение, почему некоторые части данных схожи. Например, если наблюдениями являются слова, собранные в тексты, утверждается, что каждый текст представляет собой смесь небольшого количества тем, и что появление каждого слова связано с одной из тем документа.

Метод ЛРД основан на вероятностной модели и является развитием наивного Байесовсго классификатора. Вероятность того, что в документе d встерится слово w описывается формулой [22]:

$$p\left(d,w\right)=\sum_{t\in T}p(d)p(w|t)p(t|d) \tag{1.9}$$

при этом так же формулируются следующие предположения [20]:

- векторы документов $\theta_d=(p(t|d):t\in T)$ порождаются одним и тем же вероятностным распределением на нормированных $|T|$-мерных векторах; это распределение удобно взять из параметрического семейства распределений Дирихле $Dir(\theta,\alpha),\alpha\in R^{|T|}$;
- векторы тем $\phi_t=(p(w|t):w\in W)$ порождаются одним и тем же вероятностным распределением на нормированных векторах размерности $|W|$; это распределение удобно взять из параметрического семейства распределений Дирихле $Dir(\theta,\beta),\beta\in R^{|W|}$;

Базовые вероятностные тематические модели позволяют выявлять скрытую тематику документов на основе модели документа как мешка слов. В них также предполагается существование скрытых взаимосвязей между различными объектами, которые могут проявляться в структуре словоупотребления. Семантическая близость различных объектов может оцениваться путем сравнения их тематических векторов. Основным недостатком распределения Дирихле является отсутствие убедительных лингвистических обоснований. С точки зрения

анализа текстовых документов предположение о распределении Дирихле не является обоснованным [21, 58].

1.2.3 Нейронные сети

Нейронные сети — это класс моделей, основанных на биологической аналогии с мозгом человек и предназначенных после прохождения этапа так называемого обучения на имеющихся данных для решения разнообразных задач анализа данных. При применении этих методов, прежде всего, встает вопрос выбора конкретной архитектуры сети (числа «слоев» и количества «нейронов» в каждом из них). Размер и структура сети должны соответствовать (например, в смысле формальной вычислительной сложности) существу исследуемого явления. Поскольку на начальном этапе анализа природа явления обычно известна плохо, выбор архитектуры является непростой задачей и часто связан с длительным процессом «проб и ошибок».

Нейронная сеть, полученная в результате обучения, выражает закономерности, присутствующие в данных. При таком подходе она оказывается функциональным эквивалентом некоторой модели зависимостей между переменными, подобной тем, которые строятся в традиционном моделировании [81]. Однако, в отличие от традиционных моделей, в случае нейронных сетей эти зависимости не могут быть записаны в явном виде, подобно тому, как это делается в статистике. При таком подходе сосредотачиваются исключительно на практическом результате, в данном случае на точности прогнозов и их прикладной ценности, а не на сути механизмов, лежащих в основе явления, или на соответствии полученных результатов какой-либо имеющейся теории.

Одно из главных преимуществ нейронных сетей состоит в том, что они могут аппроксимировать любую непрерывную функцию, и поэтому нет необходимости заранее принимать какие-либо гипотезы относительно модели и даже в ряде случаем о том, какие переменные действительно важны. Существенным недостатком нейронных сетей является тот факт, что окончательное решение зависит от начальных установок сети и его практически невозможно интерпретировать в традиционных аналитических терминах [57, 71], сети необходимо

проводить на примерах, которые не участвовали в ее обучении. При этом число тестовых примеров должно быть тем больше, чем выше качество обучения. Если ошибки нейронной сети имеют вероятность близкую к одной миллиардной, то и для подтверждения этой вероятности нужен миллиард тестовых примеров. Получается, что тестирование хорошо обученных нейронных сетей становится очень трудной задачей [79].

Использование нейронной сети для поддержки принятия решений схоже с классической задачей классификации, решаемой нейронными сетями. При этом классификации подлежат ситуации, характеристики которых поступают на вход нейронной сети. На выходе сети при этом должен появиться признак решения, которое она приняла. При этом в качестве входных сигналов используются различные критерии описания состояния управляемой системы [29].

Сеть Кохонена–Гроссберга

Сеть Кохонена–Гроссберга — это двуслойная сеть, используемая, в основном, в задачах классификации. Первый слой сети - сеть Кохонена, обучаемая для получения наилучшего представления векторов обучающей выборки. Сеть Кохонена обучается без учителя на основе самоорганизации [45]. В течении обучения вектора весов нейронов стремятся к центрам кластеров — групп векторов обучающей выборки [118, 119, 121]. После обучения сеть сопоставляет предъявляемый образ к одному из кластеров, то есть к одному из выходов. Каждый нейрон сети Кохонена запоминает один класс, то есть величина выхода тем выше, чем ближе предъявляемый образец к данному классу. Суть интерпретатора — выбрать номер нейрона с максимальным выходом. Выход так же можно трактовать как вероятность. Меняя количество нейронов, мы можем динамично менять количество классов.

Присвоение начальных значений происходит с помощью генератора случайных чисел — каждому весу присваивается небольшое значение.

Второй слой — сеть Гроссберга, обучаемый отображать нейроны слоя Кохонена на различные классификационные рубрики. Слой Гроссберга обучается

«с учителем». Процесс обучения обычно представляется в следующей итерационной форме:

$$w_i(t+1) = w_i(t) + \alpha\left(X_i - w_i(t)\right) \tag{1.10}$$

где w_i — весовые коэффициенты нейронов, X_i — входы нейрона [23].

Слой Гроссберга предназначен для совместной работы со слоем, дающим единственную единицу на выходе или же такой набор выходов, что их сумма равна единице. Нейроны слоя Гроссберга вычисляют взвешенную сумму своих входов. Функция активации — линейная. Слой Гроссберга дает на выходе линейную комбинацию своих векторов весов, коэффициенты комбинации задаются входами слоя Гроссберга [26]. В течении обучения вектора весов нейронов стремятся к центрам кластеров — групп векторов обучающей выборки. После обучения сеть сопоставляет предъявляемый образ к одному из кластеров, то есть к одному из выходов [46].

Таким образом, сеть Кохонена–Гроссберга позволяет выделить в пространстве входных векторов области, соответствующие каждой из предъявленных рубрик [30, 120].

1.2.4 Векторные методы

Векторные методы используют векторную модель представления текста. Как правило, для классификации используется скалярное произведение векторов. Вектор документа последовательно скалярно перемножается с векторами категорий и чем больше скалярное произведение, тем больше вероятность, что документ попадет в эту категорию.

Расстояние Хэмминга ($Hamming(X, Y)$) — это количество различающихся позиций для строк с одинаковой длинной [3]. Например, $Hamming(100, 001) = 2$.

Впервые проблема подсчета расстояния Хэмминга была поставлена М. Мински в 1969 году, где задача сводилась к поиску всех строк из базы данных, которые находятся в пределах заданного расстояния Хэмминга к запрашиваемой.

Расстояние Хэмминга уже довольно широко используется для различных задач, таких как поиск близких дубликатов, распознавание образов, классификация документов, исправление ошибок, обнаружения вирусов и т.д.

1.2.5 Латентно-семантический анализ

Метод латентно-семантического анализа (ЛСА) позволяет выявлять значения слов с учетом контекста их использования путем обработки большого объема текстов.

Модель представления текста, используемая в латентно-семантическом анализе, во многом схожа с восприятием текста человеком. Например, с помощью этого метода можно оценить текст на соответствие заданной теме.

В качестве исходной информации используется терм-документная матрица.

Терм-документная матрица — это математическая матрица, описывающая частоту терминов, которые встречаются в коллекции документов.

Строки соответствуют документам в коллекции, а столбцы соответствуют терминам. К матрице применяется сингулярное разложение.

Сингулярное разложение — это математическая операция, раскладывающая матрицу на 3 составляющих. Сингулярное разложение можно представить в виде формулы:

$$A = USV^T \tag{1.11}$$

где A — исходная матрица, U и V^T — ортогональные матрицы, а S — диагональная матрица, значения, на диагонали которой называются сингулярными коэффициентами матрицы A. Сингулярное разложение позволяет выделить ключевые составляющие исходной матрицы.

Основная идея ЛСА состоит в том, что если в качестве матрицы A использовалась терм-документная матрица, то матрица A^*, содержащая только k первых линейно независимых компонент, отражает основную структуру различных зависимостей, присутствующих в исходной матрице. Структура зависимостей определяется весовыми функциями термов [11].

Как правило, выбор k зависит от поставленной задачи и подбирается эмпирически. Если выбранное значение k слишком велико, то метод теряет свою мощность и приближается по характеристикам к стандартным векторным методам. Слишком маленькое значение k не позволяет улавливать различия между похожими термами или документами. Если же необходимо выбирать значение k автоматически, то можно, например, установить пороговое значение сингулярных коэффициентов и отбрасывать все строки и столбцы, соответствующие сингулярным коэффициентам, не превышающим данное пороговое значение.

Схожесть между любой комбинацией термов и/или документов чаще всего вычисляют с помощью скалярного произведения их векторов, однако на практике лучший результат дает вычисление схожести с помощью коэффициента корреляции Пирсона [106].

ЛСА отображает документы и отдельные слова в так называемое «семантическое пространство», в котором и производятся все дальнейшие сравнения. При этом делаются следующие предположения:

1. Документ это просто набор слов. Порядок слов в документах игнорируется. Важно только то, сколько раз то или иное слово встречается в документе.
2. Семантическое значение документа определяется набором слов, которые, как правило, идут вместе.
3. Каждое слово имеет единственное значение. Это, безусловно, сильное упрощение, но именно оно делает проблему разрешимой.

1.2.6 Деревья решений

Деревья решений — один из простейших методов машинного обучения. Это совершенно прозрачный способ классификации наблюдений, и после обучения они представляются в виде последовательности предложений if-then (если-то), организованных в виде дерева [106].

Имея дерево решений, нетрудно понять, как оно принимает решения. Достаточно проследовать вниз по дереву, правильно отвечая на вопросы, — и в конечном итоге ответ будет получен. Обратная трассировка от узла, в котором

произошла остановка, до корня дает обоснование выработанной классификации [31, 112].

1.2.7 Эволюционный анализ и генетическое программирование

Эволюционный анализ данных описывает и моделирует регулярности и тренды для объектов, чье поведение изменяется во времени. Несмотря на то, что здесь могут применяться рассмотренные до этого характеризация и дискриминация, анализ ассоциаций, классификация, кластеризация, у данного вида анализа имеются отличительные черты и свои собственные методы, которые включают анализ временных рядов, анализ последовательности и периодичности, поиск близостей [34].

Генетическое программирование — это методика машинного обучения, аналогией которой является биологическая эволюция. В общем случае все начинается с большого набора популяций (программ), сгенерированных случайным образом или написанных вручную, о которых известно, что это достаточно близкие решения.

Затем эти программы конкурируют между собой в попытке решить некоторую поставленную задачу. По завершении состязания составляется отсортированный список программ — от наилучшей к наихудшей. Затем лучшие программы копируются и модифицируются одним из двух способов. Самый простой способ называется мутацией; в этом случае некоторые части программы случайным образом и очень незначительно изменяются в надежде, что от этого решение станет лучше [34, 92].

Новые поколения создаются до тех пор, пока не будет выполнено условие завершения, которое в зависимости от задачи может формулироваться одним из следующих способов:

- Найдено идеальное решение.
- Найдено достаточно хорошее решение.
- Решение не удается улучшить на протяжении нескольких поколений.
- Количество поколений достигло заданного предела [106].

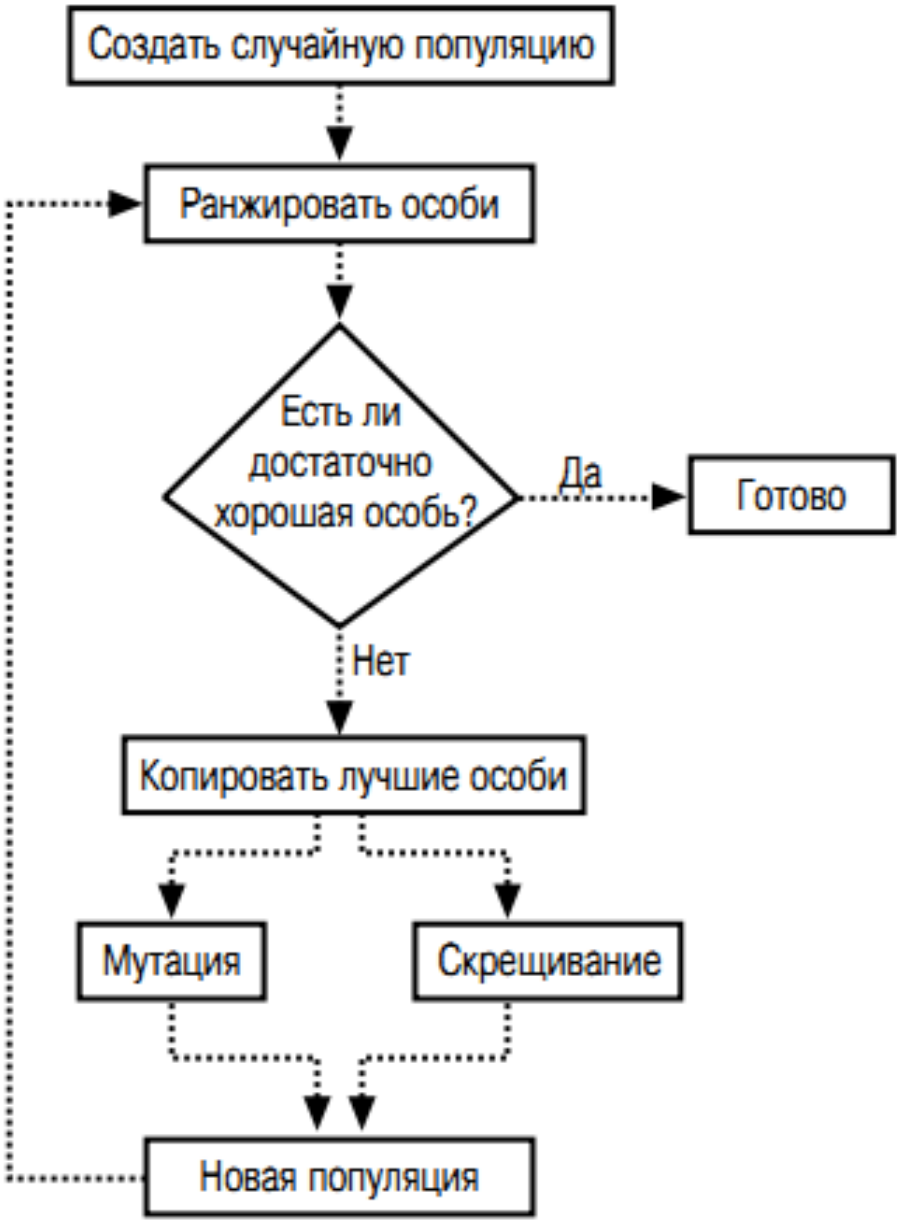

Рисунок 1.1 — Схема работы генетического алгоритма

Идея систем рассуждений на основе аналогичных случаев (case-based reasoning — CBR) заключается в следующем. Для того чтобы сделать прогноз на будущее или выбрать правильное решение, эти системы находят в прошлом близкие аналоги наличной ситуации и выбирают тот же ответ, который был для них правильным. Поэтому этот метод еще называют методом "ближайшего соседа"(nearest neighbour). В последнее время распространение получил также термин memory based reasoning, который акцентирует внимание, что решение принимается на основании всей информации, накопленной в памяти [81, 112].

Системы CBR показывают неплохие результаты в самых разнообразных задачах. Главным их минусом считают то, что они вообще не создают каких-

либо моделей или правил, обобщающих предыдущий опыт, — в выборе решения они основываются на всем массиве доступных исторических данных, поэтому невозможно сказать, на основе каких конкретно факторов CBR системы строят свои ответы.

Другой минус заключается в произволе, который допускают системы CBR при выборе меры "близости". От этой меры самым решительным образом зависит объем множества прецедентов, которые нужно хранить в памяти для достижения удовлетворительной классификации или прогноза [112].

Алгоритмы ограниченного перебора были предложены в середине 60-х годов М.М. Бонгардом для поиска логических закономерностей в данных. С тех пор они продемонстрировали свою эффективность при решении множества задач из самых различных областей.

Эти алгоритмы вычисляют частоты комбинаций простых логических событий в подгруппах данных. Примеры простых логических событий: $X = a; X < a; X > a; a < X < b$ и др., где X — какой либо параметр, a и b — константы. Ограничением служит длина комбинации простых логических событий (у М. Бонгарда она была равна 3). На основании анализа вычисленных частот делается заключение о полезности той или иной комбинации для установления ассоциации в данных, для классификации, прогнозирования и пр.

1.3 Процесс обнаружения знаний

Для обнаружения знаний в данных недостаточно просто применить методы интеллектуального анализа, хотя, безусловно, этот этап является основным в процессе интеллектуального анализа. Весь процесс состоит из нескольких этапов.

Итак, весь процесс можно разбить на следующие этапы:

- понимание и формулировка задачи анализа;
- подготовка данных для автоматизированного анализа (препроцессинг);
- применение методов интеллектуального анализа и построение моделей;
- проверка построенных моделей;
- интерпретация моделей человеком.

На первом этапе выполняется осмысление поставленной задачи и уточнение целей, которые должны быть достигнуты методами интеллектуального анализа. Важно правильно сформулировать цели и выбрать необходимые для их достижения методы, так как от этого зависит дальнейшая эффективность всего процесса [124].

Второй этап состоит в приведении данных к форме, пригодной для применения конкретных методов интеллектуального анализа. При этом вид преобразований, совершаемых над данными, во многом зависит от используемых методов, выбранных на предыдущем этапе.

Третий этап — это собственно применения методов интеллектуального анализа. Сценарии этого применения могут быть самыми и различными и могут включать сложную комбинацию разных методов, особенно если используемые методы позволяют проанализировать данные с разных точек зрения.

Следующий этап — проверка построенных моделей. Очень простой и часто используемый способ заключается в том, что все имеющиеся данные, которые необходимо анализировать, разбиваются на две группы. Как правило одна из них большего размера, другая - меньшего. На большей группе, применяя те или иные методы интеллектуального анализа, получают модели, а на меньшей - проверяют их. По разнице в точности между тестовой и обучающей группами можно судить об адекватности построенной модели.

Последний этап — интерпретация полученных моделей человеком в целях их использования для принятия решений, добавление получившихся правил и зависимостей в базы знаний и т. д. Этот этап часто подразумевает использование методов, находящихся на стыке технологий интеллектуального анализа и технологии экспертных систем. От того, насколько эффективным он будет, в значительной степени зависит успех решения поставленной задачи [33].

Для подготовки данных необходимо сделать следующее:

– Очистка данных — исключение противоречий и случайных "шумов"из исходных данных

  При анализе любой текстовой информации имеет место быть сильная зашумленность. В связи с этим, прежде, чем переходить к анализу необходимо произвести ряд действий для освобождения текста от шумов.

  Стемминг — это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с морфологиче-

ским корнем слова. Алгоритм стемматизации представляет собой давнюю проблему в области компьютерных наук. Первый документ по этому вопросу был опубликован в 1968 году. Данный процесс применяется в поисковых системах для обобщения поискового запроса пользователя [115].

Конкретные реализации стемматизации называются алгоритм стемматизации или просто стеммер. Наиболее удачный алгоритм стемминга — стеммер Портера.

Стеммер Портера — алгоритм стемминга, опубликованный Мартином Портером в 1980 году. Оригинальная версия стеммера была предназначена для английского языка. Алгоритм не использует баз основ слов, а лишь, применяя последовательно ряд правил, отсекает окончания и суффиксы, основываясь на особенностях языка, в связи с чем работает быстро, но не всегда безошибочно [115].

Семантическое ядро — это подборка понятий, имеющих существенное значение для данной предметной области. Точное определение семантического ядра зависит от области применения. Так, в лингвистике, семантическим ядром называют «не упрощаемое замкнутое подмножество языка», подразумевая при этом скорее смысловую составляющую языка, а не грамматические конструкции.

Если попытаться перейти в доступную пониманию область теории информации, оперирующую статистическими параметрами текста, то можно говорить о семантическом ядре как о подборке смысловых единиц, достаточной для классификации текста. В качестве такой единицы может выступать словоформа, лексема или другая языковая конструкция [11].

Для использования в статистическом анализе текста можно дать определение нескольких подборок смысловых единиц, сходных с семантическим ядром, например:

1. Специфичные слова предметной области

   Это такие слова, которые встречаются исключительно в текстах предметной области и позволяют установить принадлежность текста этой предметной области.
2. Высокоинформативные слова предметной области

Это такие слова, которые позволяют рубрицировать тексты внутри предметной области.

Семантическое ядро проще всего сформировать, анализируя большой объем текстов по предметной области. В него попадают слова, которые чаще всего встречаются в анализируемых текстах.

– Интеграция данных — объединение данных из нескольких возможных источников в одном хранилище
– Преобразование данных. На данном этапе данные преобразуются к форме, подходящей для анализа. Часто применяется агрегация данных, дискретизация атрибутов, сжатие данных и сокращение размерности.

Обычно в системах интеллектуального анализа текстов выделяются следующие главные компоненты [19, 41]:

1. База данных, хранилище данных или другой репозиторий информации. Это может быть одна или несколько баз данных, хранилище данных, электронные таблицы, другие виды репозиториев, над которыми могут быть выполнены очистка и интеграция. Виды баз данных:
   – Реляционные базы данных;
   – Хранилища данных;
   – Транзакционные базы данных;
   – Объектно-ориентированные базы данных;
   – Объектно-реляционные базы данных;
   – Пространственные базы данных (Spatial databases);
   – Временные базы данных (Temporal databases);
   – Текстовые базы данных;
   – Мультимедийные базы данных;
   – Разнородные базы данных;
   – Всемирная Паутина.

   Для рассматриваемой предметной области целесообразнее всего использовать реляционных хранилища и объектно-ориентированные базы данных, поскольку оптимизированы для хранения данных подобной структуры.
2. Сервер базы данных или хранилища данных. Указанный сервер отвечает за извлечение существенных данных на основании пользовательского запроса.

3. База знаний. Это знания о предметной области, которые указывают, как проводить поиск и оценивать полезность результирующих паттернов. Данный пункт тесно взаимосвязан с предыдущим, поскольку являтся его частным случаем.
4. Служба добычи знаний. Она является неотъемлемой частью системы интеллектуального анализа текстов и содержит набор функциональных модулей для таких задач, как характеризация, поиск ассоциаций, классификация, кластерный анализ и анализ отклонений.
5. Модуль оценки паттернов. Данный компонент вычисляет меры интереса или полезности паттернов.
6. Графический пользовательский интерфейс. Этот модуль отвечает за коммуникации между пользователем и системой интеллектуального анализа текстов, визуализацию паттернов в различных формах.

1.4 Проблема лексической неоднозначности

Векторная модель использует вес (частоту) терма, чтобы определить его важность в документе. Однако, термы могут быть схожи семантически, но отличаться лексикографически, и наоборот, что в свою очередь приведет к тому, что классификация, основанная на частоте термов, не даст нужного результата.

В исследованиях [65, 74, 105] предлагается использовать семантическую связанность и семантические меры для учета особенностей языка. Например, в реализации известной семантической сети WordNet, разработанной в Принстонском университете, используются так называемые «синсеты» — синонимические ряды, объединяющие слова со схожим значением. Каждый «синсет» содержит список синонимов или синонимичных словосочетаний и указатели, описывающие отношения между ним и другими «синсетами». Слова, имеющие несколько значений, включаются в несколько «синсетов» и могут быть причислены к различным синтаксическим и лексическим классам. Такой подход дает более точные результаты в сравнении с классической векторной моделью представления знаний. База WordNET свободно доступна в сети Интернет, на ее основе

было выполнено значительное число экспериментов [56, 65, 74, 105] в области информационного поиска.

Возможность свободного использования сети WordNET привела к тому, что появилось множество исследований [56, 65, 74, 105], которые использовали ее в качестве базы для обучения алгоритмов информационного поиска. В ходе этих экспериментов были выявлены недостатки этой семантической сети, которые препятствуют его эффективному применению.

Основной проблемой, связанной с применением WordNET, стала сложность описания многозначных сущностей. В современной версии этой семантической сети содержится самое часто встречающееся значение каждого слова, что дает возможность выбирать именно это значение в случае возникновения проблем с многозначными сущностями. Так же серьезной проблемой практического применения WordNET является так называемая «теннисная проблема»: синсеты, принадлежащие одной и той же предметной области в структуре WordNET, часто располагаются очень далеко, что в свою очередь приводит к затруднениям их применения в задачах разрешения лексической многозначности.

Кроме того, к настоящему моменту известно еще несколько попыток реализации российской версии WordNET:

1. RusNET, разрабатываемый на филологическом факультете СПбГУ с 1999 года [48].
2. Проект RuThes [86], который используется в УИС РОССИЯ [4]. Закрытый коммерческий ресурс.
3. Проект русского WordNET от компании «Новософт», расположенной в Новосибирске. Коммерческий ресурс, детали реализации неизвестны. Пример дерева гиперонимов из российского WordNET изображен на рисунке 1.2.

К сожалению, реальное применение данных баз ограничено, поскольку онм содержат только наиболее часто встречающиеся слова. Специализированные слова некоторых предметных областей в ней практически отсутствуют, что приводит к невозможности их обработки. Кроме того, в русских версиях данной сети описания синсетов остались на английском языке, что так же накладывает определенные ограничения на область их применения.

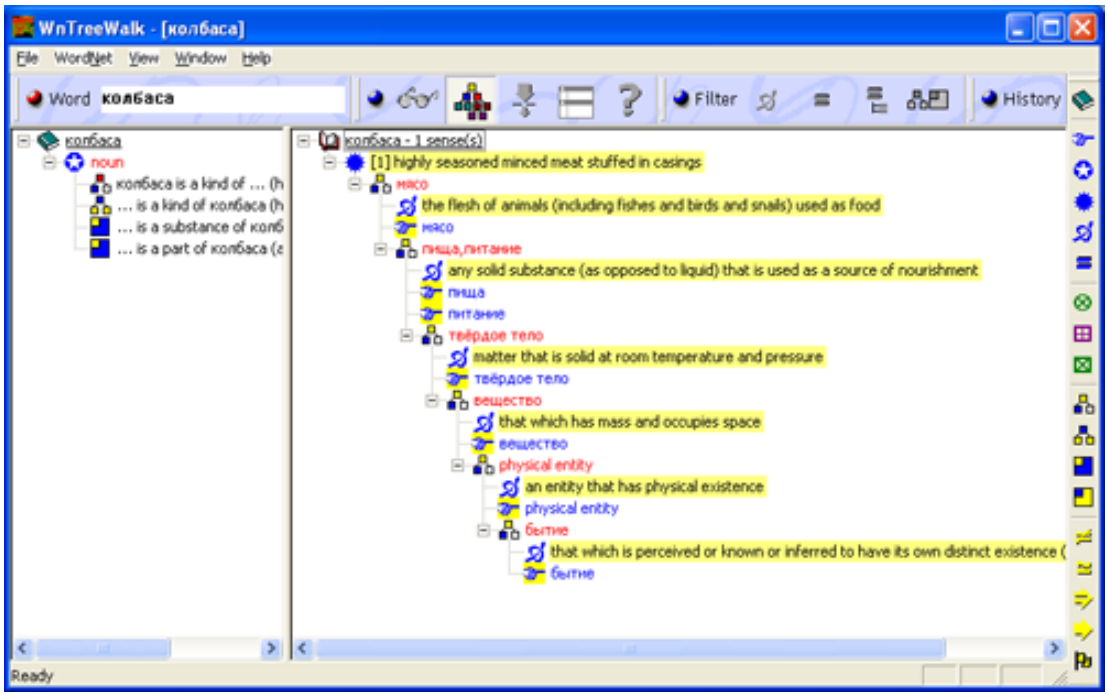

Рисунок 1.2 — Пример дерева гиперонимов из российской версии WordNET

Таким образом, реальное применение семантических сетей по типу WordNET огранично, поскольку они оперируют не какими-то определенными предметными областями, а охватывают только общую лексику и семантику естественного языка.

Похожая техника представляет собой отображение термов документа в их «смысл» и составление функциональных векторов документа. В терминах СУБД это означает, что всем словам, имеющим один и тот же смысл, приписывается некий идентификатор, который в свою очередь и становится термом. Конечно, качество обучения улучшается, однако опыт использования этой техники показывает, что улучшается оно незначительно.

