Федеральное государственное бюджетное учреждение науки
Институт системного программирования Российской академии наук

На правах рукописи

АСТРАХАНЦЕВ НИКИТА АЛЕКСАНДРОВИЧ

МЕТОДЫ И ПРОГРАММНЫЕ СРЕДСТВА ИЗВЛЕЧЕНИЯ ТЕРМИНОВ ИЗ КОЛЛЕКЦИИ ТЕКСТОВЫХ ДОКУМЕНТОВ ПРЕДМЕТНОЙ ОБЛАСТИ

Специальность 05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук,
профессор, академик РАН
Иванников Виктор Петрович

Москва – 2014

Содержание

Введение

Актуальность

Термин — это слово или словосочетание, обозначающее понятие заданной предметной области. Автоматическое извлечение терминов является важным этапом решения многих задач, связанных с обработкой текстов предметной области. К таким задачам относятся построение глоссариев, тезаурусов или онтологий, информационный поиск, машинный перевод, классификация и кластеризация документов.

К настоящему времени разработано множество методов автоматического извлечения терминов, однако их эффективность остается достаточно низкой: как правило, их точность и полнота[1] не превышают 50% — и при этом может сильно варьироваться для разных предметных областей. Кроме того, многие методы требуют размеченных вручную данных, что сужает их практическую применимость.

Одна из причин низкой эффективности методов заключается в том, что они недостаточно полным образом используют возможные источники данных.

Большинство существующих методов извлечения терминов основано на частоте вхождения кандидатов в рассматриваемую коллекцию текстовых документов. К таковым относятся, например, частота вхождений термина (Term Frequency, TF), TF-IDF [1], Domain Consensus [2], C-Value [3]. Одними из первых методов извлечения многословных терминов можно считать меры ассоциации, измеряющие, насколько случайно совместное появление слов в составе термина: взаимная информация (Mutual Information, MI), критерии Стьюдента (TTest), хи-квадрат, логарифмическое правдоподобие (Loglikelihood Ratio), LexicalCohesion и др. В отдельную подгруппу можно вынести методы на основе тематического моделирования: Term Score [4],

[1] Определения точности и полноты приводятся в разделе 1.4.

Maximum Term Frequency [4], Novel Topic Model [5] и др. Некоторые методы также учитывают контекст вхождений, например NC-Value [3] и PostRankDC (или DomainModel) [6].

В других методах — Weirdness [7], Domain Pertinence [8], Domain Relevance [9], Relevance [7] — используется частота вхождений во внешнюю коллекцию документов, не принадлежащую какой-либо определенной предметной области, например корпус новостей или художественной литературы. Иногда для извлечения терминов — как правило, двухсловных, реже многословных — применяются и другие внешние ресурсы, такие как поисковые машины интернета [10–12] или существующие тезаурусы [8, 12].

В последние годы стали появляться методы, основанные на интернет-энциклопедии Википедия [13–16]. Как правило, они используют алгоритмы поиска путей в графе категорий или случайного блуждания по этому графу и требуют вручную выбрать несколько категорий, которые соответствуют интересующей предметной области. При этом большая часть этих методов не использует коллекцию документов предметной области, опираясь исключительно на информацию Википедии; исключением можно назвать работу Вивальди и др. [16], в которой свойства путей в графе категорий (количество и длина путей) применяется для проверки терминов, определенных с помощью другого метода, однако здесь также требуется вручную задать категории Википедии, описывающие предметную область.

Некоторые работы пытаются комбинировать нескольких методов — в таком случае возникает задача преобразования вектора признаков (результатов работы каждого метода) в число, показывающее уверенность метода в том, что данный кандидат в термины является правильным термином.

Наиболее простым способом является линейная комбинация с вручную подобранными коэффициентами (как правило, равными), которая используется, например, в методах TermExtractor [9] или PostRankDC [6].

Также используется метод на основе алгоритма голосования, предложенный в работе З. Чжана и др. [17]. Данный метод не требует нормализации признаков и показывает в среднем лучшие результаты.

При наличии размеченных данных становится возможным применять алгоритмы машинного обучения с учителем, в частности AdaBoost [18], логистическую регрессию [12, 19, 20], Random forest [19], Gradient Boosting [21].

Как было показано в работе [19], классификаторы на основе машинного обучения достигают лучшей средней точности.

Таким образом, большая часть методов ограничивается текстами предметной области, которые зачастую не содержат в себе необходимого объема информации для автоматического извлечения терминов; некоторые методы также используют внешние ресурсы, такие как корпуса текстов других предметных областей, поисковые машины или созданные экспертами онтологии, однако все эти ресурсы обладают своими недостатками. Так, внешние текстовые документы, в том числе найденные поисковыми машинами, не имеют структуры и позволяют использовать только статистическую информацию о встречаемости слов и словосочетаний вне рассматриваемой предметной области, а созданные вручную онтологии обычно обладают малым объемом и покрывают лишь самые общие понятия предметных областей или только одну предметную область.

Указанных недостатков во многом лишена многоязычная интернет-энциклопедия Википедия. Ее статьи описывают понятия реального мира — как универсальные, так и специфичные для узких предметных областей; она содержит структурную информацию в виде гиперссылок между статьями; обладает очень большим размером и ежедневно пополняется сообществом пользователей.

Существующие методы, как было показано выше, недостаточно полным образом используют Википедию. Как правило, в качестве возможных терминов они рассматривают только названия существующих статей Википедии, что заведомо ограничивает полноту извлечения терминов. В частности, многие методы опираются только на информацию Википедии, например структуру категорий, не используя коллекцию документов предметной области.

Использование более полным образом имеющихся источников данных — коллекции документов, внешних корпусов, а также Википедии, включая ее структуру гиперссылок, — может значительно повысить эффективность методов автоматического извлечения терминов.

Цель диссертационной работы

Целью настоящей диссертационной работы является разработка методов и программных средств извлечения терминов из коллекции текстовых доку-

ментов предметной области с использованием структуры гиперссылок Википедии.

Разрабатываемые методы должны обладать следующими свойствами:

1. полная автоматичность, в том числе отсутствие требований к наличию размеченных вручную данных;
2. точность и полнота выше соответствующих показателей современных методов для различных предметных областей.

Для достижения цели были поставлены и решены следующие задачи:

1. Исследовать существующие методы извлечения терминов.
2. Разработать метод автоматического извлечения терминов, использующий структуру гиперссылок Википедии.
3. Реализовать разработанный метод в виде программной системы и провести экспериментальное исследование его применения с целью определения эффективности разработанного метода.

Основные положения, выносимые на защиту

1. Предложен подход к использованию информации Википедии для задачи извлечения терминов, основанный на структуре гиперссылок Википедии.
2. Предложен подход к извлечению терминов на основе алгоритма частичного обучения, не требующий размеченных данных.
3. В рамках предложенных подходов разработан метод автоматического извлечения терминов.
4. Разработана программная система извлечения терминов и проведено экспериментальное исследование, доказывающее повышение эффективности разработанного метода по сравнению с существующими методами.

Научная новизна

В настоящей работе предложен новый метод извлечения терминов из коллекции текстов предметной области, основанный на алгоритме частичного обучения и использовании структурной информации Википедии. Математически доказана оценка вычислительной сложности разработанного метода. Экспериментально подтверждено повышение эффективности разработанного метода по сравнению с существующими методами.

Разработанный метод не зависит от предметной области, не требует размеченных вручную данных, может применяться в различных задачах обработки текстов предметной области.

Теоретическая и практическая значимость

Предложенный подход к извлечению терминов и разработанные в его рамках методы могут быть использованы при решении прикладных задач автоматической и полуавтоматической обработки текстов, в том числе информационного поиска, определения ключевых фраз, классификации и кластеризации документов, машинного перевода, построения и обогащения словарей, тезаурусов, онтологий.

Созданная на основе разработанного метода программная система была включена в систему Texterra, разрабатываемую в Институте системного программирования РАН.

Апробация работы

Основные результаты работы докладывались на следующих конференциях и семинарах:

- на десятом весеннем коллоквиуме молодых исследователей в области баз данных и информационных систем (SYRCoDIS) (2013г.);
- на сто шестьдесят первом заседании Московской Секции ACM SIGMOD (2013г.);
- на двадцатой Международной конференции по компьютерной лингвистике «Диалог» (2014г.);
- на научном семинаре «Управление данными и информационные системы» Института системного программирования РАН (2014г.);

- на научном семинаре «Интернет, распределенные информационные системы и цифровые библиотеки» ВЦ РАН (2014г.)

Личный вклад

Автором проведено исследование предметной области и существующих методов, разработаны все описанные в диссертации методы, подготовлена спецификация для программной системы на основе разработанных методов, проведено экспериментальное исследование. Программная система разработана совместно с Д.Г. Федоренко.

Публикации

Основные результаты по теме диссертации изложены в 6 печатных изданиях [19, 20, 22–25], 4 из которых изданы в журналах, рекомендованных ВАК [22–25].

В обзорной работе [22] автором проведен анализ существующих работ и написаны введение и основной текст статьи, заключение написано совместно с Д.Ю. Турдаковым. В работах [19, 23] автором сформулированы общие концепции и планы статей и, совместно с Д.Г. Федоренко, проведены экспериментальные исследования. В работе [24] автором написана глава 4, посвященная базе знаний системы Текстерра, в том числе обогащению базы знаний. В работе [20] автором проведен анализ существующих работ, написан текст статьи и, совместно с Д.Г. Федоренко, проведены экспериментальные исследования.

Объем и структура работы

Диссертация состоит из введения, четырех глав, заключения и двух приложений. Полный объем диссертации составляет 133 страниц с 26 рисунками и 14 таблицами. Объем приложений составляет 15 страниц. Список литературы содержит 117 наименований.

Глава 1

Извлечение терминов

Данная глава посвящена описанию рассматриваемой задачи и существующих методов решения. Учитывая множество смыслов, вкладываемых в понятия «термин» и «предметная область», вначале приводится обзор множества разработанных определений этих понятий и выбираются рабочие определения.

Задача извлечения терминов также объединяет под собой множество различных постановок, редко выделяемых в явном виде — в настоящей главе приводится такое разделение, конкретизируется постановка задачи, принятая в данной работе, и обозреваются существующие методы извлечения терминов и метрики оценки эффективности в соответствии с этой постановкой.

1.1 Определение термина

История терминоведения насчитывает более 80 лет, за это время опубликовано множество работ (только на русском языке защищено более 2300 диссертаций [26]), большая часть которых так или иначе обсуждает определения термина.

По мнению К. Мякшина, непрекращающиеся дискуссии по этому вопросу вызваны «многогранностью феномена», тем, что термин является «языковой универсалией» [27]. Действительно, изучение термина можно назвать всесторонним: «В настоящее время термин изучается в следующих аспектах: теория термина, лингвистические аспекты, психолингвистические аспекты, социолингвистические аспекты, филологические аспекты, функционально-

стилистические аспекты, дискурсивные аспекты, диахронические аспекты, функциональные аспекты, философские аспекты, семиотические аспекты, логические аспекты, гносеологические аспекты, системные аспекты, дидактические аспекты, информационные аспекты, прагматические аспекты, переводческие аспекты. Естественно, объектом изучения термин является также в сферах исследования терминов, относимых в настоящее время к отраслям терминоведения: когнитивное терминоведение, отраслевое терминоведение, историческое терминоведение и др.» [28]

Однако несмотря на проработанность вопроса и большое количество существующих определений термина, исследователями отмечается отсутствие общеупотребительного, универсального определения: «неоднократные попытки лингвистов сформулировать удовлетворяющее всех определение понятия «термин» оказались малопродуктивными» [27]; «the notion itself of term is still not clear, both from a pure linguistic and a computational point of view» (Перевод: само понятие «термин» до сих пор остается неясным как с точки зрения классической лингвистики, так и с точки зрения компьютерной лингвистики) [29]; «Нет единицы более многоликой и неопределенной, чем термин, причем наблюдается несколько подходов к определению термина: одни исследователи пытаются дать ему достаточное логическое определение; другие - стараются описательно раскрыть содержание термина, приписав ему характерные признаки; третьи - выделяют термин путем его противопоставления какой-либо негативной единицы; четвертые ищут противоречивые процедуры выделения терминов, чтобы прийти затем к строгому определению этого понятия; пятые пытаются дать пока хотя бы “рабочее” определение» [30].

Несмотря на размытость границ между подходами к определению термина, описанными в последней цитате, представляется удобным рассмотреть существующие определения в соответствии с этими подходами. Так, ниже приводятся краткий обзор дискуссий о статусе термина, признаки термина, в том числе отличающие его от остальных лексических конструкций, и рабочие определения, в первую очередь — используемые в компьютерной лингвистике.

1.1.1 Дискуссии о статусе термина

К. Мякшин выделяет субстанциональные и функциональные точки зрения на понятие «термин» [27]. Согласно субстанциональной точке зрения, термины являются особыми словами и словосочетаниями, обладающими определенным набором критериев, такими как моносемантичность, независимость от контекста, нейтральность и т. п. Более подробно критерии, или признаки, термина описываются в следующем разделе; заранее можно отметить, что к настоящему времени не существует точного и полного набора критериев.

Приверженцы функциональной точки зрения считают, что «в роли термина может выступать любое слово» и что «термины — это не особые слова, а слова в особой функции» [31]. Данная позиция представляется более логичной, так как накладывает меньше ограничений на термины и тем самым не вычеркивает из рассмотрения наблюдаемые явления переходов терминов в общеупотребительную лексику и обратно (детерминологизации и терминологизации, соответственно), однако такой подход смещает вопрос к определению понятия «функция термина», которое остается дискуссионным среди лингвистов [27].

Несколько с другого угла проблема термина рассматривается в зарубежной лингвистике: основной вопрос, который пытаются решить исследователи, заключается во взаимоотношении между лексической единицей, представляющей собой термин, и понятием, выражаемым термином.

Так, один из основоположников терминоведения Ойген Вюстер считал, что предметные области состоят из наборов понятий, или мыслительных конструкций, а термины служат текстовым представлением этих понятий [32]. Другими словами, по О. Вюстеру, термин представляет собой нечто вроде ярлыка, обозначающего конкретное понятие, то есть имеющего с этим понятием связь один к одному. В этом смысле, термины коренным образом отличаются от обычных слов и функционируют в языке во многом как имена собственные. Сами понятия предметных областей при этом фиксированы и не зависят от контекста употребления.

Гельмут Фелбер также отделяет термин от обозначаемого им понятия [33], однако, по мнению Г. Фелбера, один термин может обозначать несколько понятий, при этом конкретное значение термина, то есть понятие,

зависит от его позиции в системе рассматриваемых понятий. Этим термин отличается от обычных слов, чьи значения полностью определяются контекстом.

В стандарте ISO 1087 Vocabulary of Terminology [34] термин также определяется через обозначаемые понятия: «term: Designation (5.3.1) of a defined concept (3.1) in a special language by a linguistic expression» (Перевод: Термин — это обозначение определенного понятия в специальном языке с помощью лингвистического выражения), где «designation» (обозначение) обозначает «any representation of a concept (1990:5)» (Перевод: любое представление понятия), а «concept» (понятие) обозначает «a unit of thought constituted through abstraction on the basis of properties common to a set of objects (1990:1)» (Перевод: мыслительная единица, образованная путем абстракции на основе свойств, общих для набора объектов). Как справедливо отмечает Дж. Пирсон [35], это определение вряд ли можно назвать адекватным, если сравнить его с определением из этого же стандарта слова «слово»: «word: smallest linguistic unit conveying a specific meaning and capable of existing as a separate unit in a sentence (1990:6)» (Перевод: слово — наименьшая лингвистическая единица, выражающее определенное значение и способное существовать как отдельная единица предложения).

Дж. Сагер отличает термины от слов по обозначаемым им понятиям: термин обозначает понятия, специфические только для одной определенной предметной области [36].

В отличие от вышепроцитированных авторов, в работе Г. Рондо [37] под термином подразумевается комбинация обозначаемого им понятия (notion) и собственно обозначения (dénomination). Рондо также пытается различать термины и остальные слова, однако ограничивается замечанием, что термины используются в специальных предметных областях.

Дженнифер Пирсон, подробно разобрав существующие определения (ее обзор [35] использовался при написании настоящего раздела), приходит к выводу, что эти определения — точнее, попытки отделить термины от общеупотребительных слов — основаны на предположении, что «terms could be recognized intuitively» (Перевод: можно интуитивно распознать термин[1]). Чтобы показать ошибочность этого предположения, выделяются следующие

[1]Судя по контексту, имеется в виду: распознать в тексте

ситуации, называемые «коммуникативными установками», в которых слова могут вести себя как термины:

1. коммуникация эксперта в предметной области с экспертом в этой же предметной области;
2. коммуникация эксперта в предметной области с начинающими специалистами в этой же предметной области;
3. коммуникация относительного эксперта в предметной области с человеком, не связанным с этой предметной областью;
4. коммуникация между учителем и учеником.

Далее Дж. Пирсон показывает, что в первой, второй и четвертой установках использование терминов более вероятно, чем в третьей, а во всех остальных ситуациях невозможно с уверенностью утверждать, что определенное слово, выглядящее как термин, действительно используется в качестве термина.

1.1.2 Признаки термина

Определения терминов через описание характерных признаков — как правило, отличительных по сравнению с общеупотребительной лексикой — представляет особенный интерес в рамках данной работы, поскольку такие признаки могут служить основой для методов автоматического извлечении терминов.

К настоящему времени исследователями сформулировано достаточно большое количество таких признаков. Так, в работе К. Мякшина «К вопросу об основных признаках термина» [38] описываются более 10 признаков. В этой же работе предлагается классификация признаков в соответствии с тремя аспектами термина, предложенными А. Хаютиным [39]: синтаксическим, семантическим и прагматическим.

Ниже приводится описание признаков в соответствии с этой классификацией.

Синтаксические признаки

К данной группе относятся признаки, обусловленные формой термина.

1. Номинативность — «в качестве терминов как специфических языковых единиц обычно рассматриваются имена существительные или построенные на их основе словосочетания» [40].

2. Нормативность — соответствие языковым нормам.

3. Терминологическая инвариантность [39] — отсутствие разнообразия в написании и произношении термина, поскольку это, приводит К. Мякшин аргумент А. Хаютина, «может препятствовать общению специалистов, не говоря уже о том, что формальная разница может стать причиной семантической дифференциации» [38].

4. Мотивированность, или самообъяснимость термина — «максимальное соответствие структуры термина содержательной структуре выражаемого им понятия» [38]. Следует отметить, что некоторые терминологи считают корректным обратный признак, то есть отсутствие выводимости значений термина из его составных частей, однако такая точка зрения менее распространена, поскольку отсутствие мотивированности приводит к отсутствию другого признака термина — системности (см. ниже).

Семантические признаки

К данной группе относятся признаки, обусловленные содержанием термина.

1. Системность — принадлежность термина к определенной терминологии, то есть системе понятий определенной предметной области или отрасли знаний.

2. Соответствие обозначаемому понятию — отсутствие противоречий между лексическим значением слов, из которых состоит термин, и значением термина в данной терминологии (сфере употребления, предметной области);

3. Однозначность, или моносемантичность термина — однозначность термина в данной терминологии (сфере употребления, предметной области). Стоит отметить, что в разных сферах употребления термин может иметь разные значения.

4. Содержательная точность — точность и ограниченность значения термина.

Прагматические признаки

К данной группе относятся признаки, обусловленные спецификой функционирования термина.

1. Внедренность, или общепринятость, или общепонятность, или общепризнанность, или международность — учитывая количество синонимов, определение представляется излишним; стоит отметить только, что многие исследователи считают этот признак «наиболее системно важным критерием».

2. Дефиницированность — поскольку содержательная точность термина (см. выше), как правило, достигается с помощью установления научного определения, само это определение, или дефиниция, может служить признаком термина.

3. Независимость от контекста — данный признак является следствием моносемантичности термина; можно сказать, что контекстом термина, определяющим его значение, служит терминология, членом которой он является.

4. Вариационная устойчивость — воспроизводимость слов и словосочетаний, образующих термин, в текстах данной предметной области, то есть высокая частота термина в этих текстах.

5. Благозвучность — удобство произношения и отсутствие нежелательных ассоциаций.

1.1.3 Рабочие определения термина

Начиная с 1970-х годов, «все большее распространение приобрела точка зрения, согласно которой термин — это слово или словосочетание, номинирующее понятие определенной области познания или деятельности» [27], и это определение стало основой для большинства работ в области извлечения терминов.

Однако это определение нельзя назвать всеобъемлющим — скорее, это «рабочее» определение в терминологии Комаровой, которое также оставляет ряд вопросов. Основной из них: что собой представляет «область познания или деятельности», или «предметная область» (domain), как более распространенный синоним? Определение, предлагаемое в Большом энциклопедическом словаре [41], — «множество всех предметов, свойства которых и отношения между которыми рассматриваются в научной теории» — является во многом рекурсивным относительно определения понятия «термин»: собственно, термины обозначают все те предметы, свойства и отношения, которые и образуют собой множество, называемое предметной областью. Таким образом, в определении «термина» можно заменить «предметную область» на «научную теорию». Однако это значительно сужает область применимости самого понятия «термин»: например, предметная область «Настольные игры» вряд ли можно считать «научной теорией» и таким образом извлекать соответствующие термины.

Отметим, что даже если не пытаться определить понятие «предметная область», посчитав его интуитивным, возникает практический вопрос: как установить (проверить) принадлежность заданного понятия определенной предметной области?

Как правило, в существующих работах по автоматическому извлечению терминов вопрос о принадлежности понятия, обозначаемого термином, к предметной области остается в ведении экспертов соответствующей предметной области. В качестве постановки задачи для экспертов часто пишутся руководства [42, 43], в которых перечисляются наиболее важные признаки терминов и примеры. При этом, поскольку примеры и многие признаки характерны только для заданной предметной области, от нее становится зависимым и само определение термина.

Некоторые исследователи [13] расширяют понятие «принадлежности к предметной области» (domain-specificity) до «релевантности предметной области» (domain-relevancy): в качестве примера приводится термин «medical negligence» (врачебная халатность), который может не принадлежать предметной области «юриспруденция», однако наверняка релевантен ей. Это позволяет уйти от наиболее сложной проблемы — рассмотрения понятий, принадлежащих условной границе предметной области, заведомо считая их правильными терминами.

В других работах [6] вводится понятие «уровень специфичности» термина и акцентируется внимание на терминах «средней специфичности». Авторы ограничиваются несколькими примерами разной специфичности, предполагая ее интуитивную понятность: так, термин «lignite» (бурый уголь) более специфичен, чем термины «natural resources» (природные ресурсы) и «mineral resources» (минеральные ресурсы), которые, в свою очередь, более специфичны, чем термин «resources» (ресурсы); при этом только «natural resources» и «mineral resources» считаются терминами средней специфичности.

Иногда понятие специфичности рассматривается не для терминов, а для предметных областей [42]: например, предметная область «предохранители электрической цепи» (emergency protective circuit arrangements) более специфичная (в статье используется термин «узкая» (narrow)), чем «электричество», которая, в свою очередь, более специфична, чем «технология». Авторы предлагают рассматривать именно эту, наиболее широкую, предметную область.

Рассмотрение только средне-специфичных терминов и широких предметных областей, во-первых, снижает требования к необходимому уровню экспертизы в предметной области и повышает согласованность действий экспертов; во-вторых, что более важно, позволяет повысить эффективность приложений, использующих извлеченные термины, поскольку в разных приложениях требуются термины разной специфичности. Например, для задач поиска экспертов и извлечения ключевых фраз требуются менее специфичные термины, чем для задачи обогащения онтологий.

Таким образом, приложения могут накладывать дополнительные ограничения на термины; другими словами, с практической точки зрения определение термина зависит от приложения, что было отмечено, например, в работе

Г. Бернье-Колборн и П. Дроин [43]. В частности, эта зависимость была подтверждена в следующем эксперименте [44]: четырем группам пользователей (терминологи, эксперты предметной области, переводчики и ученые-информатики) была поставлена задача вручную выделить термины в коллекции документов — в результате получилось четыре списка терминов, значительно отличающиеся друг от друга по количеству и типу терминов.

Учитывая зависимость определения термина от приложения и предметной области, с точки зрения задачи автоматического извлечения терминов возникает противоречие[2]: включение в определение термина особенностей приложения и предметной области априори сужает применимость разработанных методов, а невключение — затрудняет постановку задачи и потенциально снижает эффективность методов.

В данной работе, как и во многих других, указанное противоречие решается в пользу универсальности и в качестве рабочего выбирается все то же определение: термин — это слово или словосочетание, обозначающее определенное понятие заданной предметной области. При этом предметная область определяется входной коллекцией текстов, а принадлежность обозначаемого термином понятия определяется экспертами соответствующей предметной области.

1.2 Сценарии извлечения терминов

Упоминавшаяся в предыдущем разделе зависимость термина от приложения удобно рассмотреть при переходе на более практический уровень — постановку задачи, или сценарий извлечения термина. Явные разделение и формулировка сценариев извлечения терминов также позволят проводить более корректное сравнение существующих методов.

Итак, в зависимости от приложения, для которого требуются термины, можно разбить сценарии извлечения терминов, или постановки задачи, на следующие категории — точнее, типы категорий.

1. По интерпретации вхождений терминов:

[2]Похожее наблюдение, но для специфичности терминов, было сформулировано в работе Дж. Бордо и др. [45]

1) сценарии, рассматривающие (классифицирующие) каждое вхождение термина по отдельности;
2) сценарии, не различающие вхождения одного термина.

2. По способу принятия решения о количестве извлекаемых терминов:
   1) сценарии, извлекающие заданное число терминов;
   2) сценарии, в которых число извлекаемых терминов определяется алгоритмом для каждой входной коллекции.

3. По длине потенциального термина:
   1) сценарии извлечения только однословных терминов;
   2) сценарии извлечения только двухсловных терминов;
   3) сценарии извлечения только многословных терминов;
   4) сценарии извлечения терминов любой длины.

Можно выделить еще множество типов сценариев, однако большинство существующих методов, релевантных данной работе, укладываются в приведенную выше категоризацию.

В данной работе предполагается сценарий извлечения заданного количества терминов любой длины из коллекции документов, не различающий вхождения одного термина.

1.3 Обзор существующих работ

1.3.1 Существующие обзоры и экспериментальные сравнения

Один из первых обзоров [46], посвященных извлечению терминов, анализирует два направления: автоматическое индексирование и собственно извлечение терминов. Основное внимание в обзоре уделяется методом на основе мер ассоциации и модификаций TF-IDF. Авторы одними из первых вводят аспекты термина: «соединенность» (unithood) — связь слов в многословных

терминах; и «терминологичность» (termnhood) — близость термина к предметной области, — и анализируют методы извлечения терминов по тому аспекту, на который опирается метод. Кроме того, в этом обзоре выделяются два класса методов: лингвистические и статистические.

Однако, как отмечает М. Пациенца и др. в обзоре 2005 года [29], современные работы рассматривают лингвистические методы как набор фильтров и явно не проводят разделение на эти классы. В этом обзоре также основной акцент ставится на ассоциативные меры (Dice Factor, z-тест, t-тест, χ^2-тест, MI, MI^2, MI^3, отношение функций правдоподобия), а также наиболее простые методы, пытающиеся определить принадлежность термина предметной области (частота вхождений, C-Value, Co-Occurrence).

Экспериментальное сравнение 2006 года [47] под названием «You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation and term extraction» (Перевод: Нельзя победить частоту, если только не использовать лингвистическую информацию: качественная оценка мер ассоциации для извлечения терминов и коллокаций) показывает, что меры ассоциации, несмотря на теорию математической статистики, лежащую в их основе, работают примерно с такой же эффективностью, как и обычная частота вхождений.

З. Чжан и др. [17] провели экспериментальное сравнение следующих методов, поддерживающих извлечение как однословных, так и многословных терминов: TF-IDF [1], Weirdness [7], C-Value [3], Glossex [48] и TermExtractor [9]. Авторы отмечают, что результаты различаются в зависимости от наборов данных, несмотря на относительную близость предметных областей — биомедицины и зоологии. Кроме того, в этом обзоре показывается превосходство алгоритма голосования (см. подраздел 1.3.6) как метода комбинации отдельных признаков.

П. Браславский и Е. Соколов [49] сравнили четыре метода извлечения двухсловных терминов: частоту вхождений, t-тест, χ^2-тест и отношение функций правдоподобия. Авторы отмечают, что лучшие (при этом сравнимые между собой) результаты показывают первые два метода, и выделяют основной тип ошибок: «выделение устойчивых общеупотребительных словосочетаний, удовлетворяющих шаблонам».

В более поздней работе тех же авторов [50] сравниваются пять методов выделения терминов произвольной структуры: MaxLen [51], C-Value [3], k-factor [52], Window [53], именные группы, выделенные с помощью синтаксического анализатора АОТ [54]. Согласно полученным результатам, «сравниваемые методы дают в целом похожие результаты», при этом авторы все же отмечают, что наибольшую эффективностью показывают методы C-Value и k-factor, в то время как наименьшую — метод на основе синтаксического анализа.

Для оценки эффективности авторы использовали комбинацию экспертной оценки и формальной по предзаданному словарю («эталонному списку») и по результатам такой оценки делают важный вывод: «формальные методы [оценки эффективности] годятся для сравнения больших списков кандидатов в термины».

В работе [19] сравниваются два метода, основанных на комбинации нескольких признаков: алгоритм голосования и метод на основе машинного обучения с учителем (логистическая регрессия и Random forest), — и по итогам экспериментов делается вывод, что второй метод показывает лучшие результаты.

В работе 2013 года Н. Лукашевич и М. Нокеля [21] сравниваются методы извлечения однословных и двухсловных терминов для задачи построения тезауруса для информационного поиска. Авторы используют алгоритм машинного обучения с учителем (Gradient Boosting), анализируют большинство существующих признаков.

По результаты проведенного экспериментального сравнения авторы делают четыре важных вывода:

1) лучшие признаки для извлечения однословных терминов основаны на применении тематических моделей;

2) комбинация нескольких признаков во всех случаях дает существенный прирост эффективности по сравнению с использованием отдельных признаков;

3) признаки на основе внешнего корпуса дают наиболее значительный прирост эффективности для извлечения двухсловных терминов;

4) меры ассоциации не улучшают эффективность.

1.3.2 Общая схема методов извлечения терминов

Для рассматриваемого сценария — извлечения заданного количества терминов любой длины из коллекции документов, не различая вхождения одного термина, — можно выделить общую схему, в которую укладывается большая часть методов. Согласно этой схеме, метод извлечения терминов состоит из трех этапов.

1. **Сбор кандидатов**: фильтрация слов и словосочетаний, извлеченных из коллекции документов, по статистическим и лингвистическим критериям.

2. **Подсчет признаков**: перевод каждого кандидата в вектор признакового пространства.

3. **Вывод на основе признаков**: оценка вероятности быть термином для каждого кандидата на основе значений признаков, последующая сортировка всех кандидатов по этой оценке и взятие заранее определенного числа кандидатов.

Методы сбора кандидатов, в свою очередь, также состоят из нескольких шагов. На первом шаге применяются лингвистические фильтры, цель которых — оставить только существительные и именные группы, то есть словосочетания с существительным в роли главного слов, в соответствии с таким признаком термина, как номинативность (см. подраздел 1.1.2). Для этого применяется либо поверхностный синтаксический разбор (shallow parsing, chunking) [42], либо, более часто [3,19,55,56], фильтрация N-грамм по предопределенным шаблонам частей речи[3].

На последующих шагах сбора кандидатов с целью снижения шума производится дополнительная фильтрация:

1) по частоте: как правило, исключаются из рассмотрения кандидаты с числом вхождений меньше 2 или 3, так как в этом случае становятся неприменимы многие статистические признаки;

[3]Стоит отметить, что сценарии извлечения терминов определенной длины, например только многословных, обычно укладываются в приведенную выше схему, накладывая ограничения на этапе сбора кандидатов в виде порядка учитываемых N-грамм.

2) по содержанию в составе кандидата стоп-слов из заранее составленного списка [56]: многие слова, такие как «хороший» или «интересный», очень редко входят в состав терминов, при этом могут встречаться достаточно часто (например, «хороший метод»);

3) по длине слов кандидата или содержанию в них особых символов [45]: часто исключаются из рассмотрения неалфавитные символы и слова из одной буквы.

Второй этап, вычисление признаков для кандидатов в термины, представляет собой наибольший интерес и подробно разбирается ниже (см. подразделы 1.3.3–1.3.5).

Стоит отметить различие терминов «признак» и «метод»: признак — это отображение кандидата в некоторое число, а метод — это последовательность действий, позволяющая получить ранжированный список кандидатов для заданной коллекции документов, которая включает в себя вычисление одного или нескольких признаков. Тем не менее, на практике эти два термина часто используются взаимозаменяемо, поскольку любой метод может рассматриваться в качестве признака, а большая часть признаков изначально разрабатывалась как отдельные методы; кроме того, под методом может подразумеваться и более общее значение — способ вычисления признака. В данной работе термины «признак» и «метод» также будут использоваться взаимозаменяемо при отсутствии неоднозначности в контексте.

Последний этап подробно рассматривается в подразделе 1.3.6 данного раздела.

1.3.3 Методы на основе статистики вхождений

В данном подразделе собраны методы, учитывающие только частоту вхождений кандидатов в коллекции документов и, может быть, составных частей этих вхождений.

Самым первым, простым и при этом сравнительно эффективным методом можно считать частоту вхождений кандидата в термины — TF. Учитывая очевидность этого признака, определить авторство не представляется возможным.

Также к ранним признакам можно отнести и классический метод из области информационного поиска — TF-IDF:

$$TF \cdot IDF(t) = TF(t) \cdot \log \frac{1}{TF_d(t)}, \tag{1.1}$$

где $TF_d(t)$ — количество документов, в которых встретился кандидат t.