1.4.1 Подходы к устранению лексической многозначности

Проблема снятия лексической многозначности может быть переформулирована так же, как задача максимизации с использованием формализма скрытых Марковских моделей [87]. Пусть T — множество терминов, M — множество значений, соответствующих терминам. Для последовательности терминов $\tau = \{t_1,...,t_n\}$, где $\forall i\ t_i \in T$, задача состоит в нахождении наиболее вероятной

последовательности значений $\mu = \{m_i,...,m_n\}$, где $\forall i\, m_i \in M$, соответствующей входным терминам.

$$\hat{\mu} = \arg_\mu P(\mu|\tau) = \arg_\mu \left(\frac{P(\mu)P(\tau|\mu)}{P(\tau)} \right) \tag{1.12}$$

Поскольку вероятность $P(\tau)$ для входной последовательности является величиной постоянной, то задача сводится к максимизации числителя, указанного в формуле (1.12). Для решения этого уравнения делается марковское предположение, что значение i-го термина зависит только от конечного числа значений предыдущих терминов [77]:

$$\hat{\mu} = \arg_\mu \left(\prod_{i=1}^{n} P(m_i|m_{i-1},..., m_{i-k})P(t_i|m_i) \right) \tag{1.13}$$

где k — порядок модели.

Множители равенства (1.13) определяют скрытую Марковскую модель k-го порядка, где наблюдения соответствуют входным терминам, состояния соответствуют значениям терминов, $P(m_i|m_{i-1},..., m_{i-k})$ - вероятность перехода между состояниями, $P(t_i|m_i)$ — вероятность появления термина t_i в каждом состоянии m_i.

Дальнейшее использование данной модели связано со значительными трудностями, в частности с разреженностью языка. Например, чтобы построить модель перехода для Марковской модели первого порядка, необходимо оценить вероятность каждой пары состояний, что для данной задачи сводится к вероятности встречи двух терминов в конкретных значениях вместе. Для задачи устранения лексической многозначности проблема оценки параметров марковской модели является нетривиальной задачей. Это связано с большими объемами обрабатываемой информации, то есть с объемами представленных знаний и с тем, что слова в тексте на естественном языке распределяются не равномерно, а по закону Ципфа [69, 80].

Закон Ципфа — эмпирическая закономерность распределения частоты слов естественного языка: если все слова языка упорядочить по убыванию частоты их использования, то частота i-го слова в таком списке окажется приблизительно обратно пропорциональной его порядковому номеру i.

1.4.2 Использование семантических сетей для устранения лексической многозначности

Большинство подходов к устранению лексической многозначности связаны с развитием огромных баз знаний, созданных вручную, таких как WordNet. Можно указать на очевидный недостаток такого подхода — ограниченность области применения данных методов, поскольку ручное поддержание баз знаний в актуальном состоянии является очень сложной и трудозатратной задачей.

Если в семантической сети, на основе которой производится анализ (например, WordNET) используются различные значения для многозначных слов, то для анализа необходимо обеспечение автоматического выбора между этими многозначными сущностями [32]. Обычно в таких случаях используется наивный метод, который выбирает наиболее часто встречающуюся сущность. Очевидно, что использование такого подхода далеко от идеала и в ряде случаев может давать необъективные результаты. Частично эта проблема решается в концепции «универсального терминологического пространства» [38—40], однако пока о какой-либо реализации этого пространства неизвестно.

Другим подходом к решению проблемы лексической многозначности является использование внешних источников данных. С развитием сети Интернет появилось огромное количество документов, которые связаны между собой гиперссылками. Например, в работе [91, 108, 123] рассматривалась возможность использования глобальной энциклопедии «Википедия» в качестве аннотированного корпуса для обучения Марковской модели. Для методов, использующих данный подход, очень часто применяют алгоритм Леска. Алгоритм основан на предположении, что многозначное слово и его окружение относятся к одной теме [85].

Все вышеописанные методы и алгоритмы так или иначе основаны на внешних данных и имеют один общий недостаток. В основе всех этих алгоритмов явно или неявно лежит предположение, что существуют однозначные термины, на основании которых в последствии определяются многозначные термины. Это в свою очередь составляет огромную проблему, поскольку в неспециализированных текстах, таких, как объявления о поиске работы, новостных статьях, участвуют только многозначные термины, либо присутствующие од-

нозначные термины слабо связаны с темой документа и могут быть расценены классификатором, как стоп-слова. Этот факт приводит к тому, что точность и эффективность указанных методов в значительной степени ухудшается при их применении для классификации вакансий или новостей [60].

Для избавления от лексической многозначности так же используются меры семантической связности. Отметим, что семантическая близость и семантическая связность — это разные понятия. *Семантическая близость* является частным случаем семантической связности. *Семантическая связность* — это количество связей, с помощью которых связаны два слова. Перечислим наиболее известные способы вычисления семантической связности.

1. *Мера Ликока–Чодороу*

   Мера Ликока и Чодороу [108] основана на вычислении длины пути между термами. Кратчайшим путем от одного термина к другому считается путь, который использует наименьшее количество соседних термов. Данную меру можно представить в виде следующей формулы:

   $$related_{lch} = (t_1, t_2) = max\left[-\log\left(L(t_1, t_2)/(2 \cdot D)\right)\right] \tag{1.14}$$

   где $L(t_1, t_2)$ — кратчайшая длинна пути (наименьшее количество узлов) между двумя термами, D — максимальная глубина (максимальное количество узлом от корневого узла, либо количество узлов до ближайшего общего предка [18]).

2. *Мера Цеша*

   Вычисление меры Цеша [123] подобно вычислению меры Ликока–Чодороу, основное отличие заключается в том, что поиск кратчайшего пути осуществляется между термами по произвольным типам ссылок.

3. *Мера Лина*

   Вычисление семантической связности с помощью меры Лина основано на теореме близости. Она гласит, что близость между двумя термами можно вычислить с помощью коэффициента отношения количества текстов (корпусов), в которых термы встречаются вместе к частоте их встречаемости в определениях.

   Отсутствие общих терминов между двумя документами еще не означает, что они являются абсолютно несхожими. Термины могут быть синтаксически различны, но в то же самое время семантически очень

близки. Дальнейшее развитие метода анализа данных будет основано именно на этом утверждении.

1.5 Обзор работ по теме диссертации

В работе [37] представлена модель лингвистической онтологии для автоматической обработки текстов широкой предметной области, т.е. предметной области, в состав которой входят тысячи разных классов сущностей, входящих между собой в неограниченные типы отношений и ситуаций. В предложенной авторами модели используется набор отношений лингвистической онтологии, который специально подобран для описания широкой предметной области. Предложено использовать небольшой набор отношений, сопоставимый с набором отношений в традиционных информационно-поисковых тезаурусах. Однако, были введены более строгие онтологические определения используемых отношений. Предложенный подход позволяет производить обработку существенных объемов данных. При применении данного подхода для динамических предметных областей эту обработку необходимо проводить всякий раз при пополнении словарей и незначительном изменении бизнес-логики системы, что в реальных условиях является недостатком.

В работе [21] рассматривается тематическое моделирование коллекций текстовых документов. Тематическое моделирование развивается в настоящее время, главным образом, в рамках байесовского подхода и графических моделей. В данной работе предлагается альтернативный подход, свободный от избыточных вероятностных предположений. Аддитивная регуляризация тематических моделей (ARTM) основана на максимизации взвешенной суммы логарифма правдоподобия и дополнительных критериев регуляризаторов. Это упрощает комбинирование тематических моделей и построение сколь угодно сложных многоцелевых моделей. Применение результатов данного исследования ограничено на предметных областях, в которых данные тематически распределены неравномерно, т.е. документы некоторых тем преобладают. Данный факт приводит к снижению качества модели, а иногда и к невозможности ее применения.

В работах [93, 94] рассматриваются особенности применения латентно-семантического анализа для поиска и классификации веб-документов. В частности, рассматривается вопрос нахождения подобия HTML-документов. В работах так же рассматривается повышение эффективности и надежности систем классификации путем ограничения области поиска, т.е. произведение поиска не во всех коллекциях, а только в ограниченом подмножестве. В указанных работах не рассматривается вопрос улучшения качества информационного поиска. Кроме того применение методов, рассмотренных в данных работах ограничено в условиях неполноты данных.

В работе [43] описана обобщенная модель порождения текстов на основе цепей Маркова. На основе модели предложен новый метод определения веб-спама как искусственно-порожденных текстов. Результаты исследования могут быть так же использованы для автоматической рубрикации текстовых документов. Главной проблемой применения метода, описанного в статье, является относительно низкое качество классификации в задачах классификации текстов по многочисленным рубрикам. Кроме того, метод не опробован в условиях неполноты данных.

В статье [1] описываются способы ранжирования текстовых документов на основе лога действий пользователя поисковой системы. Метод дает достаточно хороший результат, однако применение метода ограничено в условиях неполноты данных, кроме того метод не решает проблему "холодного запуска".

В работе [36] предлагается новый подход к извлечению оценочных слов для различных предметных областей. В рамках этого подхода была разработана модель, включающая набор характеристик и комбинацию алгоритмов, которые позволяют извлекать оценочные слова в конкретной предметной области. Данная модель была обучена в предметной области о фильмах и затем применена в четырех других областях. Качество работы метода оценивалось на основании разметки экспертов и оставалось на высоком уровне при переносе модели на различные предметные области. Кроме того, созданная модель была использована в предметной области о фильмах на английском языке и продемонстрировала высокое качество извлечения оценочных слов. В результате применения данного подхода получается модель очень большого размера, что ограничивает ее применение на предметных областях, в которых требуется анализ большого количества текстов.

В работах [5, 6] рассматриваются вопросы построения и применения компьютерного словаря русских однобуквенных паронимов, т.е. слов, отличающихся одной буквой и получающихся друг из друга в результате замены, вставки, удаления буквы или же перестановки двух стоящих рядом букв. Словарь разрабатывался для автоматизированного исправления в тексте случайных ошибок (так называемых малапропизмов), при которых одно знаменательное слово заменяется другим похожим словом, отличным от первого по смыслу и тем самым нарушающим исходный смысл высказывания. Паронимы построенного словаря служат вариантами исправления однобуквенных ошибок, при котором не требуется изменение контекста ошибочного слова. В связи с последним требованием в работе уточняется понятие буквенных паронимов за счет дробления морфологических парадигм слов (лексем) и учета свойства параллельности морфологических парадигм. Параллельными считаются морфопарадигмы двух лексем, для которых существует элементарная редактирующая операция (вставка, удаление, замена буквы, перестановка двух соседних букв), переводящая каждую словоформу первой лексемы в соответствующую словоформу второй лексемы. В работе описывается процесс построения компьютерного словаря однобуквенных паронимов, а также формулируются основные шаги алгоритмов поиска исправляющих слов для найденных в тексте ошибок, с помощью построенного словаря – соответственно для случаев полной и неполной параллельности морфопарадигм исправляемого слова и его паронима. Приводится также статистика контента построенного компьютерного словаря однобуквенных паронимов, общий объем которого достиг 70 тыс. вокабул, в среднем с тремя паронимами на вокабулу. Применение данного словаря для исправления ошибок ограничено, поскольку он хорошо применим только для общеупотребительной лексики. При его использовании в предметных областях, где уместо применение жаргонной или любой другой разговорной лексики — применение метода малоэффективно.

В работе [2] предлагается метод для извлечения цепочек семантически близких слов и выражений, описывающих различных участников сюжета — тематических узлов. Предполагается, что выделение основных участников позволит улучшить качество обработки новостного кластера. Метод основан на структурной организации новостных кластеров и анализе контекстов вхождения языковых выражений. Контексты слов используются в качестве базиса для извлечения многословных выражений и построения тематических узлов. Оцен-

ка предложенного алгоритма производится в задаче построения обзорных рефератов новостных кластеров. Применение метода ограничено на предметных областях, в которых требуется анализ большого количества текстов, поскольку в результате построения контекстов получаются модели, хранение которых с помощью средств вычислительной техники представляет собой нетривиальную задачу.

1.6 Выводы по первой главе

В главе рассмотрены основные методы интеллектуального анализа текстов, основные модели представления знаний, а так же алгоритмы обнаружения знаний.

Анализ современных методов и моделей в применении к задаче классификации и поиска тестовых документов, рассмотренный в первой главе, показывает, что при разработке алгоритмов классификации перспективным подходом является использование комплекса подходов. Для очистки данных от шумов предлагается использовать стеммер Портера и семантическое ядро, для представления данных выбрана векторная модель, для выделения сематического ядра — модификация подхода ЛСА. При анализе частоты вхождения термов в документы используются известные семантические сети по типу WordNET. Так же отметим, что для сокращения расходов на хранение промежуточных вычислений и результатов построения модели необходимо данные размещать максимально компактно.

Глава 2. Интеллектуальный метод подбора персональных рекомендаций гарантирующий получение непустого результата

В предыдущей главе приведен краткий обзор наиболее известных методов и алгоритмов интеллектуального анализа текстов проанализированы особенности их применения, однако в ряде случаев, в том числе в случае неравномерности обучающей выборки, традиционные методы могут выдавать пользователю пустой результат, что в свою очередь для ряда предметных областей является неприемлемым. В данной главе предлагается алгоритм интеллектуального анализа текстов, который на любой запрос пользователя, независимо от размера и равномерности обучающей выборки дает пользователю непустой ответ, отсортированный по степени релевантности запросу пользователя.

2.1 Постановка задачи

Подходы машинного обучения для обеспечения системы формирования персональных рекомендаций используют набор свойств и характеристик документа из обучающей выборки. Самой важной частью этих характеристик является множество всех слов обучающей выборки (исключая стоп-слова).

Пусть имеется выборка текстовых данных (например, товары, услуги), формируемых пользователями, которые необходимо обработать и систематизировать. Пусть так же имеется выборка данных пользователей (например, покупателей, поставщиков) так же представленная в текстовом виде. Технических отличий между двумя этими выборками нет, различия скорее идеологические. Первая выборка — это то, что мы используем для построения модели, а вторая — то, для чего мы анализируем с использованием модели. Необходимо обработать вышеуказанные данные таким образом, чтобы можно было их использовать для быстрого подбора персональных рекомендаций для любого пользователя. При этом данные между категориями распределены неравномерно, и пользователи независимо от их предпочтений всегда гарантированно должны получить выборку рекомендаций определенного объема. На практике очень

часто встречаются текстовые корпусы, неравномерно распределенные между категориями. Категории в данном случае — это некие группы, число групп конечно и известно заранее, которые формируются специалистом в предметной области.

Поскольку одной из основных областей применения интеллектуального анализа текстов является анализ Интернет-запросов и программное обеспечение в среде клиент-сервер, то на все такие алгоритмы накладываются жесткие ограничения по использованию процессорного времени и оперативной памяти. Поэтому метод должен быть масштабируемым, а именно обеспечивать возможность обслуживания запросов большого количества пользователей одновременно [83].

2.2 Выбор модели представления знаний

Последние успехи применения векторных моделей представления знаний доказывают их эффективность. Например, Р. Рапп в своем исследовании [97] использовал векторное представление знаний для обучения программы для прохождения теста с возможностью нескольких вариантов ответа по английскому языку (TOEFL). Его классификатор в 92.5% случаях выбирал правильный ответ при том, что средняя оценка человека при прохождении данного теста составляет 64.5%.

Использование векторного представления знаний имеет ряд привлекательных свойств. В частности, извлечение знаний из текстового корпуса требует значительно меньше затрат, чем, например, ручное составление баз знаний или использование онтологий. Под онтологией в данном случае понимается база знаний специализированного типа, содержащая сведения о понятийной структуре и терминологическом составе предметной области.

Векторная модель является основой для решения многих задач информационного поиска, как то: поиск документа по запросу, классификация документов, кластеризация документов.

Главным достоинством векторной модели является возможность поиска и ранжирования документов по подобию, то есть по их близости в векторном

пространстве. Однако практика показывает, что при оценке близости запроса к документу результаты поиска могут быть не всегда удовлетворительными, что особенно проявляется, когда запрос содержит малое количество слов [50].

Документ в векторной модели рассматривается как неупорядоченное множество термов. Различными способами можно определить вес терма в документе – «важность» слова для идентификации данного текста. Например, можно просто подсчитать количество употреблений терма в документе, так называемую частоту терма, - чем чаще слово встречается в документе, тем больший у него будет вес. Если терм не встречается в документе, то его вес в этом документе равен нулю.

Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если теперь для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов.

Более формально это утверждение можно представить в виде формулы:

$$\vec{d_i} = (w_{i1},w_{i2},...,w_{in}) \tag{2.1}$$

где $\vec{d_i}$ — векторное представление i-го документа, w_{ij} — вес j-го терма в i-м документе, n — общее количество различных термов во всех документах коллекции.

2.3 Схема алгоритма

Условно алгоритм классификации с использованием векторной модели представления знаний можно разделить на несколько последовательных шагов:

1. Подготовка данных (для всех документов)
   - очистка от стоп-слов
   - обработка стеммером Портера (переход от слов к термам)
   - определение вхождения терма в документ

2. Получение набора термов (на основе обучающей выборки)
   - статистический анализ количества вхождений термов в документы, составление терм-документной матрицы
   - расчет матрицы корреспонденций термов (МКТ)
   - ортогональное разложение МКТ, выделение семантического ядра — отбрасывание малозначащих термов
3. Построение категориальных векторов
   - обучение — получение списка категорий (на основе обучающей выборки)
   - расчет векторных моделей категорий в пространстве термов
   - построение категориальных векторов документов базы
4. Подбор вакансий
   - расчет категориального вектора пользователя, для которого происходит подбор рекомендаций
   - расчет коэффициентов близости с категориальными векторами базы вакансий
   - сортировка по убыванию, извлечение q первых элементов

2.4 Подготовка данных к анализу

Классическая векторная модель может выдавать наиболее релевантные документы даже по неполному запросу [47], однако во многих случаях существует ненулевая вероятность, что значимое для поиска слово будет отброшено, в связи с этим предлагается осуществить предварительную обработку обучающего множества.

Очевидно, что текстовые описания формируются обычными людьми, и как следствие часто имеет место сильная зашумленность данных. В связи с этим, прежде, чем переходить к анализу данных, необходимо произвести ряд действий для освобождения текста от шумов. Для этого предлагается использовать: семантическое ядро и стемминг.

Стемминг — это процесс нахождения основы слова для заданного исходного слова.

Основа слова необязательно совпадает с морфологическим корнем слова. Алгоритм стемматизации представляет собой давнюю проблему в области компьютерных наук. Первый документ по этому вопросу был опубликован в 1968 году. Данный процесс применяется в поисковых системах для обобщения поискового запроса пользователя [30].

Конкретные реализации стемматизации называются алгоритм стемматизации или просто стеммер. Наиболее удачный алгоритм стемминга — *стеммер Портера*.

Оригинальная версия стеммера была предназначена для английского языка. Алгоритм не использует баз основ слов, а лишь, применяя последовательно ряд правил, отсекает окончания и суффиксы, основываясь на особенностях языка, в связи с чем работает быстро, но не всегда безошибочно [30].

Семантическое ядро — это подборка понятий, имеющих существенное значение для данной предметной области.

Точное определение семантического ядра зависит от области применения. Так, в лингвистике, семантическим ядром называют «не упрощаемое замкнутое подмножество языка», подразумевая при этом скорее смысловую составляющую языка, а не грамматические конструкции.

Для использования в статистическом анализе текста приведем определения нескольких подборок смысловых единиц, сходных с семантическим ядром.

1. Специфичные слова предметной области

   Это такие слова, которые встречаются исключительно в текстах предметной области и позволяют установить принадлежность текста этой предметной области.

2. Высокоинформативные слова предметной области

   Это такие слова, которые позволяют рубрицировать тексты внутри предметной области. Например, для предметной области «поиск подходящих вакансий» такими словами являются: «няня», «сантехник», «репетитор» и т.д. На рисунке 2.1 проиллюстрирована их частота в выборке из около 1 млн. текстов. В имевшейся в распоряжении тестовой выборке, данные слова встречались чаще всего (не принимая во внимание стоп слова).

Семантическое ядро проще всего сформировать, анализируя большой объем текстов по предметной области. В него попадают слова, которые чаще всего

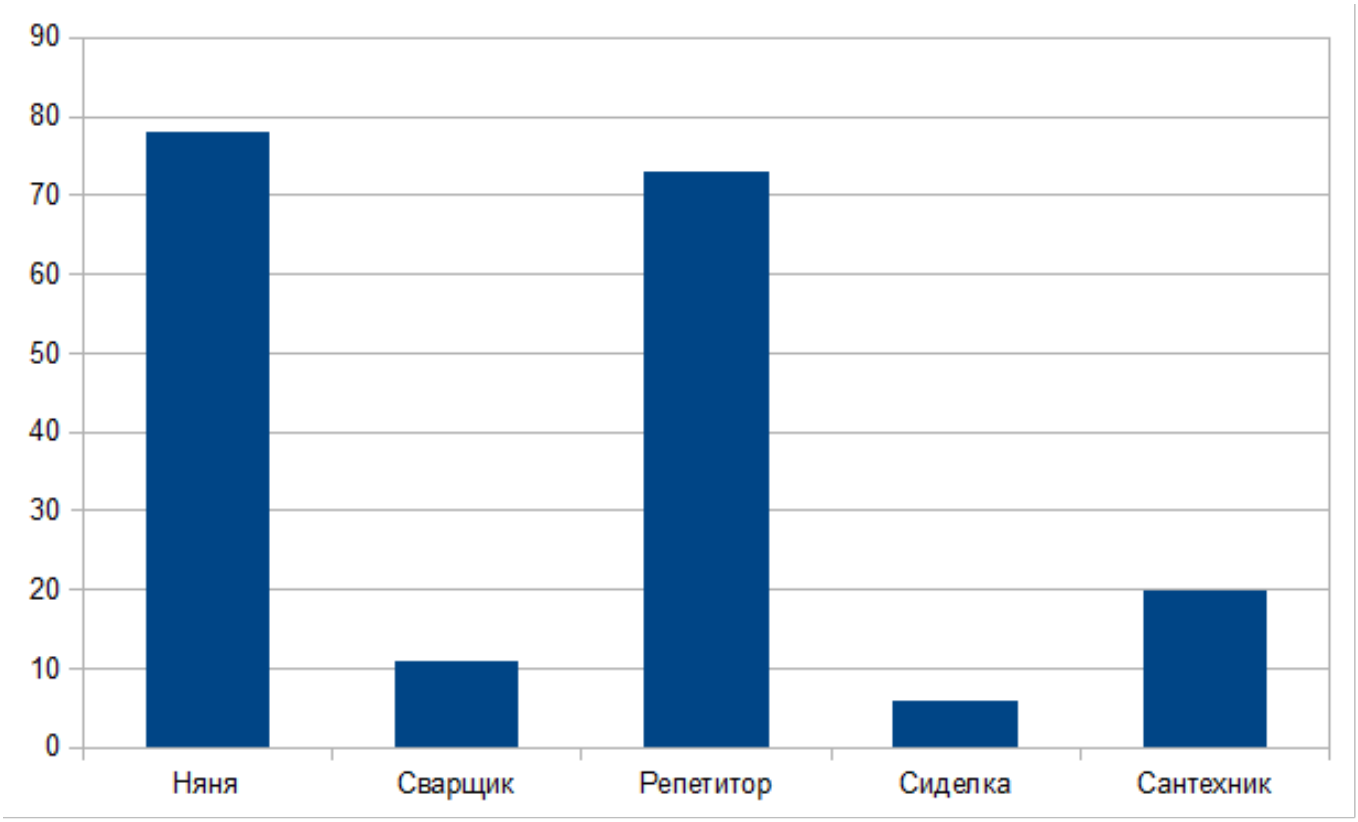

Рисунок 2.1 — Распределение наиболее популярных значимых слов (тыс. текстов)

встречаются в анализируемых текстах, исключая так называемые *стоп-слова*, например, предлоги, союзы и прочие слова, которые не несут смысловой нагрузки. Считается, что каждое из этих общих стоп-слов есть во всех документах выборки. Кроме того, в некоторых предметных областях имеет смысл удалять имена собственные. На рисунке 2.2 изображена частота самых популярных стоп слов для предметной области «подбор персональных рекомендаций в сфере поиска работы»:

В некоторых предметных областях имеет смысл поработать с так называемыми зависимыми стоп-словами. Идея зависимых стоп-слов состоит в том, чтобы не учитывать наличие некоторых слов в документе без наличия других. Например, разбирая тексты предметной области «поиск вакансий разовой работы», при анализе фразы «гибкий график», имеет смысл рассматривать слово «гибкий» только в сочетании со словом «график».

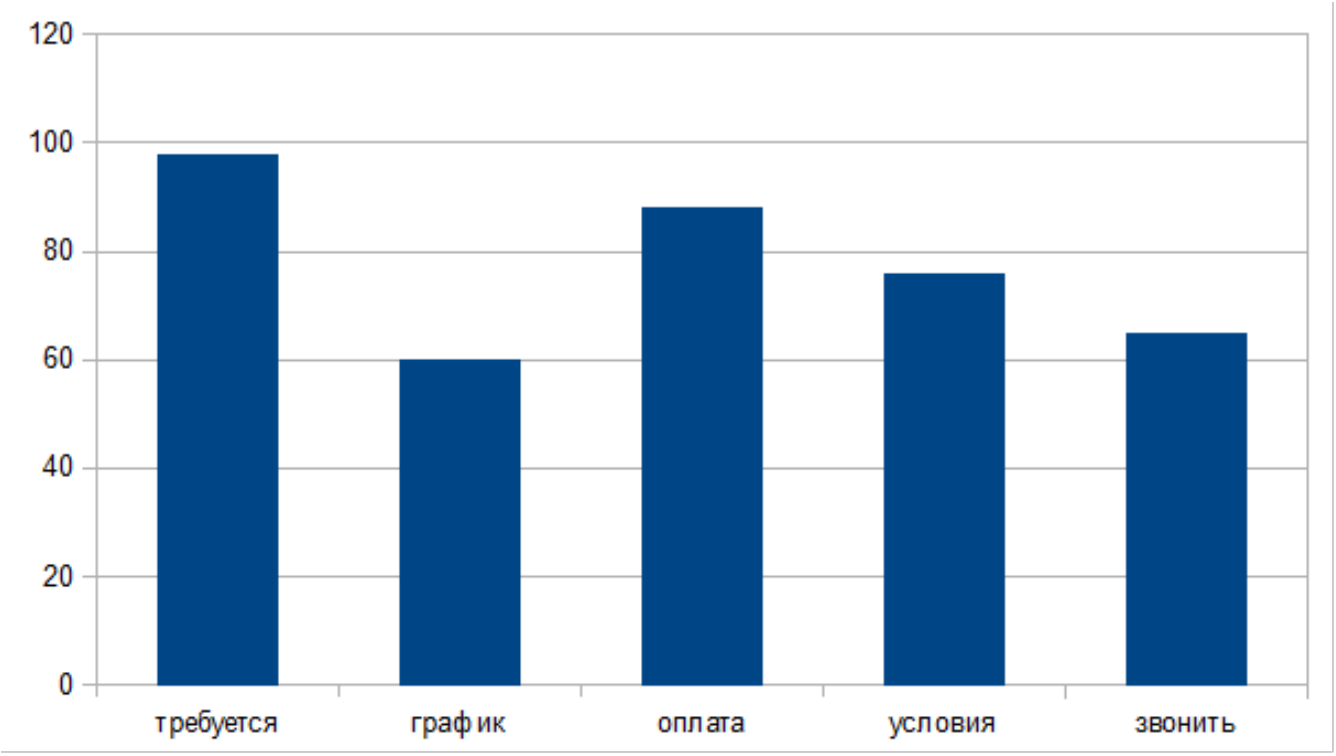

Рисунок 2.2 — Частота наиболее популярных стоп-слов (%)

2.5 ЛСА и сингулярное разложение

ЛСА отображает документы и отдельные слова в так называемое «семантическое пространство», в котором и производятся все дальнейшие сравнения [53, 54]. Для построения семантического пространства используется терм-документная матрица, отражающая количество появлений терминов (термов) в документах.

При этом делаются следующие предположения:

1. Документ это просто набор слов. Порядок слов в документах игнорируется. Важно только то, сколько раз то или иное слово встречается в документе.
2. Семантическое значение документа определяется набором слов, которые, как правило, идут вместе.
3. Каждое слово имеет единственное значение. Это, безусловно, сильное упрощение, но именно оно делает проблему разрешимой.

В качестве исходной информации используется терм-документная матрица X, которая описывает частоту термов.

Предположим у нас есть некоторая обучающая выборка текстов. Представим ее в виде матрицы X, столбцами которой являются x_i - вектора термов, n - количество термов. Вектор терма t_j представляет собой вектор-столбец:

$$\vec{x_j} = \{x_{1j}, x_{2j}, ..., x_{mj}\}^T \tag{2.2}$$

где m — количество документов, содержащихся в обучающей выборке. Частота встречаемости терма в документе равна числу вхождений терма t_j в документ d_i:

$$x_{ij} = tf(t_j, d_i) \tag{2.3}$$

где d_i — i-ый документ из обучающей выборки, $tf(t_j,d_i)$ — частота встречаемости терма t_j в документе d_i (term frequency).

Матрица $X = \{x_{ij}\}$ называется *терм-документной матрицей.*

Стандартная векторная модель представления документа основана на всех термах [104], которые встречаются в документе, либо на всех термах из корпуса. В выборках, содержащих тексты большого объема, это может привести к значительному увеличению времени обучения и объемов хранимой дополнительной информации. Применяя методы, основанные на построении семантического ядра, можно значительно увеличить общую производительность.

Одним из ключевых достоинств данного подхода является его модульность: разграничение собственно алгоритма анализа данных от статистического анализа, необходимого на предварительном этапе. Кроме того, ядра сами по себе имеют модульную структуру, что позволяет с помощью простых правил строить более сложные семантические ядра из более простых таким образом, чтобы они не выходили за границы семантического пространства [52, 109].

Предлагается процедура построения семантического ядра на основе сингулярного разложения *матрицы корреспонденций термов.*

Предположим у нас есть некоторая обучающая выборка. Представим ее в виде матрицы $X = \{x_{ij}\}$, где x_{ij} описываются формулами (2.2)-(2.3).

Тогда обучающую информацию при построении семантического ядра можно представить в виде следующей матрицы, которая состоит их всех возможных скалярных произведений между векторами термов обучающей выборки. Размерность данной матрицы получится $(n \times n)$.

$$G = ((x_i, x_j))_{i,j=1}^{n} \tag{2.4}$$

Матрица, составленная из скалярных произведений, называется матрицей Грама. Очевидно, что любая матрица Грама является симметричной.

После того, как матрица построена, можно приступать к следующему шагу, а именно к ее ортогональному разложению, которое тесно связано с сингулярным разложением ТДМ X.

Сингулярное разложение — это математическая операция, раскладывающая матрицу на три составляющие. Сингулярное разложение произвольной матрицы можно представить в виде следующей формулы.

$$X = USV^T \tag{2.5}$$

где X - исходная матрица, U и V^T — ортогональные матрицы, S — диагональная матрица. На рисунке 2.3 представлена схема сингулярного разложения.

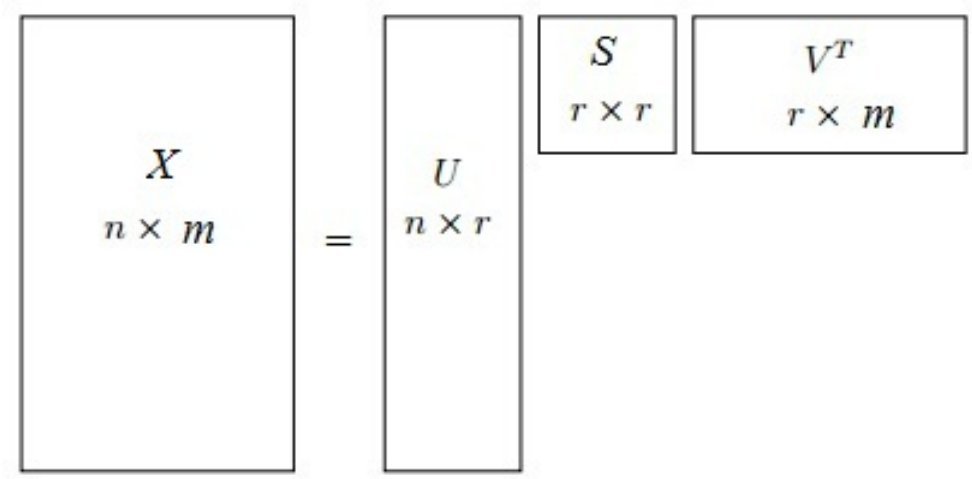

Рисунок 2.3 — Схема сингулярного разложения

Как известно [68], что для любой вещественной $(n \times n)$ матрицы X существуют матрицы U и V такие, что

$$U^T XV = S \tag{2.6}$$

где U и V — квадратные вещественные матрицы размерности $(n \times n)$, S — диагональная матрица размерности $(n \times n)$. Кроме того, матрицы U и V являются ортогональными матрицами, то есть:

$$\begin{aligned} UU^T &= E, \\ VV^T &= E \end{aligned} \tag{2.7}$$

или, что тоже самое, их обратные матрицы равны транспонированным.

Столбцы матрицы U называются *левыми сингулярными векторами* матрицы X, столбцы V (или строки V^T) — *правыми сингулярными векторами.*

Матрица S является диагональной матрицей, значения на диагонали которой называются сингулярными коэффициентами матрицы X. Сингулярные коэффициенты всегда неотрицательны. Более того, матрицы U и V можно подобрать таким образом, чтобы диагональные элементы S выглядели следующим образом:

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0 \tag{2.8}$$

где r - ранг матрицы X, то есть сингулярный коэффициент в строке матрицы X всегда больше, либо равен коэффициенту в строке ниже.

В частности, в случае, если матрица X невырождена, то:

$$\sigma_1 \geq \sigma_2 \geq .. \geq \sigma_n > 0 \tag{2.9}$$

2.6 Вычисление сингулярного разложения

Приведем алгоритм сингулярного разложения, следуя [24]. Вычисление сингулярного разложения сводится к вычислению собственных чисел и собственных векторов матриц XX^T и X^TX. Собственные вектора матрицы X^TX — это столбцы матрицы V, а собственные вектора матрицы XX^T — столбцы матрицы U. В качестве сингулярных значений берутся квадратные корни общих собственных чисел матриц XX^T и X^TX. Количество ненулевых сингулярных коэффициентов соответствует рангу матрицы X, то есть числу линейно-независимых столбцов.

Таким образом, общий алгоритм сингулярного разложения можно представить следующей последовательностью шагов:

1. Вычисление матрицы XX^T.

2. Вычисление собственных чисел и собственных векторов матрицы XX^T.
3. Вычисление матрицы X^TX.
4. Вычисление собственных чисел и собственных векторов матрицы X^TX.
5. Вычисление квадратного корня из общих собственных чисел матриц XX^T и X^TX.
6. Составление матриц U, V и S.

Рассмотрим вычисление собственных чисел отдельно. Собственными числами и собственными векторами квадратной матрицы A называются число λ и вектор $\vec{x}$, такие, что [24]:

$$Ax = \lambda x \tag{2.10}$$

Данное утверждение можно так же записать следующим способом:

$$(A - \lambda E)x = 0, x \neq 0 \tag{2.11}$$

Это означает, что для вычисления собственных чисел достаточно решить уравнение:

$$|A - \lambda E| = 0 \tag{2.12}$$

Степень получившегося многочлена равна размеру матрицы A. Это означает, что, если матрица A имеет размер $(n \times n)$, то она может иметь n собственных чисел. Сам определитель, как и получившийся многочлен, полезны для аналитических рассуждений, однако численное решение уравнения (2.12) в случае большой размерности матрицы A с помощью программного обеспечения представляет собой очень сложную задачу. Наиболее известным и эффективным способом вычисления собственных чисел и векторов считается метод QR, который был представлен Дж. Уилкинсоном в 1965 году [114]. Данный алгоритм используется в большинстве известных математических пакетов (например, MatLab) и библиотек (например, XLispStat).

Для реализации некоторых математических алгоритмов (сингулярное разложение, поиск собственных векторов), использованных в данном исследовании были использованы инструменты библиотеки XlispStat для языка программирования C++. Данная библиотека была разработана Л. Тиерни в университете Минесоты [111]. Она отвечает всем заявленным требованиям к программному обеспечению, изложенных в данной главе.

2.7 Выделение семантического ядра с помощью матрицы корреспонденций термов

2.7.1 Матрица корреспонденций термов

Для решения проблемы выделения семантического ядра введем новое понятие — *матрица корреспонденций термов.*

Определение. Матрица корреспонденций термов $G = \{g_{ij}\}$ — это квадратная матрица, элементами которой являются коэффициенты g_{ij} отражающие близость i-го и j-го термов, для которых выполняются следующие условия:

1. $g_{ij} = g_{ji}$
2. $0 \leqslant g_{ij} \leqslant 1$ для всех i и j.
3. $g_{ij} = 0$ при отсутствии взаимосвязи между термами.

Основное назначение матрицы G — отображение взаимосвязей термов внутри документов, построенное на основе знаний частоте об их совместных употреблениях. На рисунке 2.4 изображен случай, когда термы t_1 и t_2 совместно встречаются в документе d_2, а термы t_2 и t_3 - в документе d_1. Таким образом, термы t_1 и t_3 так же связаны между собой через терм t_2.

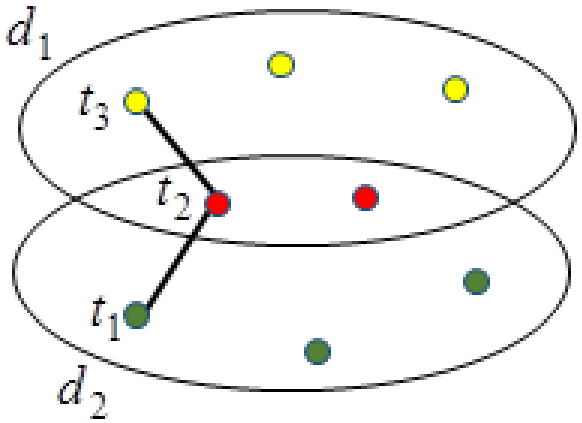

Рисунок 2.4 — Иллюстрация взаимосвязей термов

Основная идея применения ортогонального разложения состоит в том, что часть матрицы Грама содержащая только k линейно независимых компонент, отражает основную структуру зависимостей, присутствующих в исходной мат-

рице. Таким образом, термы, которые чаще встречаются в корпусе, получат в результате разложения более высокие коэффициенты.

Ортогональное разложение аналогично процедуре из метода латентно-семантического анализа (ЛСА) [11], однако, в данном случае вместо терм-документной матрицы, которая используется в ЛСА, разложению подвергается матрица корреспонденций термов G, которая показывает общее количество вхождений терма в документы.

Как правило, выбор k (количество оставляемых компонент) зависит от поставленной задачи и подбирается эмпирически. Если выбранное значение k слишком велико, то метод теряет свою мощность и приближается по характеристикам к стандартным векторным методам. Слишком маленькое значение k не позволяет улавливать различия между похожими термами или документами. Если необходимо выбирать значение k автоматически, то можно, например, установить пороговое значение сингулярных коэффициентов и отбрасывать все строки и столбцы, соответствующие сингулярным коэффициентам, не превышающим данного порогового значения.

Таким образом, в результате применения данного метода происходит переход в новое «семантическое пространство» размер которого оказывается значительно меньше исходного, что в свою очередь приводит к увеличению общей производительности текстового классификатора. Кроме прочего, малозначимые термы пропадают из общего списка термов, что в свою очередь приводит к уменьшению количества ложных выводов классификатора.

Основным недостатком данного метода является общая сложность вычислений. Хотя используемая матрица и значительно меньше, чем матрица термы-на документы, использующаяся в ЛСА, но она как правило сильно разрежена. Это приводит к сложности, а иногда и невозможности использования обучающих выборок большого объема. Чтобы преодолеть данный недостаток, необходима предварительная обработка обучающей выборки: удаление стоп-слов, стеммизация, как описано в разделе 2.2.

2.7.2 Разложение матрицы корреспонденций термов

Как было указано выше, в качестве матрицы корреспонденций термов можно брать различные матрицы. Рассмотрим подробнее случай, когда в качестве МКТ взята матрица, полученная из произведений нормированных векторов-термов.

Рассмотрим терм-документную матрицу. Построим нормированную терм-документную матрицу $Y = \{y_{ij}\}$, где

$$y_{ij} = \frac{x_{ij}}{\sum\limits_{j} x_{ij}} = \frac{x_{ij}}{n_i} \tag{2.13}$$

здесь x_{ij} — число вхождений слова в документ, а n_i - общее количество слов в документе d_j. Через y_j обозначим вектор-столбец

$$y_j = \{y_{1j}, ..., y_{mj}\} \tag{2.14}$$

Построим матрицу, состоящую из всех возможных скалярных произведений векторов термов y_j, определяемых по формулам (2.13) - (2.14):

$$G = ((y_i, y_j))_{i,j=1}^{n} = Y^T Y \tag{2.15}$$

Применим ортогональное разложение к матрице корреспонденций термов G, определенной по формуле (2.15). Сингулярное разложение аналогично процедуре из метода латентно-семантического анализа, однако, в данном случае вместо сингулярного разложения терм-документной матрицы X, которая используется в ЛСА, будем использовать ортогональное разложение матрицы корреспонденций термов G, которая отражает взаимосвязь термов в корпусе.

Матрица корреспонденций термов (МКТ) связана с моделью представления знаний через терм-документную матрицу, однако ее назначение отлично. Разложение корреспонденций термов может быть получено из разложения нормированной терм-документной матрицы Y следующим образом.

Утверждение 1. Ортогональное разложение матрицы корреспонденций термов G, определенной по формуле (2.15) имеет вид:

$$G = VZV^T \tag{2.16}$$

где V – ортогональная матрица правых сингулярных векторов в разложении нормированной терм-документной матрицы Y, матрица Z – диагональная матрица размера, на диагонали которой стоят $(\sigma_i)^2$, σ_i - сингулярные коэффициенты разложения матрицы Y.

Утверждение легко проверяется. Матрицу корреспонденций термов G можно представить следующим образом:

$$G = Y^T Y = VS_Y^T T^T T S_Y V^T \tag{2.17}$$

С учетом ортогональности матрицы D получаем:

$$G = VS_Y S_Y^T V^T = VZV^T \tag{2.18}$$

Часть матрицы, содержащая только k линейно независимых компонент, будет отражать основную структуру зависимостей, присутствующих в исходной матрице. Таким образом, термы, которые чаще встречаются в корпусе, получат в результате разложения более высокие сингулярные коэффициенты. Усеченную матрицу G до размерности k обозначим G_k:

$$G_k = V_k Z_k V_k^T \tag{2.19}$$

Замечание. Применение сингулярного разложения к стандартной терм-документной матрице X и к нормированной терм-документной матрице Y приводит, вообще говоря, к различным семантическим пространствам, поэтому преобразование T_k полученное с использованием МКТ не совпадает с матрицей U_k стандартного ЛСА.

Рассмотрим пример. Пусть терм-документная матрица X равна:

$$X = \begin{pmatrix} 100 & 100 & 100 & 0 & 0 \\ 1 & 3 & 1 & 0 & 1 \\ 2 & 0 & 3 & 1 & 0 \\ 4 & 1 & 1 & 1 & 1 \\ 4 & 3 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Сингулярное разложение терм-документной матрицы имеет вид $X = USV^T$, где:

$$U = \begin{pmatrix} -1.00 & -0.03 & 0.01 & 0.01 & 0.01 \\ -0.02 & 0.15 & 0.40 & -0.76 & -0.48 \\ -0.02 & -0.19 & -0.66 & -0.12 & -0.43 \\ -0.02 & 0.58 & -0.48 & 0.05 & -0.30 \\ -0.02 & 0.77 & 0.18 & 0.17 & 0.12 \\ -0.01 & 0.11 & -0.37 & -0.61 & 0.69 \end{pmatrix}$$

$$S = \begin{pmatrix} 173.3 & 0 & 0 & 0 & 0 \\ 0 & 3.949 & 0 & 0 & 0 \\ 0 & 0 & 3.411 & 0 & 0 \\ 0 & 0 & 0 & 1.557 & 0 \\ 0 & 0 & 0 & 0 & 0.048 \end{pmatrix}$$

Запишем нормализованную матрицу

$$Y = \begin{pmatrix} 0.333 & 0.333 & 0.333 & 0 & 0 \\ 0.116 & 0.5 & 0.116 & 0 & 0.116 \\ 0.333 & 0 & 0.5 & 0.116 & 0 \\ 0.5 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.5 & 0.375 & 0 & 0 & 0.125 \\ 0.25 & 0 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

Матрицы T и S_Y ее сингулярного разложения имеют вид:

$$T = \begin{pmatrix} -0.45 & -0.15 & -0.46 & 0.31 & -0.56 \\ -0.38 & -0.52 & -0.35 & -0.53 & 0.43 \\ -0.38 & 0.53 & -0.41 & 0.31 & 0.35 \\ -0.44 & 0.09 & 0.52 & 0.25 & 0.43 \\ -0.41 & -0.41 & 0.44 & 0.20 & -0.25 \\ -0.38 & 0.50 & 0.21 & -0.66 & -0.36 \end{pmatrix}$$

$$S_Y = \begin{pmatrix} 1.214 & 0 & 0 & 0 & 0 \\ 0 & 0.573 & 0 & 0 & 0 \\ 0 & 0 & 0.38 & 0 & 0 \\ 0 & 0 & 0 & 0.274 & 0 \\ 0 & 0 & 0 & 0 & 0.007 \end{pmatrix}$$

В приведенном примере результаты разложения нормированной и ненормированной матриц — существенно разные, так как им соответствуют разные матрицы линейных преобразований U и T. Так же различными будут размерности усеченных матриц, поскольку будет отброшено разное количество термов. В первом случае (без нормировки) будут отброшены все, кроме первого терма, во втором (с нормировкой) все, кроме первого и второго. Это в свою очередь приведет к образованию разных семантических пространств и получению разных результатов обучения. Такой результат в примере получен, прежде всего, за счет существенного различия в количествах термов в документах. Нетрудно проверить, что при одинаковом числе термов в документах оба разложения дадут один и тот же результат.

Таким образом, матрица T_k является матрицей линейного преобразования, переводящей вектора из исходного пространства в семантическое. Вычислительные эксперименты показали, что при проведении интеллектуального анализа текстов с использованием матрицы корреспонденций термов внутренние зависимости отражаются более явно, что приводит к большему снижению размерности семантического пространства. Отметим, что предлагаемый метод выделения семантического ядра на основе сингулярного разложения МКТ аналогичен методу главных компонент в статистическом анализе, если в качестве меры близости термов взять коэффициент корреляции.

Использование матрицы корреспонденций термов для выделения семантического ядра является более гибким методом по сравнению с латентно-семантическим анализом, поскольку позволяет использовать различные меры близости термов. Использование нестандартных мер может быть необходимо, например, при обработке большого количества текстов из разных предметных областей в рамках обучения одного классификатора.

2.8 Свойства матрицы корреспонденций термов

2.8.1 Свойства собственных чисел

Будем рассматривать коллекцию документов $D = \{d_1, d_2, ..., d_m\}$ и набор термов $T = \{t_1, t_2, ..., t_n\}$. Будем предполагать, что длина первых документов k, где $1 \leq k \leq n$, равна Φ, а длина оставшихся ϕ. В этом случае терм-документная матрица $X = (x_{ij})_{i=1,j=1}^{m,n}$ запишется в виде

$$x_{ij} = \begin{cases} \Phi a_{ij} & \text{для} \quad i = \overline{1,k}, \quad j = \overline{1,n}, \\ \phi b_{ij} & \text{для} \quad i = \overline{k+1,m}, \quad j = \overline{1,n}. \end{cases}$$

Относительно чисел $\Phi, \phi, a_{ij}, b_{ij}$ будем предполагать выполненными следующие условия

$$a_{ij} \geq 0 \qquad \forall i = \overline{1,k}, \quad j = \overline{1,n}, \tag{2.20}$$

$$b_{ij} \geq 0 \qquad \forall i = \overline{k+1,m}, \quad j = \overline{1,n}, \tag{2.21}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \qquad \forall i = \overline{1,k}, \tag{2.22}$$

$$\sum_{j=1}^{n} b_{ij} = 1 \qquad \forall i = \overline{k+1,m}, \tag{2.23}$$

$$\Phi > \phi \geq 1. \tag{2.24}$$

Матрицу X удобно записать в виде

$$X = \Phi A + \phi B, \tag{2.25}$$

где A - матрица $m \times n$, у которой элементами первых n строк являются элементы a_{ij}, а все элементы остальных $m - k$ строк равны нулю, B - матрица $m \times n$, у которой элементы первых k строк равны нулю, а элементами остальных $m - k$

строк являются числа b_{ij}:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kn} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ b_{k+1,1} & b_{k+1,2} & \dots & b_{k+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix}.$$

Для матрицы $G = X^T X$ получаем

$$G = \big(\Phi A + \phi B\big)^T \big(\Phi A + \phi B\big) =$$

$$\Phi^2 A^T A + \Phi\phi A^T B + \phi\Phi B^T A + \phi^2 B^T B.$$

Вводя обозначения $G_A = A^T A$, $G_B = B^T B$ и замечая, что $A^T B = B^T A = 0$, получаем

$$G = \Phi^2 G_A + \phi^2 G_B. \tag{2.26}$$

Несложно видеть, что G, G_A и G_B являются симметричными и неотрицательно определенными матрицами. Также отметим, что в силу теоремы о ранге произведения матриц [24, 114]

$$rank\ G_A \le \min\big\{rank\ A^T, rank\ A\big\} = rank\ A \le k.$$

Перечисленные свойства означают, что все собственные числа матрицы G_A являются вещественными, неотрицательными и, по крайней мере, $n - k$ из них равны нулю [24, 114].

Основной вопрос, который нас интересует, заключается в следующем: какое влияние на собственные числа матрицы G оказывают числа Φ и ϕ при условии, что Φ существенно больше чем ϕ? Для ответа на этот вопрос нам понадобятся некоторые обозначения и утверждения. В частности, для произвольной матрицы $P = (p_{ij})_{i,j=1}^n$ будем использовать величину

$$\|P\|_E = \sqrt{\sum_{i,j=1}^{n} p_{ij}^2},$$

которую называют евклидовой нормой матрицы. Отметим, что для любых матриц P и Q справедливы соотношения

$$\|PQ\|_E \le \|P\|_E \|Q\|_E, \tag{2.27}$$

$$\|P^T\|_E = \|P\|_E. \tag{2.28}$$

У симметричной матрицы P все собственные числа являются вещественными, поэтому их можно упорядочить. Обозначим через $\lambda_s(P)$ собственное число матрицы P, занимающее s-ую позицию в упорядоченном по убыванию множестве собственных чисел матрицы P. Из определения собственного числа следует, что при любом μ справедливо $\lambda_s(\mu P) = \mu\lambda_s(P)$ Для симметричных матриц справедливы следующие утверждения ([114], [24]).

Теорема 1. *Пусть P и Q — симметричные матрицы размера $n \times n$, тогда*

$$|\lambda_s(P) - \lambda_s(Q)| \le \|P - Q\|_E \qquad \forall s = \overline{1,n}.$$

Теорема 2. *Пусть P — симметричная матрица и Q — неотрицательно определенная матрица, тогда*

$$\lambda_s(P + Q) \ge \lambda_s(P) \quad \forall s = \overline{1,n}.$$

Используя эти теоремы можно оценить влияние величин Φ и ϕ на собственные значения матрицы G.

Теорема 3. *Пусть выполнено условие (2.23). Тогда для любого $s = \overline{1,n}$ справедлива оценка*

$$\Phi^2\lambda_s(G_A) \le \lambda_s(G) \le \Phi^2\lambda_s(G_A) + \phi^2(m - k). \tag{2.29}$$

Доказательство. В силу симметричности матриц G и G_A из теоремы 1 следует, что для любого $s = \overline{1,n}$

$$|\lambda_s(G) - \lambda_s(\Phi^2 G_A)| \le \|G - \Phi^2 G_A\|_E =$$

$$= \|\phi^2 G_B\|_E = \phi^2\|G_B\|_E = \phi^2\|B^T B\|_E. \tag{2.30}$$

Из условия (2.23) получаем

$$\|B\|_E = \sqrt{\sum_{i=k+1}^{m}\sum_{j=1}^{n} a_{ij}^2} \le \sqrt{\sum_{i=k+1}^{m} 1} = \sqrt{m-k}. \tag{2.31}$$

Отсюда и из свойств (2.27), (2.28) получаем

$$|\lambda_s(G) - \lambda_s(\Phi^2 G_A)| \le \phi^2 \|B^T\|_E \|B\|_E = \phi^2 \|B\|_E^2 = \phi^2(m-k). \tag{2.32}$$

Поскольку матрица $\phi^2 G_B$ является неотрицательно определенной, то из теоремы 2 мы получаем для любого $s = \overline{1,n}$

$$\lambda_s(G) \ge \lambda_s(\Phi^2 G_A).$$

Отсюда и из оценки(2.32) следует (2.29). □

Если $\lambda_s(G_A) \ne 0$, то оценку (2.29) удобно переписать в виде

$$1 \le \frac{\lambda_s(G)}{\lambda_s(\Phi^2 G_A)} \le 1 + \frac{\phi^2(m-k)}{\Phi^2 \lambda_s(G_A)}. \tag{2.33}$$

Отсюда следует, что для любого $\varepsilon > 0$ найдется такое $\Phi^* = \Phi^*(m, k, \phi, A)$, что при $\Phi > \Phi^*$ справедлива оценка

$$1 \le \frac{\lambda_s(G)}{\lambda_s(\Phi^2 G_A)} \le 1 + \varepsilon. \tag{2.34}$$

Если $\lambda_s(G_A) = 0$ (выше было показано, что это верно, в частности, при $s > n - k$), то оценка (2.29) будет иметь вид

$$0 \le \lambda_s(G) \le \phi^2(m-k). \tag{2.35}$$

2.8.2 Влияние длины документа на сингулярное разложение матрицы

Пусть X – исходная терм-документная матрица (ТДМ), Y – нормированная терм-документная матрица, $G = G_Y = Y^T Y$ – матрица корреспонденций термов. Будем сравнивать результаты сингулярного сглаживания терм-документной матрицы и ортогонального разложения МКТ. Как показано выше, разложение МКТ сводится к сингулярному разложению нормированной терм-документной матрицы Y, и, соответственно, разложение ненормированной матрицы Грамма $G_X = X^T X$ сводится к сингулярному разложению терм-документной матрицы X. Очевидно, что выполняется следующее свойство.

Утверждение 1. *Пусть длина всех документов в коллекции одинакова, т.е.*

$$\sum_{j=1}^{n} n_{ij} = n_i = \Phi,$$

тогда $X = \Phi Y$ и сингулярное разложение матрицы X имеет вид

$$X = U(\Phi S)V^T,$$

где унитарные матрицы U и V – матрицы в сингулярном разложении нормированной терм-документной матрицы, т.е. $Y = USV^T$.

Таким образом, сингулярные числа ТДМ выражаются через собственные числа МКТ по формуле:

$$\sigma_s = \Phi\sqrt{\lambda_s},$$

где λ_s собственное число матрицы G_Y, занимающее s-ую позицию в упорядоченном по убыванию множестве собственных чисел матрицы G_Y, σ_s – s-ое сингулярное число матрицы X.

Следствие 1. *Пусть все документы коллекции имеют одинаковую длину и для уменьшения размерности пространства термов на основе сингулярного разложения терм-документой матрицы X в качестве критерия отбрасывания столбцов, соответствующих малым собственным числам, выбрано усло-*

вие

$$\sigma_s < \sigma^*. \tag{2.36}$$

Тогда количество и состав оставляемых термов совпадет результатом ортогонального разложения матрицы корреспонденций термов G, при условии отбрасывания столбцов, сопутствующих собственным значениям, удовлетворяющим условию

$$\lambda_s < \lambda^*, \ \textit{где} \ \ \lambda^* = \frac{(\sigma^*)^2}{\Phi^2}.$$

Ситуация меняется, если документы имеют существенно различную длину. Рассмотрим случай, когда документы коллекции делятся на два типа:

— длинные, с количеством термов, равным Φ;

— короткие, с количеством термов в каждом документе, равным ϕ, $\phi < \Phi$.

В этом случае терм-документную матрицу X можно записать в форме блочной матрицы, определяемой соотношениями (2.20)–(2.25). Первые k строк матрицы X соответствуют длинным документам, остальные – коротким. Нормированная матрица терм-документная матрица будет равна $Y = A + B$. Если короткие документы значительно короче длинных, т. е. выполняется неравенство

$$\frac{\phi}{\Phi} < \epsilon, \tag{2.37}$$

то при сингулярном разложении МКТ и отбрасывании термов, соответствующих малым сингулярным числам, останутся только термы, соответствующие длинным документам, так как разложения матриц X и ΦA будут близки. Из теоремы 3 следует следующее утверждение.

Теорема 4. *Пусть проводится выделение семантического ядра из коллекции k длинных документов, длиной Φ, $m-k$ коротких документов, длиной ϕ, причем длина коротких документов удовлетворяет условию (2.37). Тогда сингулярные числа $\sigma_s(X)$ в разложении терм-документной матрицы X близки к сингулярным числам $\sigma_s(\Phi A)$ терм-документной матрицы ΦA, содержащей только длинные документы, при $s = \overline{1,r_A}$, где r_A – ранг матрицы A.*

Для сингулярных чисел матрицы X выполняется неравенство

$$\sigma_s(\Phi A) \le \sigma_s(X) \le \sigma_s(\Phi A)\left(1 + 0.5\epsilon^2 \cdot \frac{(m-k)}{\sigma_s^2(A)}\right), \textit{если } s \le r_A. \tag{2.38}$$

Здесь $\sigma_s(A)$ — s-ый собственный вектор матрицы A, $\Phi\sigma_s(A) = \sigma_s(\Phi A)$.