Этот признак показывает высокие значения для терминов, часто встречающихся лишь в малом числе документов. Одно из первых применений для задачи извлечения термина было в работе Д. Эванса и Р. Лефферетса [1].

Интересно отметить, что используется и в некотором смысле противоположный признак — Domain Consensus [2], — предназначенный для распознавания терминов, равномерно распределенных по всей коллекции документов:

$$DomainConsensus(t) = -\sum_{d \in Docs} \frac{TF_d(t)}{TF(t)} \log_2 \frac{TF_d(t)}{TF(t)} \tag{1.2}$$

В отдельную группу под названием «меры ассоциации» (word association measures) выделяют признаки, оценивающие, насколько сильно связаны слова в составе термина (unithood), или насколько случайно эти слова встретились вместе.

Поскольку эти методы применимы только к многословным терминам, причем зачастую только к двухсловным, и при этом в нескольких работах [21,47] было показано, что эти методы не дают прироста эффективности, ограничимся здесь перечислением наиболее распространенных методов из этой группы: z-тест [57], t-тест [58], χ^2-тест, отношение функций правдоподобия [59], взаимная информация (Mutual Information, MI) [60], MI^2, MI^3 [61], Lexical Cohesion [62], Term Cohesion [48].

К методам на основе статистики вхождений относится и наиболее популярный признак — C-Value [3]:

$$C\text{-}Value(t) = \begin{cases} \log_2 |t| \cdot f(t), & \text{если } \{s : t \subset s\} = \emptyset; \\ \log_2 |t| \cdot \left(f(t) - \frac{\sum_s f(s)}{|\{s : t \subset s\}|} \right), & \text{иначе,} \end{cases} \tag{1.3}$$

где t — кандидат в термины, $|t|$ — длина кандидата t (в словах), $f(t)$ — частота вхождений t в коллекции текстов, s — множество кандидатов, объемлющих кандидата t, то есть таких кандидатов, что t является их подстрокой.

В данном признаке вес кандидата уменьшается, если он является частью других кандидатов, поскольку в этом случае частота вхождений кандидата суммируется с частотой вхождения объемлющих кандидатов: например, словосочетание *point arithmetic* (арифметические операции с точкой) имеет не меньшую частоту вхождений, чем термин *floating point arithmetic* (арифметические операции с плавающей точкой), хотя, очевидно, не является термином.

Важно отметить, что признак C-Value предназначен для извлечения только многословных терминов: иначе выражение под логарифмом обнуляет значение признака.

В работе Баррона-Кедено и др. [56] C-Value обобщается на случай однословных терминов путем добавления константы к логарифму:

$$C\text{-}Value(t) = \begin{cases} c(t) \cdot TF(t), & \text{если } \{s : t \subset s\} = \emptyset; \\ c(t) \cdot \left(TF(t) - \frac{\sum_s TF(s)}{|\{s : t \subset s\}|} \right), & \text{иначе,} \end{cases} \tag{1.4}$$

где $c(t) = i + log_2|t|$. Авторы отмечают, что изначально пробовали значение $i = 0.1$, с тем чтобы вносить меньше искажений в исходную формулу, однако в ходе экспериментов обнаружили, что наибольшую эффективность показывает значение $i = 1$.

Вентура и др. [55] предлагают добавлять единицу перед взятием логарифма[4]:

$$C\text{-}Value(t) = \begin{cases} \log_2(|t| + 1) \cdot TF(t), & \text{если } \{s : t \subset s\} = \emptyset; \\ \log_2(|t| + 1) \cdot \left(TF(t) - \frac{\sum_s TF(s)}{|\{s : t \subset s\}|} \right), & \text{иначе.} \end{cases} \tag{1.5}$$

Дж. Бордо и др. [6] предлагают метод Basic[5] — модификацию метода C-Value для извлечения терминов средней специфичности:

[4] Отметим, что это в некотором смысле ближе к сглаживанию по Лапласу.

[5] Название взято из последующей работы [45], в первоначальной работе [6] этот метод назывался Baseline.

$$Basic(t) = |t| \log f(t) + \alpha e_t, \tag{1.6}$$

где e_t — количество кандидатов, содержащих кандидата t.

Так же, как и метод C-Value, Basic применим только для многословных терминов. Однако в отличие от C-Value, в котором учитываемая частота кандидата уменьшается, если он является частью других кандидатов, в данном признаке содержащие его кандидаты, напротив, увеличивают значение признака, поскольку средне-специфичные термины часто служат для образования более специфичных терминов.

В качестве примера авторы приводят термин *information retrieval* (информационный поиск), который может использоваться для создания таких более специфичных терминов, как *information retrieval system* (система информационного поиска), *information retrieval metric* (метрика информационного поиска) и др.

Стоит отметить, что метод Basic представляет собой часть метода Domain Model, см. следующий подраздел

Методы на основе контекстов вхождений

Методы данной группы, в частности NC-Value [3], основаны на предположении, что контексты терминов и обычных слов отличаются. Вслед за Г. Грефенстетт авторы признака NC-Value подразумевают под контекстом существительные, глаголы или прилагательные, непосредственно предшествующие или следующие за вхождением термина.

Вычисление признака состоит из трех этапов. На первом этапе извлекаются 200 лучших терминов с помощью метода C-Value, хотя, как отмечают авторы, можно использовать любой другой метод, в том числе и разметку вручную.

На втором этапе вычисляются веса для слов контекста по формуле:

$$weight(w) = \frac{t(w)}{n}, \tag{1.7}$$

где w — слово контекста (существительное, глагол или прилагательное); $t(w)$ — количество терминов, в контексте которых встретилось w (не путать

с частотой вхождений термина); n — общее количество рассматриваемых терминов.

На третьем этапе вычисляется финальное значение по формуле:

$$NC\text{-}value(t) = 0.8 \cdot C\text{-}Value(t) + 0.2 \sum_{w \in C_t} f_t(w) weight(w), \tag{1.8}$$

где t — кандидат в термины; C_t — множество слов, встречающихся в контексте кандидата t; w — слово из C_t; $f_t(w)$ — частота, с которой слово w встречается в контексте кандидата t.

В работе Дж. Бордо и др. [6] предлагается метод DomainCoherence — модификация метода NC-Value для случая извлечения средне-специфичных терминов.

Авторы вводят следующие ограничения на контекстные слова, называемые «моделью домена»:

1) вхождение не менее чем в четверть документов входной коллекции;

2) принадлежность к существительным, глаголам или прилагательным;

3) семантическая близость ко многим специфичным терминам.

Последнее ограничение по сути представляет собой способ взвешивания; в отличие от простого подсчета соотношения терминов, перед которыми или после которых встретилось слово, применяемого в NC-Value, в методе DomainCoherence предлагается использовать метрику PMI:

$$s(w) = \sum_{t \in T} PMI(t, w) = \sum_{w \in W} log \left(\frac{P(t, w)}{P(t)P(w)} \right), \tag{1.9}$$

где w — слово, рассматриваемое в качестве кандидата в модель домена; T — множество 200 лучших терминов, извлеченных с помощью метода Basic (см. предыдущий подраздел); $P(t, w)$ — вероятность появления слова w в контексте термина t; $P(t)$ и $P(w)$ — вероятности появления термина t и слова w, соответственно. Указанные вероятности оцениваются на основе частот вхождения во входной коллекции документов; в качестве контекста рассматривается окно в 5 слов.

Для вычисления финального значения признака DomainCoherence также применяется метрика PMI, вычисляемая между каждым кандидатом в термины (t) и словом из модели домена (w).

Также авторы показывают, что в ходе экспериментального исследования лучшие результаты продемонстрировала линейная комбинация признаков Basic и DomainCoherence, названная в работе PostRankDC (в настоящей диссертационной работе будет использоваться название DomainModel, поскольку это ближе к названию статьи, где был введен этот метод, и к составляющим признакам).

Методы на основе тематических моделей

Вследствие развития методов тематического моделирования — кластеризации слов по темам, а тем — по документам, — за последние годы появилось несколько признаков для извлечения терминов на его основе. Как отмечают Н. Лукашевич и М. Нокель [21], большая часть признаков является модификацией обычных методов, в которых вместо частоты вхождений используется вероятностное распределение по темам слов, представляющих собой кандидаты в термины. Из этого, в частности, следует, что эти методы применимы только для извлечения однословных или (реже) двухсловных терминов.

К таким признакам относятся, например, Term Score, изначально разработанный для визуализации тем [63] и примененный Большаковой и др. [4] для задачи извлечения терминов, а также Term Frequency (частота вхождений), максимальная частота вхождений (Maximum Term Frequency), TF-IDF и Domain Consensus [4].

С. Ли и др. [5] предлагают метод под названием Novel Topic Model, позволяющий извлекать термины любой длины. Для его вычисления необходимо с помощью тематического моделирования получить распределения слов по следующим темам:

- ϕ^t - конкретные темы в предметной области ($1 \leq t \leq T$, авторы используют $T = 20$);
- ϕ^B - общая тема предметной области (background topic);
- ϕ^D - тема, специфичная для документа.

После чего для каждой темы извлекаются 200 наиболее вероятных слов: V_t, V_B, V_D, соответственно, — и для каждого кандидата c_i, состоящего из L_i слов $(w_{i1}w_{i2}...w_{iL_i})$, его весом считается сумма максимальных вероятностей составляющих его слов (из полученных распределений):

$$NTM(c_i) = \log(tf_i) \cdot \sum_{1 \le j \le L_i, w_j \in \cup\{V_t\}_{t \in T \cup \{B,D\}}} \phi_{w_j}^{mt_{w_j}}, \tag{1.10}$$

где

$$mt_{w_j} = \arg\max_{t \in T \cup \{B,D\}} \phi_{w_j}^{t}$$

1.3.4 Методы на основе внешних ресурсов

Методы на основе внешних корпусов

Методы данной группы основаны на наблюдении, что термины предметной области встречаются в текстах этой предметной области намного чаще, чем в текстах других предметных областей, в частности — текстах так называемой общей предметной области (general domain) или текстах, не принадлежащих какой-либо предметной области.

В качестве таковых текстов, называемых обычно внешним или опорным корпусом (reference corpus), используются коллекции текстов других предметных областей [8,9], наборы электронных книг и журналов [6], коллекции новостей [64] или созданные лингвистами корпуса, такие как Open American National Corpus[6] (OANC) [6] или British National Corpus[7] (BNC) [7].

Один из наиболее простых способов учесть сформулированное выше наблюдение заключается в модификации метода TF-IDF [65], называемого иногда [21] TF-RIDF: при подсчете IDF (RIDF) — количества документов, где встретился термин, — вместо коллекции предметной области используется внешний корпус.

На не менее простой формуле основан признак Domain Pertinence [8]:

$$DR(t) = \frac{TF_{target}(t)}{TF_{reference}(t)}, \tag{1.11}$$

[6] http://www.americannationalcorpus.org/
[7] http://www.natcorp.ox.ac.uk/

где $TF_{target}(t)$ — частота вхождений кандидата t во входной коллекции текстов предметной области; $TF_{general}$ — частота во внешнем корпусе.

Похожую формулу использует и признак Domain Relevance [9]:

$$DR(t) = \frac{TF_{target}(t)}{TF_{target}(t) + TF_{reference}(t)} \tag{1.12}$$

Признак Weirdness [7] дополнительно учитывает размер коллекций:

$$W(t) = \frac{TF_{target}(t) \cdot |Corpus_{reference}|}{TF_{reference}(t) \cdot |Corpus_{target}|} \tag{1.13}$$

Признак Relevance [64] пытается уменьшить вес кандидатов, которые редко встречаются в документах предметной области, либо встречаются в очень небольшом проценте документов предметной области, либо часто встречаются во внешнем корпусе:

$$Relevance(t) = 1 - \left(\log_2\left(2 + \frac{TF_{target}(t) \cdot DF_{target}(t)}{TF_{reference}(t)}\right)\right)^{-1} \tag{1.14}$$

В признаке Domain Specificity, являющемся частью системы GlossEx [48], учитываются частоты отдельных слов, из которых состоит термин:

$$DomainSpecificity(t) = \frac{\sum_{w_i \in t} log\frac{P_d(w_i)}{P_c(w_i)}}{|t|}, \tag{1.15}$$

где $|t|$ — число слов в кандидате t; $P_d(w_i)$ — вероятность появления слова w_i, являющегося частью кандидата t, в коллекции текстов предметной области; $P_c(w_i)$ — вероятность появления слова во внешней коллекции текстов. Как и в остальных методах, вероятность оценивается как число вхождений, нормализованное на размер коллекции в словах.

Методы на основе поисковых машин

Отдельную группу образуют методы, использующие поисковые машины. Так, в работе П. Браславского и Е. Соколова [10] для извлечения двухсловных терминов применялось несколько признаков: iFreq, TF-IDF, freq/iFreq и coherence. Авторы отмечают, что эти методы работают не для каждой предметной области, и выдвигают следующую гипотезу: «метод скорее всего

будет работать для областей со специфичной терминологией (той, которая по большей части отлична от общеупотребительных выражений, редко использует выражения для универсальных понятий), к тому же не очень широко представленных в Сети».

Д. Голомазов [11] для фильтрации двухсловных терминов формирует следующие запросы к поисковым машинам: "A" (собственно термин), "A is a term", "A is a concept", "A_1", "A_2", "A_1 AND A_2", где A_1 и A_2 — слова, из которых состоит термин A.

Далее автор требует выполнения хотя бы одного из следующих условий[8], чтобы термин прошел фильтрацию:

1) $\frac{hits(\text{“}A \text{ is a term}'')}{hits(A)} > C_1,$

2) $\frac{hits(\text{“}A \text{ is a concept}'')}{hits(A)} > C_2,$

3) $\frac{hits(\text{“}A_1 \text{ AND } A_2'')}{min(hits(A_1),hits(A_2))} > C_3,$

где $hits(A)$ — количество страниц, возвращенных поисковой машиной на запрос A; $C_1, C_2, C_3 \in [0, 1]$ — параметры алгоритма.

В работе Б. Доброва и Н. Лукашевич [12] также рассматривается извлечение только двухсловных терминов с применением поисковых машин. Однако вместо частот встречаемости кандидатов в Вебе авторы активно используют поисковые сниппеты, полученные после запроса, состоящего из всего кандидата и отдельных слов кандидата. В частности, анализируются частоты кандидатов в сниппетах (FreqBySnip); количество в сниппетах предопределенных слов, характерных для предметной области (Markers); количество в сниппетах слов, часто встречающихся в словарных определениях (NearDefWords); близость сниппетов, полученных для всего кандидата и для отдельных слов (Scalar Features) и т.п.

В этой работе также рассматриваются и другие типы признаков и на основе экспериментального исследования делается вывод, что максимальная эффективность достигается при использовании всех типов признаков.

[8]Строго говоря, возможно выполнение и еще одного условия, описанного в подразделе 1.3.5

Методы на основе онтологий

Для задачи извлечения терминов онтологии используются реже, чем другие внешние ресурсы, поскольку общие онтологии слабо покрывают предметные области, ограничиваясь наиболее общими же терминами, в то время как специальные онтологии существуют лишь для очень ограниченного набора предметных областей, при этом формат и структура таких онтологий часто зависит от конкретной предметной области.

Так, большинство работ, посвященных обогащению онтологии, сосредотачивают внимание на извлечении отношений между понятиями, никак не используя на этапе извлечения терминов информацию, которая содержится в обогащаемой онтологии. Например, в работе К. Мейер и др. [8] применяются описанные выше Domain Consensus, Domain Pertinence, Lexical Cohesion; Ф. Ксю и др. [66] извлекают термины с помощью TF-IDF и различных ассоциативных мер (MI, t-тест и отношение функций правдоподобия).

Несколько выделяется уже упоминавшаяся в этом подразделе работа Б. Доброва и Н. Лукашевич [12], в которой используется существующая онтология предметной области (точнее, тезаурус для информационного поиска). Однако предлагаемые признаки применимы только к двухсловным терминам: бинарный признак SynTerm равен 1 в том и только том случае, если для каждого слова, из которого состоит термин, в тезаурусе присутствует синоним; признак Completeness суммирует число синонимов и отношений для дескрипторов, которые также ищутся в тезаурусе для отдельных слов термина.

1.3.5 Методы на основе Википедии

Как отмечалось выше, многоязычная интернет-энциклопедия Википедия обладает уникальными качествами, которые могут быть полезными для задачи извлечения терминов: ее статьи описывают как универсальные понятия, так и специфичные для узких предметных областей, причем их покрытие постоянно увеличивается; Википедия содержит структурную информацию в виде гиперссылок между статьями; обладает очень большим размером и ежедневно пополняется сообществом пользователей.

Так, в работе Д. Милна и др. [67] приводится сравнительный анализ покрытия предметной области «сельское хозяйство» Википедией и созданным экспертами тезаурусом Agrovoc[9]. Авторы показывают, что около половины всех терминов тезауруса Agrovoc содержится в Википедии, причем к ним относятся наиболее часто употребимые. Отметим, что за прошедшие 8 лет Википедия значительно выросла в размере: на момент написания настоящей диссертационной работы число статей англоязычной версии превышает 4,6 млн; в 2006 году это число составляло около 1.1 млн, — и есть основания полагать, что на текущий момент покрытие намного больше.

Однако несмотря на широкое использование Википедии для различных задач извлечения знаний [68–72], к настоящему времени разработано не так много методов извлечения терминов предметной области на основе Википедии.

В работе Д. Голомазова [11] Википедия используется лишь на этапе фильтрации терминов: если для термина есть описывающая его статья в Википедии, то термин проходит фильтрацию (см. подраздел 1.3.4).

Работа В. Ву и др. [13] также не укладывается в рассматриваемый сценарий, поскольку предполагает извлечение терминов исключительно из Википедии, а не коллекции текстов предметной области, и требует задать вручную несколько концептов (статей Википедии) в качестве положительных примеров терминов предметной области.

Более точно, авторы строят взвешенный граф, в котором узлами являются статьи и категории Википедии, а ребрами — гиперссылки между ними, после чего, используя заданные вручную концепты, применяют алгоритм случайного блуждания (Random walk) к указанному графу. Вес, назначенный каждому концепту в результате алгоритма, считается оценкой того, что соответствующий концепт выражается термином предметной области.

Дж. Вивальди и др. в серии статей [14–16] предлагают метод, соответствующий рассматриваемому сценарию: из коллекции документов предметной области извлекаются кандидаты в термины, которые затем оцениваются с помощью алгоритмов поиска путей в графе категорий Википедии.

Как и в предыдущем методе, авторы требуют задания дополнительной входной информации, а именно: одну или несколько категорий Википедии,

[9] http://aims.fao.org/agrovoc

называемых «границами предметной области» (domain borders), которые наиболее точно и полно описывают желаемую предметную область.

Алгоритм оценки кандидатов в термины состоит в следующем. Для каждого кандидата определяются все его концепты (статьи Википедии с таким названием — в общем случае может быть несколько статей для одного кандидата по причине лексической многозначности), после чего для каждого статьи определяются все категории, которым она принадлежит. Из всех полученных оценок для каждой из статей в конечном итоге выбирается лучшая.

Далее, для каждой категории запускается рекурсивный обход графа категорий (следуя только по ссылкам в категории верхнего уровня), до тех пор пока не будет достигнута заданная граница предметной области либо категория самого верхнего уровня. Свойства полученных путей в конечном итоге используются для получения оценки кандидата одним из нижеприведенных способов.

1. Количество путей (NC):

$$NC(t) = \frac{NP_{domain}(t)}{NP_{total}(t)}, \tag{1.16}$$

   где $NP_{domain}(t)$ — количество путей от категорий кандидата до границы домена; $NP_{total}(t)$ — количество путей от категорий кандидата до категории верхнего уровня.

2. Длина путей (LC):

$$LC(t) = \frac{LP_{total}(t) - LP_{domain}(t)}{LP_{total}(t)}, \tag{1.17}$$

   где $LP_{domain}(t)$ — длина путей (суммарная) от категорий кандидата до границы домена; $LP_{total}(t)$ — длина путей (суммарная) от категорий кандидата до категории верхнего уровня.

3. Средняя длина путей (LMC):

$$LC(t) = \frac{ALP_{total}(t) - ALP_{domain}(t)}{ALP_{total}(t)}, \tag{1.18}$$

где $ALP_{domain}(t)$ — средняя длина путей от категорий кандидата до границы домена; $ALP_{total}(t)$ — средняя длина путей от категорий кандидата до категории верхнего уровня.

В ходе экспериментального исследования, проводимого для одно- и двухсловных терминов, авторы показывают, что наибольшей эффективностью обладает метод на основе количества путей (NC).

1.3.6 Методы вывода на основе признаков

Последний этап, вывод на основе признаков, в случае нескольких признаков реализуется одним из следующих способов.

1. Линейная комбинация признаков с вручную подобранными коэффициентами (как правило, равными) [6, 9, 62].

2. Алгоритм голосования, предложенный в работе З. Чжана и др. [17]:

$$V(t) = \sum_{i}^{n} \frac{1}{rank(F_i(t))}, \tag{1.19}$$

   где n — число признаков, $rank(F_i(t))$ — порядковый номер кандидата t среди всех кандидатов, отсортированных по значению признака F_i.

3. Машинное обучение с учителем: на основе вручную размеченных данных строится модель классификатора. В случае сценария, позволяющего задавать число извлекаемых терминов, используется алгоритмы обучения, поддерживающие вероятностную классификацию, то есть возвращающие не бинарный ответ для каждого кандидата, а вероятность принадлежности к классу.

   Среди используемых алгоритмов обучения с учителем можно выделить Ada Boost [18], логистическую регрессию [12, 19, 20], Random forest [19], Gradient Boosting [21] и другие.

Отдельного рассмотрения заслуживают методы, основанные на алгоритмах машинного обучения с учителем, но не требующие размеченных данных. Так, в работе И. Янга и др. [73] предлагается метод, названный Fault-Tolerant

Learning и представляющий собой комбинацию бутсреппинга и алгоритмов совместного обучения (co-training).

Авторы выделяют два набора признаков: классический TF-IDF и признаки, основанные на использовании слов-разделителей, специфичных для китайского языка. С помощью каждого набора признаков все кандидаты сортируются — получается два списка (точнее, две сортировки) кандидатов, состоящие из одних и тех же элементов. Из каждого списка извлекается по 500 лучших и 500 худших кандидатов, которые рассматриваются как положительные и отрицательные примеры, соответственно, для последующего обучения с учителем. Алгоритмы обучения с учителем представляют собой метод опорных векторов (SVM) с 5 признаками:

1) частота вхождений кандидата;
2) части речи слов кандидата;
3) слова-разделители из контекстов вхождений кандидата;
4) первое слово кандидата;
5) последнее слово кандидата.

Обученные классификаторы затем применяются ко всем кандидатам в термины; кандидаты с наибольшей и наименьшей оценкой используются снова в качестве положительных и отрицательных примеров для следующей итерации обучения. При этом для предотвращения деградации процесса используется так называемая верификация обучающих наборов: в случае разных меток (термин и нетермин), присвоенных двумя классификаторами одному и тому же кандидату, этот кандидат исключается из обучающего набора.

В работе «Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents» (Перевод: Не требующая размеченных данных генерация обучающего множества для автоматического извлечения технической терминологии в патентах) [42] предлагается метод извлечения терминов, классифицирующий каждое вхождение кандидата в термины отдельно, однако принципиальная схема может быть использована и для сценария, рассматриваемого в настоящей диссертационной работе.

Авторы применяют следующие эвристики для выделения положительных и отрицательных примеров: термином считается слово или словосочетание,

непосредственно предшествующее ссылке на иллюстрацию в тексте патента; нетерминами считаются слова и словосочетания, встретившиеся в патентах лишь один раз, либо являющиеся цитатами или единицами измерения.

Получив таким образом набор положительных и отрицательных примеров, авторы используют алгоритм обучения с учителем (логистическую регрессию и Conditional random fields) с 74 признаками, среди которых части речи кандидатов, контексты и статистики вхождений, а также признаки на основе строковых метрик.

Результаты экспериментальной оценки весьма высоки (F1-мера более 75%), однако методика оценки значительно отличается от принятой в рассматриваемом сценарии: в частности, один и тот же кандидат в термины, имеющий очень много вхождений, даст больший вклад в точность и полноту, чем несколько кандидатов с небольшим числом вхождений, притом что первые кандидаты обычно проще классифицировать верно.

Другой существенный недостаток заключается в невозможности переноса на другую предметную область и другие языки по причине используемой эвристики для выделения положительных примеров.

1.4 Методы оценки эффективности

Как отмечают Г. Бернье-Колборн и П. Дроин [43], вопрос об оценке систем извлечения терминов остается нерешенным: оценки эффективности регулярно публикуются в соответствующих работах, однако методология отличается от работы к работе, затрудняя таким образом какое-либо сравнение.

Можно выделить два принципиальных подхода к оценке эффективности методов извлечения терминов:

1) оценка результатов работы метода вручную с помощью экспертов предметной области (например, [74]);

2) использование «золотого стандарта» — заранее созданного списка терминов-эталонов («формальная оценка», в терминологии П. Браславского и Е. Соколова [49, 50]).

Достоинства и недостатки каждого подхода очевидны: первый позволяет производить наиболее точную оценку, в то время как второй подход обес-

печивает повторяемость результатов и возможность настраивать параметры или сравнивать разные методы на одном наборе данных. При этом, как уже было отмечено выше, результаты сравнения методов с помощью этих подходов согласуются между собой в случае достаточно больших списков терминов [10, 49, 50].

Второй подход также можно разделить на несколько методов оценки, в зависимости от способа получения списка терминов-эталонов:

1) разметка всех документов вручную (например, [42]);

2) разметка небольшой части документов вручную (например, [20]);

3) адаптация существующих ресурсов к задаче извлечения терминологии (например, [12, 45]).

Первый метод наиболее точный, однако и наиболее ресурсоемкий при большом числе документов (а малое число искажает работу статистических признаков).

Второй метод позволяет вычислять признаки на основе всех документов, а оценивать эффективность только для тех терминов, которые встречатся в размеченных документах.

Применимость третьего метода может зависеть от рассматриваемой предметной области и приложения. Так, для некоторых предметных областей существуют созданные вручную тезаурусы или словари, которые могут быть использованы в качестве золотого стандарта [12, 50]. Иногда термины аппроксимируются ключевыми фразами или индексными терминами: например, в работе Дж. Бордо и др. [45] в качестве терминов-эталонов для коллекции статей одной научной области используется объединение множеств ключевых слов для каждой статьи. Также для этой задачи используются предметные указатели книг [10, 49]

Что касается метрик эффективности, то для рассматриваемого сценария — извлечения заданного количества терминов любой длины из коллекции документов, не различая вхождения одного термина, — обычно используются следующие метрики.

1. Точность (precision), называемая иногда также «точность на уровне N»:

$$P(N) = \frac{|Correct \cap Retrieved[1:N]|}{N}, \tag{1.20}$$

где N — количество учитываемых лучших кандидатов; $Correct$ — множество терминов-эталонов; $Retrieved[1:N]$ — множество лучших N кандидатов в термины в соответствии с весами, назначенными оцениваемым методом.

2. Полнота (recall):

$$R(N) = \frac{|Correct \cap Retrieved[1:N]|}{|Correct|} \tag{1.21}$$

3. Средняя точность (average precision):

$$AvP(N) = \sum_{i=1}^{N} P(i)(R(i) - R(i-1)) \tag{1.22}$$

Стоит отметить, что на практике полнота обычно явно не оценивается, поскольку фактически определяется заданным количеством извлекаемых терминов и точностью, и что наиболее популярной метрикой в настоящее время является средняя точность, так как представляет собой интегральную оценку по множеству значений N.

Кроме того, в некоторых работах явно исследуется зависимость точности от N [45] и даже средней точности от N [19, 21].

1.5 Выводы

В данной главе приводится обзор существующих определений базовых понятий для задачи извлечения терминов, конкретизируется сама постановка задачи и анализируются существующие методы ее решения.

Среди основных проблем, связанных с рассматриваемой задачей, можно выделить следующие:

1. Отсутствует общепринятые определения термина и предметной области; многочисленные предпринимаемые попытки сформулировать опре-

деления привели к созданию множества несогласованных, а иногда и вовсе противоречивых утверждений относительного одного и того же понятия. Определения, часто выбираемые в качестве рабочих, являются достаточно неформальными.

2. Как следствие предыдущего пункта, постановка задачи извлечения терминов также далека от полностью формальной. Это, в свою очередь, существенно затрудняет оценку эффективности и сравнение разработанных методов — в итоге на настоящий момент нет общепринятых наборов данных и методологии оценки эффективности.

3. Разработанные методы часто зависят от предметной области, языка, приложения и т.п., что затрудняет или делает невозможным перенос методов на другие предметные области, языки, приложения.

4. Существующие методы недостаточно полным образом используют Википедию. Как правило, в качестве возможных терминов они рассматривают только названия существующих статей Википедии, что заведомо ограничивает полноту извлечения терминов. В частности, многие методы опираются только на информацию Википедии, например структуру категорий, не используя коллекцию документов предметной области.

В следующей главе описываются методы извлечения терминов на основе Википедии, использующие коллекцию документов предметной области и не требующие задания вручную дополнительной информации.

Глава 2

Методы извлечения терминов на основе Википедии

Википедия[1] — «свободная общедоступная мультиязычная универсальная интернет-энциклопедия»[2]. Википедия развивается сообществом пользователей и на момент написания данной работы содержит более 30 миллионов статей, в том числе более 4 млн статей на английском языке и более 1 млн статей на русском языке.

Как и в любой энциклопедии, каждая статья Википедии описывает определенное понятие, или концепт. Более точно, статья представляет собой «озаглавленный связный текст из основного пространства имён, содержание которого отражает одно значение его заголовка»[3].

Важным отличием интернет-энциклопедий, в частности Википедии, от обычных энциклопедий является наличие структуры гиперссылок: любая статья Википедии может быть связана гиперссылкой с любой другой статьей. При этом гиперссылки также создаются пользователями на основе «Руководства по расстановке гиперссылок»[4], согласно которому следует ставить гиперссылки на статьи с релевантным содержимым.

[1] https://www.wikipedia.org/
[2] https://ru.wikipedia.org/wiki/Википедия
[3] https://ru.wikipedia.org/wiki/Википедия:Статья
[4] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking

Учитывая этот факт, становится возможным определить на основе структуры гиперссылок семантическую близость (semantic relatedness)[5] — функцию, определенную для любой пары концептов и имеющую значения от 0 до 1: чем ближе значение функции к 1, тем больше общего между концептами.

В данной главе описываются два новых метода для извлечения терминов, использующие гиперссылки Википедии. Первый метод основан на статистике употреблений гиперссылок в текстах статей Википедии; второй метод — на описанной выше функции семантической близости. В завершении главы приводится описание экспериментального исследования разработанных методов.

2.1 Метод «Вероятность быть гиперссылкой»

Рассмотрим подробнее упоминавшееся выше руководство по расстановке гиперссылок.

Согласно ему, гиперссылки следует ставить на следующие статьи:

1) содержащие релевантную информацию, которая может помочь читателям более полным образом понять статью, с которой ведет гиперссылка;

2) объясняющие технические термины, а также жаргонные или сленговые выражения;

3) представляющие собой имена собственные, которые с большой вероятностью незнакомы читателю.

Также согласно этому руководство, гиперссылками не следует делать следующие слова (если только они не являются особенно релевантными теме статьи, с которой ведет гиперссылка):

1) повседневные слова, понятные большинству читателей из контекста;

[5] В некоторых работах термин *semantic relatedness* переводится как «семантическая связность» (см. например, [75]) и противопоставляется термину *semantic similarity* (данная мера, как правило, ограничивается рассмотрением отношений гипонимии [68,76]), который и предлагается переводить как «семантическая близость». Однако, во-первых, такой подход не является общепринятым (см. например, [72]); во-вторых, слово «близость» представляется более подходящим переводом для слова *relatedness* (буквальный перевод: *родство*), чем для слова *similarity* (буквальный перевод: *похожесть*); в-третьих, термин «семантическая связность» может быть неверно соотнесен с термином *semantic connectivity* (см. например, [77]).

2) названия основных географических объектов, языков, религий, профессий;

3) распространенные единицы измерений, например времени, температуры, длины, площади или объема;

4) даты.

На основе приведенных пунктов руководства можно сделать предположение, что слова и словосочетания, выделенные как гиперссылки, чаще являются терминами (различных предметных областей). В пользу этого предположения прямо свидетельствуют пункт 1 из второго списка и пункт 2 из первого списка, если учесть, что жаргонные и сленговые выражения часто являются прообразами или синонимами терминов. Остальные пункты, как минимум, не противоречат данному предположению: имена собственные, даты и названия основных географических объектов, языков, религий, профессий не рассматриваются в качестве кандидатов в термины, а распространенные единицы измерений редко являются терминами какой-либо предметной области и в любом случае, их общее число весьма невелико.

Формализуем данное предположение, введя следующую функцию:

$$H(t) = \frac{N_{hc}(t)}{N_{wp}(t)},$$

где t — слово или словосочетание, $N_{hc}(t)$ показывает, сколько раз t встречалось в статьях Википедии в виде названия гиперссылки (hyperlink caption), $N_{wp}(t)$ — сколько раз t встречалось в статьях Википедии всего.

Следует отметить, что аналогичная функция под разными названиями использовалась в нескольких исследованиях, посвященных обработке текстов на основе Википедии. Так, в работе Р. Михалци и А. Цомай [69] относительная частота употреблений слова или словосочетания как гиперссылки обозначалась «Keyphraseness» и применялась для ранжирования ключевых фраз. Д. Милн и Я. Виттен [71] использовали эту функцию, называя ее Link Probability, как один из признаков в задаче разрешения лексической многозначности. Д. Турдаков [72], используя термин «Информативность», применял функцию $H(t)$ в смежной задаче — для определения слов и словосочетаний, не имеющих подходящего значения, то есть статьи Википедии.