При $s > r_A$ сингулярные числа матрицы X удовлетворяют неравенству

$$0 \le \sigma_s(X) \le \phi\sqrt{m-k}. \tag{2.39}$$

Доказательство. Так как квадраты сингулярных чисел матриц X и ΦA равны собственным значениям матриц $G = X^T X$ и $\Phi^2 G_A = \Phi^2 A^T A$ соответственно, то из неравенства (2.29) получаем неравенство для сингулярных чисел ТДМ X

$$\sigma_s^2(\Phi A) \le \sigma_s^2(X) \le \sigma_s^2(\Phi A) + (m-k)\phi^2.$$

Отсюда с учетом (2.37) для $\sigma_s(\Phi A) > 0$ получаем

$$\sigma_s(X) - \sigma_s(\Phi A) \le \frac{(m-k)\phi^2}{\sigma_s(X) + \sigma_s(\Phi A)} \le \frac{(m-k)\phi^2}{2\sigma_s(\Phi A)}.$$

Таким образом, для сингулярных чисел матриц X и ΦA при $s = \overline{1,r_A}$, где r_A – ранг матрицы A, выполняется

$$\sigma_s(\Phi A) \le \sigma_s(X) \le \sigma_s(\Phi A)\left(1 + 0.5\frac{(m-k)\phi^2}{\sigma_s^2(\Phi A)}\right).$$

Учитывая равенство (2.37) и то, что $\Phi\sigma_s(A) = \sigma_s(\Phi A)$ получаем

$$\Phi\sigma_s(A) \le \sigma_s(X) \le \Phi\sigma_s(A)\left(1 + 0.5\epsilon^2 \cdot \frac{(m-k)}{\sigma_s^2(A)}\right), \text{если } \sigma_s(A) \ne 0.$$

Неравенство (2.39) непосредственно следует из (2.35). □

Замечание 1. *Неравенство (2.38) можно записать уточнить, используя норму матрицы B:*

$$\Phi\sigma_s(A) \le \sigma_s(X) \le \Phi\sigma_s(A) + 0.5\epsilon\frac{\phi \parallel B \parallel^2}{\sigma_s(A)}, \textit{если } \ \sigma_s(A) \ne 0. \tag{2.40}$$

Замечание 2. *Отметим, что из неравенств (2.38) следует, что разница между сингулярными числами матриц X и ΦA, вообще говоря, увеличивается с ростом номера s и уменьшением сингулярных чисел.*

Пример 1. Рассмотрим упрощенный пример обработки 6 документов с учетом количества вхождений 5-ти термов. Под длиной документа d_i здесь по-

нимается сумма вхождений термов, подлежащих учету

$$n_i = \sum_{j=1}^{n} n_{ij},$$

где n_{ij} – количество вхождений j-го терма в i-ый документ.

Пусть коллекция документов состоит из 4-х длинных документов, длина каждого из которых $n_i = \Phi = 500$ термов ($i = 1, \ldots, 4$), и 2-х коротких, содержащих по $n_i = \varphi = 50$ термов ($i = 5, 6$). Терм-документная матрица имеет вид

$$X = \begin{pmatrix} 285 & 60 & 90 & 55 & 10 \\ 105 & 30 & 265 & 80 & 20 \\ 117 & 25 & 167 & 151 & 40 \\ 82 & 48 & 155 & 132 & 83 \\ 14 & 16 & 2 & 8 & 10 \\ 4 & 18 & 12 & 10 & 6 \end{pmatrix}$$

содержат данные о числе встречаемости 5-ти термов $t_1, \ldots, t_5$ в этих документах. Сингулярные коэффициенты матрицы X (без нормировки) равны

$$\{509.5, 197.9, 98.4, 40.1, 13.1\}.$$

Найдем сингулярное разложение матрицы ΦA, у которой первые 4 строки совпадают со строками матрицы X, а остальные состоят из нулей. Ее сингулярные числа равны

$$\{508.9, 197.7, 97.7, 37.1, 0\}.$$

Соответственно, сингулярные числа матрицы A равны $\{1.02, 0.4, 0.19, 0.07, 0\}$. Найдем относительные разности δ_s, $s = \overline{1,4}$, где

$$\delta_s = \frac{\sigma_s(X) - \sigma_s(\Phi A)}{\sigma_s(\Phi A)}.$$

Получим значения относительно небольшие значения, которые немного возрастают с ростом номера $\{0.0012, 0.0011, 0.0077, 0.078\}$. Проверка показывает, что в данном примере неравенство (2.38) выполнено.

Теорема 5. *Пусть в условиях Теоремы 4 критерий отбрасывания сингулярных чисел имеет вид (2.36) и выполняются следующие условия:*

1. *верно неравенство*

$$\sigma^* > \phi\sqrt{m-k}; \tag{2.41}$$

2. *первое отбрасываемое сингулярное число $\sigma_{s^*+1}(\Phi A)$ матрицы ΦA удовлетворяет условию*

$$\sigma_{s^*+1}(\Phi A) < \sigma^* - \Delta, \tag{2.42}$$

$$\Delta = 0.5\epsilon\frac{(m-k)\phi}{\sigma_{s^*+1}(A)}. \tag{2.43}$$

Тогда при любых результатах обработки коротких документов количество оставляемых сингулярных чисел в разложении матрицы X, содержащей все документы, и матрицы ΦA, содержащей только длинные документы, совпадает. Причем оставшиеся сингулярные числа удовлетворяют условию:

$$\sigma_s(\Phi A) < \sigma_s(X) < \sigma_s(\Phi A) + \Delta, \quad s = \overline{1,s^*}. \tag{2.44}$$

Доказательство. Из условия (2.41) и неравенства (2.39) Теоремы 4 следует, что все сингулярные числа матрицы X при $s > r(A)$ удовлетворяют условию $\sigma_s(X) < \sigma^*$.

Запишем неравенство (2.42) для сингулярного числа $\sigma_{s^*+1}(X)$ матрицы X

$$\sigma_{s^*+1}(X) \leq \sigma_{s^*+1}(\Phi A) + 0.5\epsilon\frac{(m-k)\phi}{\sigma_{s^*+1}(A)}.$$

С учетом условий (2.42) и (2.43) получаем, что

$$\sigma_{s^*+1}(X) < \sigma^* - \Delta + 0.5\epsilon\frac{(m-k)\phi}{\sigma_{s^*+1}(A)} = \sigma^*.$$

Таким образом, сингулярное число $\sigma_{s^*+1}(X)$ отбрасывается, т.к. оно меньше σ^*.

Сингулярные числа матрицы X при $s > s^*$ удовлетворяют условию

$$\sigma_s(X) \geq \sigma_s(\Phi A) \geq \sigma^*,$$

и следовательно сохраняются в разложении.

Выполнение неравенства (2.44) следует из того, что $\sigma_s(A) \geq \sigma_{s^*+1}(A)$ при $s \geq s^*$ и, следовательно,

$$0.5\epsilon\frac{(m-k)\phi}{\sigma_s(A)} \leq 0.5\epsilon\frac{(m-k)\phi}{\sigma_{s^*+1}(A)} = \Delta.$$

□

Пример 2. Продолжим рассмотрение матриц из примера 1. Пусть выбрано $\sigma^* = 110$. В разложении матрицы ΦA только 2 сингулярных числа удовлетворяют условию $\sigma_s \geq \sigma^*$, таким образом, $s^* = 2$. Найдем

$$\Delta = 0.5\epsilon\frac{(m-k)\phi}{\sigma_{s^*+1}(A)} \approx \frac{0.5 \cdot 0.1 \cdot 2 \cdot 50}{0.19} < 2.7.$$

Таким образом, при любых результатах обработки коротких документов (т.е. при любой матрице B) при сингулярном разложении матрицы X составленной по всей коллекции документов будет сохранено 2 наибольших сингулярных числа, которые будут отличатся от сингулярных чисел матрицы ΦA не более, чем на 3 единицы. Последнее утверждение было проверено рядом численных экспериментов.

2.8.3 Переход к новому базису

Рассмотрим вопрос как влияют длины документов на новый базис семантического пространства, т.е. сравним усеченные матрицы собственных векторов $V = V_X$ и $V = V_A$ матриц X^TX и A^TA.

Будем использовать следующее утверждение о свойствах собственных векторов симметричной матрицы [114], стр. 220.

Теорема 6. *Пусть x – собственный вектор матрицы A, отвечающий собственному значению λ, x_* –собственный вектор приближенной матрицы A_*, отвечающий собственному значению λ_*, тогда*

$$|\sin\alpha| < \frac{\|A - A_*\|_E}{\gamma}, \tag{2.45}$$

где α – угол между векторами x и x_*, γ – расстояние от до ближайшего не совпадающего с λ собственного значения матрицы A.

В качестве меры рассогласования собственных векторов выбран синус угла между ними, а не разность координат из-за того, что собственные векторы определяются с точностью до постоянной.

Рассмотрим вопрос о расхождении между собственными векторами, соответствующими s-м собственным числам матриц $G = X^T X$ и $\Phi^2 G_A = \Phi^2 A^T A$. Отметим, что собственные вектора матрицы $\Phi^2 A^T A$ совпадают с собственными векторами матрицы $A^T A$.

Будем рассматривать случай, когда все оставляемые после разложения сингулярные числа терм-документной матрицы $\Phi^2 A^T A$ – простые, т.е.

$$\sigma_1 > \sigma_2 > \ldots > \sigma_{s^*} \geq \sigma^*.$$

Обозначим через δ_s минимальное расстояние от собственного числа $\lambda_s = \sigma_s^2(A)$ матрицы $A^T A$ до других собственных чисел этой матрицы:

$$\delta_s(A) = \min\{\sigma_s^2(A) - \sigma_{s+1}^2(A), \sigma_{s-1}^2(A) - \sigma_s^2(A)\}, \tag{2.46}$$

Следствие 2. *Пусть проводится выделение семантического ядра из коллекции k длинных документов, длиной Φ, $m-k$ коротких документов, длиной ϕ, и длина коротких документов удовлетворяет условию (2.37). Кроме того все сингулярные числа терм-документной матрицы ΦA, составленной только из длинных документов удовлетворяющие неравенству $\sigma_s \geq \sigma^*$ различны. Тогда правые собственные вектора матриц $G = X^T X$ и $G_A = A^T A$, соответствующие оставляемым сингулярным числам удовлетворяют неравенству:*

$$|\sin \alpha_s| \leq \epsilon^2 \frac{(m-k)}{\delta_s(A)}, \tag{2.47}$$

где δ_s определяется из соотношения (2.46).

Доказательство. Из теоремы 6 следует неравенство

$$|\sin \alpha_s| \leq \frac{\|\phi^2 B^T B\|}{\delta_s(\Phi A)}.$$

Из оценки (2.31) нормы матрицы B и определения $\delta_s(A)$ получаем

$$|\sin\alpha_s| \le \frac{(m-k)\phi^2}{\Phi^2\delta_s(A)}. \tag{2.48}$$

При выполнении (2.37) выполняется неравенство (2.47). □

Пример 4. Продолжим рассмотрение Примера 3. Пусть при отбрасывании выбран критерий $\sigma_s > 100$, тогда будут оставлены по 2 сингулярных числа для терм-документной матрицы X и для ТДМ ΦA, построенной на основе только длинных документов. Новые базисы состоят в обоих случаях из двух векторов и находятся как линейная комбинация термов с помощью матриц $V = V_X$, $V = V_A$ состоящих из правых собственных векторов единичной длины, т.е. нормированных собственных векторов матриц X^TX и Φ^2A^TA соответственно. Расчет с помощью стандартного математического пакета Mathcad показывает, что эти матрицы равны равны соответственно (с точностью $0.5\cdot10^{-4}$)

$$V_X = \begin{pmatrix} -0.584 & -0.160 & -0.672 & -0.403 & -0.143 \\ 0.790 & 0.105 & -0.521 & -0.261 & -0.157 \end{pmatrix},$$

$$V_A = \begin{pmatrix} -0.584 & -0.158 & -0.672 & -0.402 & -0.142 \\ 0.790 & 0.105 & -0.520 & -0.262 & -0.159 \end{pmatrix}.$$

Видно, что эти матрицы близки, $\|V_X - V_A\|_E < 0.002$.

Проведем оценку с помощью неравенства (2.47). Собственные числа матрицы A^TA равны $\{1.036, 0.156, 0.038, 0.0052, 0\}$, поэтому расстояния до ближайших собственных чисел равны соответственно $\delta_1 = 0.880$, $\delta_2 = 0.118$, что дает следующую оценку для синусов углов между собственными векторами матриц X^TX и Φ^2A^TA (для первого и второго собственного числа)

$$\sin\alpha_1 \le 2.3\times10^{-2}, \quad \sin\alpha_2 \le 0.13.$$

Эта оценка верна для любой матрицы ϕB, составленной из документов длиной 50 термов, для конкретной матрицы оценка может быть уточнена. Расчеты показывают, что синусы углов значительно меньше в данном примере, и равны соответственно

$$\sin\alpha_1 = 2.2\times10^{-3}, \quad \sin\alpha_2 = 2.4\times10^{-3}.$$

Таким образом, выделение семантического ядра на основе всех 6 документов и только 4-х первых, имеющих большую длину, примерно совпадут.

Замечание 3. *Если часть термов содержится только в коротких документах и проводится выделение семантического ядра путем сингулярного разложения ТДМ* $X = \Phi A + \phi B$, *содержащей* k *длинных и* $m - k$ *коротких документов, то после выделения семантического ядра на основе отбрасывания сингулярных чисел по условию (2.37), при существенной разнице в длине документов будут оставлены только сингулярные числа, соответствующие термам из длинных документов. Тем не менее термы, встречающиеся только в коротких документах, войдут в итоговое семантическое пространство, правда с небольшими коэффициентами. Если проводить выделение семантического ядра на основе МКТ (что сводится к сингулярному разложению нормированной ТДМ), то термы, входящие в длинные и короткие документы учитываются одинаково.*

Теорема 7. *Пусть* K, $K < n$, *термов* $\{t_1, \ldots, t_K\}$ *содержатся только в длинных документах длины* Φ, *остальные* $n - K$ *содержатся только в коротких документах длины* ϕ *и*

$$\phi < \epsilon\Phi.$$

Тогда при выделении семантического ядра путем сингулярного разложения ТДМ $X = \Phi A + \phi B$, *с условием отбрасывания сингулярных чисел (2.37) все* $n - K$ *термов* $\{t_{K+1}, \ldots, t_n\}$ *не будут учитываться при построении семантического ядра при достаточно малых* $\epsilon > 0$.

Доказательство. В рассматриваемых условиях у матрицы A последние $n - K$ столбцов состоят из нулей. Обозначим матрицу размера $m \times K$, состоящую из первых K столбцов матрицы A через A_1. Аналогично, у матрицы B первые K столбцов - нулевые. Обозначим матрицу размера $(m-k) \times (n-K)$, содержащую последние $(n - K)$ столбцов матрицы B через B_1. Таким образом,

$$A = (A_1\ 0),\ B = (0\ B_1).$$

Получаем, что матрица $G_X = X^T X$ имеет блочную структуру

$$G_X = \begin{pmatrix} \Phi^2 A_1^T A_1 & 0 \\ 0 & \phi^2 B_1^T B_1 \end{pmatrix}$$

поэтому множество ее собственных векторов состоит из объединения собственных векторов матриц $\Phi^2 A_1^T A_1$ и $\phi^2 B_1^T B_1$ дополненных нулями до размерности n. Собственные вектора матрицы $\phi^2 B_1^T B_1$ соответствуют собственным числам $\phi^2 \lambda_s(B^T B)$, где $\lambda_s(B^T B)$ – собственные числа матрицы $B^T B$. Собственные вектора отбрасываются, если соответствующие собственные числа удовлетворяют неравенству $\lambda_s < \Phi^2 \lambda^*$, поэтому если ϵ достаточно мало и $\phi < \epsilon\Phi$, то все собственные вектора матрицы $\phi^2 B_1^T B_1$ будут отброшены, и урезанная матрица правых собственных векторов $\tilde{V}$ в сингулярном разложении ТДМ после сглаживания $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^T$ будет состоять только из собственных векторов матрицы $\Phi^2 A_1^T A_1$ (часть из которых также будет отброшена).

□

Утверждение теоремы верно, если документы имеют различную длину, но разбиваются по количеству содержащихся термов на 2 группы.

Следствие 3. *Пусть K, $K < n$, термов $\{t_1, \ldots, t_K\}$ содержатся только в k длинных документов, длины которых больше заданного значения: $\Phi_j > \Phi$, $j = 1, \ldots, k$; остальные $n - K$ термов содержатся только в коротких документах, длины которых ϕ_j, $j = k+1, \ldots,$ тменьше заданного значения, т.е.*

$$\phi_i < \epsilon\Phi.$$

Тогда при выделении семантического ядра путем сингулярного разложения ТДМ X, с условием отбрасывания сингулярных чисел (2.37), все $n - K$ термов $\{t_{K+1}, \ldots, t_n\}$ не будут учитываться при построении семантического ядра при достаточно малых $\epsilon > 0$.

2.9 Алгоритм подбора персональных рекомендаций

Обилие информации, доступной в интернете в сочетании с ее динамичностью предопределило возрастающую сложность поиска информации и повышение требований к представлению результатов поиска. В следствии чего появилась необходимость получения персонализированного доступа к информации.

Системы подбора персональных рекомендаций необходимы для того, чтобы пользователь на свой запрос получил выдачу не только соответствующую его запросу, а так же отсортированную в соответствии с его предпочтениями [98]. В большинстве случаев системы подбора персональных рекомендаций используют для формирования выдачи информацию, которую пользователь сообщил о себе ранее, например, описание его интересов, резюме и прочее.

Предлагаемый алгоритм подбора рекомендаций разделяется на несколько последовательных этапов:

1. Обучение (получение векторов термов и списка категорий).
2. Построение векторной модели обучающей выборки.
3. Получение векторных моделей обучающей и контрольной выборок и построение категориальных векторов.

2.9.1 Обучение (получение векторов термов и списка категорий)

Предположим, что проанализировано m тематических текстов, принадлежащих предметной области. Данная обучающая выборка подбирается экспертом в предметной области. Входной характеристикой данного этапа является выбранная точность, которая зависит от предметной области, качества и объема обучающей выборки.

Под выбранной точностью понимаем коэффициент k, описанный в разделе 2.4 данной главы. Число k устанавливается путем отбрасывания «незначимых» компонент матрицы корреспонденций термов. Незначимыми являются компоненты, сингулярные значения которых ниже порогового значения, и от того насколько хорошо оно будет подобрано напрямую зависит количество термов, а также точность результатов анализа.

Так же экспертом в предметной области на данном этапе формируется конечный список категорий.

Под категорией в данном случае понимается некое именованное множество текстов, объединенных по определенному признаку.

Количество категорий и объединяющие признаки зависят от поставленной задачи и выбранной предметной области. Например, для предметной обла-

сти «подбор подходящих вакансий» количество категорий составило 17, а для предметной области «подбор автозапчастей для автомобилей китайского производства» — 24.

После того, как список категорий сформирован, эксперт вручную относит каждый из текстов (из обучающей выборки) к одной или нескольким категориям. На рисунке 2.5 изображена частотная диаграмма принадлежности текстов категориям для тестовой выборки, принадлежащей предметной области «подбор персональных рекомендаций в сфере поиска работы».

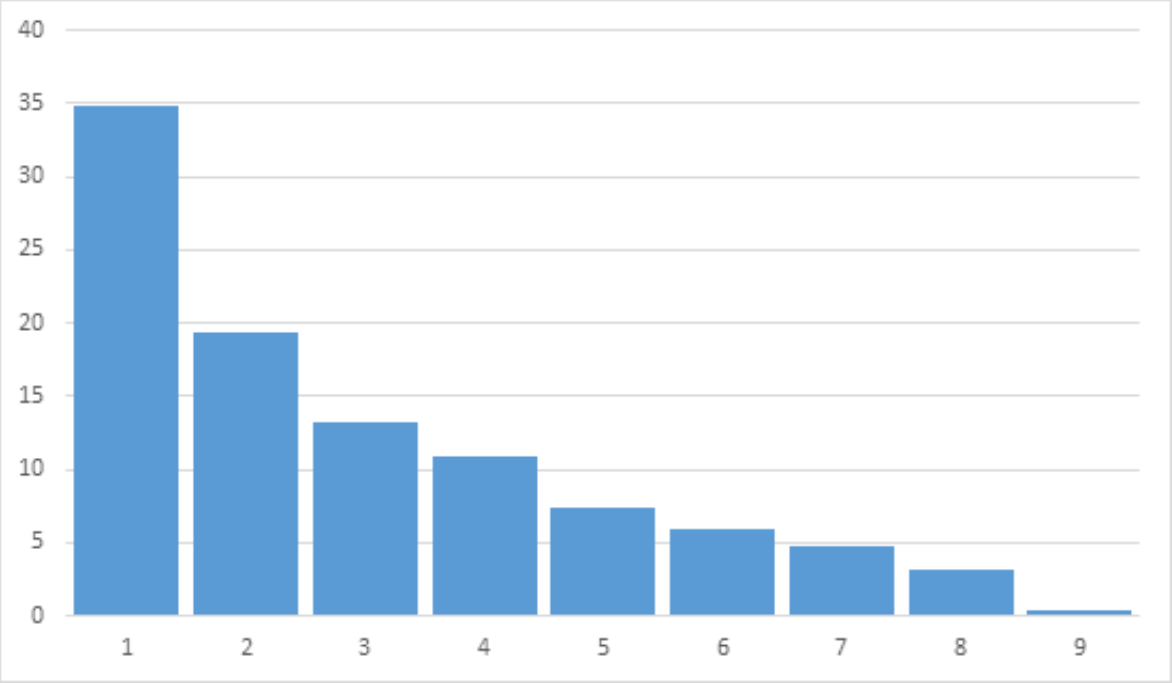

Рисунок 2.5 — Распределение текстов по категориями

Подписям соответствуют следующие категории:

1. Здоровье и красота.
2. Строительство и ландшафт.
3. Языки и лингвистика.
4. Маркетинг, реклама, копирайтинг.
5. Программирование.
6. Ремонт машин и оборудования.
7. Дизайн, рисование, фото, видео.
8. Категории менее (6 штук) 1% каждая.
9. Категории менее (4 штуки) 0.1% каждая.

На рисунке 2.5 видно, что в некоторые категории попадает очень мало текстов, и пользователь, которому необходимо будет сформировать персональные

рекомендации по этим категориям (при использовании стандартных алгоритмов) получит пустую выдачу либо выдачу малого объема.

Итак, на данном этапе мы получили множество из n термов (в ходе построения семантического ядра), множество категорий в количестве c (в ходе ручного анализа предметной области специалистом) и множества текстов для каждой из категорий.

2.9.2 Построение векторной модели обучающей выборки

На данном этапе вычисляются вектора категорий, которые будут впоследствии необходимы для определения коэффициента принадлежности анализируемого текста указанной категории.

Для получения вектора категории необходимо проанализировать все тексты, которые в нее входят, и построить для каждого из них векторную модель. Векторная модель текста представляет собой вектор размерности n (количество термов), каждая компонента которого соответствует количеству вхождений i-того терма в анализируемый текст. Множество термов было получено на предыдущем этапе. Векторная модель некоторого документа d представляет собой следующий вектор.

$$\vec{d} = \{tf(t_1), tf(t_2), ..., tf(t_n)\} \tag{2.49}$$

где $tf(t_i)$ - частота встречаемости (*term frequency*) терма t_i в документе d. На самом деле функция $tf(t_i)$ - зависит от двух аргументов: $tf(t_i, d)$, однако второй аргумент в данном случае является фиксированным, поэтому для удобства его можно опустить.

Последовательно рассматриваем термы. Пусть w_j - j-тое слово анализируемого текста d, t - текущий терм. Уточним понятие вхождения терма в тест. Считается, что T входит в d, если:

$$\exists w_j : Hamming(w_j, t) < h \cup stem(w_i) = stem(t) \tag{2.50}$$

где $stem$ — операция стемминга, $Hamming$ — операция получения расстояния Хэмминга, h — расстояние, при котором можно считать слова идентичными. На практике наиболее эффективным является принятие $h = 3$, но данное значение может значительно отличаться для различных предметных областей. Через $\cup$ обозначено логическое «или», то есть должно быть выполнено хотя бы одно из условий. Пояснения требует тот факт, что функция получения расстояния Хэмминга применяется к исходным словам, а не к словам, подвергнутым стеммингу. Именно так делается потому, что, если применить операцию стемминга к слову с орфографической ошибкой, то его основа будет определена неправильно.

В качестве альтернативы расстоянию Хэмминга можно так же рассмотреть следующие алгоритмы определения расстояний: расстояние Левенштейна, расстояние Димрау-Левенштейна и расстояние Джаро-Винклера [76]. В таблице 1 указана их сравнительная характеристика [67].

Таблица 1 — Сравнительная характеристика алгоритмов определения расстояний

Метрика	Расстояние Хэмминга	Расстояние Левенштейна	Расстояние Димерау-Левенштейна	Расстояние Джаро-Винклера
Временная сложность (где m и n — длины сравниваемых строк)	O(log n)	O(mn)	O(mn)	O(mn)
Потребление памяти	O(min(m, n))	O(mn)	O(mn)	O(mn)
Полнота результатов (количество ошибок)	98%	65%	80%	90%

Большинство алгоритмов нечеткого поиска с индексацией не являются истинно сублинейными (т.е. имеющими асимптотическое время работы $O(\log n)$ или ниже), и их скорость работы обычно напрямую зависит от длинны строки. Тем не менее, множественные улучшения и доработки позволяют добиться достаточного малого времени работы даже при весьма больших объемах словарей [67].

Метод вычисления расстояния Хэмминга в классической своей реализации не является сублинейным, однако с помощью простых правил можно добиться его ускорения [67].

После того, как вектора всех текстов рассматриваемой категории построены, проводим их нормализацию по алгоритмам, рассмотренным в разделе 2.7.2.

После проведения нормализации вектора количества вхождений термов, можно говорить, что векторная модель указанного текста построена. Процедура построения векторной модели текста одинакова для всех анализируемых единиц и будет использоваться и на последующих этапах.

Когда векторные модели текстов, входящих в определенную категорию построены, можно вычислить *векторную модель данной категории*, вычислив средний вектор между векторами ее текстов. Средний вектор - вектор, состоящий из средних арифметических соответствующих компонент векторов текстов. Обозначим данный вектор $\vec{d}_{\text{ср}}$. Указанные действия необходимо повторить для всех категорий.

Предположим, что необходимо рассчитать средний вектор первой категории $\vec{d}_{\text{ср.1}}$ в которую входит p текстов, тогда:

$$\vec{d}_{\text{ср.}i} = \left\{ \frac{\sum\limits_{i=1}^{p} tf_{\text{норм}}(t_1, d_i)}{p}, \frac{\sum\limits_{i=1}^{p} tf_{\text{норм}}(t_2, d_i)}{p}, ..., \frac{\sum\limits_{i=1}^{p} tf_{\text{норм}}(t_p, d_i)}{p} \right\} \tag{2.51}$$

где d_i — i-тый документ из первой категории.

На данном этапе мы получили векторные модели категорий, обозначим их количество буквой c. После получения данных векторных моделей, все данные, которые участвовали в их вычислении можно удалить — они больше не нужны.

2.9.3 Получение векторных моделей анализируемых текстов

На этапе 2 была рассмотрена процедура получения векторной модели текста. Используем ее для анализируемых единиц: анализируемых текстов и текстов пользователей. Категориальные вектора состоят из коэффициентов при-

надлежности указанного текста определенной категории, номер этой категории соответствует компоненте категориального вектора. Обозначим множество данный вектор Z, размерность данного вектора равна c (количество категорий).

Предположим, что нам необходимо вычислить категориальный вектор для выбранного документа d. Обозначим его $\vec{Z}$, тогда первая его компонента будет иметь вид:

$$z_1 = \frac{\left(\vec{d}_{\text{ср.1}}, \vec{d}\right)}{\left|\vec{d}_{\text{ср.1}}\right| \left|\vec{d}\right|} \tag{2.52}$$

где $\vec{d}_{\text{ср.1}}$ — средний вектор первой категории, $\vec{d}$ — вектор документа из коллекции, $\left|\vec{d}\right|$.

Будем так же называть z_1 коэффициентом принадлежности некоторого документа первой категории.

Отметим, что коэффициенты принадлежности не являются координатами разложения векторов из исходного векторного пространства R^n по базису $\vec{d}_{\text{ср.1}}, \vec{d}_{\text{ср.2}}, ..., \vec{d}_{\text{ср.}c}$, так как категориальные вектора, вообще говоря, не ортогональны.

Таким же образом поступаем для всех остальных компонент. Необходимо построить категориальные вектора для всех анализируемых единиц, включая анализируемые тексты и тексты пользователей.

2.9.4 Свойства категориальных векторов

Утверждение 2. *Пусть вектора категорий* $\left\{\vec{d}_{ср.1}, \vec{d}_{ср.2}, ..., \vec{d}_{ср.c}\right\}$, $\vec{d}_{ср.j} \in R^n$ *линейно независимы, тогда категориальные вектора* Z_1 *и* Z_2 *документов* d_1 *и* d_2 *совпадают тогда и только тогда, когда совпадают проекции векторов* $\vec{d}_1$ *и* $\vec{d}_2$ *этих документов на подпространство, натянутое на категориальные вектора.*

В силу линейной независимости векторов категорий, коэффициенты принадлежности $\{z_{i1}, z_{i2}, ..., z_{ic}\}$ взаимно однозначно определяют координаты разложения произвольного вектора $\vec{d}_i$ по базису $\vec{d}_{\text{ср.1}}, \vec{d}_{\text{ср.2}}, ..., \vec{d}_{\text{ср.}c}$.

Обозначим через C матрицу размера $n \times k$, столбцами которой являются вектора категорий $\vec{d}_{\text{ср.1}}, \vec{d}_{\text{ср.2}}, ..., \vec{d}_{\text{ср.}c}$ в пространстве R^n. Так как вектора линейно независимы, то $rank(C) = c$.

Матрица C^T задает оператор перехода из исходного векторного пространства R^n, где n – число термов, в новое векторное пространство R^c , где с - число категорий. Отметим, что количество категорий значительно меньше числа термов.

Таким образом, категориальный вектор документа d находится по формуле:

$$\vec{Z}_d = C^T \vec{d} \tag{2.53}$$

где $\vec{d}$ - векторная модель документа, найденная по формуле (2.2).

2.10 Выбор рекомендаций

Непосредственно выбор рекомендаций происходит следующим образом.

1. Рассчитывается категориальный вектор пользователя, для которого происходит подбор рекомендаций. Обозначим его $\vec{Z}_{\text{польз}}$
2. Данный вектор скалярно перемножается со всеми категориальными векторами документов (находим γ_i по формуле 2.54).
3. Полученные значения сортируются по убыванию.
4. Из отсортированной выборки извлекается q первых элементов.

Пусть c - количество категорий, $\vec{Z}_{\text{польз}}$ - категориальные предпочтения пользователя, $\vec{Z}_{\text{объект}i}$- категориальный вектор i-того объекта из базы данных предложений, тогда:

$$\gamma_i = \frac{\left(\vec{Z}_{\text{польз}}, \vec{Z}_{\text{объект}i}\right)}{\left|\vec{Z}_{\text{польз}}\right| \left|\vec{Z}_{\text{объект}i}\right|} \tag{2.54}$$

Будем называть γ_i коэффициентом близости между категориальным вектором пользователя $\vec{Z}_{\text{польз}}$ и категориальным вектором i-того документа $\vec{Z}_i$, $\vec{Z}_j \in R^c$.

Отметим, что коэффициенты близости γ_i обладают в силу формулы (2.54) следующими свойствами:

1. $-1 \leqslant \gamma_i \leqslant 1$
2. при $\gamma_i = 1$ вектора $\vec{Z}_i$, $\vec{Z}_j$ сонаправлены, т.е. $\vec{Z}_j = \lambda \vec{Z}_i$, где $\lambda > 0$.

2.11 Свойства коэффициентов близости

Утверждение 3. *Если проекции векторных моделей двух документов* $\vec{d_i} \in R^n$ *и* $\vec{d_j} \in R^n$ *на подпространство категориальных векторов сонаправлены, то эти документы имеют максимальное значение коэффициента близости* $\gamma_{ij} = 1$.

Мера близости, определяемая соотношением (2.54), называется косинусной мерой и минимизирует расстояние между векторами единичной длины, то есть верно следующее, легко проверяемое, утверждение.

Утверждение 4. *Пусть вектора* $\left\{\vec{Z}_i\right\}$, $i = \bar{1,n}$ *имеют единичную длину, тогда оптимальное решение задачи*

$$\min_{i=\bar{1,n}} \|\vec{Z}_i - \vec{Z}_0\|^2 \tag{2.55}$$

достигается на векторе $\vec{Z}_j$, *удовлетворяющему условию максимума*

$$\left(\vec{Z}_j, \vec{Z}_0\right) = \max_i \left(\vec{Z}_i, \vec{Z}_0\right) \tag{2.56}$$

Таким образом, среди векторов единичной длины минимум расстояния достигается на том же элементе, что и максимум скалярного произведения. Отметим, что для векторов единичной длины коэффициент близости, рассчитываемый по формуле (2.54), совпадает со скалярным произведением.

Так как мы используем векторную модель документов, которая не учитывает связи, основанные на порядке слов, а только отражает частоты встречаемости слов (термов) в документах, то можно считать, что два документа описываются одной и той же случайной величиной, если частоты совпадают. При анализе семантической близости документов с использованием векторной

модели документов, отражающей частоту встречаемости термов, важным является только соотношение между встречаемостью различных терминов, т.е. направление вектора документа. Поэтому в качестве меры близости используется не расстояние между категориальными векторами, а косинусная мера, отражающая близость направлений векторов.

Утверждение 5. *Максимум коэффициентов близости γ_i между вектором $\vec{Z}_0$ и набором ненулевых векторов $\left\{\vec{Z}_i\right\}$, $i = \bar{1,n}$,*

$$\max_i \frac{\left(\vec{Z}_i, \vec{Z}_0\right)}{\|\vec{Z}_i\|\|\vec{Z}_0\|} \tag{2.57}$$

достигается на векторе $\vec{Z}_j$, который является решением задачи нахождения минимума

$$\min_{i=\bar{1,n}} \left(\min_\lambda \|\vec{Z}_0 - \lambda\vec{Z}_i\|^2\right) \tag{2.58}$$

Доказательство. Найдем минимум в задаче (2.58) по параметру λ. Запишем

$$\min_\lambda \|\vec{Z}_0 - \lambda\vec{Z}_i\|^2 = \min_\lambda \left(\vec{Z}_0^2 - 2\lambda\vec{Z}^T\vec{Z}_0 + \lambda^2\vec{Z}^2\right) \tag{2.59}$$

Минимум достигается при $\lambda = \left(\vec{Z}^2\right)^{-1}\left(\vec{Z}^T\vec{Z}_0\right)$ и равен

$$\min_\lambda \|\vec{Z}_0 - \lambda\vec{Z}_i\|^2 = \vec{Z}_0^2 - \left(\frac{\vec{Z}^T\vec{Z}_0}{\vec{Z}^2}\right)^2 \tag{2.60}$$

Следовательно, вектор $\vec{Z}_j$, дающий минимальное значение в задаче (2.58) одновременно доставляет максимум коэффициента близости (2.57). □

Опыт анализа баз данных вакансий показывает, что близость к 1 коэффициента γ_{ij} отражает соответствие j-го элемента базы i-му пользовательскому запросу [12, 14].

Таким образом, метод позволяет получить выборку отсортированную по степени «полезности» конечному пользователю. Такой способ хорош в первую очередь тем, что даже в случае, когда данные распределены между категориями неравномерно, пользователь получит непустой результат.

Кроме того, использование расстояния Хэмминга при расчете вхождений термов в тексты позволяет учитывать и исправлять возможные орфографиче-

ские ошибки, которые неизбежно возникают, когда данные формируются обычными пользователями.

2.12 Выводы по второй главе

Предложен эффективный метод, позволяющий из любой непустой выборки сформировать персональные рекомендации. Предложен способ формирования семантического пространства на основе ортогонального разложения матрицы корреспонденций термов, которая подвергается ортогональному разложению. Проведено сравнение ортогонального разложения МКТ и сингулярного разложения ТДМ. Доказано, что при использовании предлагаемого метода термы, содержащиеся в коротких документах, сохраняются в семантическом ядре. В качестве недостатков метода можно выделить:

- сложность поддержания категориальных векторов в актуальном состоянии;
- большой объем промежуточных вычислений.

Эффективность метода состоит в том, что он:

- требует малый объем вычисленных данных, которые необходимо хранить;
- имеет высокую скорость получения результата;
- метод оптимизирован для реляционных хранилищ.

Так же в главе была приведена математическая формализация метода категориальных векторов, который представлен в форме линейного оператора для векторного представления документов. Исследованы свойства косинусной меры близости векторов, показано, что эта мера адекватно отражает семантическую близость между пользовательскими запросами и выбираемыми данными базы, что подтверждается эффективной работой алгоритма подбора вакансий. Основные результаты исследований данной главы опубликованы в статьях [8—16, 62].

Глава 3. Векторная модель представления знаний использующая семантическую близость термов

Во второй главе было представлено описание алгоритма, позволяющего для любого запроса получить непустой результат. Разработанный метод работает даже в случае, когда тексты обучающей выборки распределены по категориям неравномерно. Однако, экспериментальное использование алгоритма показало, что он выдает некачественные (нерелевантные запросу пользователя) выдачи в некоторых предметных областях. Анализ некачественных выдач показал, что ошибки происходят из-за особенностей естественного языка, которые не были учтены. Кроме прочего, неучет этих особенностей, а именно синонимии и полисемии, увеличивает размерность семантического пространства, от которой зависит быстродействие конечного программного продукта, разработанного на основе алгоритма. В какой-то степени проблема синонимии и полисемии решается с помощью метода поиска семантического ядра, описанного в пункте 2.7 главы 2, однако его применение обусловлено значительными затратами процессорного времени, что на практике накладывает определенные ограничения. Кроме того, результаты работы разработанного алгоритма сложно воспринимаются экспертом предметной области, который подготавливает обучающую выборку, что в свою очередь так же сказывается на качестве выдачи алгоритма. В этой главе предлагается другой способ решения проблемы полисемии и синонимии, а именно «перевзвешивание» термов с помощью вычисления их семантической близости друг с другом.

Для вычисления семантической близости термов будем использовать адаптацию расширенного алгоритма Леска [85]. Метод расчета семантической близости состоит в том, что для каждого значения рассматриваемого слова подсчитывается число слов упомянутых как в словарном определении данного значения (предполагается, что словарное определение содержит описание нескольких значений слова), так и в ближайшем контексте рассматриваемого слова. В качестве наиболее вероятного значения слова выбирается то, для которого такое пересечение оказалось больше. Векторная модель с учетом семантической близости термов решает проблему неоднозначности синонимов.

3.1 Расширенный метод Леска

Оригинальный алгоритм Леска [56] предусматривает использование только словарных значений анализируемого слова и же его ближайшего контекста. Это является существенным ограничением, поскольку словарные определения как правило являются очень короткими, и влияют на рассчитанную по алгоритму Леска близость слов только косвенно [56]. Если взять для примера одну из самых крупных баз знаний WordNet, то средняя длина определения слова в словаре равна всего семи словам [28].

Расширенный метод Леска расширяет определения сравниваемых слов и включает определения слов, которые связаны со сравниваемыми словами. Будем считать, что два термина похожи, если их определения содержат похожие слова. В самом простом случае расширенный метод Леска можно выразить следующей формулой:

$$\begin{aligned} similarity_{extLesk}(t_1, t_2) = overlap(gloss(t_1), gloss(t_2)) + \\ overlap(gloss(hyppo(t_1)), gloss(hyypp(t_2)) + \\ overlap(gloss(hyppo(t_1)), gloss(t_2)) + \\ overlap(gloss(t_1), gloss(hyppo(t_2))) \end{aligned} \tag{3.1}$$

где $overlap(t_1, t_2)$ — количество совпадений между термами t_1 и t_2, $gloss(t)$ — определение терма t, $hyppo(t)$ — гипероним слова, например для слова «красный» гиперонимом является слово «цвет», t_1 и t_2 — термины.

В классической версии алгоритма Леска гиперонимы не используются, однако их использование значительно улучшает качество выдачи алгоритма. В работах [105] и [78] используются синсеты и гиперонимы из английской версии WordNET, но синсет выбирается согласно наивному методу, после чего выбирается соответствующий синсету гипероним. Эксперименты проводились на нескольких независимых общедоступных выборках. Авторы данных работ сделали вывод, что использование гиперонимов привело к улучшению качества работы классификатора на всех обучающих множествах. Кроме того, выяснилось, что применение гиперонимов почти всегда улучшает качество работы классификатора по сравнению с применением только синсетов [78, 105].

Гиперонимы слов для первоначальной оценки можно взять, например, в российской версии WordNet, разработка которой осуществляется в Петербургском университете путей сообщения [49].

3.2 Учет семантической близости при вычислении веса терма

Чтобы учесть семантическую связь между терминами, вес термина в документе будем рассчитывать несколько иначе, чем в классической векторной модели представления знаний. Термином (термом) будем называть, как и в главе 2, слово, обработанное с помощью стеммера Портера [115] и не содержащееся в списке стоп-слов.

Настройка весов термов производится с помощью вычисления семантической близости связанных термов. Считается, что термины связаны, если они находятся в одном документе в непосредственной близости друг к другу. Новый вес терма рассчитывается следующим образом:

$$\tilde{w}_{dt_1} = w_{dt_1} + \sum_{t_1 \neq t_2} similarity(t_1, t_2) \tag{3.2}$$

где w_{dt_1} — вес терма в документе d до настройки, рассчитанный по формуле (2.3), $similarity$ — семантическая близость термов t_1 и t_2, рассчитываемая с помощью адаптации расширенного метода Леска. Суммирование происходит по всем термам документа d.

Этот шаг переопределяет классический вес терма в векторе документа и учитывает семантические связи между каждой парой термов. Для вычисления исходных весов термов в документе будем использовать меру $tf.idf$. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции [102]. Мера $tf.idf$ терма t в документе d вычисляется следующим образом:

$$tf.idf(d,t) = \ln\left(tf(d,t) + 1\right) \ln \frac{|D|}{df(t)} \tag{3.3}$$

где $df(t)$ — документная частота терма, показывающая количество документов, в которых встречается терм, $tf(t,d)$ — число появлений терма t в документе d, нормализованное общим количеством термов в документе d, $|D|$ — общее количество документов.

Предлагается использовать именно эту меру, поскольку она приписывает большие веса термам, которые редко встречаются в обучающей выборке, но часто в некоторых конкретных документах. $tf.idf$ дает примерно на 14% более точный результат, чем стандартная мера tf, основанная на частоте терма в документе [102].

В [109] показано, что каждая из категорий обычно представлена множеством «базовых» слов, а остальные слова являются слишком общими, чтобы определять категорию. Предлагается использовать общие слова для повышения значимости (весов) «базовых» слов. Данное решение в значительной степени улучшило результаты подбора персональных рекомендаций, поскольку при использовании этого подхода определение принадлежности документа некоторой категории происходит более точно.

3.3 Анализ возможности применения тезаурусов и словарей

В разделе 3.2 был предложен способ взвешивания термов векторной модели представления документа с помощью вычисления семантической близости, которая рассчитывается с помощью адаптированного метода Леска. Метод показал достаточно высокую эффективность, однако он так же имеет ряд недостатков, а именно:

- необходимость хранения дополнительных словарей (словарь гиперонимов);
- необходимость хранения определений термов;
- проблемы, связанные со сложной структурой естественного языка.

3.3.1 Обзор существущих словарей русского языка

В таблицах 2 и 3 представлены наиболее известные словари русского языка и некоторые сведения о них. Как видно, большинство словарей были составлены еще в прошлом веке, соответственно современных слов в них почти нет.

Таблица 2 — Словари русского языка (тыс. слов)

№	Словарь	Кол-во слов	Кол-во уник. слов
1	Толковый словарь русского языка под ред. Д.Н. Ушакова, 1935	88.8	87.1
2	Толковый словарь русского языка С.И. Ожегова, 1949	41.2	40.3
3	Малый академический словарь под ред. А.П. Евгеньевой, 1957	83.5	81.6
4	Большой толковый словарь русского языка под ред. С.А. Кузнецова, 1998	76.3	73.2
5	Новый словарь русского языка. Толково-словообразовательный под ред. Т.Ф. Ефремовой, 2000	135.2	123.7
6	Грамматический словарь русского языка под ред. А.А. Зализняка, 1977	93.4	93.4
7	Русский орфографический словарь под ред. В.В. Лопатина, 2001	132.4	130.1
8	Словообразовательно-морфемный словарь русского языка под ред. А.Н. Тихонова, 1985	112.3	111.0

Таблица 3 — Словари синонимов русского языка (тыс. слов)

Словарь	Кол-во слов	Кол-во уник. слов
Словарь синонимов Н. Абрамова, 1999	5.4	5.4
Словарь синонимов русского языка под ред. А.П. Евгеньевой, 1975	5.5	4.6
Словарь синонимов русского языка под ред. Л.Г. Бабенко, 1999	5.0	4.9

Поддержание используемых словарей в актуальном состоянии представляет собой достаточно сложную и затратную, с точки зрения вычислительных

ресурсов, задачу. Кроме того, все словари поставляются в необработанном виде, следовательно необходимо провести их программную обработку, что в свою очередь потребует значительных вычислительных ресурсов. В современных условиях эта задача не реализуема полностью, поскольку невозможно найти источник данных, содержащий, например, гиперонимы всех слов, особенно сленговых.

В работе [82] исследуется количество совпадающих слов в известных словарях. В результате получается всего лишь порядка 165 тыс. уникальных слов. Такое малое результирующее количество слов объясняется традиционными правилами и преемственностью проектов по составлению словарей. Кроме того, можно заметить, что словари синонимов заметно меньше по объему традиционных словарей, что приводит к сложности применения методов устранения лексической многозначности. Словари синонимов в лучшем случае покрывют 60% всех синонимных связей между словами [82].

На рисунке 3.1 представлена визуализация доли словарей в множестве уникальных слов. Словари на диаграмме обозначены Cn, где n — номер словаря в таблице 2. При рассчетах было опущено около 108 тыс. слов, которые присутствуют в двух и более словарях одновременно.

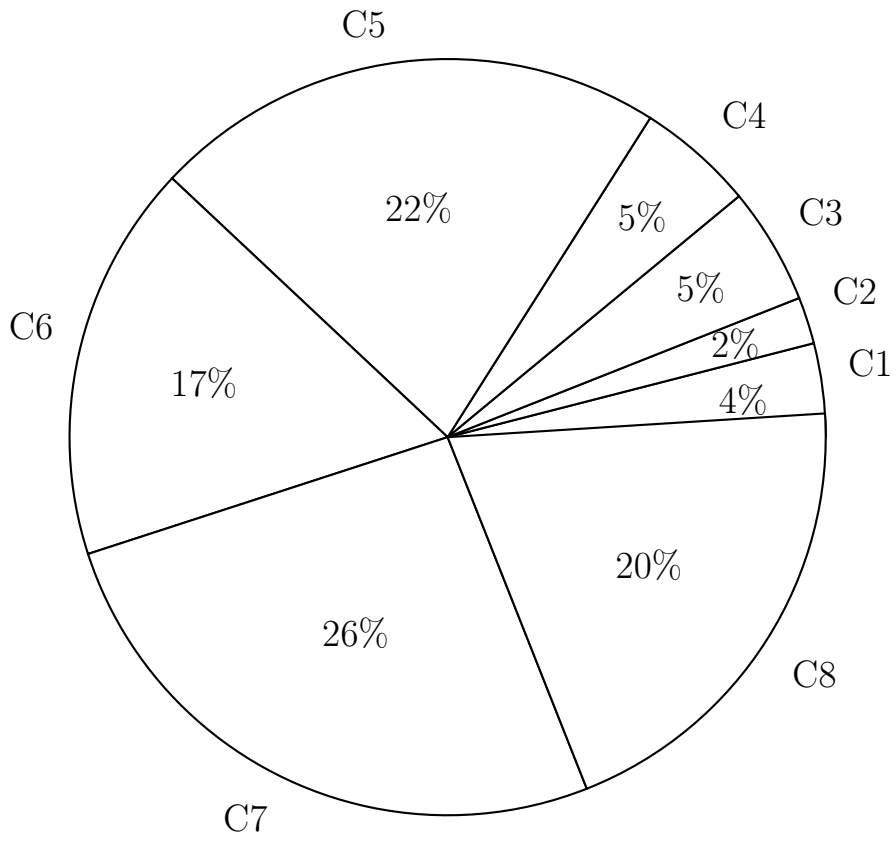

Рисунок 3.1 — Доля словарей в множестве уникальных слов

Все описанные факты говорят об ограниченности применения в алгоритмах интеллектуального поиска существующих словарей русского языка в качестве корпуса. Во-первых, словари содержат по большей части слова общеупотребительной лексики, а охват в них слов и терминов 21 века (в том числе сленговых слов) оставляет желать лучшего. Во-вторых, словарей синонимов недостаточно для построения семантических связей между словами. Как уже было отмечено, они покрывают в лучшем случае 60% всех возможных связей. В-третьих, подготовка к использованию и поддержание словарей в актуальном состоянии представляет собой отдельную и весьма сложную задачу. Таким образом, актуальной задачей является разработка алгоритмов анализа текстовых данных без использования словарей.

3.3.2 Анализ русскоязычных тезаурусов

Возможность применения русскоязычных тезаурусов так же ограничена, поскольку известные реализации тезаурусов для русского языка содержат недостаточное количество слов, что является следствием частого построения тезаурусов на основе обработки словарей. На сегодняшний день наиболее известными тезаурусными базами данных для русского языка являются RussNet [44], Russian WordNET [101], RuThes [35], RuThes-lite [35], YARN [117]. Сведения о них сведены в таблицу 4.

Среди всех, перечисленных тезаурусов, выделяется проект YARN, который наполняется, помимо автоматического анализа словарей, еще и с помощью краудсорсинга, т.е. силами его пользователей [64]. Этот факт является одновременно и достоинством и недостатком проекта: люди добавляют большое количество новых слов, но зачастую дублируют уже существующие.

На рисунке 3.2 изображена гистограмма сравнения русскоязычных тезаурусов по количеству слов, которые в них содержатся. Наибольшее количество слов содержится в семантической сети RuThes. Однако поскольку она разрабатывалась, как поисковой инструмент Университетской информационной системы Россия [51], в которой содержатся документы в основном жанра деловой прозы — нормативные акты, материалы общественно-политических СМИ и т.д.,

Таблица 4 — Сравнение тезаурусов

	Кол-во концептов	Кол-во связей	Кол-во слов
RussNet	5.5 тыс.	8 тыс.	15 тыс.
Russian WordNET	157 тыс.	-	124 тыс.
RuThes	55 тыс.	210 тыс.	158 тыс.
RuThes-lite	26 тыс.	108 тыс.	115 тыс.
YARN	44 тыс.	0	49 тыс.

применение ее для других предметных областей затруднительно. Кроме того, все перечисленные семантические базы данных, кроме YARN запрещены для коммерческого использования.

Если взять для примера две крупнейшие базы данных RuThes и Russian WordNET (на рисунке 3.3 изображено их пересечение) и попытаться определить какое количество общих слов они содержат, то получится 108467 слов. Несмотря на то, что количество уникальных слов существенно и при гипотетическом объединении этих двух баз получится достаточно большой тезаурус, этот факт не решает проблемы отсутствия слов современной лексики.

Сложность применения семантических баз данных по типу WordNET в реальных задачах обоснована, как минимум сравнительно малым количеством слов, а так же невозможностью применения в коммерческих системах в большинстве случаев. В настоящее время в российских научных работах, посвященных интеллектуальному анализу текстов, наблюдается переходный период от традиционных методов анализа к крупномасштабным проектам по созданию корпусов. В то же время, учитывая специфику этих проектов, их часто критикуют за качество. Все это делает использование таких баз данных на текущем этапе их развития в реальных задачах маловозможным.

3.3.3 Анализ применимости баз данных интернета

Еще одним подходом, основанном на применении различного рода словарей и баз данных является использование в качестве источника данных «Викисловаря». В работе [89] рассматривается сравнение данного подхода с использованием семантической базы данных WordNET в том числе и для русского языка.

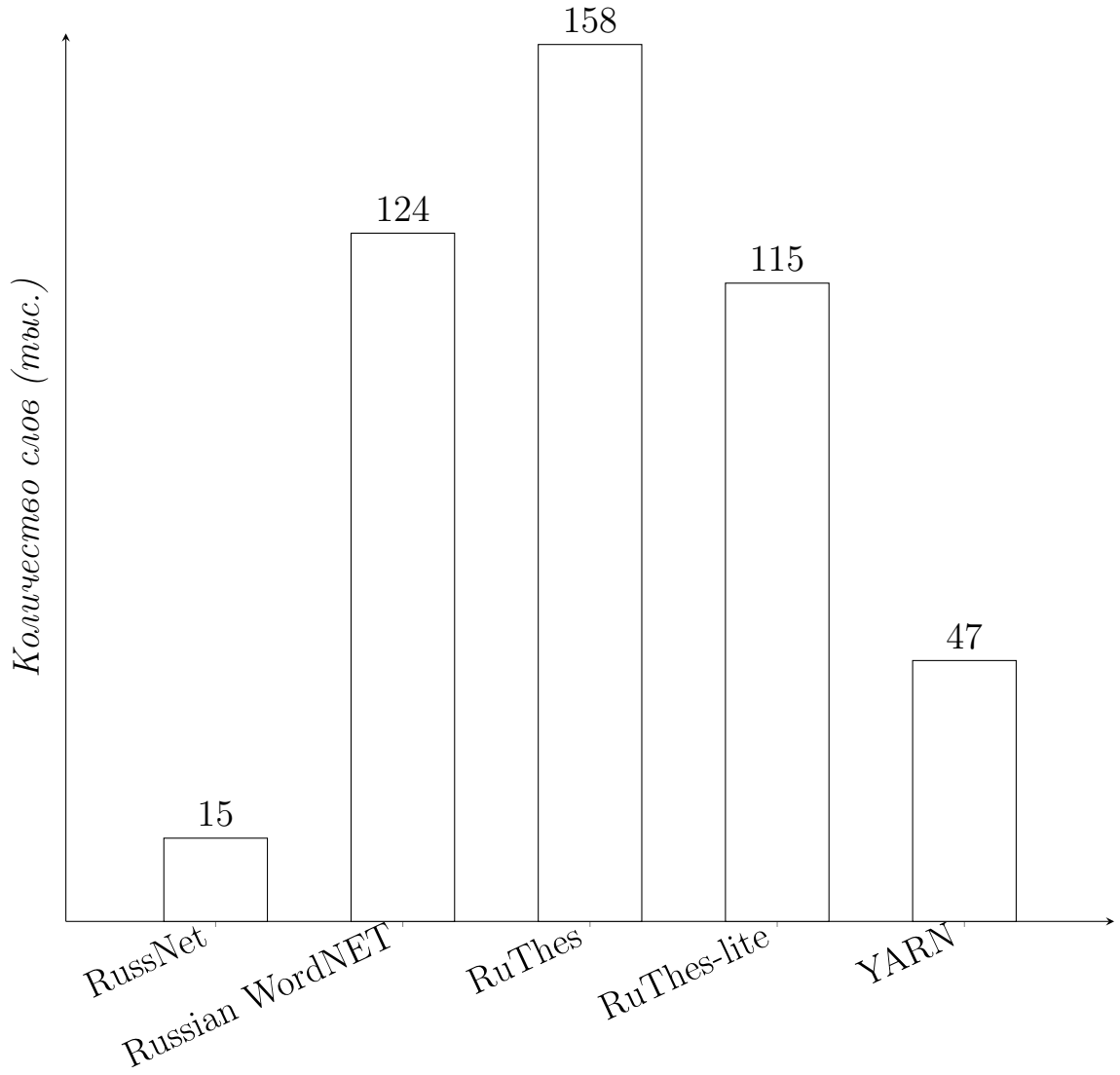

Рисунок 3.2 — Сравнение количеств слов в русскоязычных тезаурусах

В качестве примера для сравнения используется Russian WordNET. По данным 2016 года в «Викисловаре» содержится порядка 290 тыс. русских слов. Огромным достоинством «Викисловаря» по сравнению с традиционными словарями является наличие у него версии, удобной для компьютерной обработки.

По данным статьи [89] на 2010 год количество совпадающих термов в «Викисловаре» и Russian WordNET было равно порядка 18 тыс. На сегодняшний день это количество значительно выросло — проиллюстрировано рисунком 3.4.