На основе функции $H(t)$ можно ввести признак для автоматического извлечения терминов — «Вероятность быть гиперссылкой» (LinkProbability):

$$LinkProb_T(t) = \begin{cases} 0, & \text{если } t \text{ не содержится в Википедии или } H(t) < T; \\ H(t), & \text{иначе,} \end{cases} \tag{2.1}$$

где t — кандидат в термины (слово или словосочетание, прошедшее фильтрацию по частям речи, частоте и списку стоп-слов), $H(t)$ определена выше; T — параметр метода.

Значение этого признака будет ближе к нулю для слов и словосочетаний, являющихся частью общей лексики, то есть не принадлежащих какой-либо предметной области. Таким образом, мотивация использования этого признака заключается в фильтрации таких слов и словосочетаний, поскольку они, скорее всего, не принадлежат и к предметной области, для которой извлекаются термины.

Предположим, что значение параметра T равно нулю, и рассмотрим возможные значения признака для нескольких примеров. Так, слово *Gene* (ген) встречается 85972 раза в статьях Википедии и 14569 раза в виде гиперссылки на статью, описывающую соответствующее понятие. Таким образом, значение признака составит 0.16946. Для словосочетания *Last card* (последняя карта) значение признака равняется 0.012 (332 раза в статьях и всего лишь 4 раза в виде гиперссылки на статью про карточную игру с одноименным названием). Слово *Size* (размер) встречается всего 261607 раз, из которых 58 раз в виде гиперссылки, что приводит к значению 0,0002. Таким образом, при прочих равных условиях вероятность быть отнесенным к терминам для слово *Gene* выше, чем для словосочетания *Last card*, которое, в свою очередь, является более вероятным термином, чем *Size*.

Введение положительного параметра T обусловлено двумя причинами. Во-первых, пользователи иногда нарушают требования руководства по расстановке гиперссылок или ошибаются при наборе или разметке текста, например добавляют лишнее слово, пропускают нужное или просто допускают опечатку в заголовке гиперссылки. Для фильтрации шума такого рода следует игнорировать очень редкие гиперссылки — например, Р. Михалси и

А. Цомай [69] учитывают только те гиперссылки, которые встретились не менее 5 раз.

Во-вторых, это позволяет более корректно учитывать термины, которые пока еще отсутствуют в Википедии: как было показано выше, слова общей лексики могут иметь очень малые, но все же положительные значения функции $H(t)$, в то время как слова и словосочетания, новые с точки зрения Википедии — например, *electrically driven transmission* (электроприводная передача) или *2-player game* (игра для двух игроков) — будут иметь нулевые значения $H(t)$, хотя при этом могут являться более вероятными терминами.

Обнуление признака для очень малых значений становится особенно важно в случае комбинации данного признака с другими, так как упрощает задачу по борьбе с шумом методу, основанному на этих признаках.

Значение параметра T подбирается экспериментально, см. раздел 2.3.2.

2.2 Метод «Близость к ключевым концептам»

Метод «Близость к ключевым концептам» (KeyConceptRelatedness) основан на следующей интерпретации определения термина: «Термин — слово или словосочетание, обозначающее *понятие*, которое *принадлежит* заданной *предметной области*», где:

- «понятие» интерпретируется как концепт, присутствующий в Википедии в виде статьи;
- «предметная область» — набор близких по смыслу понятий;
- «принадлежность понятия к предметной области» — близость по смыслу понятия к предметной области;
- «близость по смыслу» — семантическая близость, которая определена выше (см. введение главы 2).

Кроме того, фраза «обозначающее понятие» интерпретируется как «обозначающее хотя бы одно понятие»: это позволяет решить проблему лексической многозначности кандидата в термины, то есть ситуации, когда кандидат в термины может иметь несколько значений (концептов Википедии),

путем выбора среди всех понятий термина то, которое окажется максимально близко к понятиям предметной области. Заметим, что такая интерпретация применима только для сценария, не различающего различные вхождения кандидатов в термины (см. раздел 1.2).

Для практического применения приведенной выше интерпретации необходимо решить две задачи. Во-первых, определить концепты, образующие предметную область в смысле приведенной выше интерпретации. Во-вторых, определить семантическую близость между концептом, который обозначает термин, и концептами, образующими предметную область. Эти задачи подробно описаны в следующих подразделах.

В конце раздела приводится подробное описание алгоритма предлагаемого метода.

2.2.1 Определение концептов предметной области

Само возникновение задачи извлечения терминов подразумевает отсутствие полного множества концептов, образующих собой предметную область в смысле приведенной выше интерпретации. Идея метода основана на предположении, что для вычисления близости достаточно использовать некоторую аппроксимацию полного множества, то есть представительное подмножество множества всех концептов предметной области.

Учитывая тот факт, что единственным источником информации о предметной области служит входная коллекция текстовых документов, предлагается использовать в качестве такого представительного подмножества ключевые концепты, извлеченные из документов входной коллекции.

Более точно, предлагается следующий эвристический алгоритм.

1. Извлечь d ключевых концептов из каждого документа коллекции.

2. Посчитать *встречаемость* ключевых концептов: сколько раз в коллекции каждый концепт попал в число лучших d ключевых концептов документа.

3. Взять N ключевых концептов с наибольшей встречаемостью.

Для извлечения ключевых концептов из документа используется метод, описанный в работе М. Гриневой и др. [78]. Данный метод состоит в из-

влечении всех концептов из документа (с разрешением лексической многозначности), построении семантического графа этих концептов, его кластеризации, ранжировании полученных кластеров и выборе концептов из лучших кластеров.

Важно отметить, что метод извлечения терминов предъявляет дополнительное требование к извлеченным ключевым концептам: эти концепты должны быть характерными для основной предметной области заданной коллекции, но не для посторонних предметных областей, упоминания которых могут встречаться в документах в виде второстепенных тем. Алгоритм извлечения ключевых концептов, однако, должен извлекать концепты, характерные для всего документа, а не для одной его темы, пусть даже и основной.

Решение этого противоречия заключается, во-первых, в усреднении по всей коллекции в предположении, что второстепенные темы, специфичные для какого-либо документа, не затрагиваются в большом числе документов; во-вторых, в выборе алгоритма извлечения ключевых концептов, который предпочитает концепты из основных тем (то есть кластеров семантического графа) документа [78].

2.2.2 Вычисление семантической близости

Задача вычисления семантической близости на основе Википедии широко исследована за последние годы [70, 79, 80]. В частности, в работе Д. Турдакова и П. Велихова [80] показано, что для задачи разрешения лексической многозначности на основе Википедии наибольшая эффективность достигается при вычислении семантической близости по взвешенной мере Дайса, где соседними считаются статьи, связанные хотя бы одной гиперссылкой, а вес задается типом этой ссылки: например, ссылка из основного текста статьи, ссылка из секции «См. также» (See also) и т.п.

Напомним, что семантическая близость представляет собой функцию, определенную для пары концептов. В то же время для метода извлечения терминов, как было отмечено выше, необходимо получить для каждого концепта потенциального термина одно число, характеризующее близость этого концепта ко всем концептам предметной области.

Возникает следующая задача: имея попарные значения семантической близости между концептом потенциального термина и каждым концептом

предметной области, получить агрегатную оценку, отражающую принадлежность концепта потенциального термина предметной области.

Очевидные решения — усреднить попарные значения или взять максимальное — обладают очевидными недостатками. Поясним их на примере: допустим, для предметной области «Настольные игры» были извлечены следующие ключевые концепты: Board game (настольная игра), Playing Card (игральная карта), Poker (покер), Solitaire (пасьянс Солитер), Monopoly (настольная экономическая игра Монополия), Scrabble (настольная игра в слова Скраббл). Рассмотрим взятие максимума как функцию агрегатной оценки. Тогда концепт MS Windows будет считаться ближе к предметной области «Настольные игры», чем концепт Blackjack (карточная игра Блекджек), из-за высокого значения семантической близости концепта MS Windows и концепта Solitaire (так как эта игра включена в операционную систему Windows). См. рисунок 2.1.

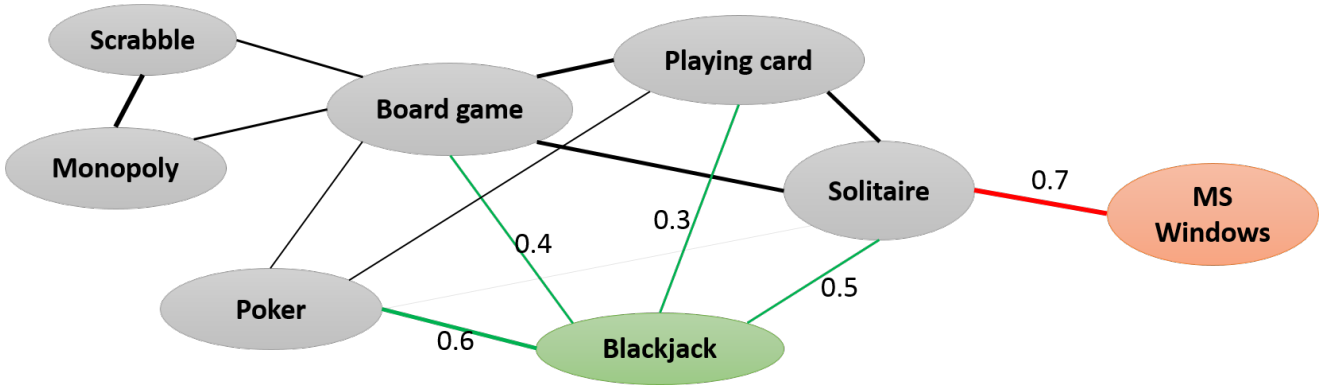

Рисунок 2.1: Взятие максимума в качестве функции агрегатной оценки

Рассмотрим теперь усреднение как функцию агрегатной оценки. В этом случае концепт Family (семья) будет считаться ближе к предметной области Настольные игры, чем концепт Settlers (настольная игра с одноименным названием), поскольку концепт Family близок, пусть и не так значительно, ко многим концептам за счет включения в число ключевых концептов нескольких семейных игр, в то время как концепт Settlers оказался близок лишь к двум, так как относится к типу настольных игр, слабо представленных среди ключевых концептов. См. рисунок 2.2.

Для решения этих недостатков предлагается учитывать при подсчете функции агрегатной оценки лишь подмножество наиболее близких концептов предметной области. Например, если учитывать только 2 ближайших

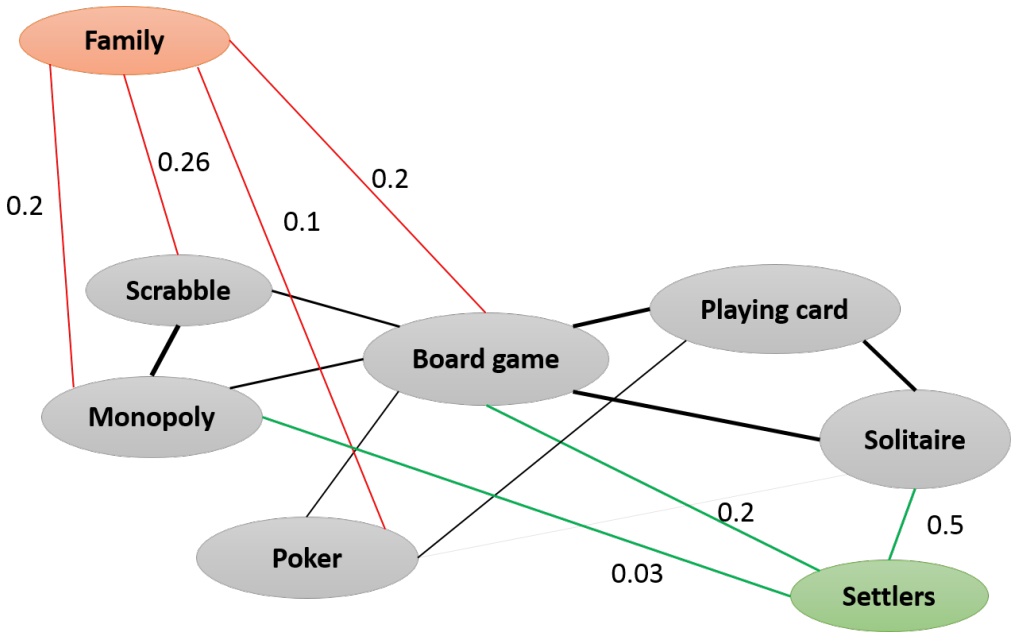

Рисунок 2.2: Усреднение в качестве функции агрегатной оценки

концепта, то в обоих приведенных выше примерах агрегатная оценка будет выше для концептов, действительно принадлежащих предметной области.

Нетрудно заметить, что предложенное решение представляет собой взвешенный вариант метода k ближайших соседей (k Nearest Neighbors, kNN), адаптированный для случая только положительных примеров:

$$sim_k(c, C_N) = \frac{1}{k} \sum_{i=1}^{k} sim(c, c_i), \tag{2.2}$$

где c — концепт термина; C_N — множество ключевых концептов, отранжированных по убыванию семантической близости к c; $sim(c, c_i)$ — функция семантической близости; k — константа, определяющее число ближайших концептов, которые учитываются при вычислении итоговой семантической близости.

При такой формулировке отмеченные выше недостатки являются известными проблемами высокого смещения (high bias) для слишком больших значений k и высокой вариации (high variance) для слишком малых значений k.

2.2.3 Описание алгоритма

Общий алгоритм метода «Близость к ключевым концептам» следующий.

1. Определить ключевые концепты на основе заданной коллекции документов:
    1) извлечь d ключевых концептов из каждого документа коллекции;
    2) посчитать *встречаемость* ключевых концептов: сколько раз в коллекции каждый концепт попал в число лучших d ключевых концептов документа;
    3) выбрать N ключевых концептов с наибольшей встречаемостью.
2. Для рассматриваемого кандидата в термины: найти все возможные концепты Википедии, такие что их названия совпадают с кандидатом в термины.
3. Для каждого найденного концепта кандидата в термины: вычислить семантическую близость к найденным ключевым концептам.
    1) вычислить попарную семантическую близость с каждым ключевым концептом;
    2) выбрать k ключевых концептов с максимальной семантической близостью;
    3) усреднить значения семантической близости к выбранным ключевым концептам.
4. Выбрать максимальное значение по всем концептам кандидата в термины.

Таким образом, значение этого признака будет близко к нулю для слов и словосочетаний, обозначающих понятия, далекие по смыслу от ключевых понятий предметной области.

Рассмотрим в качестве примера снова *Gene* и *Last card* применительно к предметной области «Настольные игры». Допустим, $k = 2$ и из коллекции текстов про настольные игры были извлечены следующие ключевые концепты:

1. Board game (собственно, Настольная игра)
2. Card game (Карточная игра)

3. Hasbro Inc. (Компания, производящая игрушки и настольные игры)

Как уже упоминалось выше, в Википедии существует статья *Last card* про одноименную игру; значения семантической близости этого концепта с указанными в списке составляют 0.001, 0.037 и 0, соответственно. Таким образом, значение признака составляет 0.019. В то же время для термина *Gene* (Ген) семантическая близость к указанным концептам будет равна нулю и, таким образом, термин Last card является более вероятным термином предметной области «Настольные игры» с точки зрения метода «Близость к ключевым концептам».

2.3 Экспериментальное исследование разработанных методов

В данном разделе представлены результаты экспериментального исследования разработанных методов. После методики тестирования — используемых наборов данных, метода сбора кандидатов и метрик эффективности — описывается выбор параметров разработанных методов и их сравнение с существующими методами.

2.3.1 Описание экспериментальной установки

Наборы данных

В данной работе экспериментальное исследование проводилось на следующих открытых наборах данных: GENIA, Krapivin, FAO, Patents и Board games.

GENIA [81] представляет собой коллекцию из 2000 размеченных документов биомедицинской тематики, она является одним из наиболее популярных наборов данных для тестирования эффективности извлечения терминов: результаты на этой коллекции приводятся, как минимум, в 6 работах [17, 19, 25, 45, 82, 83].

FAO [84] состоит из 780 размеченных вручную отчетов Продовольственной и сельскохозяйственной организации ООН (Food and Agriculture

Organization): в каждом отчете выделялось по 2 термина. Этот набор данных использовался для тестирования автоматического извлечения терминов в работе Дж. Бордо [45].

Krapivin [85] представляет собой 2304 научные статьи по информатике; в качестве эталонного множества терминов используются ключевые слова, выделенные авторами статей. При тестировании в настоящей диссертационной работе к этому множеству был добавлен словарь предметной области «Вычислительная техника» (Computing), использованный в качестве эталона в системе Protodog [86]. Этот набор данных также тестировался в работе Дж. Бордо [45].

Patents [42] — набор из 16 патентов в области электротехники, размеченных вручную.

Набор данных Board games — коллекция из 1300 описаний и рецензий настольных игр — был подготовлен в рамках данной работы. 35 документов из 1300 были размечены вручную, и для тестирования использовались только термины, имеющие хотя бы одно вхождение в размеченные документы. Это позволило использовать статистику вхождений, посчитанную на всей коллекции, и при этом оценивать эффективность работы методов с минимальной погрешностью.

В таблице 2.1 представлена статистика используемых наборов данных.

Таблица 2.1: Статистика наборов данных

	Board games	Patents	GENIA	Krapivin	FAO
Предметная область	Настольные игры	Электротехника	Биомедицина	Информатика	Сельское хозяйство
Документов (размеченных)	1300 (35)	16	2000	2304	778
Слов, тыс.	612	120	484	20566	26364
Слов (на документ)	470.9	7493.0	242.2	8926.3	33887.3
Терминов-эталонов	527	1556	30423	8692	1556

Метод сбора кандидатов

Кандидаты для всех оцениваемых методов представляют собой N-граммы (от 1 до 4) со следующими фильтрами.

1. **Фильтр по шаблонам частей речи.** Кандидатом в термины считается любая комбинация существительных и прилагательных с существительным в роли главного слова; главным словом считается последнее слово в N-грамме. К главному слову предъявляется дополнительное требование: оно должно быть именем нарицательным; остальные существительные при этом могут быть именами собственными, что позволяет извлекать, например, названия теорем или методов.

2. **Фильтр по частоте.** Исключаются из дальнейшего рассмотрения кандидаты в термины, имеющие число вхождений меньше заданного порогового значения (минимальный порог встречаемости). При этом две N-граммы считаются принадлежащим одному кандидату в термины, если они совпадают с точностью до нормализации. В данной работе в качестве нормализации используется эвристический алгоритм, который основан на морфологических свойствах существительных[6].

   Для Board game, Patents и GENIA минимальный порог встречаемости составляет 2; для Krapivin и FAO — 3.

3. **Фильтр по стоп-словам.** Если в составе N-граммы содержится хотя бы одно слово из заданного списка стоп-слов, эта N-грамма исключается из числа кандидатов в термины. В качестве списка стоп-слов используется тот же список[7], что и в работы Дж. Бордо [45].

В таблице 2.2 показана статистика извлеченных кандидатов в термины по мере применения описанных фильтров. Как можно видеть, применение фильтров значительно уменьшает число кандидатов, тем самым повышая производительность методов и потенциальную точность. Вызванное этими фильтрами потенциальное падение полноты можно оценить в таблице 2.3, где показана статистика терминов-эталонов среди извлеченных кандидатов в термины.

Статистика по количеству N-грамм каждого порядка среди эталонов представлена в таблице 2.4. Как можно заметить, соотношение отличается для разных наборов данных: так, для наборов данных Board games и FAO ко-

[6] Для английского языка алгоритм состоит в отбрасывании окончаний множественного числа (*s*, *es* и т.п.); реализация алгоритма взята из системы Текстерра [24]

[7] http://www.lextek.com/manuals/onix/stopwords1.html

Таблица 2.2: Статистика извлеченных кандидатов в термины

	Board games	Patents	GENIA	Krapivin	FAO
Кандидатов всего	29590	5807	48579	778133	1065931
Кандидатов после фильтра по частоте	10289	2888	14983	161728	197036
Кандидатов после всех фильтров	8580	2540	14385	141259	171551
Кандидатов, присутствующих в Википедии	9809	2189	8922	67202	119644

Таблица 2.3: Статистика эталонов среди извлеченных кандидатов в термины

Эталонов среди кандидатов	Board games	Patents	GENIA	Krapivin	FAO
всего	262	1235	20455	7099	1450
после фильтра по частоте	215	886	9571	6147	1355
после всех фильтров	201	795	8482	5899	1324
присутствующих в Википедии	168	397	3407	3489	1160

личество однословных терминов приблизительно равно количеству двухсловных, в то время как для остальных наборов данных количество однословных терминов в 2-3 раза меньше. Наборы данных Board games и FAO выделяются также и соотношением трехсловных терминов: в то время как остальные наборы данных содержат сопоставимые числа однословных и трехсловных терминов, в наборах данных Board games и FAO практически нет трехсловных терминов.

Таблица 2.4: Статистика по количеству N-грамм каждого порядка среди эталонов

	Board games	Patents	GENIA	Krapivin	FAO
N-grams	262	1198	18573	7099	1450
1-grams	144	223	2784	1449	653
2-grams	106	696	9963	4604	771
3-grams	10	236	4879	992	25
4-grams	2	43	947	54	1

Метрики оценки эффективности

Напомним, что результатом работы каждого оцениваемого метода является список кандидатов, отранжированный по вероятности быть термином. Эффективность оценивалась с помощью средней точности (см. раздел 1.4).

Для всех наборов данных, кроме GENIA, число N — общее количество оцениваемых кандидатов, или длина верхней части списка отранжированных кандидатов — равнялось числу терминов в списке эталонов. Для GENIA N было выбрано равным 10000, так как итоговый список кандидатов после всех фильтров насчитывает всего 15 тысяч терминов (см. таблицу 2.3), что в 2 раза меньше размера списка эталонов.

2.3.2 Выбор параметров

Метод «Вероятность быть гиперссылкой»

Для выбора оптимального значения параметра T (порога, ниже которого обнуляется значение всего признака) был произведен перебор значений от 0 до 0.1 с шагом 0.0001, то есть для каждого такого значения параметра T проводилась оценка эффективности работы метода LinkProbability для каждого набора данных.

Результаты для наборов данных Patents, Board games и GENIA приведены на рисунке 2.3. Для наборов данных Krapivin и FAO значения метода не менялись при указанном изменении параметра T, так как количество терминов-эталонов, а следовательно и учитываемых кандидатов в термины (число N в метрике средняя точность) представляет собой слишком малое число по сравнению с общим количеством кандидатов в термины; другими

словами, значение признака «Вероятность быть гиперссылкой» для кандидата в термины, последнего в списке учитываемых метрикой, намного больше значения 0.1. По этой причине далее будет рассматриваться подбор параметра исключительно для наборов данных Patents, Board games и GENIA.

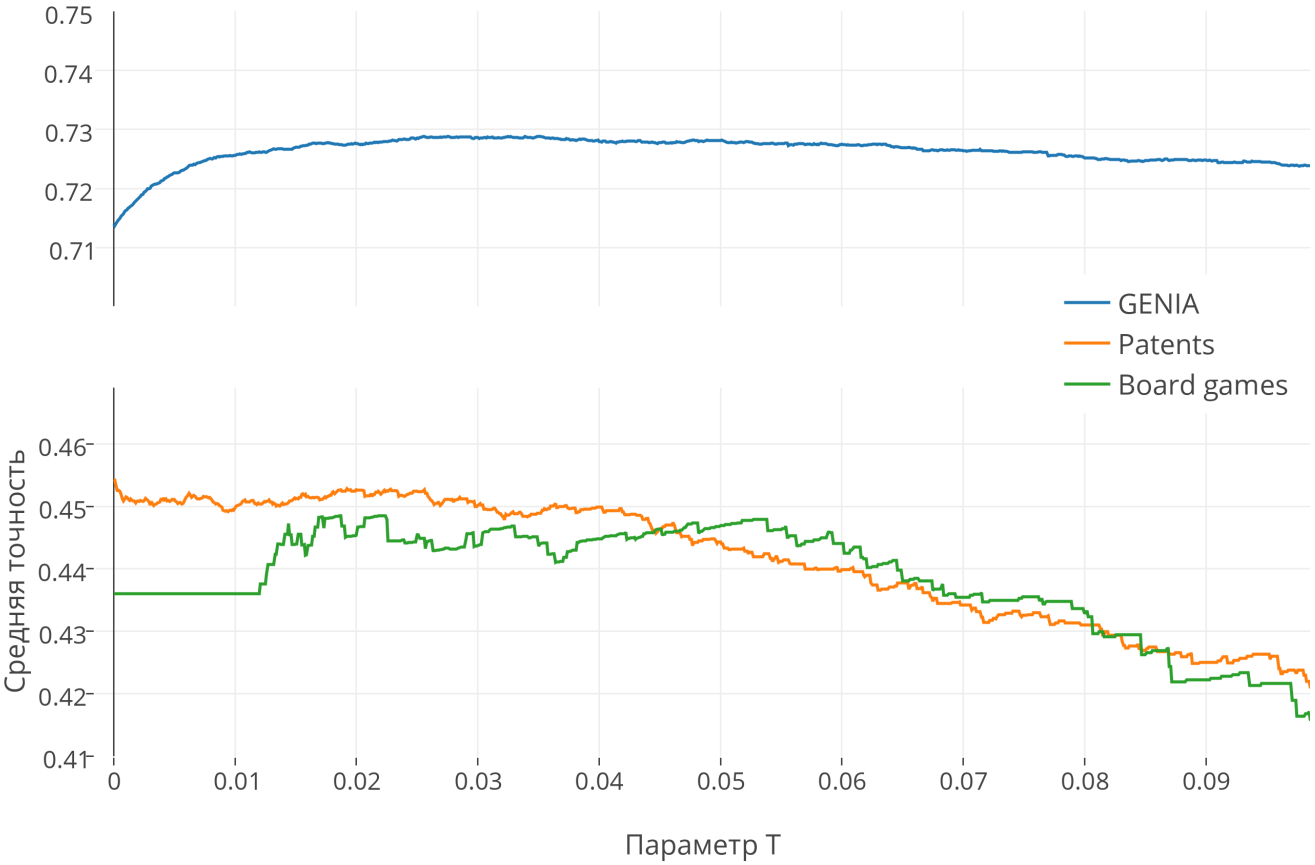

Рисунок 2.3: Выбор параметра T

Как видно из графика, оптимальные значения параметра T лежат в диапазоне от 0.01 до 0.03: после этого эффективность начинает уменьшаться для всех наборов данных.

Увеличим масштаб для этого диапазона.

Важно отметить, что в данном эксперименте термины с нулевым значением признака, в том числе и те термины, у которых значение признака было меньше T, выбираются случайно, поэтому в приведенных значениях средней точности также присутствует доля случайности. Кроме того, слишком боль-

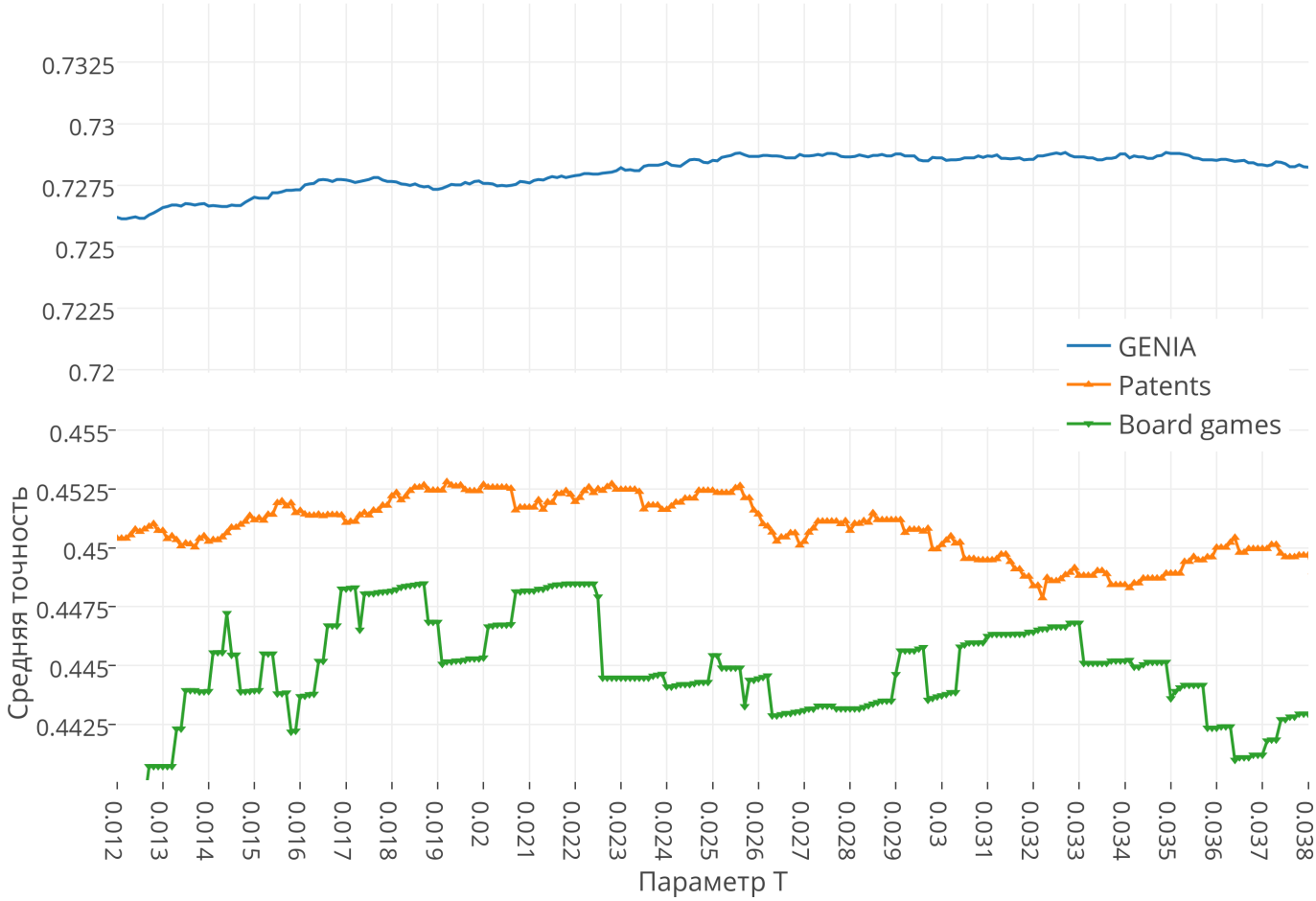

Рисунок 2.4: Выбор параметра T: увеличенный масштаб

шое значение параметра T может привести к потери важной информации в случае комбинации данного признака с другими.

Учитывая эти замечания, выбранное значение параметра T составило 0.018: это один из двух максимумов для набора данных Board games (зоны рядом с 0.018 и 0.022), причем с меньшим значением параметра T, к тому же при данном значении достаточно высоки показатели средней точности и для остальных наборов данных.

Метод «Близость к ключевым концептам»

Алгоритм метода KeyConceptRelatedness содержит три параметра:

1) d — количество ключевых концептов, извлекаемых из каждого документа входной коллекции;

2) N — общее количество ключевых концептов, извлекаемых из входной коллекции;

3) k — количество ключевых концептов с максимальной семантической близостью, учитываемых при вычислении итогового значения признака.

Как упоминалось в разделе 2.2.1, извлекаемые из документа ключевые концепты должны принадлежать основной теме документа, а с ростом параметра d, то есть числа концептов, извлекаемых из одного документа, растет и вероятность нарушить это требование, особенно в случае небольших документов. С другой стороны, извлечение слишком малого числа концептов из одного документа приведет к низкой полноте: темы, принадлежащие заданной предметной области, будут слабо представлены. Например, в случае предметной области «Настольные игры» может быть не извлечено ни одного концепта, связанного с карточными игры.

С учетом вышеизложенного, было выбрано значение $d = 3$ как компромисс между точностью и полнотой извлечения ключевых концептов предметной области. При этом в случае коллекции с очень малым числом больших документов это значение повышается: так, для набора данных Patents, содержащего всего 16 документов, использовалось значение $d = 15$, т.к. оно потенциально позволяет получить 240 ключевых концептов, что почти соответствует верхней границе в диапазоне изменений N, см. ниже).

Для выбора оптимальных значений параметров N и k проводилась оценка эффективности работы метода «Близость к ключевым концептам» для каждого набора данных на следующих значениях параметров:

1) N: от 25 до 250 с шагом 25: $\{25, 50, 75, 100, 125, 150, 175, 200, 225, 250\}$;

2) k: от 2 до 50 с меняющимся шагом: $\{2, 3, 5, 7, 10, 15, 25, 50\}$.

Для визуализации результатов — средней точности для каждой комбинации — используется теплокарта, на которой значения параметров представляют собой координаты, а температура цвета — значение средней точности: чем светлее цвет, тем выше средняя точность.

Результаты для всех пяти наборов данных представлены на рисунках 2.5, 2.6, 2.7, 2.8, 2.9.