В «Викисловаре» гораздо больше слов современной лексики, чем в других семантических базах данных, однако они не разнесены по предметным областям, что в значительной степени затрудняет его обработку. Для преемлемого времени ответа интеллектуальной системы на запрос пользователя достаточно

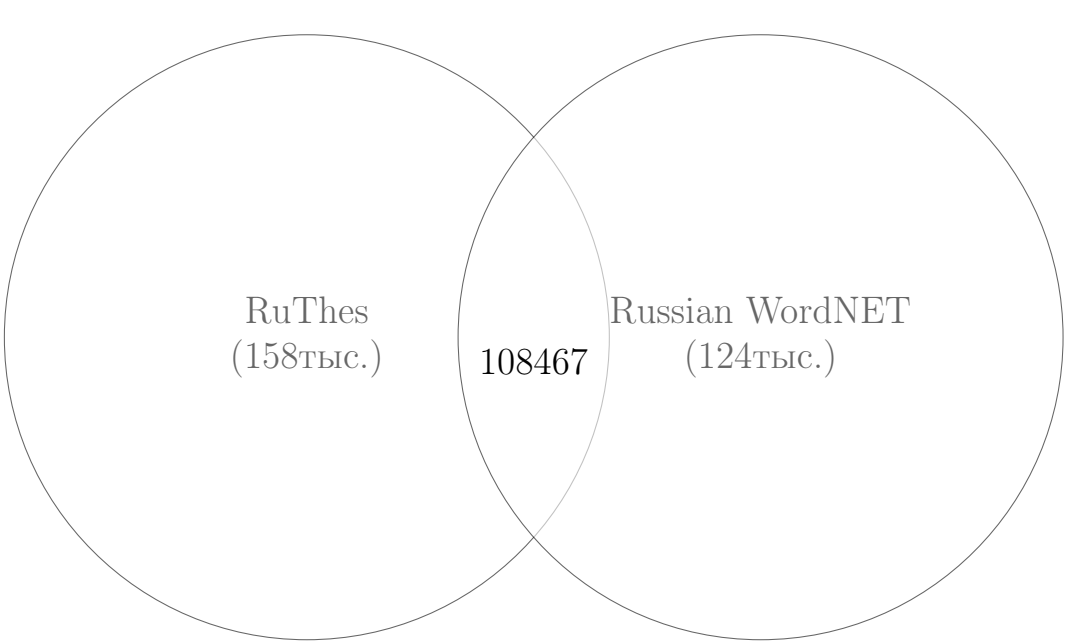

Рисунок 3.3 — Пересечение RuThes и Russian WordNET

сложно организовать хранение словаря такого объема. Кроме того, получившийся индекс будет иметь довольно большой размер. Так же, «Викисловарь» не лишен «теннисной» проблемы, характерной для WordNET, т.е. термы одной предметной области часто располагаются в графе связей достаточно далеко друг от друга, что в свою очередь говорит об их слабой семантической взаимосвязи.

3.4 Анализ проблемы синонимии и полисемии

Рассмотрим подход основанный на использовании внешних источников, расположенных в сети интернет (web-источники). Существует два фактора, которые негативно влияют на вычисление семантической близости с использованием данного подхода. Этими факторами являются синонимия и полисемия.

Синонимия — случай, когда несколько слов имеют схожий смысл, например, «автомобиль» и «машина».

Полисемия — случай, когда одно слово имеет несколько смыслов, например, «диск».

Проблема синонимии состоит в том, что если документ уже содержит один из синонимов, то вероятность, что он так же содержит другой синоним мала.

Рисунок 3.4 — Пересечение Викисловаря и Russian WordNET

Это приводит к тому, что связность между синонимами, вычисленная только с помощью подхода, основанного на использовании веб-контента, получается меньше, чем она есть на самом деле. Например, поиск в Google слова «происшествие» даст примерно 440 000 результатов, а поиск его синонима слова «инцидент» — 592 000 результатов, и примерно 34 000 результатов, где они встречаются вместе. NSG можно вычислить по следующей формуле:

$$NSG(t_1, t_2) = \frac{max\{lgf(t_1), lgf(t_w)\} - lg(f(t_1, t_2))}{lgN - min\{lgf(t_1), lgf(t_2)\}} \tag{3.4}$$

где N — общее количество веб-страниц, обрабатываемых поисковой системой Google, $f(t_1)$ и $f(t_2)$ — количество страниц, на которых термы t_1 и t_2 находятся по отдельности, $f(t_1, t_2)$ — количество страниц, на которых термы t_1 и t_2 находятся вместе.

Вычислим *NGD (Normalized Google Distance)* между этими связанными словами [66]:

$$NGD(\textit{происшествие}, \textit{инцидент}) \approx 0.8365$$

Если принять результат за достоверный, то можно сделать вывод, что связи между словами «происшествие» и «инцидент» фактически нет. NGD семантически идентичных слов равна 0, а семантически не связанных — 1.

Полисемия дает обратный эффект, состоящий в том, что в документе одно и то же слово может употребляться в нескольких смыслах. Например, слово

«диск» может употребляться в различных смыслах: «колесный диск», «компьютерный диск» и т. п. Поиск в Google слова «диск» дает примерно 32 720 000 результатов, однако, допустим, нас интересуют только «колесные диски». По данному запросу мы получим только 238 000 результатов. Наблюдения производились в ноябре 2014 года.

Таким образом, вычисление семантической близости термов, основанное на результатах выдачи глобальной поисковой системы, зачастую не дает ожидаемого результата. Конечно, это утверждение верно, если помимо глобальной поисковой системы никаких других источников данных о связи между словами не используется. Однако извлечь только нужные данные из этих источников и интерпретировать их для вычисления семантической близости достаточно сложно.

Теперь рассмотрим подход вычисления семантической близости основанный на семантической сети WordNet. Вычислим связь между близкими словами «университет» и «экзамен», используя «синсеты» WordNet и метод Леска [85] и метод Резника [99]. Результаты данного вычисления указаны в таблице 5:

Таблица 5 — Близость между словами «университет» и «экзамен»

Метод	Леска	Резника
Близость	20	0

Теперь вычислим те же самые метрики близости для слов, которые имеют совершенно различные значения: «университет» и «растение». Результаты указаны в таблице 6:

Таблица 6 — Близость между словами «университет» и «растение»

Метод	Леска	Резника
Близость	28	2,3447

Сравнивая результаты, приведенные в таблице 5 и в таблице 6 можно прийти к выводу, что слово «университет» ближе к слову «растение», чем к слову «экзамен», хотя очевидно, что данное утверждение далеко от истины.

3.5 Алгоритм построения контекстного множества терма

Анализ, проведенный в разделе 3.3, показал, что применение тезаурусов и словарей возможно, но связано с определенными техническими сложностями. Все эти факты значительно ограничивают область применения метода, полученного в разделе 3.2.

Исследование семантической близости термов является неотъемлемой частью теории обработки текстов на естественном языке. Семантическая близость между двумя сущностями с течением времени может изменяться в связи с изменениями корпусов и словарей [98]. Кроме того, семантическая близость двух сущностей может быть различна в различных предметных областях. Например, слово «одноклассники» в российском интернете чаще всего будет связано с фразой «социальная сеть», однако данный смысл вряд ли будет представлен в каком-либо из словарей или корпусов. Человека, ищущего данное слово в интернете скорее всего интересует именно этот смысл данного слова, чем какой-либо другой. Поддержание словарей в актуальном состоянии является достаточно сложной задачей, потому что предметные области меняются и очень часто создаются новые слова и новые смыслы приписываются новым словам.

Семантическая близость термов может быть вычислена с помощью специализированных баз данных и статистических корпусов, так же огромное число современных подходов к вычислению семантической близости основано на вычислении расстояний между словами в известной семантической сети WordNet. При расчете семантической близости может использоваться так же связность между словами в контексте, а также ее важность. Чтобы показать разницу между связностью и близостью Резник Ф.А. приводил в пример слова «автомобиль» и «бензин». Данные слова не являются синонимами и их значения далеки друг от друга, однако очевидно, что эти термины все-таки имеют что-то общее. Эти слова могут иметь сильную функциональную взаимосвязь в контексте, в пример можно привести фразу «автомобили используют бензин в качестве топлива» [99].

Предлагается другой способ вычисления семантической близости, основанный на предположении, что семантически близкие термы употребляются в одинаковых или схожих контекстах. Главная идея состоит в том, что связность

между словами не является численной величиной, удобнее ее представлять в виде множества слов, связанных с заданным термином. Для расчета семантической близости введем понятие *«контекстное множество». Контекстное множество* — это множество термов, с которыми целевой терм встречается в одном контексте.

Определение. Будем называть множество слов, связанных с заданным термом, *контекстным множеством* терма. Чтобы решить данную проблему предлагается использовать метод, использующий множество связанных слов, которое, как уже было сказано выше, предлагается назвать *контекстным множеством* слова. Используя распространенные корпусы и словари, можно сформировать данное множество для любого слова.

Идея данного подхода строится на следующих предположениях:

– слова всегда можно понять из контекста;
– чтобы найти семантически схожие слова, необходимо найти слова, которые употребляются в таком же контексте.

Приведем простой пример:

– Стакан яблочного **сока** стоит на столе.
– Все любят фруктовый **сок**.
– Мы делаем **сок** только из свежих фруктов.

Из примера видно, что слова «яблочный» и «фрукты» могут быть семантически связаны друг с другом, поскольку употребляются в схожем контексте.

Предлагается следующий алгоритм построения контекстного множества терма. Возьмем матрицу корреспонденций термов G, описанную в главе 2 в разделе (2.4). Элементы данной матрицы отражают частоту встречаемости пары термов совместно. Предположим, необходимо построить контекстное множество i-того терма, в этом случае необходимо рассматривать i-тую строку, исключая i-тый элемент, поскольку он показывает встречаемость терма с самим собой. Обозначим эту строку G_i, ее элементами будут скалярные произведения:

$$G_i = \{g_{ij}\} = \{(x_i, x_j)\}_{i \neq j}^{n} \tag{3.5}$$

где x_i, x_j — векторы термов, n — количество термов.

После этого вычислим среднее арифметическое среди элементов вектора из формулы (3.5). Обозначим среднее буквой $\bar{G}_i$.

$$\bar{G}_i = \frac{\sum_{j \neq i} g_{ij}}{n-1} \tag{3.6}$$

Далее, отбросим все компоненты вектора G_i, меньшие среднего значения $\bar{G}_i$. Контекстное множество i-того терма будет состоять из термов, соответствующих оставшимся компонентам вектора G_i, для которых выполняется неравенство $g_{ij} \geq \bar{G}_i$.

3.5.1 Пример построения контекстного множества

Предположим у нас имеется выборка из заголовков новостей.

1. Британская полиция знает о местонахождении основателя WikiLeaks.
2. В суде США начинается процесс против россиянина, рассылавшего спам.
3. Церемонию вручения Нобелевской премии мира бойкотируют 19 стран.
4. В Великобритании арестован основатель сайта Wikileaks Джулиан Ассандж.
5. Украина игнорирует церемонию вручения Нобелевской премии.
6. Шведский суд отказался рассматривать апелляцию основателя Wikileaks.
7. НАТО и США разработали планы обороны стран Балтии против России.
8. Полиция Великобритании нашла основателя WikiLeaks, но, не арестовала.
9. В Стокгольме и Осло сегодня состоится вручение Нобелевских премий.

Семантическое ядро, в результате анализа, изложенного во второй главе, будет состоять из термов (обработанных стеммером Портера), указанных в таблице 7.

Таблица 7 — Пример семантического ядра

wikileaks	арестова	великобритан	вручен	нобелевс	основател	полиц
прем	прот	стран	суд	сша	церемон	

Матрица корреспонденций термов в этом случае приведена в таблице 8.

Таблица 8 — Пример матрицы корреспонденций термов

0.27	0.13	0.13	0.00	0.00	0.27	0.13	0.00	0.00	0.00	0.07	0.00	0.00
0.22	0.22	0.22	0.00	0.00	0.22	0.11	0.00	0.00	0.00	0.00	0.00	0.00
0.22	0.22	0.22	0.00	0.00	0.22	0.11	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.25	0.25	0.00	0.00	0.25	0.00	0.08	0.00	0.00	0.17
0.00	0.00	0.00	0.25	0.25	0.00	0.00	0.25	0.00	0.08	0.00	0.00	0.17
0.27	0.13	0.13	0.00	0.00	0.27	0.13	0.00	0.00	0.00	0.07	0.00	0.00
0.25	0.13	0.13	0.00	0.00	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.25	0.25	0.00	0.00	0.25	0.00	0.08	0.00	0.00	0.17
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.17	0.17	0.33	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.00	0.33	0.00
0.17	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.17	0.00	0.33	0.17	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.17	0.17	0.33	0.00
0.00	0.00	0.00	0.22	0.22	0.00	0.00	0.22	0.00	0.11	0.00	0.00	0.22

Предположим, мы хотим построить *контекстное множество* для терма «нобелевс». Для этого рассмотрим пятую строку матрицы корреспонденций термов из таблицы 8 и исключим из нее пятую компоненту, так как она показывает встречаемость терма самим с собой. Вектор терма «нобелевс» будет иметь вид:

$$G_5 = \{0.00, 0.00, 0.00, 0.25, 0.00, 0.00, 0.25, 0.00, 0.08, 0.00, 0.00, 0.17\} \qquad (3.7)$$

Рассчитаем среднее арифметическое среди компонент этого вектора, получим $\bar{G_5} \approx 0.06$. Далее отбросим все термы, компоненты которых в данном векторе меньше 0.06, оставшиеся термы составят контекстное множество терма **нобелевс**, обозначим его C_5.

$$C_5 = \{\textit{вручен}, \textit{прем}, \textit{стран}, \textit{церемон}\} \qquad (3.8)$$

Чтобы графически интерпретировать данный факт, проведем отрогональное разложение матрицы корреспонденций термов из таблицы 8. Отбросим все последние столбцы матрицы V (формула 2.16), оставив только первые два (формула 3.9).

$$V = \begin{pmatrix} -0.276 & 0.202 \\ -0.318 & 0.215 \\ -0.338 & 0.315 \\ -0.069 & -0.213 \\ -0.069 & -0.353 \\ -0.356 & 0.172 \\ -0.370 & 0.220 \\ -0.069 & -0.313 \\ -0.258 & -0.304 \\ -0.167 & -0.383 \\ -0.333 & -0.086 \\ -0.258 & -0.344 \\ -0.031 & -0.390 \end{pmatrix} \tag{3.9}$$

На рисунке 3.5 можно увидеть картину расположения слов в семантическом пространстве. Термы, которые составляют контекстные множества друг друга, располагаются в семантическом пространстве рядом. На рисунке можно заметить, что все термы образуют три независимые группы:

– термы, вокруг *wikileaks* (документы, где идет обсуждение wikileaks);
– термы, вокруг *прем* (документы, где идет обсуждение Нобелевской премии);
– термы, вокруга *сша* (остальные документы).

Этот факт можно использовать для автоматического выделения рубрик и последующей автоматической рубрикации. В реальных задачах групп получается намного больше, чем две, и пространство будет многомерным, а не двумерным, но идея подхода остается той же.

Несложно заметить, что подавляющее число ячеек матрицы корреспонденций термов из таблицы 8 содержит нули. Данная матрица сильно разрежена, и использование контекстного множества позволяет значительно улучшить производительность и снизить потребление памяти.

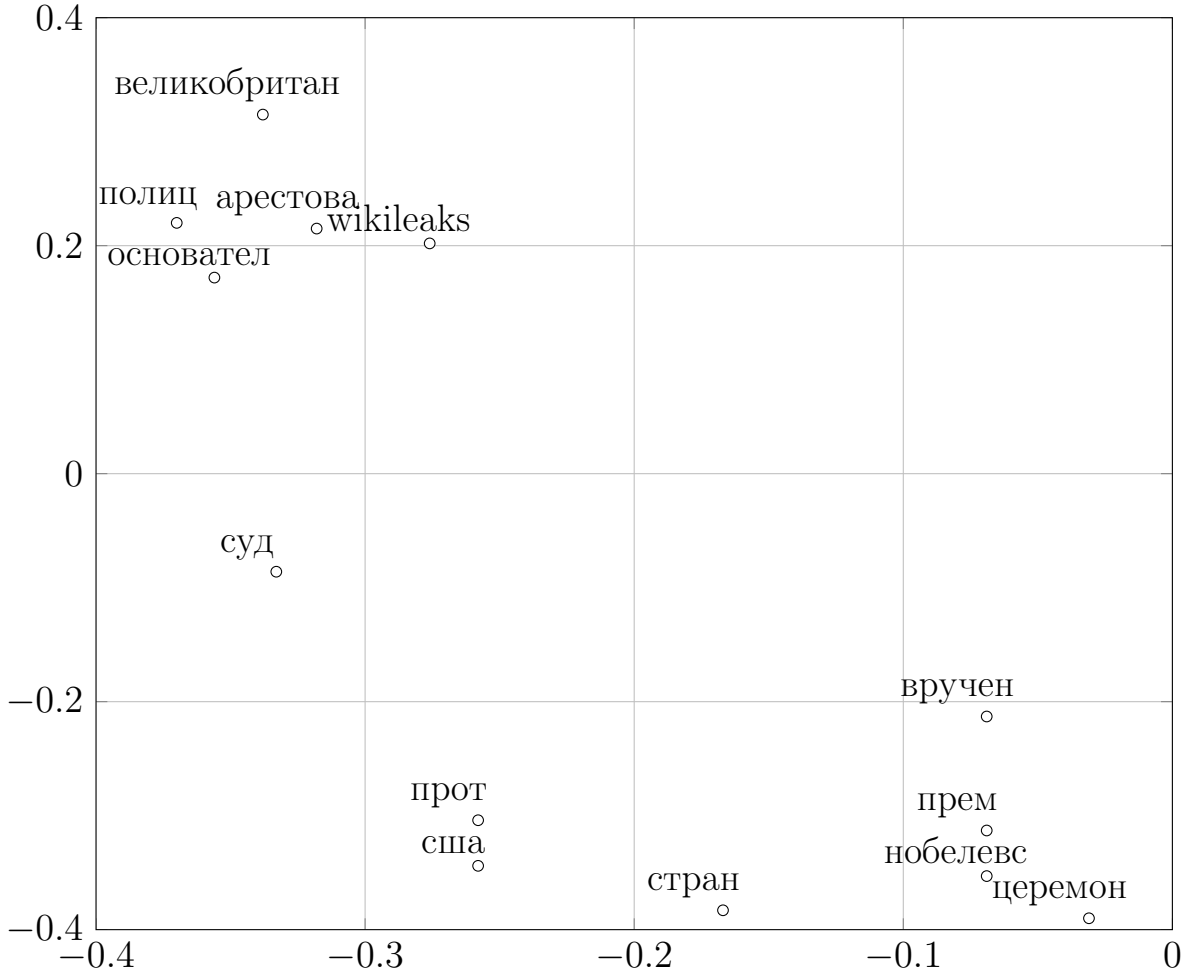

Рисунок 3.5 — Графическое изображение термов в семантическом пространстве

3.6 Предлагаемый метод вычисления семантической близости

В предыдущем разделе было дано понятие контекстного множества, а так же введен алгоритм его построения. Контекстное множество можно использовать при вычислении семантической близости между двумя термами для ее использования в формуле 3.2.Предположим, что w_1 и w_2 — слова, для которых необходимо вычислить семантическую близость. Предлагаемый метод можно условно разделить на несколько шагов:

1. Формирование контекстных множеств слов w_1 и w_2 по алгоритму, описанному в разделе 3.5. Пусть $C_1 = \{c_{11}, c_{12}, ..., c_{1n}\}$ и $C_2 = \{c_{21}, c_{22}, ..., c_{2m}\}$ — контекстные множества слов w_1 и w_2 соответственно. Данные множества содержат слова, с которыми слова w_1 и w_2 часто

употребляются в одном контексте. Затем мы формируем общее контекстное множество слов:

$$C = C_1 \cup C_2 \tag{3.10}$$

Очевидно, что мощность данного множества будет не меньше чем $n+m$.

2. Вычисление нормализованных близостей между общим контекстным множеством и каждым из слов w_1 и w_2:

$$\begin{aligned} \textit{близость}(c_i, w_1) &= \frac{\textit{частота}(c_i, w_1)}{\textit{макс.частота}(w_1)} \\ \textit{близость}(c_i, w_2) &= \frac{\textit{частота}(c_i, w_2)}{\textit{макс.частота}(w_2)} \end{aligned} \tag{3.11}$$

где $\textit{частота}(c_i,w_1)$ — количество документов, где c_i и w_1 встречаются вместе, а $\textit{макс.частота}(w_j)$ рассчитывается по формуле как максимум частот по всем словом из объединенного контекстного множества C:

$$\textit{макс.частота}(w_j) = \max\left\{\textit{частота}(c_i,w_j)\right\}, c_i \in C \tag{3.12}$$

3. Расчет семантической близости.

   Рассмотрим расчет семантической близости между словами w_1 и w_2. Для этого рассчитаем коэффициенты R_i для всех слов из контекстного множества C по формуле:

$$R_i = \frac{\min\left\{\textit{близость}(c_i, w_1), \textit{близость}(c_i, w_2)\right\}}{\max\left\{\textit{близость}(c_i, w_1), \textit{близость}(c_i, w_2)\right\}} \tag{3.13}$$

   Обозначим p_i — коэффициент совместной встречаемости w_1 и w_2 во всей выборке, равный 2 в случае, когда оба слова встречаются в одном документе и 1 в противном случае, s коэффициент синонимии, равный 1, если слова w_1 и w_2 являются синонимами и 0, в противном случае. Тогда семантическая близость слов w_1 и w_2 рассчитывается по формуле:

$$\textit{сем.близость}(w_1,w_2) = \frac{\sum\limits_{i=1}^{k}\left(\frac{p_i R_i}{1+R_i} + s\right)}{1+s} \tag{3.14}$$

В результате применения формулы (3.14) получится число в промежутке $[0, 0.75 \cdot (n+m)]$, чтобы получить семантическую близость в диапазоне $[0, 1]$ необходимо разделить получившийся результат на $0.75 \cdot (n+m)$. Для семантически близких слов, коэффициент близок к 1.

3.6.1 Пример расчета семантической близости

Рассчитаем семантическую близость для очевидно близких термов *нобелевс* (1) и *прем* (1) и для двух далеких термов *нобелевс* (2) и *wikileaks* (3) из раздела 3.5, обозначим их соответственно C_1, C_2 и C_3. Их контекстные множества соответственно равны:

$$\begin{gathered} C_1 = \{\textit{вручен}, \textit{прем}, \textit{стран}, \textit{церемон}\} \\ C_2 = \{\textit{вручен}, \textit{нобелевс}, \textit{стран}, \textit{церемон}\} \\ C_3 = \{\textit{великбритан}, \textit{полиц}, \textit{арестова}, \textit{основател}\} \end{gathered} \tag{3.15}$$

Как видно из формулы 3.15, контекстные множества C_1 и C_2 отличаются только одним словом, поэтому общее контекстное множество будет равно:

$$C_{12} = \{\textit{вручен}, \textit{прем}, \textit{стран}, \textit{церемон}, \textit{нобелевс}\}$$

Контекстные множества C_1 и C_3 отличаются значительно, их общее контекстное множество равно:

$$C_{13} = \{\textit{вручен}, \textit{прем}, \textit{стран}, \textit{церемон}, \textit{великбритан}, \textit{полиц}, \textit{арестова}, \textit{основател}\}$$

После построения общих контекстных множества можно вычислить нормализованные близости между общим контекстным множеством и каждым из термов (*нобелевс*, *прем*) и термов (*нобелевс*, *wikileaks*). Воспользуемся для этого формулой 3.11. В таблицах 9 и 10 представлены частоты совместной встречаемости в документах каждого из термов с каждым словом из общего контекстного множества.

Таблица 9 — Частоты совместной встречаемости термов (*нобелевс*, *прем*)

вручен	прем	стран	церемон	нобелевс
2	2	1	1	2
2	2	1	1	2

Таблица 10 — Частоты совместной встречаемости термов (*нобелевс*, *wikileaks*)

вручен	прем	стран	церемон	великбритан	полиц	арестова	основател
2	2	1	1	0	0	0	0
0	0	0	0	2	2	2	3

Так же требуется определить максимальную частоту совместной встречаемости в документах для каждого из термов, для которых необходимо вычислить семантическую близость. Для этого воспользуемся формулой 3.12. Максимальная частота для слова *нобелевс* равна 2, для *прем* - 2, для *wikileaks* - 3. Теперь можно рассчитать нормализованные близости между каждым словом из общего контекстного множества и каждым из термов, для которых требуется определить семантическую близость по формуле 3.11. Результаты этого вычисления представлены в таблицах 11 и 12.

Таблица 11 — Нормализованные частоты совместной встречаемости термов (*нобелевс*, *прем*)

вручен	прем	стран	церемон	нобелевс
1	1	0.5	0.5	1
1	1	0.5	0.5	1

Далее необходимо рассчитать коэффициенты R_i для всех слов из контекстных множеств C_{12}, C_{13} и каждым из термов, для которых необходимо вычислить семантическую близость по формуле 3.13.

После вычисления всех коэффициентов R_i можно переходить к вычислению семантической близости по формуле 3.14. В итоге семантическая близость между термами *нобелевс* и *прем* получится равной 0.8, а между термами *нобелевс* и *wikileaks* равной 0.23. Если взглянуть рисунок 3.5, можно убедиться, что полученные результаты близки к достоверным. Примечательно, что даже на обучающей выборке такого малого объема, получились результаты близкие к достоверным. Это говорит о том, что метод пригоден для работы в условиях неполноты данных.

Таблица 12 — Нормализованные частоты совместной встречаемости термов (*нобелевс*, *wikileaks*)

вручен	прем	стран	церемон	великбритан	полиц	арестова	основател
1	1	0.5	0.5	0	0	0	0
0	0	0	0	0.66	0.66	0.66	1

Таблица 13 — Коэффициенты R_i для термов (*нобелевс*, *прем*)

вручен	прем	стран	церемон	нобелевс
1	1	1	1	1

3.7 Выводы по третьей главе

В главе предложена векторная модель представления знаний, использующая семантическую близость термов. Модель помогает решить проблему лексической неоднозначности терминов, а также находит скрытые семантические связи между документами, сравнивая семантически близкие термы.

Так же была проанализирована возможность применения словарей и тезаурусных баз данных для вычисления семантической взаимосвязи между термами. Было выявлено, что их применение ограничено из-за того, что они охватывают в основном общеупотребительную лексику. Зачастую словари, как традиционные, так и современные, имеют значительный объем, который может оказаться слишком большим для использования в прикладных задачах. Более того, вычисление семантической близости между термами может оказаться не очень точным, поскольку семантически близкие термы часто располагаются в тезаурусах достаточно далеко.

В главе предложен статистический способ вычисления семантической близости между словами, который предлагается в качестве альтернативы использованию известных семантических сетей по типу WortdNET. Метод основан на сборе контекстных множеств термов в форме набора слов. Экспериментальные результаты показывает, что метод является эффективным. Однако задача под-

Таблица 14 — Коэффициенты R_i для термов (*нобелевс*, *wikileaks*)

вручен	прем	стран	церемон	великбритан	полиц	арестова	основател
2	2	1	1	0	0	0	0

бора контекстных множеств термов является достаточно сложной. Основные результаты, представленные в данной главе опубликованы в [60, 61].

Глава 4. Вычислительные эксперименты

В данной главе приводятся результаты вычислительных экспериментов по исследованию эффективности разработанных в диссертации моделей, методов и алгоритмов. В качестве тестовых данных использовались открытые наборы данных, такие как: USENET, ClueWeb09, ClueWeb12, NBER Patent Citations, сборник статей русскоязычной википедии.

4.1 Выбор порогового значения сингулярных коэфициентов

Рассмотрим вычислительные эксперименты по установлению порогового значения сингулярных коэффициентов для отбрасывания малозначимых термов при выделении семантического ядра методом разложения матрицы корреспонденций термов (глава 2 раздел 2.4).

Для предметной области «поиск вакансий разовой работы» при установке порогового значения сингулярных коэффициентов равным 1 и при объеме обучающей выборки около 100000 текстов, термов получилось немногим более 1000. Для тестирования результатов работы алгоритма было разработано специальное программное обеспечение. На рисунке 4.1 изображена зависимость количества термов, полученных в результате обучения, от порогового значения сингулярных коэффициентов.

В таблице 15 представлены результаты обучения при некоторых точностях.

Так как количество термов меняется существенно, то на рисунке 4.1 приведен график зависимости $\lg n$ (по вертикальной оси) от пороговых значений сингулярных коэффициентов, здесь n — количество оставляемых термов. Зависимость хорошо описывается убывающей экспонентой. Такой вид зависимости $\lg n$ характерен и для других предметных областей. Основываясь на практическом опыте применения данного алгоритма, можно сказать, что оптимальной является точность, при которой получается около 1000 термов [10].