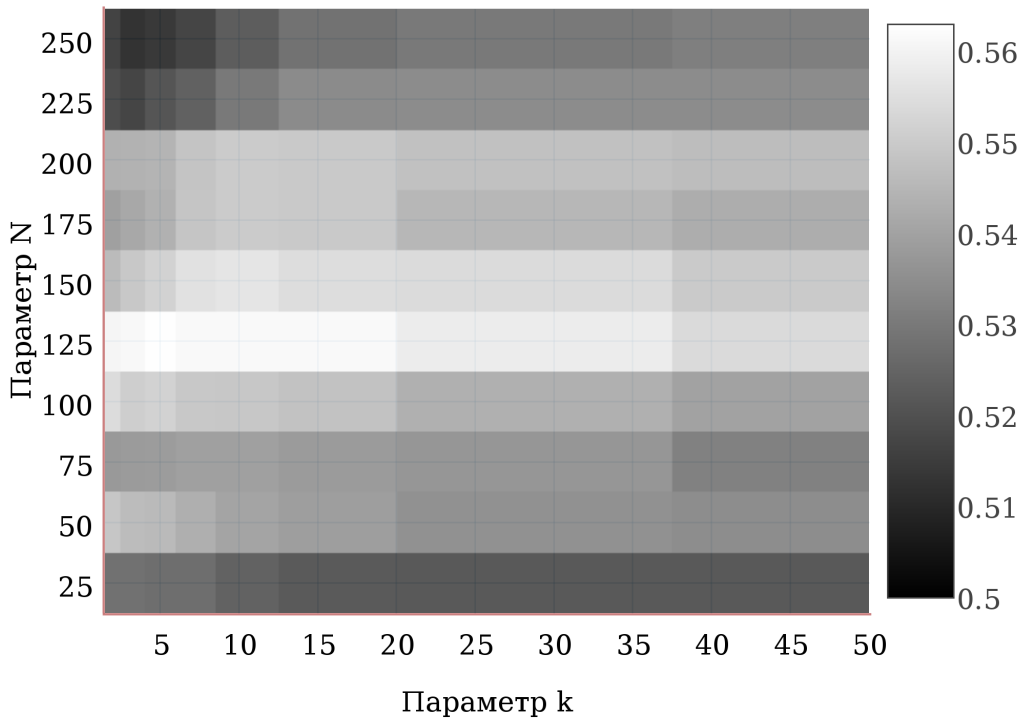

Рисунок 2.5: Зависимость средней точности от k и N (Board games)

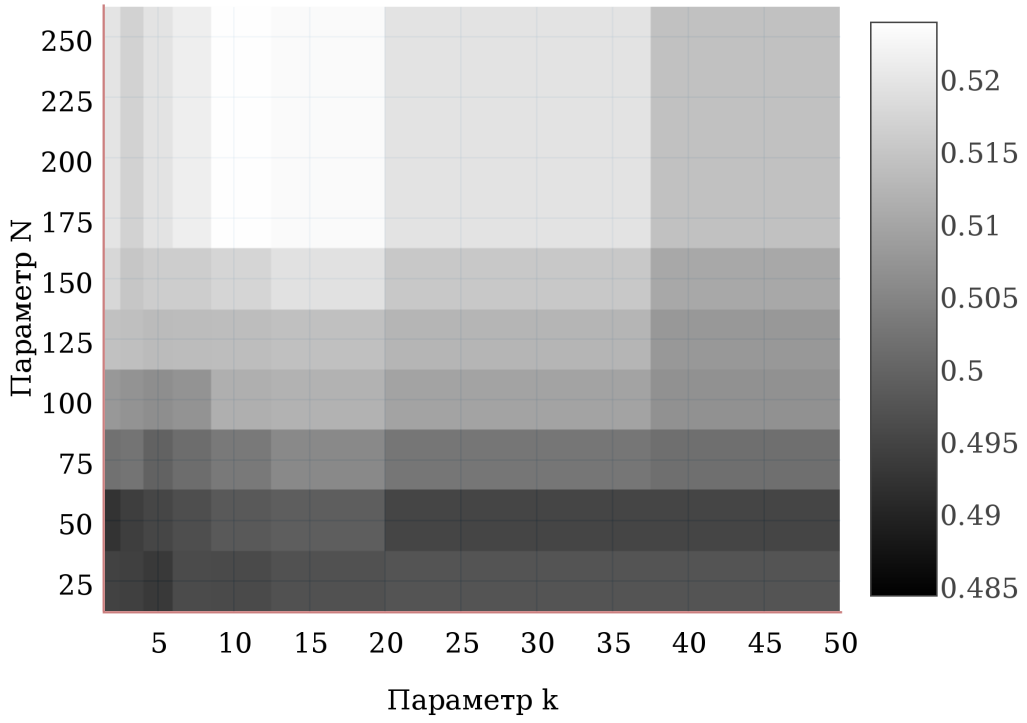

Рисунок 2.6: Зависимость средней точности от k и N (Patents)

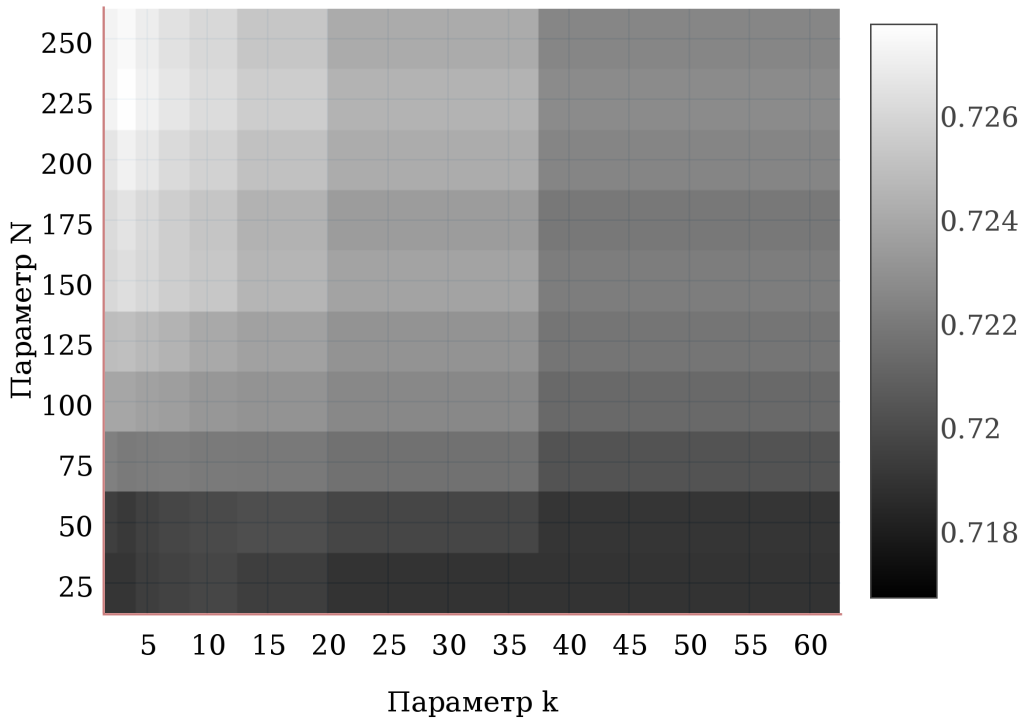

Рисунок 2.7: Зависимость средней точности от k и N (GENIA)

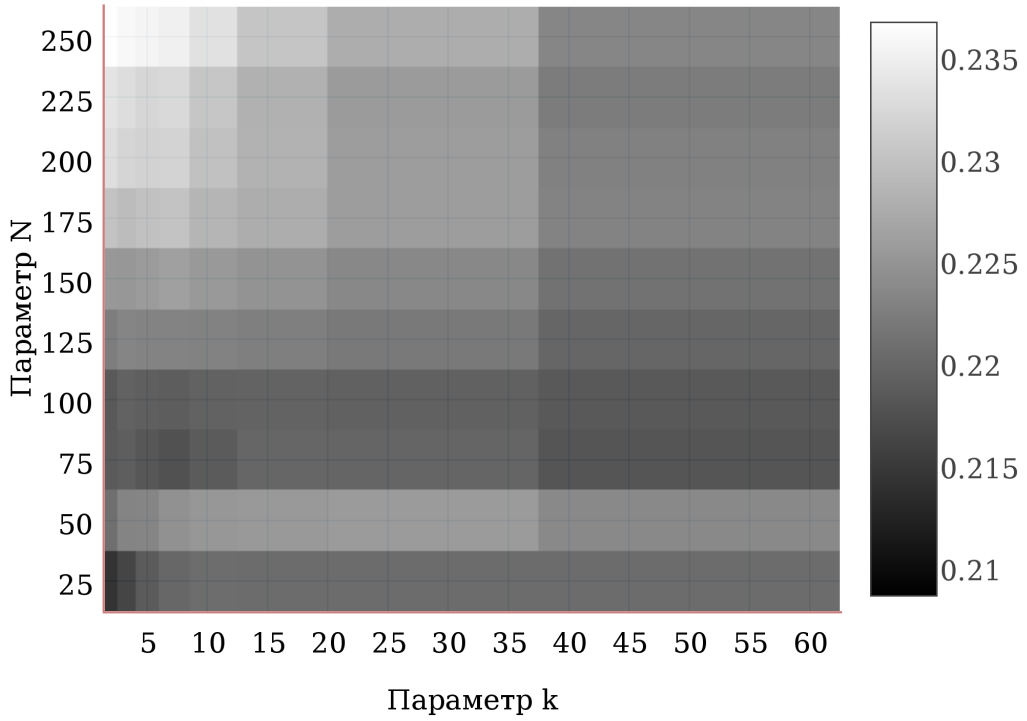

Рисунок 2.8: Зависимость средней точности от k и N (Krapivin)

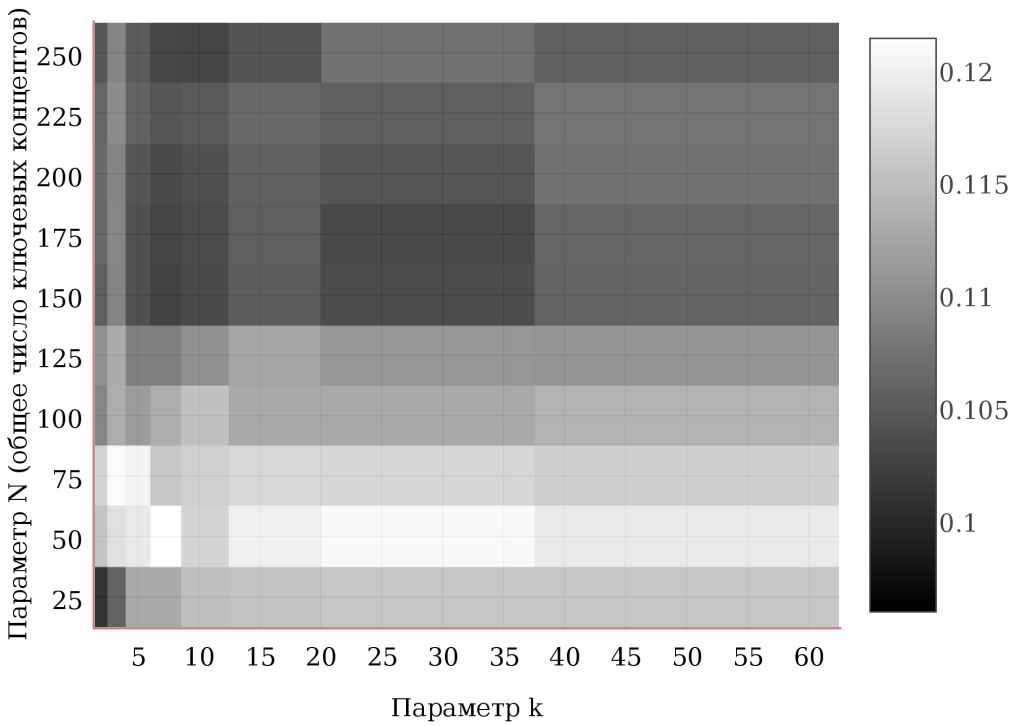

Рисунок 2.9: Зависимость средней точности от k и N (FAO)

Как видно из графиков, оптимальные значения параметров отличаются для разных наборов данных: наборы данных с большим числом кандидатов и эталонов (GENIA и Krapivin) демонстрируют лучшую эффективность для больших N и малых k; для небольших наборов данных оптимальные значения представляют собой числа из середины диапазона: для Board games $125 \leq N \leq 150$ и $2 \leq k \leq 15$ и для Patents $N \geq 175$ (всего извлечено меньше 200 концептов) и $5 \leq k \leq 25$; отдельного рассмотрения заслуживает набор данных FAO, показавший низкие и нестабильные результаты для всех комбинаций параметров: можно только отметить, что средняя точность выше для небольших N (50–75).

По результатам этого исследования были выбраны значения $N = 200$ и $k = 10$, поскольку они позволяют достичь хороших результатов на большинстве наборов данных: значения N больше 200 существенно уменьшают эффективность для набора данных Board games, а также увеличивают вычислительную сложность; при этом значения меньше 200 уменьшают эффективность на больших наборах данных.

При $N = 200$ оптимальным значением для наборов данных Board games и Patents является $k = 10$; при этом для набора данных GENIA эффективность меняется незначительное (на десятые доли процента); результаты на наборах данных Krapivin и FAO учитывались с меньшим весом, так как в них размечены лишь небольшая часть эталонов и поэтому оценка точности имеет бóльшую погрешность.

Итак, выбранные значения для метода «Близость к ключевым концептам»:

1) $N = 200$;

2) $k = 10$;

3) $d = 3$ (или больше, если количество извлекаемых концептов меньше 200; для набора данных Patents использовалось $d = 15$).

2.3.3 Сравнение с существующими методами

В таблице 2.5 представлены результаты сравнения разработанных методов с существующими методами извлечения терминов из коллекции документов. Ниже изложены некоторые особенности реализации этих методов.

Для обобщения метода C-Value на случай однословных терминов применялся подход Ж. Вентуры и др. [55], однако в качестве сглаживающего коэффициента было выбрано значение 0.1 (в оригинальной работе предлагалось значение 1). Этот подход показал лучшую эффективность на всех наборах данных по сравнению с подходом А. Баррон-Цедено и др. [56].

Для методов Domain Relevance, Relevance, Weirdness, требующих корпус текстов из другой (общей) предметной области, использовалась статистика из Корпуса современного американского английского языка[8] (Corpus of Contemporary American English).

Для метода Novel Topic Model производилась фильтрация слов по частоте (с порогом встречаемости 2) и по частям речи: оставлялись только существительные, прилагательные, глаголы и наречия.

В методе WikiCategories (NC) вместо полного перебора всех категорий при поиске в ширину производилась остановка при нахождении 100 путей до категории верхнего уровня (пробовалось также ограничение в 1000 путей,

[8] http://www.ngrams.info

но на наборах данных Board games, Patents и GENIA результаты отличались незначительно и, как правило, в худшую сторону).

Кроме того, дополнительно к методу Domain Model, в котором используется линейная комбинация двух признаков (Basic и DomainCoherence), приводятся также результаты с комбинацией этих признаков алгоритмом голосования, а также результаты одного только признака Basic.

Таблица 2.5: Сравнение разработанных методов с существующими методами

Метод	Board games	Patents	GENIA	Krapivin	FAO
TF-IDF	0.3882	0.4922	0.6936	0.3619	**0.4270**
C-Value	0.3350	**0.6271**	0.7294	0.3706	0.3731
Ventura C-Value (0.1)	0.3967	0.5874	0.7376	0.3911	0.4080
Weirdness	0.3121	0.3972	0.5289	0.2562	0.2853
TermExtractor	0.3526	0.4974	0.7331	0.3209	0.1543
Relevance	0.3797	0.4104	0.5338	0.3175	0.3488
Domain Relevance	0.3253	0.4943	**0.7425**	0.3218	0.1534
Domain Consensus	0.2976	0.2618	0.6296	0.2246	0.1535
Domain Model	0.3680	0.3992	0.6720	0.3929	0.3389
Domain Model (Vote)	0.3821	0.4594	0.6729	**0.4609**	0.3447
Basic (Domain Model)	0.3539	0.5051	0.6475	0.4141	0.3486
Novel Topic Model	0.3684	0.5426	0.7129	0.1271	0.0593
Wiki Categories (NC)	**0.4110**	0.2934	0.7157	0.1671	0.0765
LinkProbability	0.4482	0.4522	0.7276	0.1815	0.0169
KeyConceptRelatedness	**0.5470**	0.5237	0.7253	0.2282	0.1089

Как видно из таблицы, разработанные методы обладают сравнимой эффективностью с существующими методами на трех наборах данных (Board games, Patents, GENIA), в частности, на этих наборах данных метод «Близость к ключевым концептам» занимает 1, 4 и 5 место, соответственно.

Низкие результаты на остальных наборах данных (Krapivin, FAO) можно объяснить двумя основными отличиями этих наборов данных: размером и природой терминов-эталонов. Так, большой размер наборов данных Krapivin и FAO дает преимущество методам, основанным на статистике вхождений, в том числе с учетом контекстов, внешних корпусов или тематических моделей. При этом разработанные методы никак не используют вхождения кандидатов в термины помимо собственно выражающей их текстовой строки.

Что касается природы терминов-эталонов, то в отличие от Board games, Patents и GENIA, в наборах данных Krapivin и FAO в качестве эталонов выделены исключительно ключевые термины (точнее, их подмножество), которые являются терминами средней специфичности (по терминологии Дж. Бордо [45], см. раздел 1.1.3). В то же время, разработанные методы либо не учитывают специфичность терминов («Близость к ключевым концептам»), либо предпочитают термины высокой специфичности («Вероятность быть гиперссылкой»).

2.4 Выводы

В данной главе приведено краткое описание Википедии с акцентом на ее структуру гиперссылок; представлены новые методы, основанные на использовании структуры гиперссылок Википедии, а также приведено их экспериментальное исследование.

Результаты экспериментального исследования показывают, что для некоторых наборов данных разработанные методы показывают сравнимые показатели эффективности, однако для наборов данных, обладающих большим размером и предполагающих выделение терминов средней специфичности, эффективность разработанных методов оказывается неудовлетворительной. В следующей главе приводится описание метода, устраняющего эти недостатки.

Глава 3

Метод извлечения терминов на основе алгоритма частичного обучения

В настоящей главе описывается новый метод извлечения терминов, основанный на алгоритме обучения на основе положительных и неразмеченных примеров (Positive-unlabeled learning, PU-learning) — частного случая алгоритма частичного обучения (Semi-supervised learning).

В первом разделе описывается общая схема метода, формализуется его модель. Во втором и третьем разделе подробно анализируются основные этапы подхода — извлечение терминов-положительных примеров и обучение модели алгоритма PU-learning. Четвертый раздел посвящен экспериментальному исследованию разработанного метода.

3.1 Общая схема подхода

Известно, что для эффективного извлечения терминов необходим учет множества факторов [12, 87], то есть комбинация различных признаков. При этом, как было рассмотрено в разделе 1.3.2, в настоящее время существует всего несколько методов комбинации признаков, каждый из которых обладает своими недостатками.

Так, методы, основанные на алгоритмах машинного обучения с учителем, требуют большого числа размеченных данных, получить которые на практике

зачастую слишком накладно, с учетом необходимости обучать классификаторы для каждой предметной области.

Линейная комбинация с одинаковыми коэффициентами и алгоритм голосования не требуют размеченных данных, но обладают меньшей эффективностью [19], поскольку предполагают одинаковую важность каждого признака. Подбор же коэффициентов вручную, в том числе в методе на основе правил, на практике представляется не менее сложной задачей, чем разметка данных для алгоритма обучения с учителем.

Одновременно с этим можно отметить, что многие существующие методы извлечения терминов обладают высокой точностью для небольшого числа терминов (100-200 лучших терминов), которая к тому же может быть дополнительно повышена за счет фильтрации терминов, отсутствующих в Википедии: практически любая предметная область обладает достаточным покрытием в Википедии для извлечения такого количества терминов.

Исходя из этих наблюдений формулируются три предположения:

1. Кандидаты, близкие в признаковом пространстве, обладают одинаковыми метками (термин и не-термин) — на данном предположении основано применение алгоритмов машинного обучения с учителем; подтверждением данного предположения может служить эффективность методов машинного обучения с учителем для задачи извлечения терминов.

2. Можно выбрать (разработать) специальный метод извлечения терминов, такой что извлеченные с его помощью 100-200 лучших кандидатов в термины будут содержать достаточно информации о скрытых взаимосвязях признаков, то есть с их помощью можно будет обучить эффективный классификатор.

3. Среди 100-200 лучших кандидатов, извлеченных с помощью специального метода, содержится достаточно малое число ошибок (не терминов), и этим шумом можно пренебречь — подтверждением данного предположения может служить эффективность методов NC-Value и Domain Model, в которых лучшие 200 кандидатов по результатам методов C-Value и Basic, соответственно, считаются правильными терминами.

Разработанный метод основан на данных предположениях и состоит в следующем: с помощью специального метода извлечения терминов определя-

ются S лучших кандидатов, которые затем используются как положительные примеры для построения модели алгоритма обучения на основе положительных и неразмеченных примеров (Positive-unlabeled learning, PU-learning). В данном случае неразмеченными примерами служат все остальные кандидаты в термины. Построенная модель алгоритма обучения далее используется для вероятностной классификации каждого кандидата в термины.

Для формализации метода вводятся следующие понятия.

- $W = \{w_i\}$ — все слова и словосочетания естественного языка.
- $y_d : W \to [0, 1]$ — функция, оценивающая вероятность того, что слово или словосочетание w_i является термином заданной предметной области d. Имеется в виду «истинная» оценка без учета осуществимости ее получения, например такой оценкой для определенного w_i может быть число, относительно которого согласны все возможные пользователи приложения, для которого извлекаются термины.
- e_d — экспертная оценка значения $y_d(w)$ для каждого w_i. В данной работе, как и в большинстве существующих работ, $e_d(w)$ представляет собой булеву функцию ($e_d : W \to \{0, 1\}$), возвращающую 1 для тех и только тех слов и словосочетаний, которые принадлежат заранее определенному экспертами списку правильных терминов-эталонов, однако возможна и более точная оценка ($e_d : W \to [0, 1]$), например, с помощью усреднения решений нескольких экспертов по данному слову или словосочетанию.

  В любом случае, удобно представить $e_d(w)$ в виде $e_d(w) = y_d(w) + \varepsilon_d$, где ε_d — разного рода ошибки экспертов.
- $X_{T,K} = \{x_i\}, X_{T,K} \subset W$ — кандидаты в термины, отобранные из заданной коллекции текстовых документов T с помощью заданного метода извлечения кандидатов K;
- $f : X_{T,K} \to [0, 1]$ — искомая функция, оценивающая вероятность того, что кандидат x_i является термином заданной предметной области. В данной диссертационной работе используется вещественнозначная оценка вероятности вместо бинарной классификации, так как рассматривается сценарий извлечения заранее заданного числа терминов

(см. раздел 1.2). Поскольку функцию y_d невозможно получить на практике, оценка эффективности функции f производится с помощью экспертной оценки e_d.

В разработанном методе предлагается ввести функцию $s : X_{T,K} \to \{u, 1\}$ — метод извлечения терминов для их использования в качестве положительных примеров, где значение 1 соответствует положительному примеру, а u — неразмеченному. Функция $f(x)$ строится на результатах $s(x)$ с помощью алгоритма PU-learning.

В следующем разделе подробно описывается алгоритм вычисления функции $s(x)$, после чего приводится описание алгоритма вычисления функции $f(x)$.

3.2 Автоматическое извлечение положительных примеров

Поскольку извлекаемые термины будут похожи на положительные примеры, формулируются следующие требования к методу извлечения положительных примеров:

1. Высокая точность: ошибочное отнесение кандидата к положительным примерам приведет к снижению точности функции $f(x)$.

2. Настраиваемая специфичность: как отмечалось выше, для разных приложений может требоваться разный уровень специфичности извлекаемых терминов; поскольку положительные примеры служат для обучения модели извлечения терминов, на практике полезно иметь возможность корректировать специфичность извлекаемых терминов.

   Кроме того, не исключено, что в качестве положительных примеров эффективнее рассматривать менее или более специфичные термины, что можно будет определить только в ходе экспериментального исследования.

3.2.1 Специфичность терминов

Для работы со специфичностью (в частности, для проверки второго требования) необходимо формализовать это понятие.

Во многих работах [6, 45, 88] это понятие никак не формализуется, если не считать нескольких примеров терминов разной специфичности (см. раздел 1.1.3).

Часто специфичность терминов или концептов отождествляется с гипонимией, или отношением «является» (IS-A) [89]. Однако такое ограничение сильно сужает понятие специфичности: например, оно не позволяет сравнить термины «скорость передачи данных» и «передача данных», так как они не находятся в отношении гипонимии-гиперонимии, хотя первый термин интуитивно представляется более специфичным, поскольку описывает один из важных атрибутов второго термина.

П.-М. Рю и К.-С. Чой также полагают специфичность лишь «одним из необходимых условий для образования иерархии терминов» [90], то есть если термин t_1 стоит выше термина t_2 в иерархии терминов, то специфичность t_1 должна быть меньше, чем специфичность t_2. Другими словами, каждый гипоним является более специфичным термином, но не каждый более специфичный термин является гипонимом.

Сами П.-М. Рю и К.-С. Чой предлагают следующее определение специфичности: «quantity of domain specific information contained in the term» (Перевод: количество информации о предметной области, содержащейся в термине) [91]. Похожее определение, но без отсылки к предметной области, используется и в работе С. Гаудана и др.: «количество информации о предметной области, содержащейся в термине» [92].

К сожалению, данное определение не является конструктивным для задачи определения специфичности. В частности, в той же работе [91] для разработки методов определения специфичности П.-М. Рю и К.-С. Чой опираются уже на другое определение, а именно — стандарт ISO 704:2000(E) — и используют следующую формулировку: «Let's consider two concepts X and Y. X is an existing concept and Y is a newly created concept by adding new features to X. In this case, X is a hyper concept of Y, and the feature set of X is a subset of that of Y» (Перевод: Рассмотрим два концепта X и Y, такие что X — существующий концепт, Y — новый концепт, созданный добавлением

новых признаков к X. В этом случае X является объемлющим концептом для Y, и набор признаков концепта X является подмножеством признаков концепта Y).

Далее, переходя от концептов к обозначающих их терминам, авторы выделяют два возможных случая:

1) у концептов X и Y есть общие слова;

2) у концептов X и Y нет общих слов.

И если во втором случае для определения специфичности необходимо анализировать контексты употребления терминов, в первом случае, отмечают авторы, важнее анализ композиционной структуры термина, поскольку новые слова в более специфичном концепте X как раз и обозначают новые признаки, отсутствующие у концепта Y.

В данной диссертационной работе предлагается следующее определение: термин t_1 называется более специфичным, чем термин t_2, если термин t_1 является гипонимом или меронимом[1] термина t_2 (более точно: если концепт, обозначаемый термином t_1, является гипонимом или меронимом концепта, обозначаемого термином t_2; для краткости будет использоваться первая формулировка при отсутствии неоднозначности).

Такое определение согласуется со сформулированным выше свойством «если термин t_1 стоит выше термина t_2 в иерархии терминов, то специфичность t_1 должна быть меньше, чем специфичность t_2». При этом значительно расширяется область применения понятия специфичности за счет включения меронимии в число допустимых отношений: так, в рассмотренном выше примере термин «скорость передачи данных» является меронимом термина «передача данных» и, таким образом, более специфичным термином.

Для практического определения специфичности термина полезно рассмотреть упомянутое выше понятие композиционной структуры термина. В работе «The head-modifier principle and multilingual term extraction» (Перевод: Принцип главного слова-модификаторов и извлечение многоязычных терминов) [93] подробно рассматриваются способы образования составных существительных и именных групп; в частности, вслед за С. Джонсом, авторы формулируют так называемый универсальный принцип главного слова-

[1] Отношение «часть-целое» (part-of, substance of, member-of).

модификатора: линейный порядок элементов в составных существительных или именных группах отражает часть информации, передаваемой этим словом или словосочетанием. Например, существительное *speedboat* (быстроходный катер) состоит из главного слова *boat* (катер) и модификатора *speed* (скорость, быстрота), причем значение слова *speedboat* можно вывести, зная значения составных частей; в словосочетании *boat speed* (скорость катера) главное слово и модификатор меняются местами: другими словами, порядок слов определяет значение составного слова или словосочетания.

Универсальность понимается в самом широком смысле: принцип остается справедливым для любой предметной области и любого языка (авторы подробно рассматривают английский и китайский языки и предлагают ссылки на исследования для французского и испанского).

Более того, отмечают авторы, в случае именных групп модификаторы служат для образования либо гипонимов, либо меронимов, и для некоторых языков семантика может определяться относительным положением модификаторов. Так, для английского языка модификаторы слева от главного слова служат для образования гипонимов: speedboat $\rightarrow$ competition speedboat (спортивный катер); модификаторы справа — меронимов: speedboat $\rightarrow$ speedboat engine (двигатель катера).

Однако не все исследователи согласны с однозначностью такой интерпретации: П.-М. Рю и К.-С. Чой предлагают в качестве примера словосочетание *vampire slayer*, которое можно интерпретировать с одинаковым успехом как «убийца вампиров» и как «вампир-убийца».

Отметим, что в русском языке гипонимы могут образовываться как добавлением модификаторов слева («метод касательных», «теорема об отрезках хорд»), так и справа («сходящийся интеграл»).

Из этого можно сделать вывод, что если термин t_1 образован от термина t_2 путем добавления модификаторов, то термин t_1 является более специфичным в смысле принятого определения, чем термин t_2.

На практике важен и обратный вопрос: если термин t_1 является более специфичным в смысле принятого определения, чем термин t_2, какова вероятность, что t_1 образован от термина t_2 путем добавления модификаторов? Очевидно, что эта вероятность не равна единице; например, термин «млекопитающее» более специфичен, чем «животное». Для удобства назовем *коэф-*

фициентом терминообразования эту вероятность, а именно P(«t_1 образован добавлением модификаторов к t_2»|«t_1 специфичнее t_2»).

Т.М. Юдина отмечает, что активность подобного рода терминообразования «обычно характерна для начального периода складывания терминосистем» [94].

Дж. Джастесон и С. Кац [95], проанализировав технические словари, утверждают, что большинство технических терминов представляют собой словосочетания на основе существительных, то есть образуются путем добавления слов-модификаторов к существующим терминам.

На основе приведенных наблюдений в данной работе делается предположение, что упомянутый выше коэффициент терминообразования является характеристикой предметной области, причем для технических и развивающихся предметных областей — именно тех, для которых обычно требуется автоматическое извлечение терминов — он представляет собой достаточно большую, то есть близкую к единице, величину.

3.2.2 Описание метода извлечения положительных примеров

При разработке специального метода извлечения положительных примеров уместно вспомнить признаки Domain Model и NC-Value, поскольку они в некотором смысле устроены похожим образом: на первом этапе извлекаются 200 лучших кандидатов, которые считаются правильно определенными терминами и используются на втором этапе для подсчета финального значения признака (см. раздел 1.3.3). И если второй этап любого из этих признаков и второй этап предлагаемого в данной работе метода значительно отличаются, то первый этап в обоих случаях представляется весьма похожим.

Из этих двух признаков рассмотрим более подробно метод Basic, часть метода Domain Model, по двум причинам. Во-первых, он более современный (2013 год по сравнению с 1998 годом) и поэтому авторы имели возможность выбрать свой метод извлечения положительных примеров среди множества существующих. Более того, этот метод представляет собой развитие метода C-Value, то есть основы метода NC-Value. И хотя авторы аргументировали

свой выбор не вполне корректной интерпретацией работы [47][2], их метод показал неплохие результаты.

Во-вторых, этот метод явным образом учитывает уровень специфичности извлекаемых терминов, пусть и недостаточно гибким образом.

Напомним, метод Basic вычисляется по формуле:

$$Basic(t) = |t| \log f(t) + \alpha e_t,$$

где t — кандидат в термины, $|t|$ — длина кандидата t (в словах), $f(t)$ — частота вхождений t в коллекции текстов, e_t — количество кандидатов, содержащих кандидата t.

Именно e_t повышает значение признака для средне-специфичных терминов, поскольку, как было отмечено в предыдущем разделе, термины высокой специфичности часто образуются путем добавления модификаторов к другим терминам — каковые часто и представляют собой термины средней специфичности.

В данной работе перед методом извлечения положительных примеров ставится более сложная задача: иметь возможность настраивать специфичность извлекаемых терминов. Заметим, что просто уменьшать значение параметра α не позволит достичь этой цели, потому что в этом случае будет меньший акцент на извлечение средне-специфичных терминов и больший — на извлечение частых и длинных терминов.

Предлагается ввести дополнительное слагаемое e'_t — число кандидатов, *содержащихся* в кандидате t. Назовем получившийся метод ComboBasic:

$$ComboBasic(t) = |t| \log f(t) + \alpha e_t + \beta e'_t \tag{3.1}$$

Для таким образом введенного метода справедлива следующая теорема.

[2] Как пишет Дж. Бордо [45], «more sophisticated statistical and information-theoretic measures are proposed for term extraction, it has been shown that these measures show little improvement when compared with plain frequency counts» (Перевод: было показано, что более сложные статистические и теоретико-информационные подходы, предложенные для извлечения терминов, дают малый прирост по сравнению с простыми значениями частоты) — однако в работе «You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation and term extraction» (Перевод: Нельзя победить частоту, если только не использовать лингвистическую информацию: качественная оценка мер ассоциации для извлечения терминов и коллокаций) производилось сравнение только с мерами ассоциации, как это явствует из самого названия, которыми, очевидно, методы извлечения терминов не исчерпываются.

Теорема 1. *Пусть d — коллекция документов из предметной области, такой что ее коэффициент терминообразования равен γ;*
t_1 и t_2 — кандидаты в термины, извлеченные из коллекции документов d, причем t_1 более специфичен, чем t_2;
$C_{\alpha,\beta}(t)$ — значение признака ComboBasic с коэффициентами α и β для кандидата t, посчитанное на основе коллекции d;
$\alpha, \beta_1, \beta_2 \in R^+$ — параметры ComboBasic, причем $\beta_1 < \beta_2$.

Тогда $P(C_{\alpha,\beta_2}(t_1) - C_{\alpha,\beta_1}(t_1) > C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2) \geq 0) = \gamma$.

Доказательство. По определению коэффициента терминообразования, P(«t_1 образован добавлением модификаторов к t_2»|«t_1 специфичнее t_2») $= \gamma$. Так как по условию теоремы t_1 специфичнее t_2, получим: $P(t_1$ образован добавлением модификаторов к $t_2) = \gamma$.

Условие «t_1 образован добавлением модификаторов к t_2» означает $e'_{t_1} > e'_{t_2}$, поскольку $e'_{t_1} = e'_{t_2} + f(t_2) + \varepsilon$, где $f(t_2)$ — частота вхождения термина t_2, ε — частота других кандидатов, содержащихся в термине t_1, причем $\varepsilon \geq 0$, а $f(t_2) > 0$, так как любой кандидат в термины встречается не менее одного раза.

Итак, $P(t_1$ образован добавлением модификаторов к $t_2) = P(e'_{t_1} > e'_{t_2}) = = \gamma$.

Домножим неравенство внутри вероятности на число $\beta_2 - \beta_1$, неотрицательное по условию теоремы:

$$P((\beta_2 - \beta_1)e'_{t_1} > (\beta_2 - \beta_1)e'_{t_2}) = \gamma$$

$$P((\beta_2 - \beta_1)e'_{t_1} + (|t_1| \log f(t_1) + \alpha e_{t_1}) - (|t_1| \log f(t_1) + \alpha e_{t_1}) >$$
$$> (\beta_2 - \beta_1)e'_{t_2} + (|t_2| \log f(t_2) + \alpha e_{t_2}) - (|t_2| \log f(t_2) + \alpha e_{t_2})) = \gamma$$

Перегруппировав слагаемые в обеих частях неравенства, получим:

$$P(C_{\alpha,\beta_2}(t_1) - C_{\alpha,\beta_1}(t_1) > C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2)) = \gamma$$

Осталось доказать выполнение второго неравенство в выражении для вероятности. Распишем значение признаков ComboBasic с разными β для термина t_2:

$$C_{\alpha,\beta_1}(t_2) = |t_2| \log f(t_2) + \alpha e_{t_2} + \beta_1 e'_{t_2}$$

$$C_{\alpha,\beta_2}(t_2) = |t_2| \log f(t_2) + \alpha e_{t_2} + \beta_2 e'_{t_2}$$

Вычтем из второго первое:

$$C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2) = \beta_2 e'_{t_2} - \beta_1 e'_{t_2} = (\beta_2 - \beta_1) e'_{t_2}$$

Так как $\beta_1 < \beta_2$ (по условию теоремы) и $e'_{t_2} \geq 0$ (по определению e'_t — частота кандидатов не может быть отрицательной), получаем:

$$(\beta_2 - \beta_1) e'_{t_2} \geq 0 \Rightarrow C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2) \geq 0$$

Или, что то же самое, $P(C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2) \geq 0) = 1$.

Учитывая, что из $P(X) = \gamma$ и $P(Y) = 1$ следует $P(XY) = \gamma$, получаем утверждение теоремы:

$$P(C_{\alpha,\beta_2}(t_1) - C_{\alpha,\beta_1}(t_1) > C_{\alpha,\beta_2}(t_2) - C_{\alpha,\beta_1}(t_2) \geq 0) = \gamma$$

□

Из доказанной теоремы следует, что с вероятностью γ увеличение параметра β в методе ComboBasic при прочих равных условиях приводит к извлечению им более специфичных терминов.

Таким образом, если предположить, что в заданной предметной области большая часть специфичных терминов образуется путем добавления модификаторов (коэффициент терминообразования больше 0.5), то, увеличивая параметр β в методе ComboBasic, можно извлекать более специфичные термины.

Рисунок 3.1 иллюстрирует влияние каждого компонента формулы метода ComboBasic на специфичность извлекаемых терминов[3].

На рисунке специфичность терминов увеличивается слева направо: от терминов, не принадлежащих заданной предметной области, до словосочетаний, слишком специфичных, чтобы быть терминами. Чем больше площадь под отрезком кривой, попадающей в какую-либо зону, например, «Общие тер-

[3]Похожий рисунок использовался в работе Дж. Бордо и др. [45].

мины», тем больше значение функции, например $log f(t)$, для таких терминов, или, что то же самое, тем больше терминов такого типа будет извлечено при ранжировании только по этой функции.

В первых четырех строках показано влияние компонент формулы, в последней — количество различных терминов (не путать с числом вхождений).

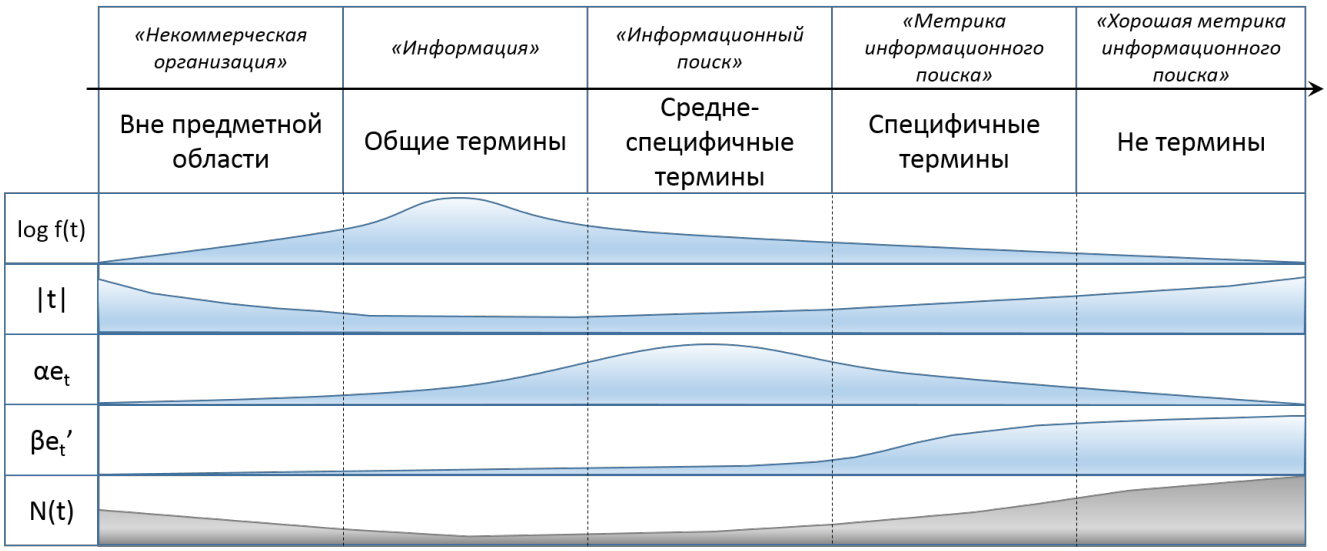

Рисунок 3.1: Влияние компонентов формулы ComboBasic на специфичность извлекаемых терминов

Заметим также, что увеличение параметра α приводит к более явному акценту на средне-специфичные термины: «горб» в соответствующей области становится «острее», см. рисунок 3.2. Очевидно, изменение параметра β приводит к аналогичным последствиям в крайних правых областях.

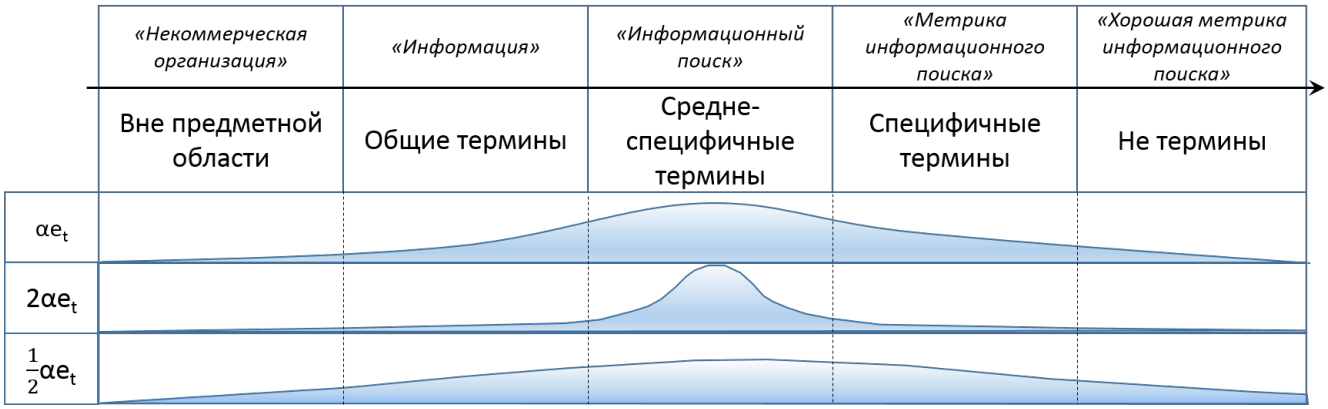

Рисунок 3.2: Влияние компонентов формулы ComboBasic на специфичность извлекаемых терминов

Для метода извлечения положительных примеров интерес представляют отрезки со второго по четвертый, то есть общие, средне-специфичные и спе-

цифичные термины, причем особенный интерес — последние два, так как их больше всего. По этой причине в формулы включены все эти компоненты.

Однако, как можно заметить, текущая формула не позволяет эффективно бороться с извлечением кандидатов, попадающих в последний отрезок, то есть словосочетаний, которые не являются терминами. Для борьбы с этим в методе извлечения положительных примеров дополнительно может производиться фильтрация по Википедии: среди результатов оставляются только те термины, которые присутствуют в Википедии в виде названий статей или текстов гиперссылок на статьи, поскольку такие термины с большей вероятностью попадают в первые четыре отрезка. Указанная фильтрация позволяет значительно повысить точность, особенно в случае наборов данных с большим покрытием Википедией.

3.3 Обучение на положительных и неразмеченных примерах

3.3.1 Обзор существующих алгоритмов PU-learning

Среди существующих алгоритмов обучения на основе положительных и неразмеченных примеров можно выделить две основные категории:

1) двухшаговые алгоритмы (называемые также иногда эвристическими [96]);

2) алгоритмы, основанные на смещенных отрицательных примерах[4] (перевод названия «Techniques Based on Biased Negative», предложенного в обзорной работе Б. Джанга и В. Зуо [97]).

Далее эти категории будут рассмотрены подробнее.

Двухшаговые алгоритмы

Алгоритмы из данной категории основаны на следующей идее: на первом шаге выбрать среди неразмеченных примеров те, что с большой вероятностью

[4]Здесь и далее в этом разделе термин «пример» соответствует англоязычному термину *instance* и обозначает «объект, экземпляр из выборки».

являются отрицательными (так называемые «надежные отрицательные», или Reliable Negatives, RN), чтобы на втором шаге использовать их вместе с имеющимися положительными примерами для создания модели алгоритма машинного обучения с учителем.

Наиболее простой алгоритм, реализующий эту схему, устроен следующим образом (обозначим как P размеченные положительные примеры; U — неразмеченные примеры).

1. Обучить классификатор, то есть обычный алгоритм машинного обучения с учителем, на P как положительных и U как отрицательных.

2. Применить обученный классификатор к $U \setminus RN$ (на первой итерации $RN = \emptyset$.

3. Добавить к RN примеры, классифицированные как отрицательные

4. Обучить классификатор на P и RN.

5. Повторять шаги 2–5, пока извлекаются новые RN.

Данный алгоритм был упомянут в работе М. Монтеса и Х. Эскаланте [98] как традиционный алгоритм обучения на положительных и неразмеченных примерах без ссылки на чье-либо авторство, поэтому далее в данной диссертационной работе этот алгоритм будет называться Traditional.

Д. Фусилиэр и др. [99] предложили вместо итеративного увеличения числа надежных отрицательных примеров итеративно улучшать[5] множество надежных отрицательных примеров путем последовательного удаления из этого множества наименее вероятных (то есть наименее надежных) отрицательных примеров.

Схема алгоритма, названного авторами PU-LEA, представлена ниже.

1. Обучить классификатор на P как положительных и U_i как отрицательных.

2. Применить обученный классификатор к U_i.

3. Извлечь положительные примеры W_i из размеченного U_i.

[5] Авторы используют термин refine, что можно дословно перевести как «очищать»

4. Убрать из множества неразмеченных примеров положительные примеры: $U_{i+1} \leftarrow U_i - W_i$.

5. Повторять шаги 1–4, пока число положительных примеров, извлеченных на данном шаге, не превышает число положительных, извлеченных на предыдущем шаге: $|W_i| \leq |W_{i-1}|$.

Б. Лю и др. [100] приводят теоретическое обоснование разрешимости задачи обучения на положительных и неразмеченных примерах, причем для случаев как полностью корректных меток, так и наличия шума в разметке, однако сами отмечают, что, несмотря на это теоретическое обоснование разрешимости, «the constrained optimization problem required appears to be difficult to solve in practice» (Перевод: задача условной оптимизации на практике оказывается сложной для решения) [100].

Авторы предлагают эвристический метод, называемый S-EM (от Spy[6] EM). Основная идея заключается в том, чтобы использовать небольшую часть положительных примеров для определения порога классификации.

Более подробно схема извлечения RN с помощью данного алгоритма представлена ниже.

1. Выбрать случайным образом $s\%$ (10-15%) из всех положительных примеров: $S \leftarrow Sample(P, s\%)$ — так называемые «шпионы».

2. Обучить классификатор (EM-алгоритм с наивным байесовским классификатором) на $P \setminus S$ как положительных и $U \cup S$ как отрицательных.

3. Применить обученный классификатор к $U \cup S$.

4. Определить порог t на основе значений вероятности $P(positive)$ для примеров из S: отсортировать по возрастанию все примеры из S по вероятности и выбрать значение на уровне $l\%$ (15%), то есть близкое к минимуму, но допускающее некоторый уровень шума.

5. Положить в RN только те примеры из $U \cup S$, для которых $P(positive) > t$.

[6] Шпион (англ.)

В оригинальной работе на втором шаге, то есть построении классификатора с помощью P и RN, предлагается использовать также EM-алгоритм, однако в последующей работе тех же авторов [101] анализируется применение других классификаторов помимо EM-алгоритма на втором шаге и показывается, что в случае большого количества положительных примеров (больше 20%) лучшие результаты демонстрирует метод опорных векторов (SVM); в то же время сам алгоритм S-EM показывает стабильные результаты при разных условиях и является одним из наиболее эффективных алгоритмов при малом числе положительных примеров (10-20%).

Для однозначности именования будем обозначать через Spy те алгоритмы, которые используют тот же способ извлечения RN, что и S-EM.

Также к категории двухшаговых алгоритмов относятся методы NB [101], PEBL [102], Roc-SVM [103] и другие; однако, поскольку они не показывают превосходящих результатов [101], их рассмотрение выходит за рамки данной работы.

Алгоритмы, основанные на смещенных отрицательных примерах

Алгоритмы из данной категории рассматривают неразмеченные примеры как смещенные, или зашумленные, отрицательные примеры. Это позволяет применять алгоритмы машинного обучения с учителем, предназначенные для работы с зашумленными наборами данных, — так, на модификации метода опорных векторов основан алгоритм Biased SVM [101].

На такой трактовке неразмеченных примеров также основаны алгоритмы, назначающие веса неразмеченным примерам и использующие взвешенные примеры в специальным образом модифицированных алгоритмах машинного обучения с учителем: в частности, логистической регрессии в работе В. Ли и Б. Лю [104] и метода опорных векторов в работе Дж. Лю и др. [105].

Ч. Элкан и К. Ното [96] совершенствуют схему взвешивания и выделяют следующие ключевые отличия: каждому неразмеченному примеру назначается собственная пара весов, оценивающая принадлежность примера к положительным и отрицательным; кроме того, в отличие от предыдущих работ, подбирающих веса экспериментально на основе выборки, Ч. Элкан и К. Ното предлагают теоретически обоснованный способ вычисления весов в предположении, что положительные примеры выбираются случайно, что позволяет

значительно сократить вычислительную сложность. В качестве алгоритма машинного обучения с учителем в данной работе также используется метод опорных векторов.

Развитием этого подхода является метод RSVM [106]. Авторы также основываются на предположении о случайном выборе положительных примеров, однако вместо взвешенного метода опорных векторов применяют алгоритм ранжирования.

Более точно, предлагаемый авторами классификатор представляет собой следующую функцию:

$$f(x) = w^T x + \theta, \tag{3.2}$$

где $f(x) = 0$ — граница принятия решений (decision boundary), то есть положительными считаются те и только те примеры, для которых $f(x) > 0$.

На основе леммы, доказанной в работе Ч. Элкана и К. Ното [96], авторы показывают, что если найти такие w^T, при которых положительные примеры будут располагаться выше неразмеченных, то это будет означать, что положительные примеры будут располагаться выше отрицательных. Для поиска w^T, то есть обучения модели ранжирования, авторы предлагают использовать ранжирующий метод опорных векторов.

При этом подбор значений порога классификации θ и регуляризационного гиперпараметра C алгоритма ранжирующего метода опорных векторов осуществляется методом перекрестной проверки, максимизирующей так называемую «представительную F-меру» (proxy F-Measure) — аппроксимацию F-меры, использующую при вычислении только положительные и неразмеченные примеры, которая была предложена в работе В. Ли и Б. Лю [104].

3.3.2 Адаптация алгоритмов PU-learning

В данном разделе описывается выбор алгоритмов обучения на основе положительных и неразмеченных примеров и их адаптация к задаче извлечения терминов.

Напомним, что искомая функция $f(x)$ должна не просто классифицировать каждый кандидат x_i на термин и не термин, но проводить вероятностную классификацию, то есть оценивать вероятность того, что кандидат x_i является термином заданной предметной области.

Для дальнейшего экспериментального исследования из категории двухшаговых алгоритмов были выбраны два наиболее простых — Traditional и PU-LEA, — а также один из наиболее эффективных алгоритмов, т.е. Spy (с разными классификаторами).

Из второй категории был выбран алгоритм RSVM, так как он является наиболее современным и превосходит остальные методы, основанные на тех же принципах, а кроме того, за счет использования ранжирования перед классификацией, он естественным образом поддерживает вероятностную классификацию.

Адаптация двухшаговых алгоритмов

Как было показано выше, в алгоритмах Traditional, PU-LEA и Spy в качестве итоговой модели классификации используется модель алгоритма обучения с учителем.

Таким образом, возникает задача выбора конкретного алгоритма обучения с учителем, для чего формулируются следующие требования:

1) поддержка вероятностной классификации;
2) высокая эффективность при малом числе признаков;
3) высокая эффективность при относительно малом объеме данных для обучения и зашумленной выборке.

На основе первого требования и с учетом остальных были выбраны следующие алгоритмы:

1) наивный байесовский классификатор (NB);
2) логистическая регрессия (LR);
3) Random forest (RF).

Часто используемый в алгоритмах обучения на основе положительных и неразмеченных примеров, метод опорных векторов не рассматривался по ряду причин. Во-первых, SVM не поддерживает вероятностную классификацию и, хотя существуют методы ее оценки (например, [107, 108]), они вносят

дополнительную погрешность и уменьшают размер выборки, доступной для непосредственного построения модели.

Во-вторых, алгоритм на основе SVM, пусть и в виде модификации для ранжирования, используется в методе RSVM, который будет экспериментально исследоваться.

В-третьих, SVM чувствителен к значениям гиперпараметров (особенно в случае зашумленных данных), что значительно осложняет его применение на практике (особенно в случае данных небольшого объема).

Адаптация алгоритма RSVM

Как было отмечено выше, функция $f(x)$ вместо бинарной классификации производит ранжирование кандидатов по оценке вероятности быть термином предметной области и, таким образом, не использует порог классификации, вычисляемый методом RSVM в ходе построения модели. По этой причине нет необходимости оценивать параметр θ в выражении 3.2.

Это позволяет уменьшить вычислительную сложность подбора гиперпараметров, так как вместо оптимизации по θ, используется заранее заданное число кандидатов, которые необходимо извлечь из входной коллекции документов. Тем не менее, остается необходимость оптимизировать регуляризационный гиперпараметр C.

3.3.3 Выбор признаков

Как отмечалось выше, для автоматического извлечения терминов необходимо учитывать множество факторов, то есть признаков для алгоритма обучения; с другой стороны, увеличение количества признаков может привести к повышению корреляции между ними, в то время как вероятностная классификация более эффективна при низкой корреляции между признаками [109–111].

Решение этого противоречия основано на предположении, что признаки из разных категорий (например, на основе тематических моделях и на основе контекстов вхождений) обладают меньшей корреляцией. С учетом этого, были выбраны лучшие признаки из каждой категории, поддерживающие работу с терминами любой длины:

1. C-Value в модификации для поддержки однословных терминов — признак на основе частот вхождений.

2. Novel Topic Model — признак на основе тематических моделей.

3. Relevance — признак на основе анализа внешней коллекции документов.

4. Domain Model — признак на основе контекстов вхождений.

5. Близость к ключевым концептам — признак на основе Википедии, учитывающий близость к предметной области.

6. Вероятность быть гиперссылкой — признак на основе Википедии, учитывающий специфичность термина.

3.4 Экспериментальное исследование разработанного подхода

В данном разделе представлены результаты экспериментального исследования разработанного подхода. В ходе исследования используется та же методика тестирования — наборы данных, метод сбора кандидатов и метрика эффективности, — что и в предыдущей главе. И так же, как и в предыдущей главе, сначала описывается выбор параметров разработанного подхода, после чего производится его сравнение с существующими методами.

Кроме того, в конце раздела проверяется статистическая значимость полученных результатов и приводится сравнение с методами на основе алгоритмов машинного обучения с учетами.

3.4.1 Выбор параметров

Разработанный подход на основе частичного обучения содержит следующие параметры:

1) $S \in \mathbb{N}$ — количество кандидатов в термины, извлекаемых с помощью метода ComboBasic и используемых в качестве положительных примеров;

2) $\alpha \in \mathbb{R}^+$ — одноименный параметр в методе ComboBasic, соответствующий слагаемому e_t — количеству кандидатов, содержащих кандидата t;

3) $\beta \in \mathbb{R}^+$ — одноименный параметр в методе ComboBasic, соответствующий слагаемому e'_t — количеству кандидатов, содержащихся в кандидате t;

4) $WikiFilter \in \{true, false\}$ — булева переменная, означающая использование или неиспользование фильтра по наличию кандидата в Википедии при извлечении положительных примеров;

5) $puAlgo \in \{Traditional, PU\text{-}LEA, S\text{-}EM, Spy, RSVM\}$ — алгоритм обучения на основе положительных и неразмеченных примеров; при этом у алгоритмов Traditional, PU-LEA, S-EM и Spy имеется дополнительный параметр $puAlgoInner \in \{NB, LR, RF\}$ — используемый алгоритм машинного обучения с учителем.

В данной работе предлагается считать параметр $WikiFilter$ настраиваемым, то есть выбрать два набора остальных параметров для каждого значения $WikiFilter$, предпочитая одни и те же значения, если это возможно.

Иначе говоря, будет получено два метода — с фильтрацией и без фильтрации положительных примеров по Википедии. В первом случае подавляющее большинство извлеченных терминов будет содержаться в Википедии, однако некоторые специфичные термины могут быть пропущены. Если обратиться к иллюстрации из предыдущего раздела, то метод с фильтрацией по Википедии будет смещен в левую сторону, к терминам из другой предметной области; метод без фильтрации — в правую, к слишком специфичным словосочетаниям, не являющимися терминами.

Для выбора остальных параметров в данной работе используется следующая последовательность действий:

1) зафиксировать значение параметра S равным 100, предполагая, что число положительных примеров является характеристикой используемых признаков, а не предметной области или размера набора данных, и что значение $S = 100$ позволит достичь хорошей точности;

2) выбрать значение параметра $puAlgo$;

3) выбрать значения параметров α и β.

Рассмотрим второе и третье действия подробнее.

Выбор значения параметра $puAlgo$

Параметры $puAlgo$, α и β очевидным образом зависят друг от друга, в то время как оптимизация по всем разумным комбинациям не представляется возможной. По этой причине на первом шаге выбирается $puAlgo$ для начального приближения α и β — достаточно небольшого числа возможных значений α и β, — после чего производится более точная оптимизация для выбранного алгоритма PU-learning.

В качестве начального приближения α и β использовались все возможные комбинации следующих значений:
$\alpha \in \{0, 0.1, 0.25, 0.5, 1\}$;
$\beta \in \{0, 0.5, 1, 2, 3\}$.

В таблицах 3.1 и 3.2 представлены, соответственно, максимальные и средние значения средней точности для каждого алгоритма PU-learning при использовании фильтрации по Википедии. В таблицах 3.3 и 3.4 — также максимальные и средние значения средней точности для каждого алгоритма PU-learning, но без фильтрации. Курсивом выделены наибольшие значения для набора данных; жирным шрифтом — значения, отличающиеся от наибольших на процент и менее.

Алгоритм RSVM не тестировался на наборах данных Krapivin и FAO из-за большой вычислительной сложности (как временной, так и пространственной), а также из-за низкой эффективности на первых трех наборах данных.

Максимальные значения средней точности позволяют определить потенциальную эффективность алгоритма PU-learning, в то время как средние значения — получить представление о чувствительности алгоритма к конкретным значениям α и β.

На основе анализа таблиц можно сделать вывод, что показатели эффективности алгоритмов отличаются довольно сильно, причем большее влияние оказывает используемый алгоритм обучения с учителем, и что нельзя выделить один алгоритм, лучший на всех наборах данных.

Так, на наборе данных Patents лучшей эффективностью обладают алгоритмы на основе наивного байесовского классификатора, что можно объяс-

Таблица 3.1: Сравнение максимальных значений алгоритмов обучения на основе положительных и неразмеченных примеров с фильтрацией по Википедии

Метод	Board games	Patents	GENIA	Krapivin	FAO
Traditional NB	0.5107	***0.6467***	**0.7808**	0.4841	0.5134
Traditional LR	***0.5962***	0.6163	**0.7867**	0.5124	**0.5259**
Traditional RF	0.4820	0.6180	0.6907	0.5265	0.4419
PU-LEA NB	0.5114	**0.6432**	**0.7807**	0.4816	0.5101
PU-LEA LR	0.5803	0.6338	**0.7870**	0.5130	0.5139
PU-LEA RF	0.4868	0.6192	0.6897	0.5271	0.4369
S-EM	0.5113	**0.6416**	**0.7807**	0.4827	0.5112
Spy NB	0.5170	**0.6436**	**0.7807**	0.5104	0.5203
Spy LR	0.5474	0.6303	***0.7874***	0.5091	***0.5341***
Spy RF	0.5554	**0.6428**	0.6973	***0.5681***	0.4849
RSVM	0.4785	0.5133	0.7262	-	-

Таблица 3.2: Сравнение средних значений алгоритмов обучения на основе положительных и неразмеченных примеров с фильтрацией по Википедии

Метод	Board games	Patents	GENIA	Krapivin	FAO
Traditional NB	0.4620	**0.6149**	0.7609	0.4772	0.5102
Traditional LR	***0.5394***	0.6045	***0.7714***	0.4825	0.4934
Traditional RF	0.4428	0.5905	0.6638	**0.5211**	0.4353
PU-LEA NB	0.4605	**0.6137**	0.7605	0.4757	0.5067
PU-LEA LR	0.5142	**0.6146**	**0.7707**	0.4818	0.4728
PU-LEA RF	0.4443	0.5907	0.6635	**0.5207**	0.4321
S-EM	0.4578	**0.6147**	0.7605	0.4764	**0.5091**
Spy NB	0.4587	***0.6165***	0.7609	0.4835	**0.5061**
Spy LR	0.5135	0.6060	**0.7695**	0.4906	***0.5130***
Spy RF	0.4612	0.6012	0.6655	***0.5220***	0.4584
RSVM	0.3852	0.4807	0.6665	-	-

Таблица 3.3: Сравнение максимальных значений алгоритмов обучения на основе положительных и неразмеченных примеров без фильтрации по Википедии

Метод	Board games	Patents	GENIA	Krapivin	FAO
Traditional NB	0.4695	*0.5813*	0.7481	0.4833	0.4986
Traditional LR	***0.5466***	0.6291	**0.7845**	0.5105	0.4830
Traditional RF	0.4549	0.5408	0.6666	0.5169	0.4106
PU-LEA NB	0.4597	0.5794	0.7475	0.4794	0.4946
PU-LEA LR	**0.5466**	**0.6351**	0.7729	0.5096	0.4753
PU-LEA RF	0.4503	0.5361	0.6673	0.5148	0.4156
S-EM	0.4668	0.5761	0.7479	0.4811	0.4984
Spy NB	0.4684	0.5802	0.7453	0.4814	**0.5061**
Spy LR	***0.5496***	***0.6407***	***0.7828***	0.4623	***0.5095***
Spy RF	0.4647	0.5507	0.6738	***0.5493***	0.4466
RSVM	0.4100	0.5032	0.6783	-	-

Таблица 3.4: Сравнение средних значений алгоритмов обучения на основе положительных и неразмеченных примеров без фильтрации по Википедии

Метод	Board games	Patents	GENIA	Krapivin	FAO
Traditional NB	0.4242	*0.5251*	0.7351	0.4542	0.4449
Traditional LR	***0.4663***	**0.5549**	**0.7477**	**0.4842**	0.4143
Traditional RF	0.4033	0.5050	0.6430	0.4433	0.3339
PU-LEA NB	0.4254	0.5254	0.7350	0.4527	0.4428
PU-LEA LR	0.4449	**0.5574**	0.7367	**0.4839**	0.4120
PU-LEA RF	0.4036	0.5058	0.6428	0.4428	0.3357
S-EM	0.4227	0.5255	0.7345	0.4511	**0.4458**
Spy NB	0.4269	0.5252	0.7352	0.4603	**0.4453**
Spy LR	0.4547	***0.5557***	***0.7539***	0.4290	***0.4537***
Spy RF	0.4049	0.5223	0.6457	***0.4900***	0.3755
RSVM	0.3213	0.4258	0.6248	-	-

нить тем, что на малых объемах выборки наивный байесовский классификатор часто показывает лучшие результаты. Алгоритм Random forest значительно опережает другие алгоритмы обучения с учителем на наборе данных Krapivin, однако показывает удивительно низкие результаты на наборах данных GENIA и FAO, где все остальные алгоритмы показывают очень близкую эффективность.

В целом же наиболее стабильные результаты показывает алгоритм Traditional с логистической регрессией в качестве алгоритма обучения с учителем.

Выбор значений параметров α и β

Поскольку зависимость параметров α и β наиболее сильна, для выбора их значений был произведен решетчатый поиск (grid search) со следующими узлами:

$\alpha \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.75, 1\}$

$\beta \in \{0, 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3\}$

— всего 90 комбинаций.

Так же, как и в разделе 2.3.2, для визуализации результатов использовалась теплокарта — см. рисунки 3.3-3.12.

Для метода с использованием фильтра по Википедии были выбраны значения $\alpha = 0.15$ и $\beta = 3$ (крайний справа и четвертый снизу прямоугольник) — это локальный максимум для наборов данных Krapivin, Board games и Patents и достаточно хорошее значения для FAO и GENIA (результаты хуже максимума примерно на 1 процент).

Для метода без использования фильтра по Википедии были выбраны значения $\alpha = 1$ и $\beta = 0.1$ (второй слева и самый верхний прямоугольник) — это локальный максимум для наборов данных Patents и FAO; для Krapivin и GENIA результаты хуже максимума примерно на 2 процента; для Board games результаты значительно хуже — на 8 процентов — однако зоны максимумов Board games сильно отличаются от зон максимумов GENIA и Patents.