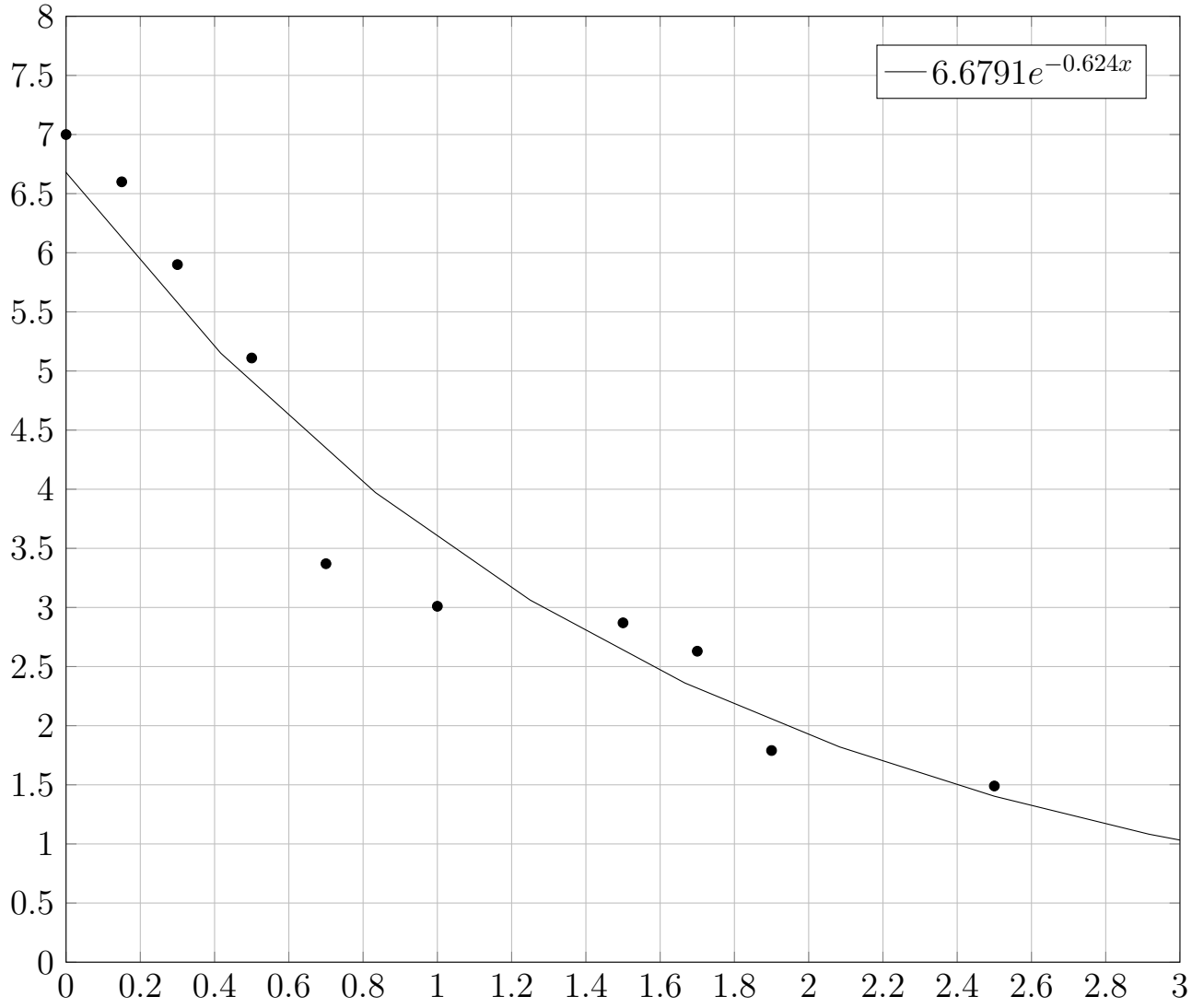

Рисунок 4.1 — Зависимость количества термов от порогового значения сингулярных коэффициентов

4.2 Сравнение с другими алгоритмами

Особенностью алгоритма, изложенного в главе 2, по сравнению с известными методами является то, что, во-первых, используется ортогональное разложение МКТ, чтобы избавиться от малозначимых слов, во-вторых, происходит переход к новому «семантическому пространству», размерность которого значительно меньше исходного. Однако следует заметить, что качество выдачи алгоритма может снижаться в случае, если в какую-либо из рубрик (категорий) попадет незначительное количество текстов. Преимущества алгоритма так же иллюстрирует таблица 16:

Таблица 15 — Результаты обучения

Пороговое значение сингулярных коэффициентов	Количество термов
0	10000000
0.15	4000000
0.3	981532
0.5	130054
0.7	2345
1	1043
1.5	749
1.7	436
1.9	63
2.5	31

Таблица 16 — Сравнительная характеристика алгоритмов подбора персональных рекомендаций

Метрика 1млн. анализируемых единиц (текстов)	Разработанный алгоритм	Стандартный векторный метод	Метод латентно-семантического анализа	Распознавание с помощью многослойного перцептрона
Размер индекса	65мб	170мб	170мб	85мб
Время создания индекса	30с	32с	33с	60с
Время поиска	71-110мс	320-330мс	0.6с	39-46мс
Полнота результатов	100%	65%	63%	56.4%
Наличие непустой выдачи в 100% случаев	Да	Нет	Нет	Нет

Примечание: результаты, представленные в таблице 16 были получены на веб-сервере, обладающим следующими техническими характеристиками и программным обеспечением:

– Операционная система Debian 7.0 Wheezy;
– Система управления базами данных MariaDB 10.0;
– Языки разработки C++/PHP;
– RAM 1Gb;
– CPU Intel Celeron Dual-Core.

Стандартные алгоритмы, описанные в таблице 16, были реализованы в соответствии с их описаниями в статьях [3], [63] и [73]. Разработанный алгоритм был опробован на нескольких предметных областях, а именно:

– подбор подходящих вакансий;
– подбор рекомендуемых автозапчастей для автомобилей китайского производства;
– фильтрация спама;
– подбор рекомендуемой литературы по уже прочитанным книгам.

4.3 Оценка результатов работы алгоритма с переопределением весов термов

В алгоритм, изложенный в главе 2, внесены изменения, позволяющие повысить эффективность метода и качество результирующей выборки. Данные изменения характеризуются иным способом вычисления весов термов на этапе построения векторных моделей. Таким образом на этапе 2 алгоритма, описанного в главе 2, после построения и нормализации векторов обучающей выборки происходит вычисление новых весов термов по алгоритму представленному в разделе 3.1.

Для оценки эффективности алгоритма классификации представления знаний использующую семантическую близость термов по сравнению с алгоритмом, представленным в главе 2, была проверена ее работа алгоритмов на различных множествах текстов:

1. Объявления о работе

2. Новости
3. Литературные аннотации

В таблице 17 представлены некоторые сведения о выборках:

Таблица 17 — Сведения о выборках

Выборка	Количество текстов	Количество категорий	Распределение
Объявления о работе	около 700 тыс.	17	неравномерное
Новости	около 1,2 млн.	10	равномерное
Литературные аннотации	около 20 тыс.	13	равномерное

В качестве мер оценок результатов использовались *F-measure* [95] и *purity* [116]:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.1}$$

где *precision* — количество правильных результатов в выдаче алгоритма, *recall* — общее количество результатов выдачи.

$$purity(W,C) = \sum_k \max_j |w_k \cup c_j| \tag{4.2}$$

где W — множество документов, w_k — k-тый документ, C — множество категорий — множество документов, отнесенных классификатором к категории k, c_j — множество документов, отнесенных к категории j экспертом.

Меры, описанные формулами (4.1) и (4.2), показывают, насколько результаты работы разработанного классификатора соответствуют представлениям экспертов в предметной области.

В таблице 18 представлены результаты данных оценок. Взяты средние значения оценок 15 текстов:

На рисунке 4.2 изображена визуализация сравнения алгоритма из главы 2 и алгоритма с использованием семантической близости (закрашенные столбики).

Результаты экспериментов показывают, что использование векторной модели с вычислением семантической близости помогает улучшить результаты работы классификатора по сравнению с векторной моделью без учета семанти-

Таблица 18 — Оценка результатов работы алгоритма классифиции

	Алгоритм из главы 2		Алгоритм с использованием семантической близости	
Множество	F-measure	Purity	F-measure	Purity
Объявления о работе	0.31	0.33	0.65	0.66
Новости	0.56	0.58	0.61	0.64
Литературные аннотации	0.56	0.57	0.63	0.67

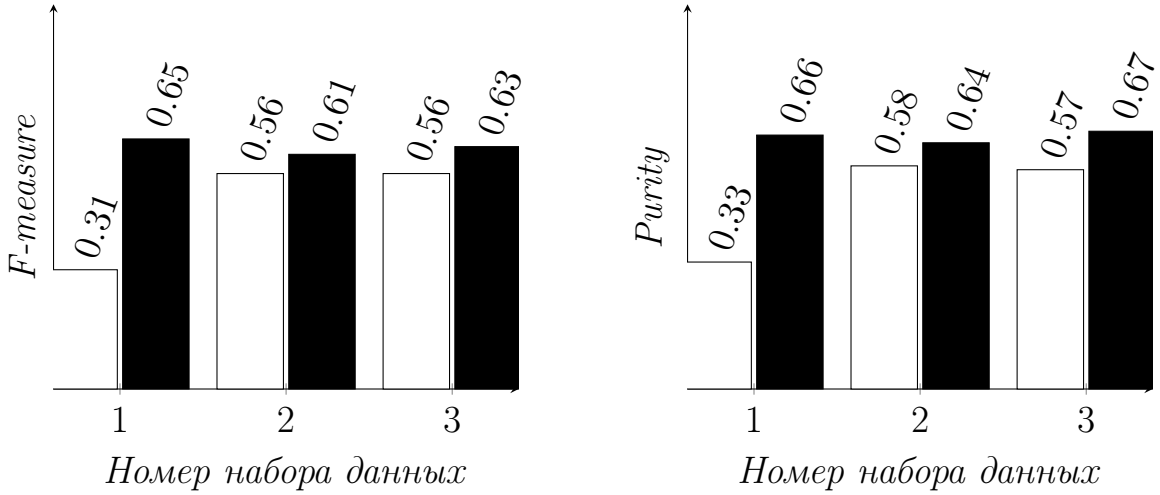

Рисунок 4.2 — Визуализация сравнения работы алгоритмов

ческой близости. В среднем категориальная векторная модель с использованием семантической близости дает на 8-10% более точный результат. Это связано с тем, что предложенная в главе 3 модель менее чувствительна к «шумам» за счет настройки весовых коэффициентов с помощью вычисления семантической близости. Новые весовые коэффициенты векторов документов учитывают контекст появления термов. Более высокие веса связаны с термами, которые сильнее семантически связаны с другими термами.

Эксперименты были проведены над выборками разного рода и объема, на всех из них метод отработал эффективно. Так же часть выборок была распределена неравномерно, метод и на них показал хороший результат в то время, как результаты векторной модели без учета семантической близости термов оказались неудовлетворительными.

4.4 Оценка результатов работы алгоритма вычисления семантической близости термов

Проверим эффективность метода расчета семантической близости (3.5) — (3.12) на словах «автомобиль» и «поезд», обучив на основе новостей, представленных на сайте одного федерального СМИ. Примем, что w_1 — «машина», w_2 — «поезд». Составим так же контекстные множества для данных слов с помощью алгоритма, описанного в разделе 3.5. Получим:

$$\begin{aligned} C_1 &= \{\textit{автомобиль}, \textit{мотор}, \textit{колесо}, \textit{пассажир}, \textit{двигатель}\} \\ C_2 &= \{\textit{рельсы}, \textit{транспорт}, \textit{двигатель}, \textit{груз}, \textit{пассажир}\} \end{aligned} \tag{4.3}$$

Поскольку слова «машина» и «автомобиль» — синонимы, то будем считать, что если документ содержит слово «автомобиль», то он содержит и слово «машина». В таблице 19 представлены нормализованные близости между общим контекстным множеством и словами, вычисленные по формуле 3.11

Таблица 19 — Нормализованные близости между общим контекстным множеством и словами «машина» (1) и «поезд» (2)

№	рельсы	трансп-т	двигатель	груз	пассажир	автомоб.	мотор	колесо
1	0,19	1,00	0,15	0,07	0,11	0,63	0,89	0,11
2	0,13	0,89	0,63	0,06	0,10	1,00	0,68	0,30

Далее рассчитаем коэффициенты R_i, результаты указаны в таблице 20. Расчет был произведен по формуле (3.13) для каждого слова из общего контекстного множества.

Таблица 20 — Коэффициенты R_i

рельсы	трансп-т	двигатель	груз	пассажир	автомоб.	мотор	колесо
0,71	0,89	0,24	0,84	0,93	0,63	0,77	0,37

После вычисления данных коэффициентов можно переходить к непосредственному вычислению семантической близости. Рассчитаем близость между словами «машина» и «поезд» по формуле (3.14), получим 3.1 (0.52 после нормализации). Рассчитаем так же семантическую близость с помощью расстояния Жаккара [109], получим 0.55. Без проведения дополнительных расчетов очевидно, что результаты достаточно близки друг к другу.

Проверка. В таблице 21 представлен результат сравнения эффективности представленного метода в сравнении с расстоянием Жаккара. Правая колонка иллюстрирует среднее арифметическое оценок людей. Группе из 50 экспертов было предложено оценить близость между двумя словами по стобальной шкале. В столбце представлен средний и нормализованный результат. Строка «корреляция» показывает коэффициент корреляции между результатами, полученными в результате применения каждого методов и оценкой реальных людей.

Таблица 21 — Эффективность представленного метода

Пара слов	Расстояние Жаккара	Представленный метод	Средняя оценка 50 человек
Веревка — Улыбка	0.102	0.137	0.16
Побережье — Лес	0.016	0.649	0.41
Бухта — Холм	0.444	0.559	0.87
Машина — Путешествие	0.071	0.443	0.33
Фрукт — Еда	0.753	0.685	0.55
Автомобиль — Машина	0.654	0.939	1
Полдень — Обед	0.106	0.876	0.97
Джем — Варенье	0.295	0.836	0.84
Корреляция	0.45	0.851	

Довольно высокий коэффициент корреляции показывает, что результаты предложенного метода, ближе к объективным, чем метод основанный на вычислении расстояния Жаккара. Исходя из полученных результатов, можно судить, что представленный метод является эффективным. При этом решены проблемы, возникшие в главе 3, связанные с необходимостью хранения и построения словарей гиперонимов и словарей определений.

К сожалению, при применении предложенного подхода возникает новая проблема, состоящая в сложности подбора определителей термов. Для этого, например, могут быть использованы любые внешние источники данных (веб-сайты, журналы, книги, словари, корпусы). Наибольшую эффективность метод показывает при использовании в качестве источников данных веб-сайтов с ясной структурой (например, веб-энциклопедии).

4.5 Сравнение работы на известных наборах данных

В качестве тестовых данных использовались известные открытые наборы данных, такие как: USENET, ClueWeb09, ClueWeb12, NBER Patent Citations. Кроме того, отличие данного эксперимента от предыдущих, заключается в том, что в данном разделе тестируется метод целиком, а в предыдущих разделах — его составные части. Каждый из этих наборов состоит из трех выборок: обучающая, валидационная и тестовая. Все эти выборки размечены (распределены по рубрикам), что позволяет оценить количество верных и неверных срабатываний алгоритма. Оценка качества считается с помощью *f-measure* и *purity*. Так же оценивается размерность пространства, получаемая при построении моделей хранения знаний. Сравнение производилось со следующими алгоритмами:

1. Метод, разработанный в рамках диссертационной работы.
2. Векторная модель представления знаний, основанная на представлении bag-of-words.
3. Метод латентно-семантического анализа.
4. Метод, основанный на использовании нейронной сети.
5. Латентное размещение Дирихле
6. Эволюционный подход

Таблица 22 — Размер модели представления знаний

Набор данных	Выборка	1	2	3	4	5	6
USENET	train	2.7G	7.8G	3.3G	4G	2.8G	0.5G
ClueWeb09	train	3.1G	8G	4.5G	4.1G	3.1G	0.7G
ClueWeb12	train	4G	9.7G	7G	5.9G	4.5G	0.8G
NBER Patent Citations	train	2G	5G	3.8G	6G	2.2G	0.2G

Анализируя результаты, представленные в таблице 22 и на рисунке 4.3, можно увидеть, что метод, разработанный в диссертационной работе, использует, как минимум, в 2 раза меньше памяти, чем исходная векторная модель. Кроме, того можно заметить, что в большистве случаев метод так же оказывается эффективнее других методов сжатия семантического пространства. Можно так же заметить, что размерность семантического пространства при использовании эволюционных подходов получается очень маленькой, но такая модель очень сложно интерпретируется и не всегда дает точный результат.

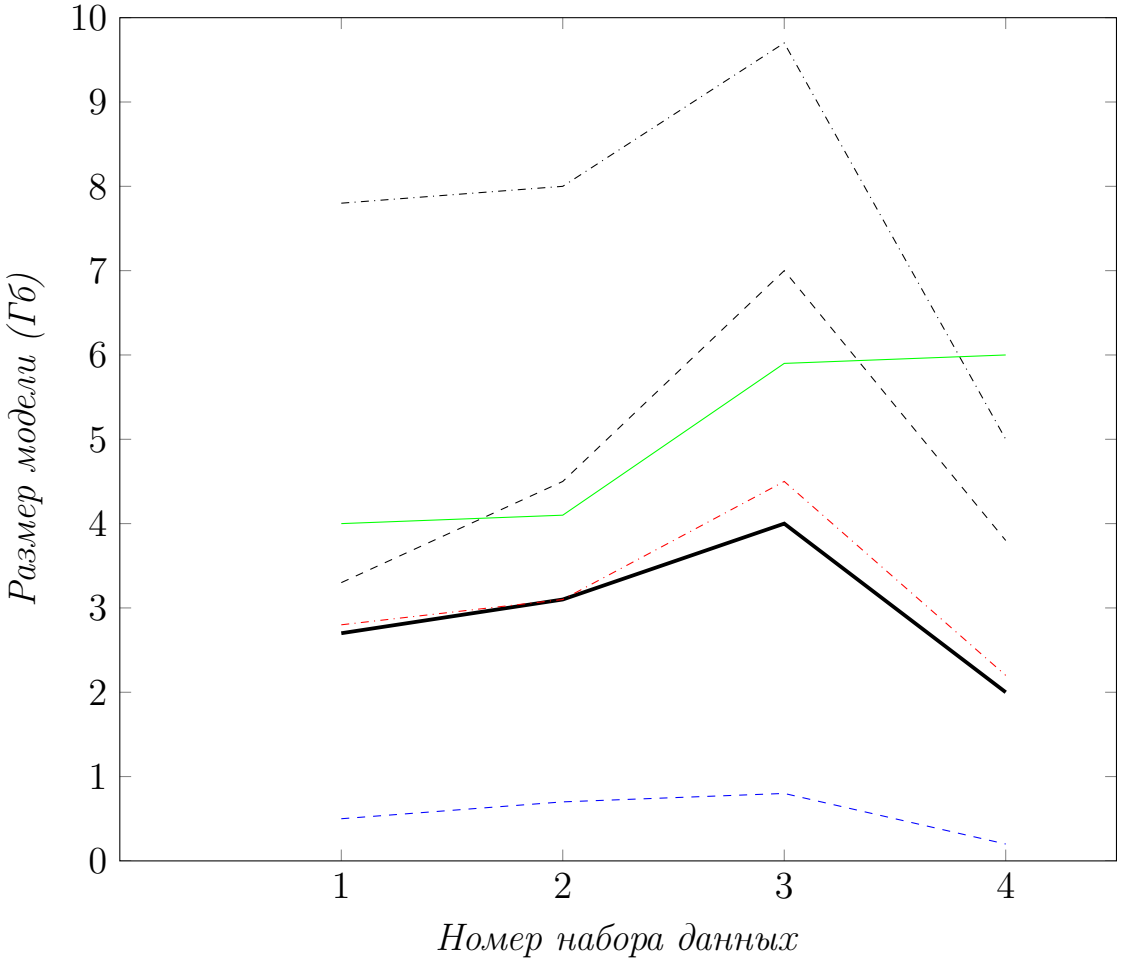

Рисунок 4.3 — Размер модели представления знаний

Следующим этапом производилось тестирование времени построения индекса и времени выдачи результата. Для каждого набора данных и для каждого алгоритмов проводилась серия экспериментов. В таблицах 23 и 24 показаны средние оценки по времени среди 10000 экспериментов. Результаты экспериментов так же иллюстрирует рисунок 4.4. Для хранения знаний использовалась документ-ориентированная СУБД MongoDB.

Таблица 23 — Среднее время построения индекса (мин.)

Набор данных	Выборка	1	2	3	4	5	6
USENET	test	7.5	26	6	8	7	56
ClueWeb09	test	8	30	7.6	10	7.6	70
ClueWeb12	test	8.5	35	9	13	9	83
NBER Patent Citations	test	5	14	4.3	11	4	43

Здесь можно отметить, что построение индекса алгоритма, разработанного в диссертационной работе, производится в ряде случаев несколько медленнее конкурентов. Однако, поскольку это действие производится только один раз

Таблица 24 — Среднее время выдачи результата (сек.)

Набор данных	Выборка	1	2	3	4	5	6
USENET	test	0.2	0.6	0.3	0.2	0.22	2
ClueWeb09	test	0.21	0.7	0.32	0.23	0.22	3
ClueWeb12	test	0.21	0.7	0.32	0.33	0.23	3.3
NBER Patent Citations	test	0.19	0.56	0.3	0.21	0.21	2

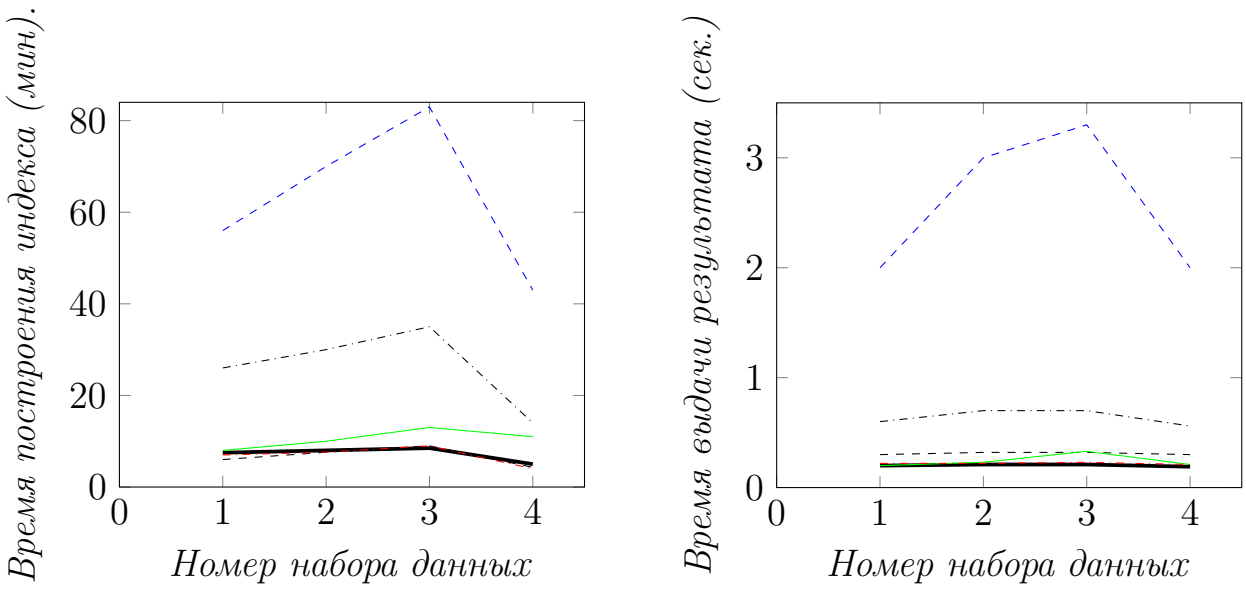

Рисунок 4.4 — Визуализация временных оценок

(на стадии обучения) и разница во времени невелика, относительно объемов данных, которые обрабатывается этот факт можно считать несущественным. Гораздо большее значение имеет тот факт, что среднее время генерации ответа в большинстве случаев происходит быстрее, чем у других методов.

Последним этапом производилось тестирование качества алгоритма с помощью мер *f-measure* (таблица 25) и *purity* (таблица 26). Визуализация результатов представлена на рисунке 4.5. Для тестирования была проведена серия из 10000 экспериментов. Здесь можно отметить, что качество работы метода в большинстве оценок было значительно выше, чем у остальных, рассматриваемых алгоритмов.

Таблица 25 — Средняя оценка f-measure

Набор данных	Выборка	1	2	3	4	5	6
USENET	test	0.8	0.37	0.7	0.5	0.75	0.1
ClueWeb09	test	0.79	0.4	0.72	0.53	0.7	0.12
ClueWeb12	test	0.6	0.45	0.6	0.53	0.54	0.2
NBER Patent Citations	test	0.53	0.47	0.5	0.41	0.31	0.07

Таблица 26 — Средняя оценка purity

Набор данных	Выборка	1	2	3	4	5	6
USENET	test	0.81	0.33	0.69	0.53	0.75	0.08
ClueWeb09	test	0.77	0.30	0.6	0.48	0.71	0.11
ClueWeb12	test	0.63	0.42	0.62	0.48	0.53	0.19
NBER Patent Citations	test	0.52	0.41	0.53	0.44	0.31	0.09

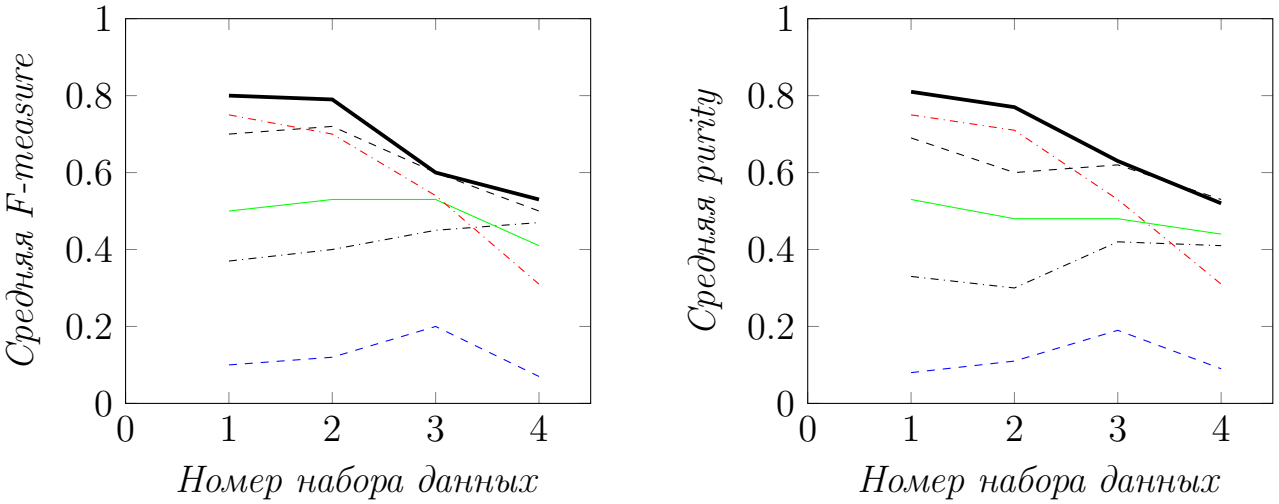

Рисунок 4.5 — Средние F-measure и Purity

На рисунках 4.3, 4.4 и 4.5 изображены визуализации полученных результатов. Самой толстой линией на всех графиках обозначен алгоритм, разработанный в рамках диссертационной работы. Рисунки показывают достоинства алгоритма по сравнению с другими, рассмотренными методами.

4.6 Выводы по четвертой главе

В данной главе были описаны, во-первых, эксперименты по выбору различных характеристик алгоритма, влияющих на его работоспособность, во-вторых, сравнительные эксперименты, исследующие его эффективность.

Эксперименты показали, что:

1. Использование матрицы корреспонденций термов в качестве модели хранения знаний позволяет добиться значительного сокращения семантического пространства и при этом качество обучения алгоритма получается, как минимум, не хуже исходного.

2. Использование мер семантической близости позволяет повысить качество информационного поиска.
3. Использование статистического подхода к вычислению семантической близости термов позволяет снизить размерность промежуточных данных.

Основные результаты исследований данной главы опубликованы в статьях [8, 12, 14].

Заключение

Для того, чтобы сделать работу систем информационного поиска, систем поддержки принятия решений, систем формирования персональных рекомендаций более качественной необходимо разрабатывать новые специализированные модели и алгоритмы.

В диссертационной работе были рассмотрены вопросы разработки и исследования методов и моделей интеллектуального анализа текстов. Рассмотрено и доказано, что: при выделении семантического ядра с помощью латентно-семантического анализа, т.е. путем сингурярного разложения терм-документной матрицы, длинные документы сильнее влияют на семантическое пространство, чем короткие. Доказано, что использование нормализованной терм-документной матрицы позволяет уравнять влияние длинных и коротких документов на семантическое пространство. Предложено и обосновано использование матрицы корреспонденций термов вместо терм-документной матрицы для формирования семантического пространства. Предложен алгоритм сортировки результирующей выборки по степени релевантности запросу пользовтеля. Предложены метод перевзвешивания термов векторной модели документа с помощью семантической близости и метод вычисления семантической близости с помощью построения контекстного множества. Эффективность разработанных методов подтверждена вычислительными экспериментами.