Выбранные значения параметров

Итак, были выбраны следующие значения параметров:

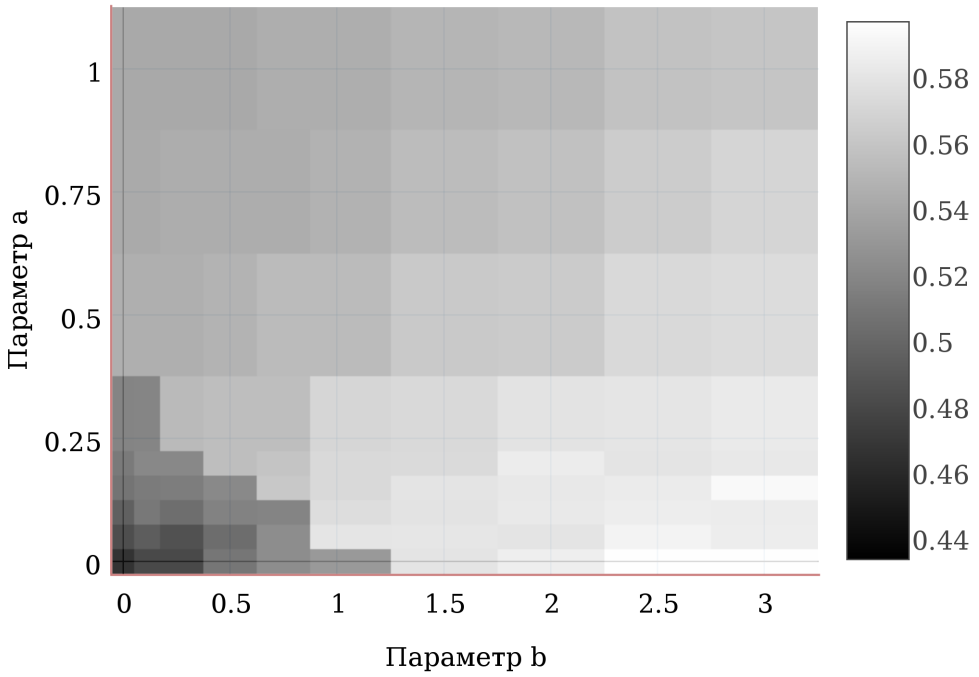

Рисунок 3.3: Зависимость средней точности от значений параметров α и β для набора данных Board games с фильтром по Википедии

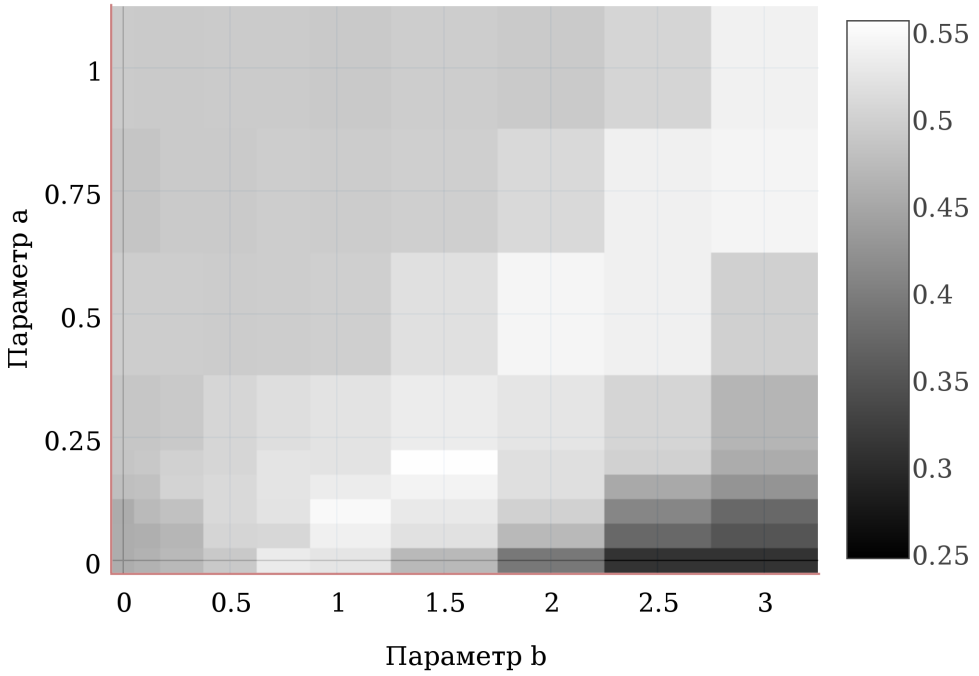

Рисунок 3.4: Зависимость средней точности от значений параметров α и β для набора данных Board games без фильтра по Википедии

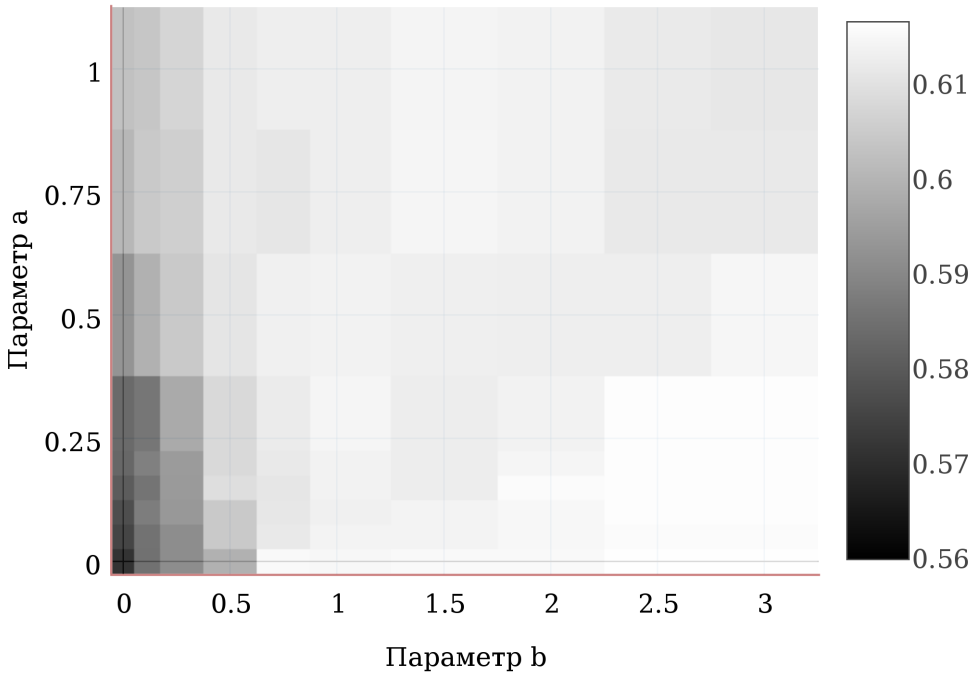

Рисунок 3.5: Зависимость средней точности от значений параметров α и β для набора данных Patents с фильтром по Википедии

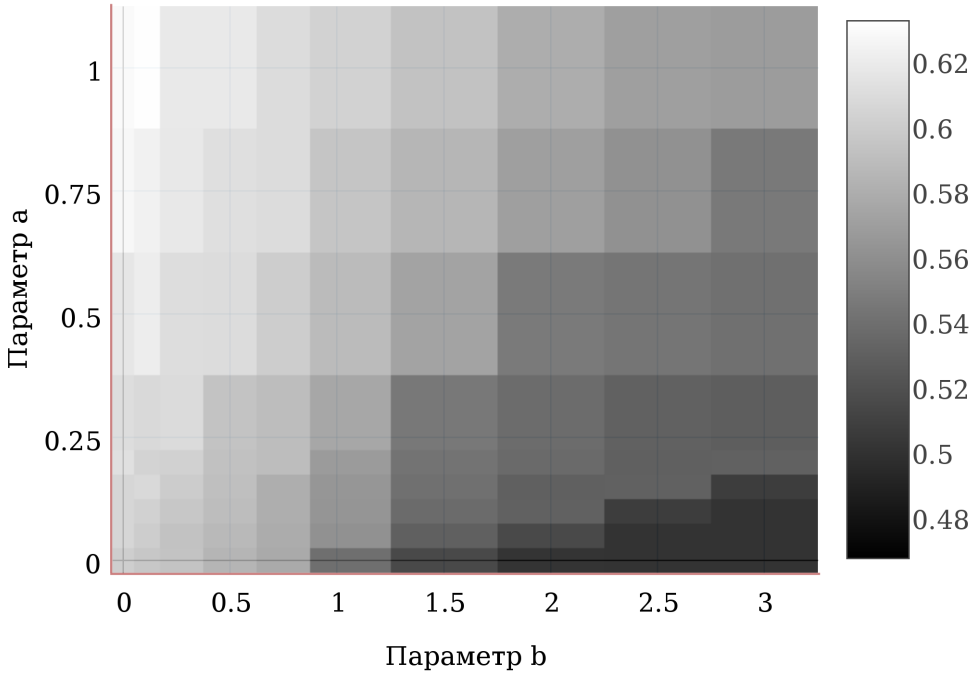

Рисунок 3.6: Зависимость средней точности от значений параметров α и β для набора данных Patents без фильтра по Википедии

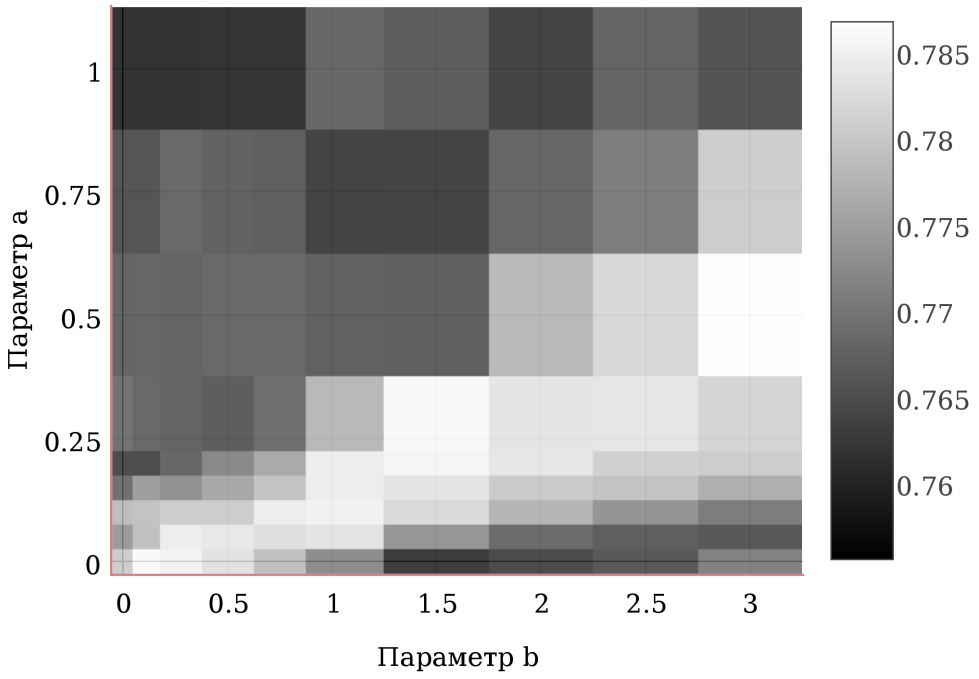

Рисунок 3.7: Зависимость средней точности от значений параметров α и β для набора данных GENIA с фильтром по Википедии

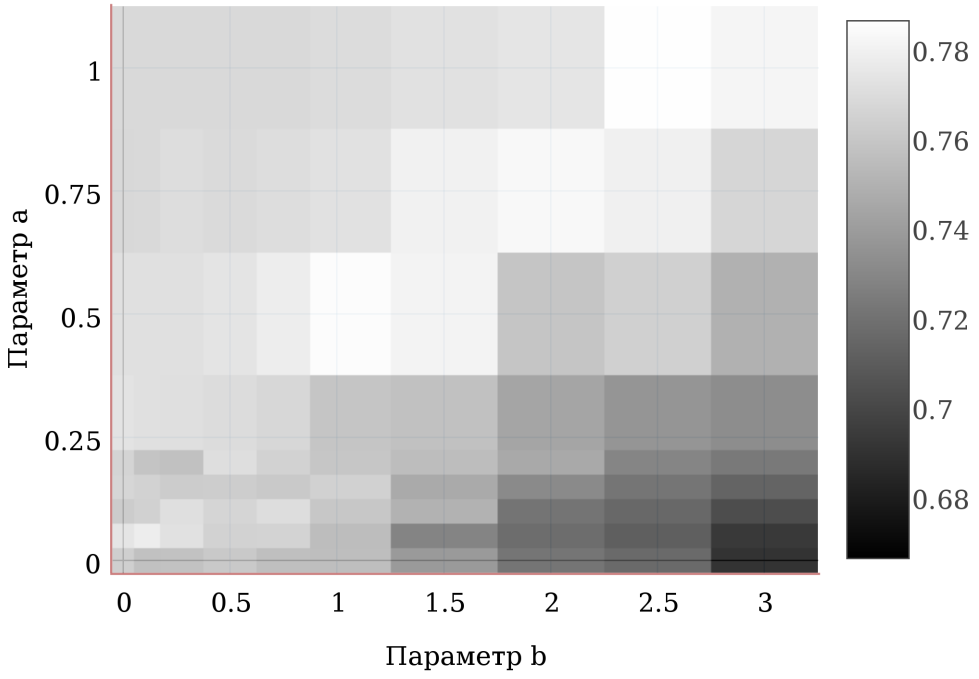

Рисунок 3.8: Зависимость средней точности от значений параметров α и β для набора данных GENIA без фильтра по Википедии

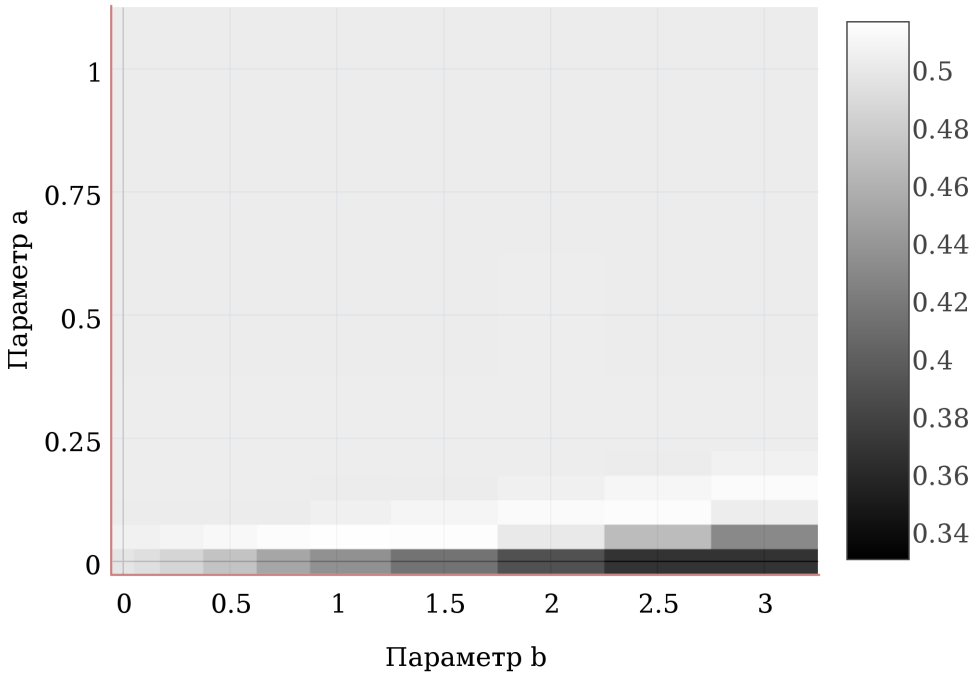

Рисунок 3.9: Зависимость средней точности от значений параметров α и β для набора данных Krapivin с фильтром по Википедии

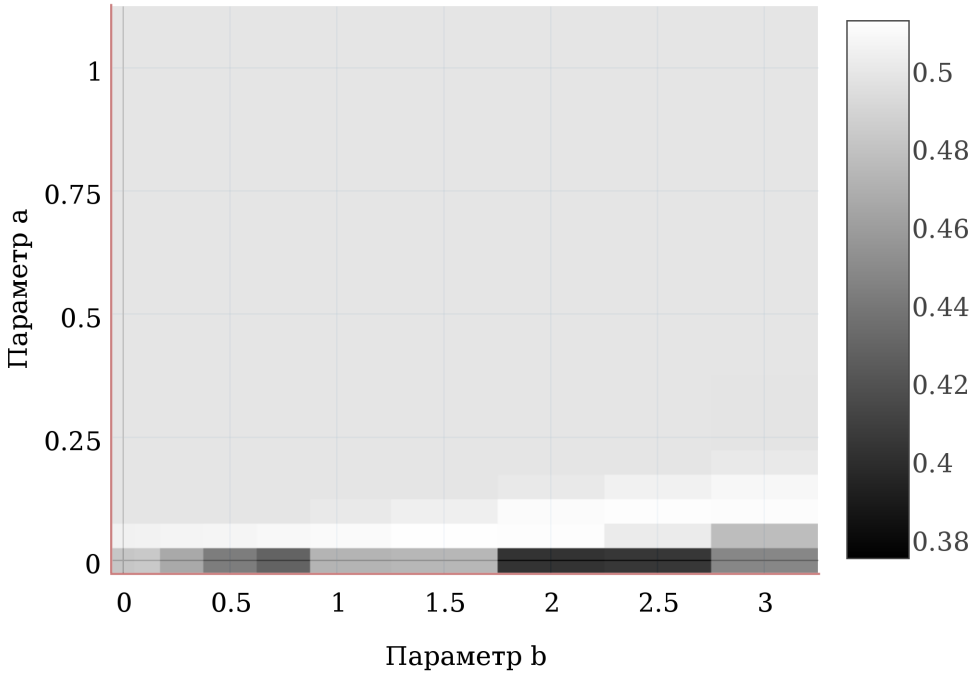

Рисунок 3.10: Зависимость средней точности от значений параметров α и β для набора данных Krapivin без фильтра по Википедии

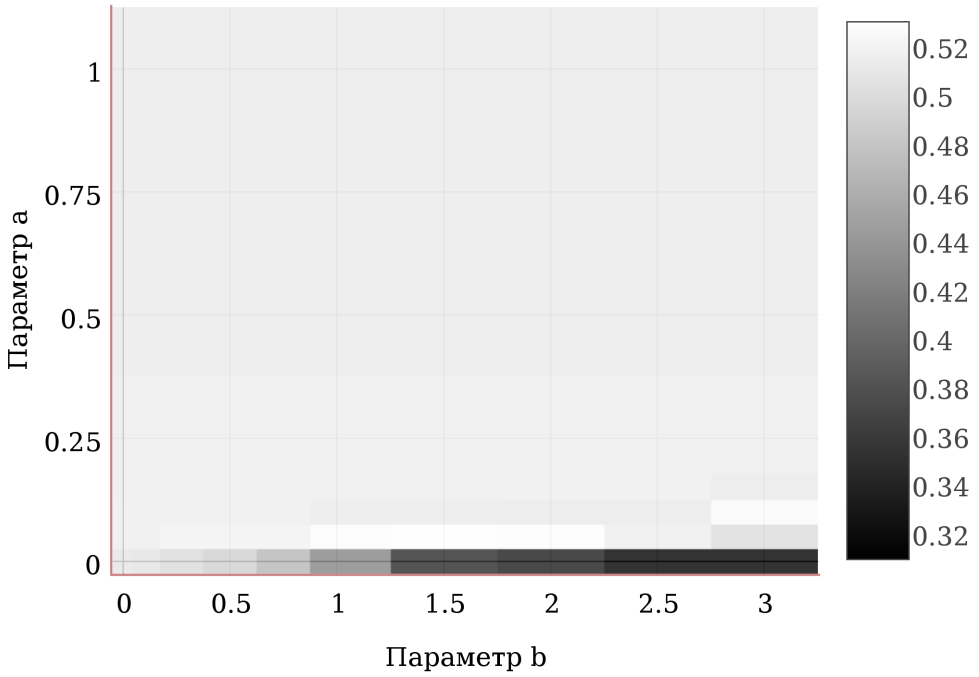

Рисунок 3.11: Зависимость средней точности от значений параметров α и β для набора данных FAO с фильтром по Википедии

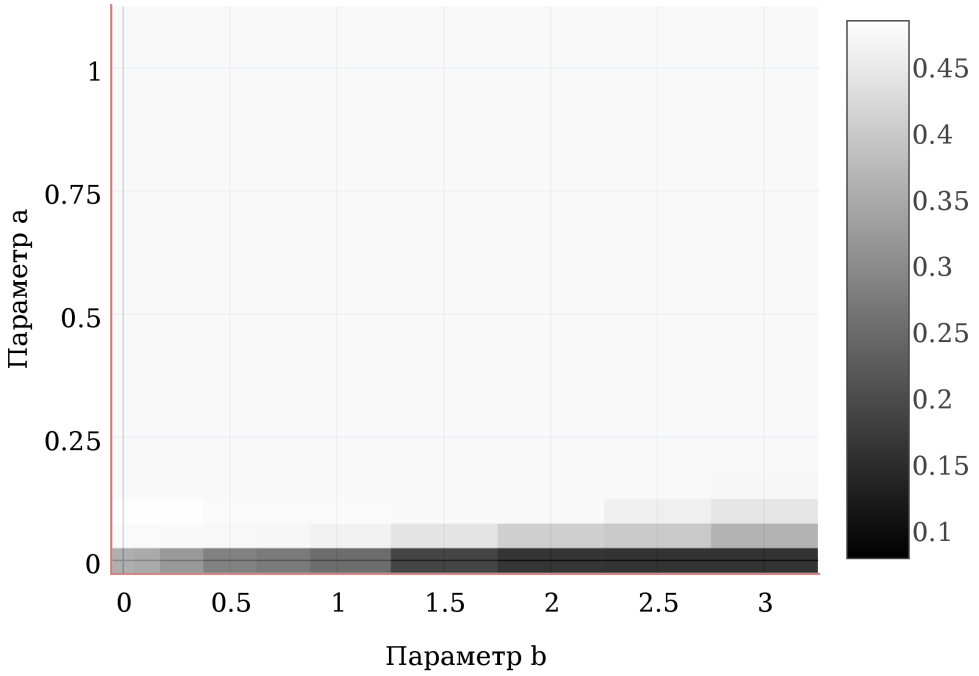

Рисунок 3.12: Зависимость средней точности от значений параметров α и β для набора данных FAO без фильтра по Википедии

1) $S = 100$;

2) $puAlgo = TraditionalLR$;

3) если $WikiFilter = true$, то $\alpha = 0.15$, $\beta = 3$;

4) если $WikiFilter = false$, то $\alpha = 1$, $\beta = 0.1$.

Примеры результатов работы предложенного подхода для выбранных значений параметров см. в приложении A.

3.4.2 Сравнение разработанного подхода с существующими методами

В таблице 3.5 представлены результаты сравнения разработанного подхода с существующими методами, детали реализации которых были изложены в разделе 2.3.3.

Предпоследняя строка таблицы — Traditional LR Wiki — представляет результаты (среднюю точность) для разработанного метода с фильтрацией положительных примеров по Википедии; последняя строка — Traditional LR — без фильтрации положительных примеров по Википедии.

Как видно из таблицы, разработанные методы превосходят по средней точности лучшие из существующих методов на всех наборах данных.

При этом метод, использующий фильтрацию положительных примеров по Википедии (Traditional LR Wiki), показывает более высокую эффективность для четырех из пяти наборов данных по сравнению с методом без такой фильтрации. Если же учесть, что на единственном исключении — наборе данных Patents — лучший результат по сравнению с методом Traditional LR Wiki показал метод C-Value, который извлекает только многословные термины, то можно сказать, что разработанный метод, использующий фильтрацию положительных примеров по Википедии, показывает наибольшую эффективность среди методов, извлекающих термины любой длины.

В приложении B показаны зависимости точности от числа извлекаемых терминов разработанного подхода и трех лучших существующих методов для каждого набора данных.

Таблица 3.5: Сравнение разработанного подхода с существующими методами

Метод	Board games	Patents	GENIA	Krapivin	FAO
TF-IDF	0.3882	0.4922	0.6936	0.3619	**0.4270**
C-Value	0.3350	**0.6271**	0.7294	0.3706	0.3731
Ventura C-Value (0.1)	0.3967	0.5874	0.7376	0.3911	0.4080
Weirdness	0.3121	0.3972	0.5289	0.2562	0.2853
TermExtractor	0.3526	0.4974	0.7331	0.3209	0.1543
Relevance	0.3797	0.4104	0.5338	0.3175	0.3488
Domain Relevance	0.3253	0.4943	**0.7425**	0.3218	0.1534
Domain Consensus	0.2976	0.2618	0.6296	0.2246	0.1535
Domain Model	0.3680	0.3992	0.6720	0.3929	0.3389
Domain Model (Vote)	0.3821	0.4594	0.6729	**0.4609**	0.3447
Basic (Domain Model)	0.3539	0.5051	0.6475	0.4141	0.3486
Novel Topic Model	0.3684	0.5426	0.7129	0.1271	0.0593
Wiki Categories (NC)	**0.4110**	0.2934	0.7157	0.1671	0.0765
LinkProbability	0.4482	0.4522	0.7276	0.1815	0.0169
KeyConceptRelatedness	**0.5470**	0.5237	0.7253	0.2282	0.1089
Traditional LR Wiki	**0.5925**	0.6161	**0.7745**	**0.5128**	**0.5117**
Traditional LR	**0.4756**	**0.6321**	**0.7645**	**0.4957**	**0.4742**

3.4.3 Проверка статистической значимости

В данном разделе рассматривается вопрос, насколько случайно — точнее, насколько статистически значимо — превосходство разработанного подхода над существующими методами.

Для проверки статистической значимости необходимо сформулировать статистические гипотезы, для чего, в свою очередь, необходимо определить, что является наблюдением, выборкой и распределением.

Естественное решение: считать *наблюдением* результат конкретного метода на конкретном наборе данных, *выборкой* — множество результатов конкретного метода на указанных 5 наборах данных, *распределением* — множество результатов конкретного метода на всевозможных наборах данных, — в данном случае некорректно, так как параметры разработанного подхода оптимизировались для набора данных и поэтому выборка не является репрезентативной.

В данной диссертационной работе предлагается использовать метод генерации повторной выборки (resampling), а именно — метод расщепления выборки, или метод «складного ножа» (jackknife) [112, 113], который заключается в следующем: исходная выборка разбивается на N равных частей (как правило, N равно числу элементов в выборке), после чего формируется N новых выборок путем выбрасывания из исходной выборки по одной части, и уже над этими N выборками считаются различные статистики.

Перед определением наблюдения, выборки и распределения в нашем случае, следует отметить, что оценка эффективности (в частности, средней точности) любого метода извлечения терминов определяется двумя событиями: коллекцией входных документов, из которых извлекались термины, и терминами-эталонами, с которыми сравниваются извлеченные термины. Другими словами, метод может показать превосходящую среднюю точность из-за особенностей входных текстов и/или размеченных экспертами терминов — к особенностям последних можно отнести ошибки экспертов либо выделение только части терминов, как в наборах данных Krapivin и FAO. Таким образом, для применения метода расщепления выборки необходимо учитывать как множество входных документов, так и множество терминов-эталонов.

Итак, в соответствии с методом расщепления выборки разобьем множество документов на N равных частей и множество терминов-эталонов также на N равных частей. Будем называть *сэмплом набора данных* пару (d_i, e_i), где d_i — коллекция документов, полученная путем выбрасывания из всего множества документов i-той части; e_i - множество терминов, также полученная путем выбрасывания из всего множества терминов-эталонов i-той части.

Тогда *наблюдение* — это результат конкретного метода на конкретном сэмпле набора данных; *выборка* — множество результатов конкретного метода на всех N сэмплах наборах данных; *распределение* — множество результатов конкретного метода на всевозможных семплах набора данных (или на всех данных такого типа, то есть таких размеров, предметной области и т.п.)

Гипотеза H_0 состоит в том, что выборки получены из одинаковых распределений (или результаты методов одинаковы на данных такого типа). Гипотеза H_0 (доминирования) — разработанный подход работает лучше (на данных такого типа).

В качестве статистического критерия будем использовать критерий знаковых рангов Уилкоксона [114].

Результаты сравнения разработанных методов с лучшими из существующих для каждого набора данных представлены в таблице 3.6. Для наборов данных Patents, Board games и GENIA $N = 16$ (количество документов в наборе данных Patents), для Krapivin и FAO $N = 5$ ввиду значительно большего объема и, как следствие, времени работы методов.

Таблица 3.6: Оценка статистической значимости превосходства разработанного подхода над существующими методами

Набор данных	Сравниваемый метод	p-value для Traditional LR Wiki	p-value для Traditional LR
Board games	Wiki Categories	0.00001526	0.00001526
Patents	C-Value	**1**	**0.4301**
Patents	Ventura C-Value	0.00001526	0.00001526
GENIA	Domain Relevance	0.00001526	0.00001526
Krapivin	Domain Model (Vote)	0.03125	0.03125
FAO	TF-IDF	0.03125	0.03125

Как видно из таблицы, для всех наборов данных, кроме Patents, разработанный подход превосходит лучшие из существующих методов с уровнем значимости 0.05.

В случае набора данных Patents статистически значимо превосходство разработанного подхода над лучшим из существующих методов, предназначенных для извлечения терминов любой длины. При этом превосходство метода C-Value над Traditional LR Wiki является статистически значимым (p-value = 0.00003), но превосходство метода C-Value над методом без фильтрации уже не является таковым (p-value = 0.5896).

3.4.4 Сравнение разработанного метода с методом на основе обучения с учителем

Определенный интерес представляет сравнение разработанного метода, основанного на алгоритме машинного обучения и не требующего размеченных данных, с методами на основе машинного обучения с учителем.

Для экспериментального исследования использовалась перекрестная проверка (cross-validation) по множеству кандидатов в термины, которая применялась во многих работах [19–21] и состоит в следующем. Из всей коллекции документов извлекаются кандидаты в термины, для которых вычисляются значения признаков — также на основе всей коллекции. После этого множество кандидатов разбивается на k равных части, каждая из которых поочередно используется как тестовое множество, в то время как остальные части служат тренировочным множеством; результаты, полученные на тестовых множествах, усредняются и считаются финальной оценкой. В случае алгоритма, не требующего обучения, тренировочные множества игнорируются.

Заметим, что разработанный подход на основе алгоритма PU-learning хоть и не требует размеченных данных, однако использует информацию, содержащуюся во всех кандидатах в термины, — в некотором смысле, «обучается» на множестве кандидатов, причем тестовом множестве, в отличие от алгоритмов машинного обучения с учителем, обучающихся на тренировочном множестве. Отсюда следует, что для корректного сравнения тренировочное и тестовое множество не должны отличаться по каким-либо параметрам, в том числе размеру, поэтому в данном случае использовалось разбиение на 2 части (2-fold cross-validation).

В таблице 3.7 приведены значения средней точности с тем же порогом, то есть длиной верхней части списка отсортированных кандидатов, что и в предыдущих экспериментах. В верхней части таблицы показаны результаты для отдельных признаков; в средней части — для методов, комбинирующих выбранные 5 признаков (C-Value, Novel Topic Model, Domain Model, LinkProbability, KeyConceptRelatedness) и не требующих обучения; в нижней части — для методов, основанных на машинном обучении с учителем. В качестве алгоритмов обучения с учителем использовались наивный байесовский классификатор, логистическая регрессия и Random forest по тем же причинам, которые обусловили выбор этих алгоритмов как части двухшаговых методов обучения на основе положительных и неразмеченных примеров (см. раздел 3.3.2).

Как видно из таблицы, разработанные методы значительно превосходят алгоритм голосования и несколько уступают методам на основе алгоритмов

Таблица 3.7: Сравнение разработанного подхода с методами на основе машинного обучения с учителем

Метод	Board game	Patents	GENIA	Krapivin	FAO
TF-IDF	0.2850	0.4096	0.6801	0.2978	0.3469
C-Value	0.2453	**0.5537**	0.6899	0.3251	**0.3517**
Ventura C-Value	0.3080	0.5303	**0.7153**	0.3353	0.3503
Weirdness	0.2838	0.3005	0.5490	0.1893	0.2712
TermExtractor	0.2944	0.4417	0.7013	0.2628	0.1121
Relevance	0.3188	0.3044	0.5521	0.2521	0.3085
Domain Relevance	0.2742	0.4541	0.7112	0.2630	0.1122
Domain Consensus	0.2032	0.2482	0.6116	0.1845	0.1482
Domain Model (Vote)	0.3059	0.3961	0.6505	**0.3980**	0.3200
Novel Topic Model	0.3112	0.4834	0.6821	0.1204	0.0474
Wiki Categories (NC)	0.3711	0.2828	0.6442	0.1583	0.0650
Link Probability	0.3664	0.3864	0.6868	0.1627	0.0209
KeyConceptRelatedness	**0.4672**	0.4060	0.6887	0.2039	0.1168
Алгоритм голосования	0.3939	0.4789	0.7155	0.2667	0.2333
Traditional LR Wiki	**0.4363**	0.5191	0.7309	0.4370	0.4373
Traditional LR	0.3452	0.5696	0.7355	0.4232	0.4148
Наивный байесовский классификатор	0.3091	0.5609	**0.7794**	0.4033	0.4747
Логистическая регрессия	0.2996	**0.5803**	0.7666	0.4323	0.4349
Random forest	0.1767	0.5457	0.7523	**0.4541**	**0.4853**

обучения с учителем, причем для набора данных Board games результаты разработанных методов существенно выше, что можно объяснить высоким покрытием Википедией этого набора данных.

3.5 Выводы

В данной главе представлен новый подход к извлечению терминов из коллекции текстов предметной области, использующий алгоритм обучения на основе положительных и неразмеченных примеров.

Разработанный подход обладает лучшей средней точностью по сравнению с существующими методами, не требующими размеченных данных, и срав-

нимой — с методами, основанными на алгоритмах машинного обучения с учителем.

Кроме того, разработанный подход включает в себя параметры, изменение которых имеет интуитивную интерпретацию. Так, увеличение параметра β приводит к извлечению более специфичных терминов; увеличение параметра α — терминов средней специфичности; включение фильтрации положительных примеров по Википедии снижает количество извлеченных слов и словосочетаний, не являющихся терминами, однако может приводить к извлечению терминов других предметных областей и к пропуску терминов высокой специфичности. Кроме того, метод без фильтра по Википедии может быть более эффективен при извлечении большого числа терминов (превышающего покрытие Википедией для данной предметной области), см. приложение В.

Возможность априорной настройки параметров исходя из знаний об особенностях конкретных приложений и предметных областей может быть особенно полезно при применении подхода на практике.

Глава 4

Программная система извлечения терминов

Настоящая глава посвящена программной системе, реализующей разработанные методы.

В первом разделе описывается общая архитектура системы, приводятся разбиение на модули и диаграммы классов основных модулей. Во втором разделе анализируется вычислительная сложность использованных алгоритмов: доказываются оценки временной и пространственной сложности вычисления каждого признака и разработанного подхода на основе алгоритма PU-learning. В последнем разделе описываются особенности программной системы, использованные технологии и оптимизации.

4.1 Общая архитектура программной системы

На рисунке 4.1 показана диаграмма потоков данных.

В соответствии с основными процессами обработки данных выделяются следующие модули программной системы.

1. Модуль обработки документов, производящий чтение документов, а также — с помощью системы Текстерра (см. 4.3.1) — разбиение их на предложения и токены, определение частей речи и ключевых концептов.

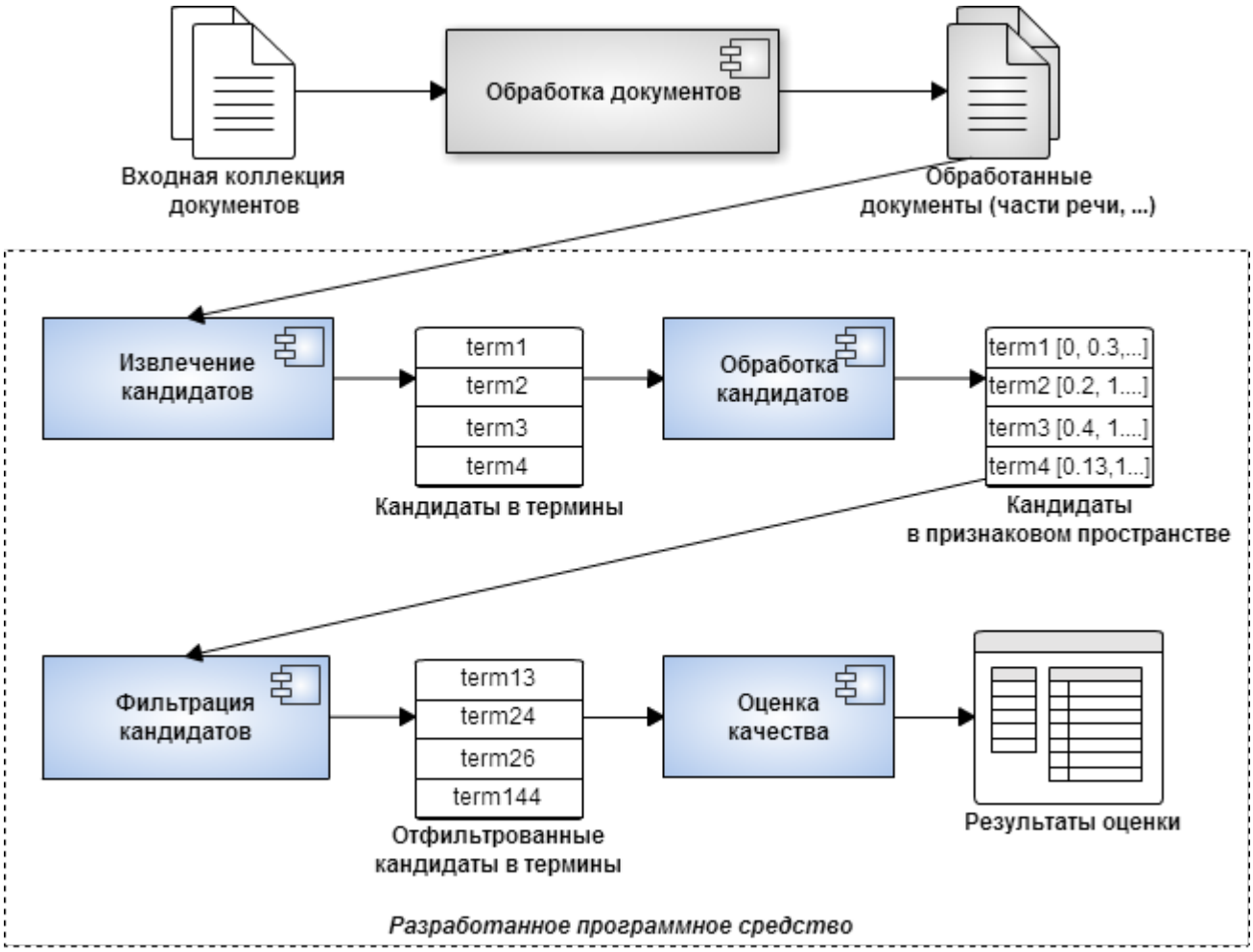

Рисунок 4.1: Диаграмма потоков данных

2. Модуль извлечения кандидатов в термины, собирающий N-граммы из токенов и фильтрующий их по шаблонам частей речи, частоте вхождений и наличию стоп-слов.

3. Модуль обработки терминов, вычисляющий признаки и производящий вероятностную классификацию.

4. Модуль фильтрации терминов, производящий отбор кандидатов на основе значений, полученных на этапе обработки терминов.

5. Модуль оценки эффективности извлеченных терминов, подсчитывающий метрики точности, полноты и средней точности.

Архитектура модели данных представлена на рисунке 4.4.

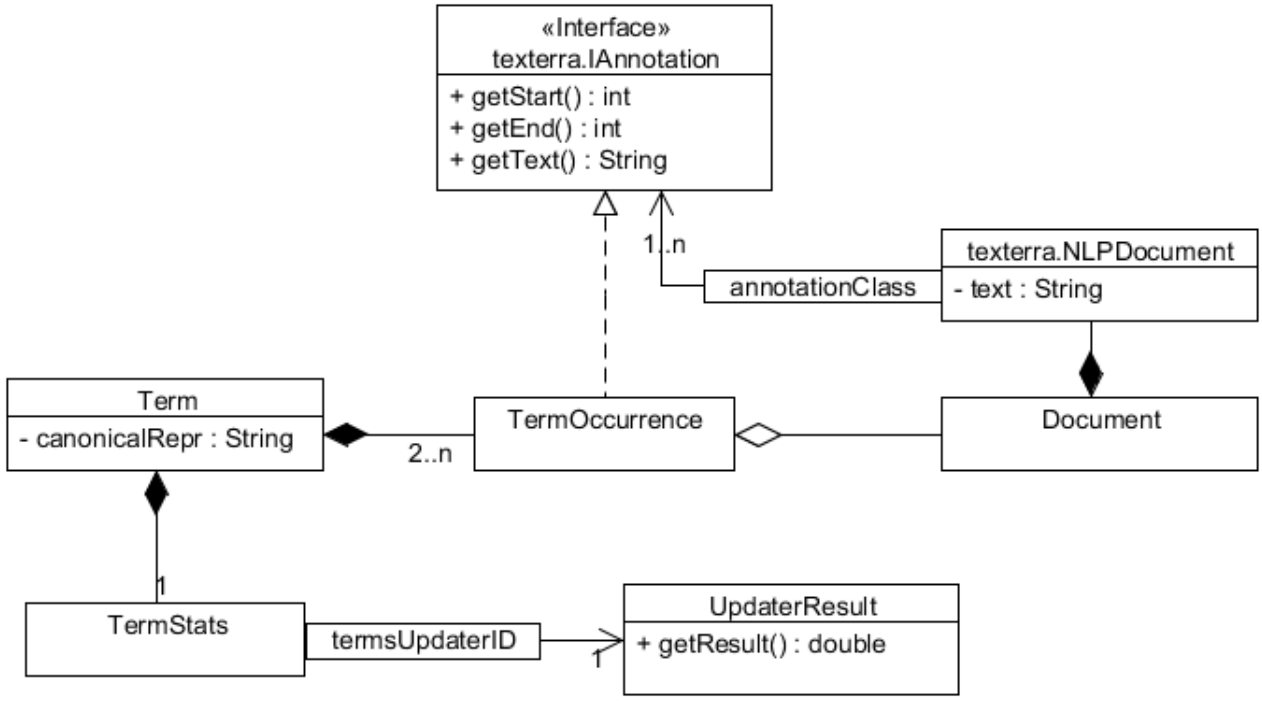

Рисунок 4.2: Архитектура модели данных

Term является базовым классом системы, представляющим собой как кандидаты в термины, извлекаемые в самом начале работы, так и финальные термины, возвращаемые в качестве результата работы.

Каждый объект класса *Term* обладает уникальным «каноническим» текстовым представлением и содержит набор вхождений — объектов класса *TermOccurrence*, — а также — в объекте класса *TermStats* — вычисленные значения признаков, в том числе и результаты вероятностной классификации, в виде отображения из уникального идентификатора класса, вычисляющего признак, в значение (*UpdaterResult*).

Класс *TermOccurrence* представляет собой реализацию интерфейса *IAnnotation* из системы Текстерра и хранит ссылку на документ, в котором и встретилось это вхождение. Отметим, что класс *Document* инкапсулирует в себе объект класса *NLPDocument* — документ с необходимыми аннотациями (границы предложений и токенов, части речи, ключевые концепты), полученными с помощью системы Текстерра.

Архитектура модуля извлечения кандидатов в термины представлена на рисунке 4.3.

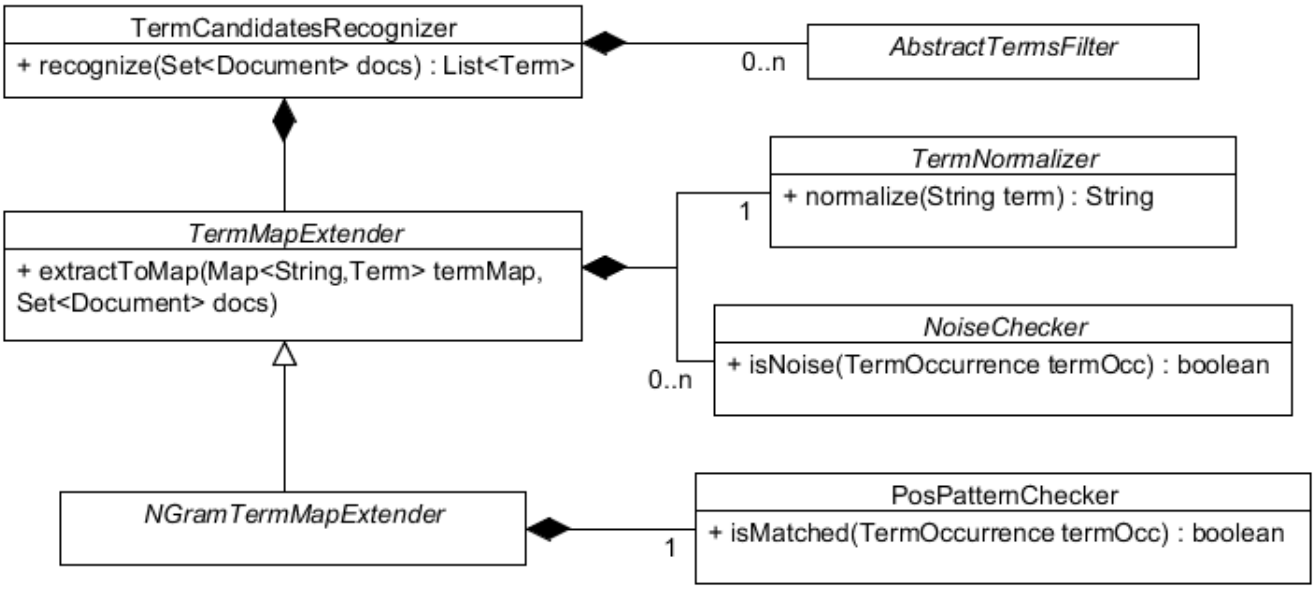

Рисунок 4.3: Архитектура модуля извлечения кандидатов

TermCandidatesRecognizer собирает термины с помощью объектов класса *TermMapExtender* и производит их первоначальную фильтрацию (*AbstractTermsFilter*), например по частоте вхождений.

В свою очередь, наследники абстрактного класса *TermMapExtender* собирают отображение из канонических представлений термина, получаемых с помощью нормализации текстовых строк (*TermNormalizer*), в объекты класса *Term*; при этом на данном этапе происходит фильтрация ошибочных вхождений (*NoiseChecker*), например имеющих длину меньше двух символов.

Так, используемый по умолчанию абстрактный класс *NGramTermMapExtender* собирает все N-граммы (порядок N определяется в наследниках), фильтруя их по предопределенным шаблонам частей речи (*PosPatternChecker*).

Архитектура модуля обработки терминов представлена на рисунке 4.4.

AbstractTermsUpdater представляет собой абстрактный класс, позволяющий вычислить значение признака для каждого кандидата в термины, в том числе используя результаты других объектов класса *AbstractTermsUpdater*.

ConsecutiveTermsUpdater предназначен для вычисления признаков, зависящих только от текущего кандидата. К таковым относятся большая часть признаков, например, C-Value, Novel Topic Model, Relevance, LinkProbability, KeyConceptRelatedness и другие.

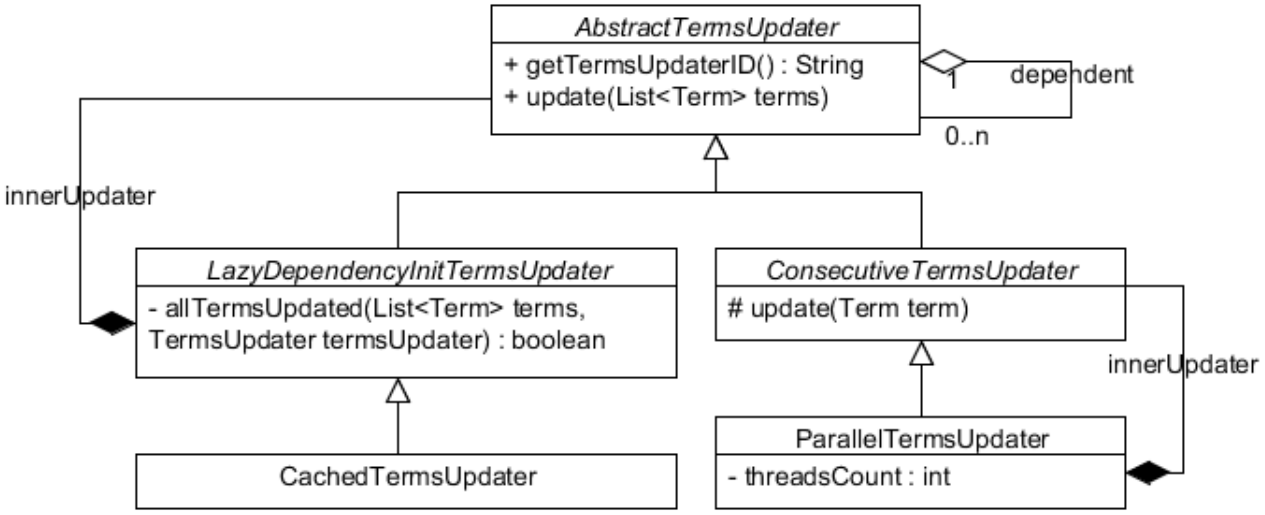

Рисунок 4.4: Архитектура модуля обработки терминов

ParallelTermsUpdater реализует шаблон проектирования «Декоратор», позволяя проводить параллельное вычисление признака, зависящего только от текущего кандидата.

LazyDependencyInitTermsUpdater реализует шаблон проектирования «Декоратор», для заданного объекта класса *AbstractTermsUpdater* автоматически вызывая зависимые объекты того же класса, если они не были уже вызваны.

CachedTermsUpdater расширяет функциональность класса *LazyDependencyInitTermsUpdater*, позволяя сохранить в объектах класса *Term* посчитанные значения признаков.

Архитектура модуля фильтрации терминов представлена на рисунке 4.5.

AbstractTermsFilter представляет собой абстрактный класс, позволяющий отфильтровать термины на основе значения определенного признака (*AbstractTermsUpdater*).

Так, *TopTermsFilter* сортирует все термины (или кандидаты в термины) по значению определенного признака и оставляет лучшие *topCount* терминов.

Большая часть остальных фильтров является наследником абстрактного класса *ConsecutiveTermsFilter*, в котором решение, оставить термин или нет, зависит только от этого термина и не зависит от остальных. К таковым фильтрам относятся, например, *ThresholdsTermsFilter*, оставляющий только те термины, значения признака для которых попадает в определенный интервал, или *EqualityTermsFilter*, используемый обычно для фильтрации на основе бинарных признаков.

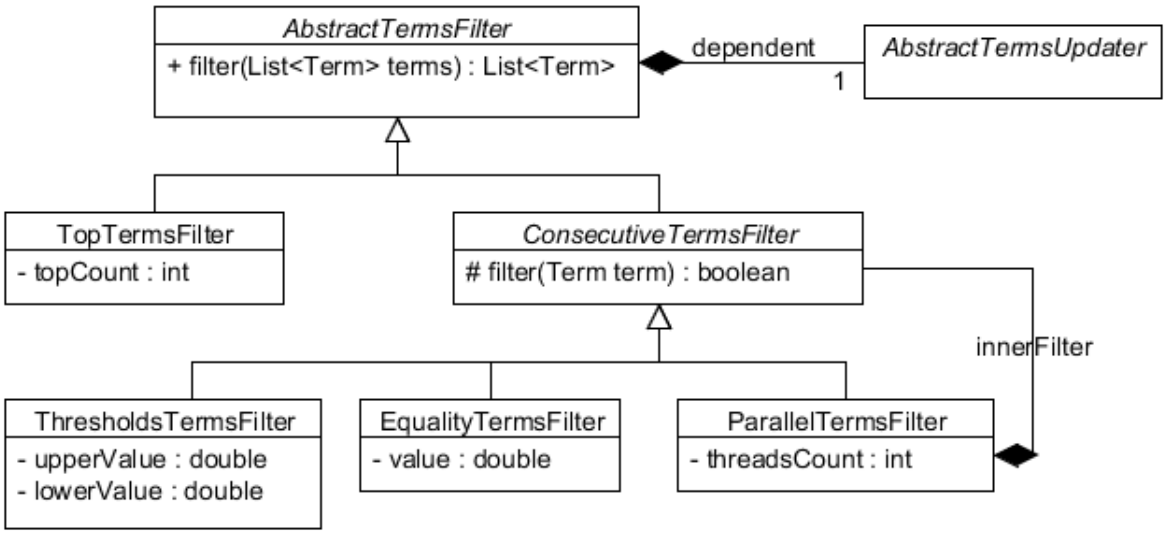

Рисунок 4.5: Архитектура модуля фильтрации терминов

ParallelTermsFilter реализует шаблон проектирования «Декоратор», позволяя проводить параллельную фильтрацию терминов.

Модуль оценки эффективности использует фреймворк QTF (см. 4.3.1). Для данного фреймворка были написаны классы-адаптеры, реализующие следующую функциональность, специфичную для приложения:

1) чтение набора данных для тестирования эффективности — обработка документов и извлечение кандидатов в термины;

2) ранжирование кандидатов в термины — применение существующих и разработанных методов вероятностной классификации кандидатов в термины.

Кроме того, для каждого сценария экспериментального исследования (например, подбор параметров для того или иного метода) был разработан класс в соответствии с шаблоном проектирования «Строитель» (Builder).

4.2 Анализ вычислительной сложности алгоритмов

В данном разделе приводится оценка вычислительной сложности (временной и пространственной) разработанного алгоритма. Поскольку основное

предназначение этой оценки заключается в сравнении с другими методами, при ее расчете не будут учитываться такие этапы, как обработка документов (за исключением этапа определения ключевых концептов) и извлечение кандидатов, поскольку они являются общими для всех методов.

Для каждого термина все его вхождения вычисляются на этапе извлечения кандидатов, поэтому не будем учитывать пространственную и временную сложность для хранения и создания такой структуры.

Используемые обозначения представлены в таблицах 4.1 и 4.2.

Таблица 4.1: Обозначения для параметров входных данных

Обозначение	**Обозначаемое**
n	Число терминов (кандидатов в термины)
m	Среднее число вхождений терминов
D	Число документов входной коллекции
n_{lt}	Среднее число терминов, содержащих другие термины
w	Среднее число слов в термине
D_{avg}	Среднее число документов, в которых встречается один и тот же термин
V_{avg}	Среднее число различных слов в документе после фильтров по частоте и стоп-словам (для построения тематической модели)
V	Размер словаря, или общее число разных слов
n_c	Среднее количество концептов термина
w_f	Число слов после фильтров по частям речи и распространенности (для построения модели домена)
v_{all}	Общее число слов (токенов) во всех документах

Лемма 1. *Временная сложность признака C-Value составляет* $O(n+nn_{lt}+nw^2)$.

Пространственная сложность признака C-Value составляет $O(nn_{lt})$.

Доказательство. Для вычисления уменьшаемого в признаке C-Value (см. формулу 1.3) требуется $O(n)$ операций (логарифм и умножение на частоту вхождений); для вычисления вычитаемого необходимо $O(nn_{lt})$. Пространственная сложность собственно вычисления признака константна.

Таблица 4.2: Обозначения для параметров алгоритмов

Обозначение	**Обозначаемое**
r	Объем внешнего корпуса (число N-грамм)
c_T	Число тем (topics) (как и в оригинальной работе [5], используется: 20 основных тем; 1 тема, специфичная для документа; 1 тема, специфичная для коллекции)
c_{iterTM}	Число итераций алгоритма построения тематической модели
c_S	Число ключевых концептов предметной области (по результатам экспериментального исследования было выбрано значение $c_S = 200$)
c_{key}	Число ключевых концептов, определяемых в документе (в экспериментах использовалось $c_{key} = 15$)
c_{basic}	Число терминов, выбранных с помощью метода DomainBasic (как и в оригинальной работе [45], используется 200 терминов)
c_{ws}	Размер окна (как и в оригинальной работе [45], $c_{ws} = 5$)
c_{wdm}	Размер модели домена (как и в оригинальной работе [45], $w_{dm} = 50$)
c_{iterLR}	Максимальное число итераций алгоритма оптимизации, применяемого при обучении модели логистической регрессии (в экспериментах использовалось $c_{iterLR} = 100$)
c_{iterPU}	Максимальное число итераций алгоритма PU-learning (в экспериментах использовалось $c_{iterPU} = 10$)
f	Число признаков (в экспериментах использовалось $f = 6$)

Кроме того, для подсчета этого признака необходимо для каждого термина вычислять само множество терминов, содержащих другие термины. В данной работе перед собственно вычислением признака для каждого термина создается и хранится отображение, позволяющее для каждого термина получить множество содержащих их терминов. Вычисление этого отображения обходится в $O(nw(w-1)) = O(nw^2)$ операций и занимает $O(nn_{lt})$ ячеек памяти.

Таким образом, временная сложность составляет $O(n + nn_{lt} + nw^2)$; пространственная сложность — $O(nn_{lt})$. □

Лемма 2. *Временная сложность признака Relevance составляет* $O(n + nD_{avg})$.

Пространственная сложность признака Relevance составляет $O(nD_{avg} + r)$.

Доказательство. Для вычисления признака Relevance (см. формулу 1.14) требуется $O(n)$ операций, при условии что выражения $TF_{target}(t)$, $DF_{target}(t)$ и $TF_{general}(t)$ вычисляются за константное время. Вычислительная сложность создания и хранения выражения $DF_{target}(t)$ — отображения из терминов в число документов, где эти термины встречаются, — равняется $O(nD_{avg})$.

Пространственная сложность последнего выражения ($TF_{general}(t)$) составляет $O(r)$, поскольку нужно хранить отображение из каждой N-граммы в число раз, сколько она встречалась; пространственная сложность выражения $TF_{target}(t)$ учтена на этапе извлечения кандидатов.

Суммируя соответствующие выражения, получаем оценки сложностей из условия леммы. □

Лемма 3. *Временная сложность признака Novel Topic Model составляет* $O(nwc_T + c_T V_{avg} D c_{iterTM})$.

Пространственная сложность признака Novel Topic Model составляет $O((V + D)c_T)$.

Доказательство. На этапе вычисления признака Novel Topic Model необходимо для каждого слова каждого термина определить в тематической модели максимальное значение среди всех тем, то есть пройтись по всем словам всех терминов и для каждого слова пройти по всем темам; таким образом, временная сложность составляет $O(nwc_T)$.

Создание тематической модели коллекции документов обходится в $O(c_T V_{avg} D c_{iterTM})$ операций и в $O((V + D)c_T)$ ячеек памяти[1].

Суммируя соответствующие выражения, получаем оценки сложностей из условия леммы. □

[1]Оценки вычислительной сложности для построения тематической модели взяты из «Руководства для пользователей» используемого фреймворка: https://github.com/ispras/tm/tree/master/documentation/userGuide

Лемма 4. *Временная сложность признака LinkProbability составляет* $O(n)$.

Пространственная сложность признака LinkProbability составляет $O(1)$.

Доказательство. Для вычисления признака LinkProbability требуется для каждого термина определить, сколько раз он встречался в качестве гиперссылки в статьях Википедии по отношению к общему числу вхождений в статьи Википедии. Поскольку система Текстерра содержит именно такой интерфейсный метод, сложность вычисления признака составляет $O(n)$ операций обращения к базе знаний системы Текстерра; пространственная сложность определяется пространственной сложностью базы знаний системы Текстерра и не зависит от размеров входных данных. □

Лемма 5. *Временная сложность признака KeyConceptRelatedness составляет* $O(nn_c c_S + Dc_{key} log(Dc_{key}) + v_{all} + c_{key}^2)$.

Пространственная сложность признака KeyConceptRelatedness составляет $O(Dc_{key} + c_{key}^2)$.

Доказательство. Для вычисления признака KeyConceptRelatedness требуется для каждого концепта каждого термина вычислить семантическую близость к ключевым концептам предметной области, сложность этого процесса составляет $O(nn_c c_S)$ операций обращения к базе знаний. Пространственная сложность является константной по тем же причинам, что и для признака LinkProbability.

Сложность извлечения ключевых концептов предметной области определяется сложностью извлечения ключевых концептов из документов и сложностью сортировки множества всех ключевых концетов всех документов.

Для извлечения ключевых концептов из документа необходимо определить все концепты этого документа, для чего, в свою очередь, необходимо определить для каждого токена, является ли он частью названия концепта Википедии и разрешить лексическую многозначность — общая временная сложность является линейной по числу токенов $O(v_{all})$ [72], пространственная сложность константна. Сложность определения (временная и пространственная) ключевых концептов является линейной по числу пар концептов, между которыми ненулевая семантическая близость [78], что можно оценить сверху как квадрат числа концептов: $O(c_{key}^2)$.

Учитывая оценки вычислительной сложности для алгоритма сортировки TimSort [115], применяемого в Java начиная с версии 7, и мощность множества ключевых концептов (Dc_{key}) получаем следующие оценки: временная сложность составляет $O(Dc_{key}log(Dc_{key}))$ (в худшем случае), пространственная — $O(Dc_{key})$. При этом после вычисления необходимо хранить не более S концептов.

Суммируя соответствующие выражения, получаем оценки сложностей из условия леммы. □

Лемма 6. *Временная и пространственная сложности вычисления признака Domain Basic совпадают с таковыми признака C-Value.*

Доказательство. Формула признака Domain Basic (см. формулу 1.3) является модификацией формулы C-Value (см. формулу 1.3); при этом сложность вычисления первого слагаемого равна $O(n)$, второго слагаемого — $O(nn_{lt})$.

При этом для вычисления второго слагаемого необходима та же структура, что и в признаке C-Value, поэтому сложность ее создания и хранения равна таковой у признака C-Value.

Таким образом, получаются такие же оценки, как у метода C-Value.

□

Лемма 7. *Временная сложность вычисления признака Domain Model:* $O(v_{all} + c_{basic}mw_fc_{ws} + w_fm + w_flog(w_f) + n + nn_{lt} + nw^2 + nmc_{wdm}c_{ws})$.

Пространственная сложность: $O(nn_{lt} + c_{basic}w_f + nc_{wdm})$.

Доказательство. Признак Domain Model представляет собой линейную комбинацию двух признаков — Domain Basic и Domain Coherence, поэтому оценка его сложности равна сумме оценок сложностей этих признаков.

Сложность вычисления признака Domain Basic доказана в предыдущей лемме.

Вычисление признака Domain Coherence также состоит из двух этапов. На первом этапе вычисляется модель домена, для чего применяется следующая последовательность действий:

1. все слова коллекции документов проходят через несколько фильтров, каждый из которых имеет линейную сложность по множеству всех слов: сложность $O(v_{all})$;

2. извлекается небольшое число терминов с помощью все того же метода Domain Basic: сложность складывается из вычисления признака Domain Basic и отбора лучших c_{basic}, для чего все кандидаты сортируются по значению признака: $O(C_{basic} + c_{basic}log(c_{basic}))$, где C_{basic} — сложность вычисления признака Domain Basic;

3. для оставшихся слов с помощью метрики PMI измеряется их близость к небольшому числу терминов, выбранным с помощью все того же метода Domain Basic: $O(C_{PMI} + w_f m)$, где C_{PMI} — сложность вычисления метрики PMI, в предположении что имеется структура данных, позволяющая за константное время узнать PMI между словом, прошедшим фильтрацию, и термином, извлеченным с помощью метода Domain Basic — сложность создания и хранения такой структуры описывается ниже, при определении сложности метрики PMI;

4. слова сортируются по близости и выбирается предопределенное число лучших слов: $O(w_f log(w_f))$.

Для вычисления метрики PMI необходимо подсчитать, сколько раз каждое слово, прошедшее фильтрацию, встретилось в одном окне заданного размера с вхождением каждого термина из выбранных с помощью метода Domain Basic. Это имеет сложность $O(c_{basic} m w_f c_{ws})$.

Итого временная сложность построения модели домена составляет $O(v_{all} + c_{basic} m w_f c_{ws} + w_f m + w_f log(w_f)) = O(v_{all} + m w_f (c_{basic} c_{ws} + 1) + w_f log(w_f)) = O(v_{all} + m w_f c_{basic} c_{ws} + w_f log(w_f))$, т.к. $c_{basic} c_{ws}$ не меньше единицы и поэтому отбрасывание единицы в коэффициенте $(c_{basic} c_{ws} + 1)$ не искажает оценку.

Пространственная сложность хранения модели домена определяется размером матрицы совместной встречаемости: $O(c_{basic} w_f)$.

Второй этап — использование модели домена для определения близости к ней каждого кандидата в термины — обладает аналогичной сложностью со следующими исключениями:

– нет этапа фильтрации (v_{all});

– вместо всех слов, прошедших фильтрацию, (w_f) рассматривается полученная модель домена (c_{wdm});

– вместо лучших терминов (c_{basic}), определенных с помощью метода Domain Basic, рассматриваются все кандидаты в термины (n);

– сортировка не выполняется, так как это происходит на последующих стадиях извлечения терминов.

То есть временная сложность составляет $O(nmc_{wdm}c_{ws} + c_{wdm}m) = O(mc_{wdm}(nc_{ws} + 1)) = O(nmc_{wdm}c_{ws})$, т.к. nc_{ws} не меньше единицы.

Пространственная сложность — $O(nc_{wdm})$.

Суммируя оценки для временной сложности, получим $O(v_{all} + c_{basic}mw_fc_{ws} + w_flog(w_f) + n + nn_{lt} + nw^2 + nmc_{wdm}c_{ws}) = O(v_{all} + c_{basic}mw_fc_{ws} + w_flog(w_f) + nn_{lt} + nw^2 + nmc_{wdm}c_{ws})$ — оценка из условия леммы.

Аналогично, суммируя выражения для оценки пространственной сложности, получаем оценку из условия леммы. □

Лемма 8. *Временная сложность признака ComboBasic составляет* $O(n + nn_{lt} + nw^2)$.

Пространственная сложность признака ComboBasic составляет $O(nn_{lt} + n)$.

Доказательство. Рассмотрим последнее слагаемое формулы 3.1 — e'_t — число кандидатов, содержащихся в кандидате t. Для вычисления этого слагаемого необходимо для каждого термина пройти по всем его подстрокам (временная сложность $O(nw^2)$) и проверить наличие подстроки среди множества всех кандидатов в термины (константная сложность при использовании подходящей структуры данных, например хеш-таблицы; при этом вычислительная сложность создания и пространственная сложность хранения такой структуры данных составляют не более $O(n)$).

Суммируя соответствующие выражения для временной и пространственной сложностей, получаем оценки из условия леммы. □

Теорема 2. ***Временная сложность*** *разработанного метода на основе алгоритма PU-learning составляет:*

$$O(n(1 + n_{lt} + w^2 + D_{avg} + wc_T + n_cc_S + mc_{wdm} + log(n) + c_{iterPU}c_{iterLR}f) +$$

$$+c_TV_{avg}Dc_{iterTM} + c_{key}Dlog(c_{key}D) + c^2_{key} + v_{all} + v_{all} +$$

$$+c_{basic}c_{ws}mw_f + w_f m + w_f log(w_f) + c_{ws} + c_{iterPU}c_{iterLR}f^2)$$

Или, без учета констант:

$$O(n(log(n)+n_{lt}+D_{avg}+n_c+m)+D(V_{avg}+log(D))+w_f(m+log(w_f))+v_{all}) \quad (4.1)$$

Пространственная сложность *разработанного метода на основе алгоритма PU-learning составляет:*

$$O(n(1 + n_{lt} + D_{avg} + c_{wdm}) + r + (V + D)c_T + Dc_{key} + c_{key}^2 + c_{basic}w_f)$$

Или, без учета констант:

$$O(n(n_{lt} + D_{avg}) + V + D + w_f) \quad (4.2)$$

Доказательство. Разработанный подход работает в три этапа, сложность которых определяет итоговую. На первом этапе вычисляется значение признака ComboBasic для всех кандидатов, после чего они сортируются по этому значению. Учитывая лемму и упомянутую выше сложность сортировки, получим следующую оценку временной сложности: $O(n+nn_{lt}+nw^2+nlog(n)) = O(nn_{lt} + nw^2 + nlog(n))$. При использовании фильтрации по Википедии, временная сложность этой фильтрации составляет $O(n)$ обращений к базе знаний системы Текстерра.

Пространственная сложность составляет $O(n)$.

На втором этапе происходит вычисление признаков C-Value, Relevance, Novel Topic Model, LinkProbability, KeyConceptRelatedness и Domain Model. Оценки доказаны в леммах 1-7.

Третий этап состоит в применении алгоритма Traditional, который состоит, в свою очередь, из итеративного применения алгоритма обучения логистической регрессии. При этом число итераций алгоритма Traditional ограничено константой c_{iterPU} (на практике число итераций для каждого набора данных не превышало 5).

В качества алгоритма оценки параметров модели логистической регрессии использовался один из Квазиньютоновских методов, а именно метод Активных множеств с итерационным методом оптимизации Бройдена — Флетчера

— Гольдфарба — Шанно (BFGS) — сложность каждой итерации которого составляет $O(f^2 + nf)$ по времени и $O(n)$ по памяти [116]. Общее число итераций также было ограничено константой (c_{iterLR}).