Основные результаты, полученные в ходе выполнения диссертационного исследования являются новыми и не покрываются ранее опубликованными научными работами других авторов, обзор которых был дан в разделе 1.5 Отметим основные отличия.

В модели лингвистической онтологии, предложенной в [37], используется набор отношений лингвистической онтологии, который специально подобран для описания широкой предметной области. Однако, при использовании данного подхода для пополняемых рекомендательных систем требуется регулярное переопределение отношений, что требует существенных вычислительных ресурсов. Разработанная в рамках диссертационной работы модель не требует постоянного переобучения.

В работе [21] предлагается аддитивная регуляризация тематических моделей (ARTM), которая основана на максимизации взвешенной суммы логарифма правдоподобия и дополнительных критериев-регуляризаторов. Применение результатов данного исследования ограничено на предметных областях, в которых данные тематически распределены неравномерно, т.е. документы некоторых тем преобладают.

В работах [93, 94] рассматриваются особенности применения латентно-семантического анализа для поиска и классификации веб-документов. В данных работах не рассматривается применение латентно-семантического анализа в условиях неравномерости данных. Подход, описанный в диссертационной работе, позволяет на любой запрос пользователя вернуть выборку, отсортированную по степени полезности.

В работе [43] описана обобщенная модель порождения текстов на основе цепей Маркова. В отличии от модели, полученной в рамках диссертационной работы, не рассматривается работа в условиях неполноты данных и ограничено применение в рекомендательных системах с многочисленными рубриками.

В статье [1] описываются способы ранжирования текстовых документов на основе лога действий пользователя поисковой системы. Главный недостаток данного подхода по сравнению с алгоритмом, полученным в рамках диссертационного исследования, состоит в том, что метод не решает проблему «холодного запуска»

В работе [36] предлагается новый подход к извлечению оценочных слов для различных предметных областей. В результате применения данного подхода получается модель очень большого размера, что ограничивает ее применение на предметных областях, в которых требуется анализ большого количества текстов.

Результаты, полученные в ходе настоящего диссертационного исследования, могут применяться при создании рекомендательных, поисковых и прочих систем, связанных с поиском, рубрикацией и фильтрацией информации.

В качестве направлений дальнейших исследований можно выделить следующие.

– Разработка и исследование методов учета поведенческих характеристик пользователя для улучшения релевантности поиска.

– Исследование эффективности подходов и методов, предложенных в диссертации, для различных систем управления базами данных.
– Обобщение подходов, моделей и методов, предложенных в работе, для текстов различных предметных областей.

Результаты диссертационного исследования были внедрены в следующих российских компаниях (данный факт подтверждается наличием соответствующих актов):

– в системе автоматизированного лингвистического анализа ЗАО «Селфхендер» для: автоматического определения категориального положения объявления, автоматического отсеивания нежелательных объявлений (фильтрация спама) и автоматического отсеивания предложений, не являющихся объявлениями о поиске сотрудников;
– в подсистеме системы электронного документооборота ООО «Восточный экспресс» для автоматического определения адресата документа и автоматической классификации входящих документов;
– на веб-сайте ООО «КитАвтоИмпорт» для: автоматической рубрикации поступающих товарных позиций, автоматизированного нахождения дублирующих товарных позиций и автоматического нахождения похожих или сопутствующих товарных позиций в системе учета товаров.

Список литературы

1. *Агеев М.* Ранжирование документов по запросу на основе лога действий пользователей поисковой системы // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). — 2012. — Т. 13. — С. 559—571.
2. *Алексеев А. А.*, *Лукашевич Н. В.* Комбинирование признаков для извлечения тематических цепочек в новостном кластере // Труды Института системного программирования РАН (электронный журнал). — 2012. — Т. 23. — С. 257—276.
3. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. — Москва : Мир, 1986. — 576 с.
4. *Богомолова А. В.*, *Дышкант Н. Ф.*, *Юдина Т. Н.* Университетская информационная система РОССИЯ: ресурсы и сервисы для поддержки общественного участия и задач государственного управления // Труды XI Всероссийской объединенной конференции "Интернет и современное общество". — Санкт Петербург, 2008. — С. 196—199.
5. *Большакова Е. И.*, *Большаков И. А.* Алгоритмы построения компьютерного словаря русских буквенных паронимов и его применение // Эвристические алгоритмы и распределенные вычисления. — 2015. — Т. 2015, № 3. — С. 8—22.
6. *Большакова Е. И.*, *Большаков И. А.* Аффиксальный критерий паронимии для построения компьютерного словаря паронимов русского языка // Научно-техническая информация. Серия Информационные процессы и системы. — 2015. — № 11. — С. 28—35.
7. *Большакова Е.*, *Лукашевич Н.*, *Нокель М.* Извлечение однословных терминов из текстовых коллекций на основе методов машинного обучения // Информационные технологии. — 2013. — С. 31—37.
8. *Бондарчук Д. В.* Алгоритм построения семантического ядра для текстового классификатора // В мире научных открытий. — 2015. — Т. 68, № 8.2. — С. 713—724.

9. *Бондарчук Д. В.* Выбор оптимального метода интеллектуального анализа данных для подбора вакансий // Информационные технологии моделирования и управления. — 2013. — 6(84). — С. 504—513.

10. *Бондарчук Д. В.* Интеллектуальный метод подбора персональных рекомендаций, гарантирующий получение непустого результата // Информационные технологии моделирования и управления. — 2015. — Т. 2(92). — С. 130—138.

11. *Бондарчук Д. В.* Использование латентно-семантического анализа в задачах классификации текстов по эмоциональной окраске // Бюллетень результатов научных исследований. — 2012. — 2(3). — С. 146—151.

12. *Бондарчук Д. В.*, *Тимофеева Г. А.* Выделение семантического ядра на основе матрицы корреспонденций термов // Системы управления и информационные технологии. — 2015. — Т. 61, № 3.1. — С. 134—139.

13. *Бондарчук Д. В.*, *Тимофеева Г. А.* Математические основы метода категориальных векторов в интеллектуальном анализе данных // Вестник Уральского государственного университета путей сообщения. — 2015. — 4(28). — С. 4—8.

14. *Бондарчук Д. В.*, *Тимофеева Г. А.* Применение машинного обучения для формирования персональных рекомендаций в сфере трудоустройства // Экономика и менеджмент систем управления. — 2015. — Т. 18, № 4.2. — С. 215—221.

15. *Бондарчук Д.* Оптимальный метод интеллектуального анализа данных для подбора вакансий // Отечественная наука в эпоху изменений: постулаты прошлого и теория нового времени. — 2015. — С. 81—84.

16. *Бондарчук Д.* Система интеллектуальной классификации и ранжирования веб-контента // Сборник материалов конференции ДНИ НАУКИ ОТИ НИЯУ МИФИ-2012. — Озерск, 2012. — С. 47—49.

17. *Вапник В. Н.*, *Стерин А. М.* Об упорядоченной минимизации суммарного риска в задаче распознавания образов // Автоматика и телемеханика. — 1978. — № 10. — С. 83—92.

18. *Варламов М. И.*, *Коршунов А. В.* Расчет семантической близости концептов на основе кратчайших путей в графе ссылок Википедии // Машинное обучение и анализ данных. — 2014. — № 8. — С. 1107—1125.

19. *Веретенников А.* Использование дополнительных индексов для более быстрого полнотекстового поиска фраз, включающих часто встречающиеся слова // Системы управления и информационные технологии. — 2013. — 2(52). — С. 61—66.

20. *Воронцов К.* Вероятностное тематическое моделирование [Электронный ресурс]. — URL: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf (дата обр. 16.04.2016).

21. *Воронцов К.*, *Потапенко А. А.* Аддитивная регуляризация тематических моделей // Доклады Академии наук. — 2014. — Т. 456, № 3. — С. 268—271.

22. *Воронцов К.*, *Фрей А.*, *Ромов П.* BigARTM: библиотека с открытым кодом для тематического моделирования больших текстовых коллекций // Аналитика и управление данными в областях с интенсивным использованием данных. — 2015. — С. 28—36.

23. *Галушкин А.* Нейронные сети. Основы теории. — Москва : Горячая линия – Телеком, 2012. — 496 с.

24. *Гантмахер Ф. Р.* Теория матриц. — М. : Наука, 1966. — 576 с. — ISBN 5-9221-0524-8.

25. *Гмурман В. Е.* Теория вероятностей и математическая статистика. — Москва : Высшая школа, 2013. — 479 с.

26. *Горелик С.*, *Марков Я.*, *Чернышкова М.* Мониторинг сложных систем на основе феноменологической модели // Современные наукоёмкие технологии. — 2016. — № 1. — С. 13—18.

27. *Джарратано Д.*, *Райли Г.* Экспертные системы: принципы разработки и программирование. — 4-е изд. — М. : Вильямс, 2006. — 1152 с. — ISBN 978-5-8459-1156-8.

28. *Добров Б. В.*, *Лукашевич Н. В.*, *Невзорова О. А.* Технология разработки онтологий новых предметных областей // Труды Казанской шкрлы по компьютерной лингвистике TEL-2002. Выпуск 7. — 2002. — С. 90—106.

29. *Жианчанг М.*, *Дж. Э.* Введение в искусственные нейронные сети. — Открытые системы, 1997.

30. *Зиновьев А. Ю.* Визуализация многомерных данных. — Красноярск : Изд. Красноярского государственного технического университета, 2000. — 180 с.

31. *Ильвовский Д.* Применение семантически связанных деревьев синтаксического разбора в задаче поиска ответов на вопросы, состоящие из нескольких предложений // Научно-техническая информация. Серия 2: Информационные процессы и системы. Т. 2. — 2014. — С. 28—37.

32. *Клещев А.*, *Шалфеева Е.* Классификация свойств онтологий. Онтологии и их классификации // НТИ сер. 1. — 2005. — № 9. — С. 16—22.

33. *Куприянов М.*, *Першин А.* Методика моделирования агентных поисковых систем с самовосстановлением // Известия СПбГЭТУ «ЛЭТИ». Серия «Информатика, управление и компьютерные технологии». — 2010. — С. 61—66.

34. *Курейчик В. М.* Гибридные генетические алгоритмы // Известия Южного федерального университета. Технические науки. — 2007. — Т. 7, № 2. — С. 5—12.

35. Лингвистическая онтология "Тезаурус РуТез". — URL: http://www.labinform.ru/ruthes/index.htm.

36. *Лукашевич Н. В.*, *Четверкин И. И.* Построение модели для извлечения оценочной лексики в различных предметных областях // Моделирование и анализ информационных систем. — 2013. — Т. 20, № 2. — С. 70—79.

37. *Лукашевич Н.*, *Добров Б.* Проектирование лингвистических онтологий для информационных систем в широких предметных областях // Онтология проектирования. — 2015. — Т. 5, № 1. — С. 47—69.

38. *Мальковский М.*, *Арефьев Н.* Семантические ограничения в словаре сочетаемости: эксперименты по разрешению синтаксической неоднозначности // Сборник научных трудов SWorld по материалам международной научно-практической конференции. — 2011. — Т. 4, № 1. — С. 21—25.

39. *Мальковский М.*, *Соловьев С.* Универсальное терминологическое пространство // Труды международного семинара - 311 - Компьютерная лингвистика и интеллектуальные технологии. — 2002. — Т. 1. — С. 266—270.

40. *Мальковский М.*, *Старостин А.*, *Миняйлов В.* Восстановление эллипсиса как задача автоматической обработки текстов // Программные продукты и системы. — 2014. — № 3. — С. 32—36.

41. *Орлов А.* Системная нечеткая интервальная математика (СНИМ) — перспективное направление теоретической и вычислительной математики // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. — 2013. — № 7. — С. 230—255.

42. *Павлов А.*, *Добров Б.* Обнаружение поискового спама в Вебе на основе анализа разнообразия текстов // Труды Института системного программирования РАН (электронный журнал). — 2011. — Т. 21. — С. 277—296.

43. *Павлов А.*, *Добров Б.* Метод обнаружения массово порожденных неестественных текстов на основе анализа тематической структуры // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). — 2011. — Т. 12, № 2. — С. 58—72.

44. Проект RussNet. — URL: http://www.russnet.org/.

45. *Розенблатт Ф.* Принципы нейродинамики: Перцептроны и теория механизмов мозга. — М. : Мир, 1965. — 480 с.

46. *Ручкин В.*, *Злобин В.* Нейросети и нейрокомпьютеры. — С-Петербург : БХВ-Петербург, 2011. — 256 с.

47. *Сегалович И.*, *Маслов М.* Некоторые аспекты полнотекстового поиска и ранжирования Яндекса // Российский семинар по Оценке Методов Информационного Поиска, Труды РОМИП-2004. — 2004. — С. 100—109.

48. *Синопальникова А. А.*, *Азарова И. В.*, *Яворская М. В.* Принципы построения wordnet-тезауруса RussNet // Компьютерная лингвистика и интеллектуальные технологии: труды международной конференции Диалог 2004. — 2004. — С. 542—547.

49. *Сухоногов А. М.*, *Яблонский С. А.* Разработка русского WordNET // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды шестой всероссийской научной конференции RDCL-2004. — 2004. — С. 113—117.

50. *Тузовский А.* Формирование семантических метаданных для объектов управления знаниями // Известия Томского политехнического университета. — 2007. — Т. 310. — С. 108—112.

51. Университетская информационная система Россия (УИС РОССИЯ). — URL: http://uisrussia.msu.ru.

52. *Хомоненко А.*, *Бубнов В.*, *Краснов С.* Модель функционирования системы автоматической рубрикации документов в нестандартном режиме // Проблемы информационной безопасности. Компьютерные системы. — 2011. — № 4. — С. 16—23.

53. *Хомоненко А.*, *Краснов С.* Применение латентно-семантического анализа для автоматической рубрикации документов // Известия Петербургского университета путей сообщения. — 2012. — 2(31). — С. 125—132.

54. *Хомоненко А.*, *Логашев С.*, *Краснов С.* Автоматическая рубрикация документов с помощью латентно-семантического анализа и алгоритма нечеткого вывода Мамдани // Труды СПИИРАН. — 2016. — 1(44). — С. 5—19.

55. *Amaravadi C. S.* Knowledge Management for Administrative Knowledge // Expert Systems. — 2005. — 25(2). — Pp. 53–61.

56. *Banerjee S.*, *Pedersen T.* An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet // Lecture Notes In Computer Science. — 2002. — Vol. 2276. — Pp. 136–145.

57. *Bishop C.* Neural Networks for Pattern Recognition. — Oxford : Oxford University Press, 1995. — 177 pp.

58. *Blei D. M.* Probabilistic topic models // Communications of the ACM. — 2012. — Vol. 55, no. 4. — Pp. 77–84.

59. *Blei D. M.*, *Ng A. Y.*, *Jordan M. I.* Latent Dirichlet Allocation // Journal of Machine Learning Research. — 2003. — 3(4–5). — Pp. 993–1022. — DOI: 10.1162/jmlr.2003.3.4-5.993.

60. *Bondarchuk D. V.*, *Timofeeva G. A.* Vector space model based on semantic relatedness // AIP Conference Proceedings 1690, 020005. — 2015. — DOI: 10.1063/1.4936683.

61. *Bondarchuk D.* Vector space model using semantic relatedness // Abstracts of the International Conference and PhD Summer School «Groups and Graphs, Algorithms and Automata», August, 9-15. — Yekaterinburg, Russia, 2015. — P. 30.

62. *Bondarchuk D.*, *Martynenko A.* Spectral properties of a matrix of correspondences between terms // CEUR Workshop Proceedings, Vol. 1662, Proceedings of 47th International Youth School-Conference "Modern Problems in Mathematics and its Applications" (MPMA 2016). — 2016. — Pp. 186–190.

63. *Brachman R. J.* What IS-A is and isn't. An Analysis of Taxonomic Links in Semantic Networks // IEEE Computer. — 1983. — 16(10).

64. *Braslavski P.*, *Ustalov D.*, *Mukhin M.* A Spinning Wheel for YARN: User Interface for a Crowdsourced Thesaurus // Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. Gothenburg, Sweden: Association for Computational Linguistics. — 2014. — C. 101—104.

65. *Budanitsky A.*, *Hirst G.* Evaluating WordNet-based Measures of Lexical Semantic Relatedness // Computational Linguistics. — 2006. — Vol. 32. — Pp. 13–47.

66. *Cilibrasi R. L.*, *Vitanyi P. M.* The Google Similarity Distance, ArXiv.org or Clustering by Compression // IEEE Trans. Information Theory. — 2004. — No. 51. — Pp. 1523–1545.

67. *Cohen W. W.*, *Ravikumar P.*, *Fienberg S. E.* A comparison of string distance metrics for name-matching tasks // KDD Workshop on Data Cleaning and Object Consolidation. — No. 3. — Pp. 73–80.

68. *Forsythe G. E.*, *Malcolm M. A.*, *Moler C. B.* Computer Methods for Mathematical Computations // Prentice-Hall. — 1977.

69. *Galitsky B.*, *Ilvovsky D.*, *Kuznetsov S. O.* Style and Genre Classification by Means of Deep Textual Parsing // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2016". — 2016. — С. 171—181.

70. *Hainaut J.*, *Hick J.*, *Englebert V.* Understanding Implementations of IS-A Relations // ER 1996. — 1996. — Pp. 42–57.

71. *Han J.*, *Kamber M.* Data mining: Concepts and Techniques. — Morgan Kaufmann Publishers, 2001.

72. *Hastie T.*, *Tibshirani R.*, *Friedman J.* The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — Verlag : Springer, 2009. — 746 pp.

73. *Helbig H.* Knowledge Representation and the Semantics of Natural Language. — Berlin, Heidelberg, New York : Springer, 2006.

74. *Hotho A.*, *Staab S.*, *Stumme G.* WordNet Improve Text Document Clustering // Special Interest Group on Knowledge Discovery in Data 2003 Semantic Web Workshop. — 2003. — Pp. 541–544.

75. *Imai K.*, *King G.*, *Lau O.* Toward a Common Framework for Statistical Analysis and Development // Journal of Computational and Graphical Statistics. — 2008. — Vol. 17, no. 4. — Pp. 1–22.

76. *Jaro M. A.* Advances in record linkage methodology as applied to the 1985 census of Tampa Florida // Journal of the American Statistical Association. — 1989. — 84 (406). — Pp. 414–420. — DOI: 10.1080/01621459.1989.10478785.

77. *Jeh G.*, *Widom J.* SimRank: a measure of structural-context similarity // Proceedings of the 8th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining. — 2002. — Pp. 271–279.

78. *Jensen L.*, *Martinez T.* Improving text classification by using coceptual and contextual features // In Proceedings of the Workshop on Text Mining at the 6th Association for Computing Machinery's Special Interest Group

on Knowledge Discovery and Data Mining Int. Conference on Knowledge Discovery and Data Mining (KDD 00). — 2000. — Pp. 101–102.

79. *Jordan M. I.*, *Mitchell T. M.* Machine learning: Trends, perspectives, and prospects // Science. — 2015. — Vol. 349, no. 6245. — Pp. 255–260.

80. *Kechedzhy K. E.*, *Usatenko O.*, *Yampolskii V. A.* Rank distributions of words in additive many-step Markov chains and the Zipf law // Phys. Rev. E. 2005. — 2005. — Vol. 72. — Pp. 381–386.

81. *Konar A.* Artifcial intelligence and soft computing: behavioral and cognitive modeling of the human brain. — Boca Raton, Florida : CRC Press LLC, 2000.

82. *Krizhanovsky A.*, *Krizhanovskaya N.*, *Bravslavsky P.* Russian Lexicographic Landscape: a Tale of 12 Dictionaries // Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue". — 2015. — С. 254—271.

83. *Kuznetsov S.*, *Nezhanov A.*, *J. P.* A system for knowledge discovery in big dynamical text collections // CEUR Workshop Proceedings, Proceedings of the International Workshop "What Can FCA Do for Artificial Intelligence"(FCA4AI 2012). — 2012. — С. 81—87.

84. *Kuznetsov S.*, *Poelmans J.* Knowledge representation and processing with formal concept analysis // Wiley interdisciplinary reviews: Data mining and knowledge discovery. — 2013. — № 3. — С. 200—215.

85. *Lesk M.* Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone // SIGDOC 86. Proceedings of the 5th Annual International Conference on Systems Documentation. — 1986. — Pp. 24–26. — DOI: 10.1145/318723.318728.Lesk:1986:ASD:318723.318728.

86. *Loukachevitch N.*, *Dobrov B.* Development and Use of Thesaurusof Russian Language RuThes // In Proceedings of workshop on WordNet Structures and Standartisation, and How These Affect WordNet Applications and Evaluation. (LREC2002). — 2002. — Pp. 65–70.

87. *Loupy C.*, *El-Beze M.*, *Marteau P. F.* Word Sense Disambiguation using HMM Tagger // In Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC. — 1998. — Pp. 1255–1258.

88. *Markman A. B.* How do indirect measures of evaluation work? Evaluating the inference of prejudice in the Implicit Association Test // Journal of personality and social psychology. — 2001. — Vol. 81, no. 5. — Pp. 760–773.

89. *Matuschek M.*, *Gurevych I.* High Performance Word Sense Alignment by Joint Modeling of Sense Distance and Gloss Similarity // In Proceedings of the the 25th International Conference on Computational Linguistics (COLING 2014). — 2014. — С. 245—256.

90. *McAuley J. J.*, *Leskovec J.*, *Jurafsky D.* Learning attitudes and attributes from multi-aspect reviews // In Proceedings of International Conference on Data Mining. — 2012. — Pp. 1020–1025.

91. *Mihalcea R.* Using Wikipedia for Automatic Word Sense Disambiguation // Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2007. — 2007. — Pp. 196–203.

92. *Negnevitsky M.*, *Ledwich G.* Optimal distributed generation parameters for reducing losses with economic consideration // In Proceedings of Power Engineering Society General Meeting. — 2007. — Pp. 1–8.

93. *Nekrestyanov I.*, *Novikov B.*, *Pavlova E.* An analysis of alternative methods for storing semistructured data in relations // Lecture Notes in Computer Science. — 2000. — С. 354—361.

94. *Nekrestyanov I.*, *Panteleeva N.* Text retrieval systems for the web // Programming and Computer Software. — 2002. — Т. 28, № 4. — С. 207—225.

95. *Powers D. M.* Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. — 2011. — 2(1). — Pp. 37–63.

96. *Quillian M. R.* Semantic memory // in Semantic information processing. — 1968. — Pp. 227–270.

97. *Rapp R.* Word sense discovery based on sense descriptor dissimilarity // In Proceedings of the ninth machine translation summit. — New Orleans, 2003. — Pp. 315–322.

98. *Resnick P.*, *Varian H.* Recommender Systems // Communications of the ACM. — 1997. — 40(3). — Pp. 56–58.

99. *Resnik P.* Semantic similarity in taxonomy: An information-based measure and its application to problems of ambiguity in natural language // Journal of Artificial Intelligence Research. — 1999. — No. 11. — Pp. 95–130.

100. *Roussopoulos N.* Conceptual Modeling: Past, Present and the Continuum of the Future // Conceptual Modeling: Foundations and Applications. — 2009. — Pp. 139–152.

101. Russian WordNET. — URL: http://wordnet.ru/.

102. *Salton G.* Improving retrieval performance by relevance feedback // Readings in information retrieval. — 1997. — Vol. 24. — Pp. 1–5.

103. *Salton G.*, *Buckley C.* Term-weighting approaches in automatic text retrieval // Information Processing and Management. — 1988. — 24(5). — Pp. 513–523.

104. *Salton G.*, *Wong A.*, *Yang C. S.* A vector space model for automatic indexing // Communications of the ACM. — 1975. — 18(11). — Pp. 613–620.

105. *Sedding J.*, *Dimitar K.* WordNet-based Text Document Clustering // COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data. — 2004. — Pp. 104–113.

106. *Segaran T.* Programming Collective Intelligence. — LA : O'REILLY, 2008. — 368 pp.

107. *Sowa J. F.* Cognitive Architectures for Conceptual Structures // In Proceedings of International Conference on Computational Science (ICCS-2011). — 2011. — Pp. 35–49.

108. *Strube M.*, *Ponzetto S. P.* WikiRelate Computing semantic relatedness using Wikipedia // In Proceedings of the American Association for Artificial Intelligence (AAAI-2006). — 2006. — Pp. 1419–1424.

109. *Tan P. N.*, *Steinbach M.*, *Kumar V.* Top 10 algorithms in data mining // Knowledge and information systems. — 2008. — Vol. 14, no. 1. — Pp. 1–37.

110. *Teevan J.* Improving inforamtion retrieval with textual analysis: Bayesian models and beyond: MA thesis / Teevan J.B. — Master's Thesis, Department of Electrical Engineering, Computer Science, Massachusetts Institute of Technology, 2001.

111. *Tierney L.* Some notes on the past and future of LISP-STAT // Journal of Statistical Software. — 2013. — Pp. 1–15.

112. *Usama F.*, *Smyth P.*, *Piatetsky–Shapiro G.* From Data Mining to Knowledge Discovery in Databases // Artifical intelligence Magazine. — 1996. — 17(3). — Pp. 34–54.

113. *Walker A.*, *McCord M.*, *Sowa J. F.* Knowledge Systems and Prolog, Second Edition. — Addison-Wesley, 1990.

114. *Wilkinson J.* The Algebraic Eigenvalue Problem. — Oxford : Clarendon Press, 1965.

115. *Willett P.* The Porter stemming algorithm: then and now // Program: Electronic Library and Information Systems. — 2006. — Vol. 40, no. 3. — Pp. 219–223.

116. *Xiong H.*, *Wu J.*, *Chen J.* K-means clustering versus validation measures: A data distribution perspective // Conference on Knowledge Discovery and Data Mining. — 2006. — Pp. 877–886.

117. Yet Another RussNet. — URL: https://russianword.net/.

118. *Zadeh L.* Fuzzy Logic // Computer. — 1988. — 1(4). — Pp. 83–93.

119. *Zadeh L.* Fuzzy sets // Information and Control. — 1965. — No. 8. — Pp. 338–353.

120. *Zadeh L.* Knowledge representation in fuzzy logic // IEEE Transactions on Knowledge and Data Engineering. — 1989. — No. 1. — Pp. 89–100.

121. *Zadeh L.* Outline of a new approach to the analysis of complex systems and decision processes // IEEE Transactions on Systems, Man, and Cybernetics. — 1973. — 3(1). — Pp. 28–44.

122. *Zadeh L.* The concept of a linguistic variable and its Application to approximate reasoning // Information Sciences. — 1975. — No. 8. — Pp. 199–249.

123. *Zesch T.*, *Muller C.*, *Gurevych I.* Using Wiktionary for computing semantic relatedness // In Proceedings of the 23rd AAAI Conference on Artificial Intelligence. — 2008. — Pp. 861–866.

124. *Zhou Z.* Three perspectives of data mining // Artif. Intell. — 2003. — No. 46. — Pp. 139–143.

Список рисунков

Список таблиц

Приложение А

Список сокращений и условных обозначений

ЛСА — Латентно-семантический анализ
LDA — Латентное размещение Дирихле
МКТ — Матрица корреспонденций термов
ТДМ — Терм-документная матрица
TF-IDF — **TF** — term frequency, **IDF** — inverse document frequency

Приложение Б

Словарь терминов

Терм-документная матрица — это математическая матрица, описывающая частоту терминов, которые встречаются в коллекции документов.

Сингулярное разложение — это математическая операция, раскладывающая матрицу на 3 составляющих.

Векторная модель — в информационном поиске представление коллекции документов векторами из одного общего для всей коллекции векторного пространства.

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса.

Документ — неупорядоченное множество термов.

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (ребра) задают отношения между ними.

Стемминг — это процесс нахождения основы слова для заданного исходного слова.

Семантическое ядро — это подборка понятий, имеющих существенное значение для данной предметной области.

Расстояние Хэмминга — это количество различающихся позиций для строк с одинаковой длинной.

Онтология — это база знаний специализированного типа, содержащая сведения о понятийной структуре и терминологическом составе предметной области.

Средний вектор — вектор, состоящий из средних арифметических соответствующих компонент векторов текстов.

Синсеты — синонимические ряды WordNet, объединяющие слова со схожим значением

Семантическая связность — это количество связей, с помощью которых связаны два слова.

Контекстное множество — множество слов, связанных с заданным термом.

Синонимия — случай, когда несколько слов имеют схожий смысл, например, «автомобиль» и «машина».

Полисемия — случай, когда одно слово имеет несколько смыслов, например, «диск».