Итак, суммируем полученные выражения для временной сложности: $O(n + nn_{lt} + nw^2 + nD_{avg} + nwc_T + c_T V_{avg} D c_{iterTM} + nn_c c_S + Dc_{key} log(Dc_{key}) + + V + c_{basic} m w_f c_{ws} + w_f m + w_f log(w_f) + nmc_{wdm} c_{ws} + nlog(n) + c_{iterPU} c_{iterLR}(f^2 + nf))$

Сгруппировав слагаемые с множителем n и заметив, что $m \geq 1$, так как любой термин встречается не меньше одного раза, получим оценку из условия теоремы.

Аналогично получим оценку пространственной сложности:

$O(nn_{lt} + nD_{avg} + r + (V + D)c_T + Dc_{key} + c_{basic} w_f + nc_{wdm} + n)$

И также сгруппировав слагаемые с множителем n, получим оценку из условия теоремы. □

4.3 Особенности программной системы

Программная система разработана на языке программирования Java 7. Общий объем кодовой базы без учета сторонних модулей составляет около 29 тысяч строк кода (417 классов), из них около 13 тысяч строк кода (234 класса) относится к модулю оценки эффективности.

Использованные технологии и оптимизации подробно описаны в следующих подразделах.

4.3.1 Примененные технологии

Программная система реализована с применением следующих технологий:

– Текстерра [24] — фреймворк для обработки текстов, разработанный в Институте системного программирования РАН и поддерживающий REST- и Веб-интерфейс[2];

– QTF — фреймворк для тестирования эффективности алгоритмов классификации и ранжирования, разработанный в Институте системного программирования РАН;

[2] https://api.ispras.ru/

– TM[3] — открытый фреймворк для построения многоязыковых регуляризованных робастных тематических моделей, разработанный в Институте системного программирования РАН;

– Weka [117] — открытая библиотека машинного обучения, разработанная в университет Ваикато.

Отдельно стоит отметить, что для получения разметки набора данных Board games использовался инструмент AnnotaMe, разработанный в Институте системного программирования РАН.

4.3.2 Использованные оптимизации

Как было показано в разделе 4.1, для большинства методов вычисления признаков и фильтрации терминов существует возможность параллельного выполнения (используя декораторы *ParallelTermsUpdater* и *ParallelTermsFilter*).

Также значение любого признака может быть закэшировано (с помощью декоратора *CachedTermsUpdater* и отсутствия повторных вызовов зависимых объектов класса *AbstractTermsUpdater*, которое обеспечивается классом *LazyDependencyInitTermsUpdater*).

Кроме того, дополнительные структуры, создаваемые при вычислении некоторых признаков и обладающие наибольшей вычислительной сложностью, также могут быть закэшированы.

Так, обработанная коллекция документов может быть сериализована со всеми найденными аннотациями (границы предложений, токены, части речи, ключевые концепты). Аналогично может быть закэширована на дисковой памяти вычисленная тематическая модель (для признака Novel Topic Model), а также посчитанные частоты совместной встречаемости терминов и слов модели домена (для признака Domain Model).

В итоге с применением кэширования дополнительных структур данных временная сложность уменьшается на следующие величины:

– $O(n(wc_T) + c_T V_{avg} Dc_{iterTM}$ — благодаря кэшированию тематической модели;

[3] https://github.com/ispras/tm

– $O(v_{all} + c_{basic}c_{ws}mw_f + w_f m + w_f log(w_f) + c_{ws})$ — благодаря кэшированию совместной встречаемости терминов и слов модели домена.

Одно из важных преимуществ кэширования заключается в возможности отбора признаков путем полного перебора, поскольку подсчет каждого признака будет произведен только один раз, а именно этап вычисления признаков обладает наибольшей вычислительной сложностью.

В частности, при использовании системы извлечения терминов в рамках практического приложения, поддерживающего собственную быструю оценку эффективности, становится возможным выбрать наилучшие комбинации признаков и значений параметров. Например, если извлечение терминов применяется для последующего определения ключевых фраз и существует набор документов с эталонным множеством ключевых фраз, то можно перебрать несколько тысяч комбинаций признаков и значений параметров метода извлечения терминов и выбрать ту комбинацию, на которой достигается максимальная эффективность.

4.4 Выводы

В настоящей главе описана программная система, реализующая разработанные методы.

Показано, что вычислительная сложность является лог-линейной по кандидатам в термины и линейной по остальным параметрам входных данных.

Описаны оптимизации, позволяющие существенно ускорить разработанные методы в случае повторного применения к той же входной коллекции — распространенного варианта использования ввиду большого числа параметров — и приведены изменения оценок сложности в таком случае.

Заключение

Основные результаты работы заключаются в следующем.

1. Предложен подход к использованию информации Википедии для задачи извлечения терминов, основанный на структуре гиперссылок Википедии.

2. Предложен подход к извлечению терминов на основе алгоритма частичного обучения, не требующий размеченных данных.

3. В рамках предложенных подходов разработан автоматический метод извлечения терминов.

4. Разработано программное средство и проведено экспериментальное исследование, доказывающее улучшение эффективности разработанного метода по сравнению с существующими методами.

Литература

1. Evans D. A., Lefferts R. G. Clarit-trec experiments // Information processing & management. 1995. Vol. 31, no. 3. P. 385–395.

2. Navigli R., Velardi P. Semantic interpretation of terminological strings // Proc. 6th Int'l Conf. Terminology and Knowledge Eng. 2002. P. 95–100.

3. Frantzi K., Ananiadou S., Mima H. Automatic recognition of multi-word terms:. the c-value/nc-value method // International Journal on Digital Libraries. 2000. Vol. 3, no. 2. P. 115–130.

4. Bolshakova E., Loukachevitch N., Nokel M. Topic models can improve domain term extraction // Advances in Information Retrieval. Springer, 2013. P. 684–687.

5. A novel topic model for automatic term extraction / S. Li, J. Li, T. Song et al. // Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval / ACM. 2013. P. 885–888.

6. Bordea G., Buitelaar P., Polajnar T. Domain-independent term extraction through domain modelling // the 10th International Conference on Terminology and Artificial Intelligence (TIA 2013), Paris, France / 10th International Conference on Terminology and Artificial Intelligence. 2013.

7. University of surrey participation in trec8: Weirdness indexing for logical document extrapolation and retrieval (wilder) / K. Ahmad, L. Gillam, L. Tostevin et al. // The Eighth Text REtrieval Conference (TREC-8). 1999.

8. Meijer Kevin, Frasincar Flavius, Hogenboom Frederik. A semantic approach for extracting domain taxonomies from text // Decision Support Systems. 2014. T. 62. C. 78–93.

9. Sclano F., Velardi P. Termextractor: a web application to learn the shared terminology of emergent web communities // Enterprise Interoperability II. 2007. P. 287–290.

10. Браславский П.И., Соколов Е.А. Автоматическое извлечение терминологии с использованием поисковых машин Интернета // Компьютерная лингвистика и интеллектуальные технологии. Тр. Международной конференции «Диалог». 2007. С. 89–94.

11. Голомазов Д. Д. Методы и средства управления научной информацией с использованием онтологий // Диссертация кандидата физико-математических наук. Москва. 2012.

12. Dobrov B. V., Loukachevitch N. V. Multiple Evidence for Term Extraction in Broad Domains. // Recent Advances in Natural Language Processing / Citeseer. 2011. P. 710–715.

13. Extracting Domain-Relevant Term Using Wikipedia Based on Random Walk Model / W. Wu, T. Liu, H. Hu et al. // ChinaGrid Annual Conference (ChinaGrid), 2012 Seventh / IEEE. 2012. P. 68–75.

14. Vivaldi J., Rodríguez H. Using Wikipedia for term extraction in the biomedical domain: first experiences // Procesamiento del Lenguaje Natural. 2010. Vol. 45. P. 251–254.

15. Vivaldi J., Rodríguez H. Extracting terminology from Wikipedia // Procesamiento del lenguaje natural. 2011. Vol. 47. P. 65–73.

16. Using Wikipedia to Validate the Terminology found in a Corpus of Basic Textbooks. / J. Vivaldi, L. A. Cabrera-Diego, G. Sierra et al. // LREC. 2012. P. 3820–3827.

17. Zhang Z., Brewster C., Ciravegna F. A Comparative Evaluation of Term Recognition Algorithms // Proceedings of the Sixth International Confer-

ence on Language Resources and Evaluation (LREC08), Marrakech, Morocco. 2008.

18. Patry A., Langlais P. Corpus-based terminology extraction // Terminology and Content Development–Proceedings of 7th International Conference On Terminology and Knowledge Engineering, Litera, Copenhagen. 2005.

19. Fedorenko D., Astrakhantsev N., Turdakov D. Automatic recognition of domain-specific terms: an experimental evaluation // Proceedings of SYRCoDIS 2013. 2013. P. 15–23.

20. Astrakhantsev N., Fedorenko D., Turdakov D. Automatic Enrichment of Informal Ontology by Analyzing a Domain-Specific Text Collection // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue”. 2014. Vol. 13. P. 29–42.

21. Nokel M., Loukachevitch N. An Experimental Study of Term Extraction for Real Information-Retrieval Thesauri // Proceedings of 10th International Conference on Terminology and Artificial Intelligence. 2013. P. 69–76. (The paper is devoted to development of machine-learning extraction of domain-specific terms intended for informaiton-retrieval thesauri.)

22. Астраханцев Н. А., Турдаков Д. Ю. Методы автоматического построения и обогащения неформальных онтологий // Программирование. 2013. Т. 39, № 1. С. 23–34.

23. Федоренко Д., Астраханцев Н. Автоматическое извлечение новых концептов предметно-специфичных терминов // Труды Института системного программирования РАН. 2013. Т. 25. С. 167–178.

24. Texterra: инфраструктура для анализа текстов / Д. Турдаков, Н. Астраханцев, Я. Недумов [и др.] // Труды Института системного программирования РАН. 2014. Т. 26, № 1. С. 421–438.

25. Астраханцев Н. Автоматическое извлечение терминов из коллекции текстов предметной области с помощью Википедии // Труды Института системного программирования РАН. 2014. Т. 26, № 4. С. 7–20.

26. Гринев-Гриневич СВ. Терминоведение // М.: Академия. 2008. Т. 304.

27. Мякшин К.А. Разнообразие подходов к определению понятия «термин» // Альманах современной науки и образования, сер. «Языкознание и литературоведение в синхронии и диахронии и методика преподавания языка и литературы». 2007. Т. 3, № 3. С. 175–178.

28. Татаринов В.А. Терминологическая лексика русского языка: Эволюция проблем и аспектов изучения // Русский язык в современном обществе: Функциональные и статусные характеристики / РАН. ИНИОН; Отв. ред. Опарина Е.О., Казак Е.А. Теория и история языкознания. ИНИОН РАН, Москва, 2006. С. 133–164.

29. Pazienza M., Pennacchiotti M., Zanzotto F. Terminology extraction: an analysis of linguistic and statistical approaches // Knowledge Mining. 2005. P. 255–279.

30. Комарова Р.И. Терминосистема подъязыка эвристики (на материале англ. яз.): автореф. дис. канд. филол. наук текст. Одесса, 1991. С. 18.

31. Винокур Г.О. Грамматические наблюдения в области технической терминологии // Труды МИИФЛИ. 1939. Т. 5.

32. Wüster E. Einführung in die allgemeine Terminologielehre und terminologische Lexikographie (1979) // København: Handelshøjskolen. 1985.

33. Felber H. Basic principles and methods for the preparation of terminology standards // Standardization of Technical Terminology: Principle and Practices. ASTM STP. 1982. Vol. 806. P. 3–13.

34. Terminology - Vocabulary: Standard: Geneva, CH: International Organization for Standardization, 1990.

35. Pearson J. Terms in context. John Benjamins Publishing, 1998. Vol. 1.

36. Sager J. C. A practical course in terminology processing. John Benjamins Publishing, 1990.

37. Rondeau G. Introduction ala terminologie // 2e éd. Québec: Gaëtan Morin. 1984.

38. Мякшин К.А. К вопросу об основных признаках термина // Альманах современной науки и образования, сер. «Языкознание и литературоведение в синхронии и диахронии и методика преподавания языка и литературы». 2008. Т. 2, № 21. С. 17–22.

39. Хаютин А. Д. Составные термины - функциональный тип сложных лингвистических единиц (СЛЕ) с позиций лексикографии // Отраслевая терминология и лексикография. Воронеж, 1981.

40. Ахманова О. С. Терминология лингвистическая // Лингвистический энциклопедический словарь. Москва, 1990.

41. Большой энциклопедический словарь. 2-е изд., перераб. и доп. Москва: Большая Рос. энцикл., 2000. С. 1452.

42. Judea A., Schütze H., Bruegmann S. Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, 2014. August. P. 290–300.

43. Bernier-Colborne G., Drouin P. Creating a test corpus for term extractors through term annotation // Terminology. 2014. Vol. 20, no. 1. P. 50–73. URL: http://www.jbe-platform.com/content/journals/10.1075/term.20.1.03ber.

44. Bagot R. E. Les unités de signification spécialisées élargissant l'objet du travail en terminologie // Terminology. 2002. Vol. 7, no. 2. P. 217–237.

45. Bordea G. Domain adaptive extraction of topical hierarchies for Expertise Mining. Ph.D. thesis. 2013.

46. Kageura K., Umino B. Methods of automatic term recognition: A review // Terminology. 1996. Vol. 3, no. 2. P. 259–289.

47. Wermter J., Hahn U. You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation

and term extraction // Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics / Association for Computational Linguistics. 2006. P. 785–792.

48. Glossary extraction and utilization in the information search and delivery system for IBM Technical Support / L. Kozakov, Y. Park, T. Fin et al. // IBM Systems Journal. 2004. Vol. 43, no. 3. P. 546–563.

49. Браславский П.И., Соколов Е.А. Сравнение четырех методов автоматического извлечения двухсловных терминов из текста // Компьютерная лингвистика и интеллектуальные технологии. Тр. Международной конференции «Диалог». 2006. С. 88–94.

50. Браславский П., Соколов Е. Сравнение пяти методов извлечения терминов произвольной длины // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог»(Бекасово, 4–8 июня 2008). № 7. 2008. С. 14.

51. Bourigault D. Surface grammatical analysis for the extraction of terminological noun phrases // Proceedings of the 14th conference on Computational linguistics-Volume 3 / Association for Computational Linguistics. 1992. P. 977–981.

52. Baroni M., Bernardini S. BootCaT: Bootstrapping Corpora and Terms from the Web // LREC. 2004. P. 1313–1316.

53. Добров Б.В., Лукашевич Н.В., Сыромятников С.В. Формирование базы терминологических сочетаний по текстам предметной области // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды пятой Всероссийской научной конференции. 2003. С. 201–210.

54. Синтаксический анализ. Проект AOT: Tech. Rep.: URL: http://www.aot.ru/docs/synan.html.

55. Combining C-value and Keyword Extraction Methods for Biomedical Terms Extraction / J. A. L. Ventura, C. Jonquet, M. Roche et al. // LBM'2013:

International Symposium on Languages in Biology and Medicine. 2013. P. 45–49.

56. An improved automatic term recognition method for Spanish / A. Barrón-Cedeno, G. Sierra, P. Drouin et al. // Computational Linguistics and Intelligent Text Processing. Springer, 2009. P. 125–136.

57. Dennis S. F. The construction of a thesaurus automatically from a sample of text // Proceedings of the Symposium on Statistical Association Methods For Mechanized Documentation, Washington, DC. 1965. P. 61–148.

58. 6. using statistics in lexical analysis / K. Church, W. Gale, P. Hanks et al. // Lexical acquisition: exploiting on-line resources to build a lexicon. 1991. P. 115.

59. Dunning T. Accurate methods for the statistics of surprise and coincidence // Computational linguistics. 1993. Vol. 19, no. 1. P. 61–74.

60. Church K. W., Hanks P. Word association norms, mutual information, and lexicography // Computational linguistics. 1990. Vol. 16, no. 1. P. 22–29.

61. Daille B. Combined approach for terminology extraction: lexical statistics and linguistic filtering. Ph.D. thesis: Ph. D. thesis, University Paris 7. 1994.

62. Park Y., Byrd R., Boguraev B. Automatic glossary extraction: beyond terminology identification // Proceedings of the 19th international conference on Computational linguistics-Volume 1 / Association for Computational Linguistics. 2002. P. 1–7.

63. Blei D. M., Lafferty J. D. Topic models // Text mining: classification, clustering, and applications. 2009. Vol. 10. P. 71.

64. Corpus-based terminology extraction applied to information access / A. Peñas, F. Verdejo, J. Gonzalo et al. // Proceedings of Corpus Linguistics / Citeseer. Vol. 2001. 2001.

65. Manning C., Schütze H. Foundations of statistical natural language processing. MIT press, 1999.

66. A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping. / Feiyu Xu, Daniela Kurz, Jakub Piskorski [и др.] // LREC. 2002.

67. Milne D., Medelyan O., Witten I. H. Mining domain-specific thesauri from wikipedia: A case study // Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence. IEEE Computer Society, 2006. P. 442–448. URL: http://dl.acm.org/citation.cfm?id=1249168.

68. Strube M., Ponzetto S. P. WikiRelate! Computing semantic relatedness using Wikipedia // AAAI. Vol. 6. 2006. P. 1419–1424.

69. Mihalcea R., Csomai A. Wikify!: linking documents to encyclopedic knowledge // Proceedings of the sixteenth ACM conference on Conference on information and knowledge management / ACM. 2007. P. 233–242.

70. Milne D. Computing semantic relatedness using wikipedia link structure // Proceedings of the new zealand computer science research student conference / Citeseer. 2007.

71. Milne D., Witten I. H. Learning to link with wikipedia // Proceedings of the 17th ACM conference on Information and knowledge management / ACM. 2008. P. 509–518.

72. Турдаков Д. Ю. Методы и программные средства разрешения лексической многозначности терминов на основе сетей документов. Ph.D. thesis. 2010.

73. Fault-Tolerant Learning for Term Extraction / Y. Yang, H. Yu, Y. Meng et al. 2011. URL: http://www.aclweb.org/anthology/Y10-1036.

74. Liu X., Kit C. An Improved Corpus Comparison Approach to Domain Specific Term Recognition. // PACLIC. 2008. P. 253–261.

75. Анисимов АВ, Марченко АА, Кисенко ВК. Метод вычисления семантической близости-связности между словами естественного языка // Кибернетика и системный анализ. 2011. № 47,№ 4. С. 18–27.

76. Semantic Measures for the Comparison of Units of Language, Concepts or Instances from Text and Knowledge Representation Analysis / S. Harispe, S. Ranwez, S. Janaqi et al. 2013. URL: http://arxiv.org/pdf/1310.1285.

77. Carley K. M., Kaufer D. S. Semantic connectivity: An approach for analyzing symbols in semantic networks // Communication Theory. 1993. Vol. 3, no. 3. P. 183–213.

78. Grineva M., Grinev M., Lizorkin D. Extracting key terms from noisy and multitheme documents // Proceedings of the 18th international conference on World wide web / ACM. 2009. P. 661–670.

79. Witten I., Milne D. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links // Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA. 2008. P. 25–30.

80. Turdakov D., Velikhov P. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. 2008.

81. GENIA corpus–a semantically annotated corpus for bio-textmining / J.-D. Kim, T. Ohta, Y. Tateisi et al. // Bioinformatics. 2003. Vol. 19, no. Suppl 1. P. i180–i182.

82. Nenadié G., Ananiadou S., McNaught J. Enhancing automatic term recognition through recognition of variation // Proceedings of the 20th international conference on Computational Linguistics / Association for Computational Linguistics. 2004. P. 604.

83. Krauthammer M., Nenadic G. Term identification in the biomedical literature // Journal of biomedical informatics. 2004. Vol. 37, no. 6. P. 512–526.

84. Medelyan O., Witten I. H. Domain-independent automatic keyphrase indexing with small training sets // Journal of the American Society for Information Science and Technology. 2008. Vol. 59, no. 7. P. 1026–1040.

85. Krapivin M., Autaeu A., Marchese M. Large dataset for keyphrases extraction. 2009. URL: http://eprints.biblio.unitn.it/1671/1/disi09055-krapivin-autayeu-marchese.pdf.

86. Faralli Stefano, Navigli Roberto. Growing Multi-Domain Glossaries from a Few Seeds using Probabilistic Topic Models. // EMNLP. 2013. C. 170–181.

87. Vivaldi J., Rodríguez H. Improving term extraction by combining different techniques // Terminology. 2001. Vol. 7, no. 1. P. 31–48.

88. Shamsfard M., Nematzadeh A., Motiee S. Orank: An ontology based system for ranking documents // International Journal of Computer Science. 2006. Vol. 1, no. 3. P. 225–231.

89. Tag-based information retrieval of video content / M. Melenhorst, M. Grootveld, M. van Setten et al. // Proceedings of the 1st international conference on Designing interactive user experiences for TV and video / ACM. 2008. P. 31–40.

90. Ryu P.-M., Choi K.-S. Determining the Specificity of Terms based on Information Theoretic Measures // insulin. 2004. Vol. 18, no. 452.297. P. 267.

91. Ryu P.-M., Choi K.-S. Measuring the specificity of terms for automatic hierarchy construction // Proceedings of ECAI-2004 Workshop on Ontology Learning and Population. 2004.

92. Combining Evidence, Specificity, and Proximity Towards the Normalization of Gene Ontology Terms in Text / S. Gaudan, A. J. Yepes, V. Lee et al. // EURASIP J. Bioinformatics Syst. Biol. New York, NY, United States, 2008. Vol. 2008. P. 4:1–4:9.

93. Hippisley A., Cheng D., Ahmad K. The head-modifier principle and multilingual term extraction // Natural Language Engineering. 2005. Vol. 11, no. 02. P. 129–157.

94. Юдина Т. М. Гиперо-гипонимические отношения терминов в горнозаводской терминологии начала XVIII в // XLIII Международная филологическая научная конференция. 2014. С. 400.

95. Justeson J. S., Katz S. M. Technical terminology: some linguistic properties and an algorithm for identification in text // Natural language engineering. 1995. Vol. 1, no. 01. P. 9–27.

96. Elkan C., Noto K. Learning classifiers from only positive and unlabeled data // Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. 2008. P. 213–220.

97. Zhang B., Zuo W. Learning from positive and unlabeled examples: A survey // Information Processing (ISIP), 2008 International Symposiums on / IEEE. 2008. P. 650–654.

98. Montes M., Escalante H. J. Novel representations and methods in text classification // 7th Russian Summer School in Information Retrieval Kazan. 2013. URL: http://romip.ru/russiras/doc/russir2013/target4-3.pdf.

99. Montes-y Gómez M., Rosso P. Using PU-Learning to Detect Deceptive Opinion Spam // WASSA 2013. 2013. P. 38.

100. Partially supervised classification of text documents / B. Liu, W. S. Lee, P. S. Yu et al. // ICML / Citeseer. Vol. 2. 2002. P. 387–394.

101. Building text classifiers using positive and unlabeled examples / B. Liu, Y. Dai, X. Li et al. // Data Mining, 2003. ICDM 2003. Third IEEE International Conference on / IEEE. 2003. P. 179–186.

102. Yu H., Han J., Chang K. C.-C. PEBL: positive example based learning for web page classification using SVM // Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. 2002. P. 239–248.

103. Li X., Liu B. Learning to classify texts using positive and unlabeled data // IJCAI. Vol. 3. 2003. P. 587–592.

104. Lee W. S., Liu B. Learning with positive and unlabeled examples using weighted logistic regression // ICML. Vol. 3. 2003. P. 448–455.

105. Partially supervised classification–based on weighted unlabeled samples support vector machine / Z. Liu, W. Shi, D. Li et al. // Advanced Data Mining and Applications. Springer, 2005. P. 118–129.

106. Sellamanickam S., Garg P., Selvaraj S. K. A pairwise ranking based approach to learning with positive and unlabeled examples // Proceedings of the 20th ACM international conference on Information and knowledge management / ACM. 2011. P. 663–672.

107. Kwok J.-Y. Moderating the outputs of support vector machine classifiers // Neural Networks, IEEE Transactions on. 1999. Vol. 10, no. 5. P. 1018–1031.

108. Rüping S. A simple method for estimating conditional probabilities for svms: Tech. Rep.: : Technical Report/Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen, 2004.

109. Salazar D. A., Vélez J. I., Salazar J. C. Comparison between SVM and Logistic Regression: Which One is Better to Discriminate? // Número especial en Bioestadıstica. 2012. Vol. 35. P. 223–237.

110. Gregorutti B., Michel B., Saint-Pierre P. Correlation and variable importance in random forests // arXiv preprint arXiv:1310.5726. 2013. URL: http://arxiv.org/pdf/1310.5726.

111. Schaefer R. L. Alternative estimators in logistic regression when the data are collinear // Journal of Statistical Computation and Simulation. 1986. Vol. 25, no. 1-2. P. 75–91.

112. Quenouille M. H. Notes on bias in estimation // Biometrika. 1956. P. 353–360.

113. Tukey J. W. Bias and confidence in not-quite large samples // Annals of Mathematical Statistics. Vol. 29. 1958. P. 614–614.

114. Wilcoxon F. Individual comparisons by ranking methods // Biometrics bulletin. 1945. P. 80–83.

115. Tim P. [Python-Dev] Sorting: Tech. Rep.: : Python Developers Mailinglist, 2011. URL: http://mail.python.org/pipermail/python-dev/2002-July/026837.html.

116. Minka T. P. A comparison of numerical optimizers for logistic regression // Unpublished draft. 2003. URL: http://research.microsoft.com/en-us/um/people/minka/papers/logreg/minka-logreg.pdf.

117. The WEKA data mining software: an update / M. Hall, E. Frank, G. Holmes et al. // ACM SIGKDD explorations newsletter. 2009. Vol. 11, no. 1. P. 10–18.

Приложение А

Примеры результатов работы предложенного подхода

В данном приложении приводятся примеры результатов работы предложенного подхода — а именно, лучшие кандидаты в термины — для наборов данных Board games, Patents, GENIA, Krapivin и FAO. Во всех случаях используются значения параметров, выбранные в разделе 3.4.1.

В нижеприведенных таблицах для каждого кандидата в термины указан его порядковый номер, полученный после ранжирования с помощью разработанного подхода, текстовое представление первого вхождения и правильный ответ (T — если кандидат присутствует в списке эталонов, и F — иначе). Неверные ответы дополнительно выделены жирным шрифтом.

Таблица A.1: Результаты работы предложенного подхода с фильтрацией по Википедии на наборе данных Patents

1	battery	T	36	**packet writing**	F
2	secondary battery	T	37	wind power system	T
3	charge level	T	38	capacitor	T
4	temperature rise	T	39	electric power	T
5	temperature rise value	T	40	battery charge	T
6	electric motors	T	41	Uninterruptible power supply	T
7	CD-R/RW drive unit	T	42	DC power	T
8	nickel metal hydride battery	T	43	hard disk	T
9	battery charger	T	44	AC power	T
10	reactive power	T	45	active power	T
11	power generator	T	46	DC/DC converter	T
12	main battery	T	47	AC adapter	T
13	battery packs	T	48	electric potential difference	T
14	**information processing**	F	49	bicycle dynamo	T
15	voltage	T	50	composite material	T
16	nickel cadmium battery	T	51	power converter	T
17	hard disk drive	T	52	motor controller	T
18	disk-at-once	T	53	rotational speed	T
19	DC voltage	T	54	electrical charge	T
20	**nickel-metal hydride**	F	55	CD-RWs	T
21	temperature rise pattern	T	56	voltage divider	T
22	battery temperature	T	57	wind turbines	T
23	constant voltage	T	58	dry battery	T
24	threshold value	T	59	AC voltage	T
25	**rise value**	F	60	battery level	T
26	dynamo	T	61	drive unit	T
27	hybrid vehicle	T	62	**regenerative braking**	F
28	disk drive	T	63	electrical generator	T
29	temperature	T	64	AC/AC converter	T
30	**depth-of-discharge**	F	65	nickel hydrogen battery	T
31	**power**	F	66	**control unit**	F
32	information processing apparatus	T	67	carbon fibre	T
33	power saving mode	T	68	doubly-fed induction generator	T
34	wind power	T	69	smart battery	T
35	**pulse-width modulation**	F	70	host computer	T

Таблица А.2: Результаты работы предложенного подхода без фильтрации по Википедии на наборе данных Patents

1	charge level	T	36	**determination flag**	F
2	battery	T	37	power saving	T
3	secondary battery	T	38	**processing apparatus**	F
4	temperature rise	T	39	temperature	T
5	temperature rise value	T	40	nickel metal hydride battery	T
6	CD-R/RW drive unit	T	41	battery package	T
7	**power**	F	42	hybrid construction machine	T
8	electric motors	T	43	**saving mode**	F
9	information processing apparatus	T	44	AC power-supply source	T
10	battery temperature	T	45	terminal voltage	T
11	threshold value	T	46	battery information	T
12	power saving mode	T	47	manipulating lever	T
13	**rise value**	F	48	rotor blade	T
14	drive unit	T	49	**current shut-off switch**	F
15	temperature rise pattern	T	50	battery packs	T
16	active power	T	51	**control unit**	F
17	battery charge	T	52	disk rotational speed	T
18	battery level	T	53	voltage	T
19	main battery	T	54	lower threshold value	T
20	**information processing**	F	55	manipulating signal	T
21	depth-of-discharge limit control mode	T	56	temperature value	T
22	reactive power	T	57	airflow path	T
23	power sources	T	58	**temperature rise value outputting**	F
24	**CD-R/RW drive**	F	59	wind power system	T
25	charge	T	60	voltage difference VA	T
26	target temperature value	T	61	air-blowing motor	T
27	power generator	T	62	**current reference signal**	F
28	**level power saving mode**	F	63	power-supply information	T
29	**current value**	F	64	electric power	T
30	battery charger	T	65	**rise value outputting section**	F
31	power storage unit	T	66	disk drive	T
32	fibre material	T	67	abnormal condition	T
33	rotational speed	T	68	temperature detecting section	T
34	**value**	F	69	wind turbines	T
35	host computer	T	70	module control units	T

Таблица А.3: Результаты работы предложенного подхода с фильтрацией по Википедии на наборе данных Board games

1	Board Game	T	36	sieve rule	T
2	game	T	37	**Single Player**	F
3	Card Game	T	38	**Solitaire**	F
4	players	T	39	**game designers**	F
5	Mayfair Games	T	40	**Image courtesy**	F
6	International Gamers Award	T	41	abstract game	T
7	Scrabble	T	42	52-card deck	T
8	Euro Games	T	43	Fantasy Flight	T
9	card	T	44	card management	T
10	BattleLore	T	45	**TSR**	F
11	Spiel des Jahres award	T	46	deck	T
12	**German board game**	F	47	Scrabble Flash	T
13	designer games	T	48	SdJ	T
14	Backgammon	T	49	**card deck**	F
15	Mancala	T	50	Cribbage	T
16	Monopoly	T	51	**tableau**	F
17	Party Game&#	T	52	board	T
18	Strategy Game	T	53	Twilight Struggle	T
19	**Word Freak**	F	54	Strategy	T
20	**Catan**	F	55	**game pieces**	F
21	**Scrabble Players**	F	56	Dice	T
22	word game	T	57	**Poker**	F
23	Trivial Pursuit	T	58	train game	T
24	**family game**	F	59	Checkers	T
25	Pictionary	T	60	wargames	T
26	Ravensburger	T	61	**significant games**	F
27	German-style games	T	62	Scattergories	T
28	German games	T	63	Dice Game	T
29	solitaire card game	T	64	Scrabble Words	T
30	Standard 52-card deck	T	65	Othello	T
31	game board	T	66	**Multiplayer**	F
32	**SIEVE**	F	67	Rummikub	T
33	Bohnanza	T	68	tokens	T
34	BoardGameGeek.com	T	69	**doubling cube**	F
35	card-driven games	T	70	**Historical Simulation Game**	F

Таблица А.4: Результаты работы предложенного подхода без фильтрации по Википедии на наборе данных Board games

1	players	T	36	designer games	T
2	game	T	37	**complete rules**	F
3	Card Game	T	38	rules	T
4	Board Game	T	39	**discard pile**	F
5	card	T	40	solitaire card game	T
6	**Image courtesy**	F	41	**Publisher**	F
7	board	T	42	**designer**	F
8	Party Game	T	43	**Game Review**	F
9	Strategy Game	T	44	hand	T
10	**significant games**	F	45	opponent	T
11	word game	T	46	**Toy**	F
12	Scrabble	T	47	**ticket**	F
13	game board	T	48	Backgammon	T
14	Monopoly	T	49	Mancala	T
15	Dice Game	T	50	**Games magazine**	F
16	Mayfair Games	T	51	pieces	T
17	**game designers**	F	52	tiles	T
18	**International Toy**	F	53	**Edition**	F
19	**family game**	F	54	**site**	F
20	**ages**	F	55	war	T
21	**web site**	F	56	Clue	T
22	International Gamers Award	T	57	**top card**	F
23	**Word**	F	58	German games	T
24	**Scrabble Players**	F	59	Risk	T
25	**SIEVE**	F	60	Stones	T
26	Dice	T	61	**tableau**	F
27	sieve rule	T	62	Euro Games	T
28	Fantasy Flight	T	63	Standard 52-card deck	T
29	Trivial Pursuit	T	64	**Poker**	F
30	Strategy	T	65	**spy**	F
31	BoardGameGeek.com	T	66	Checkers	T
32	**time**	F	67	52-card deck	T
33	Scrabble Words	T	68	BattleLore	T
34	**Catan**	F	69	**two-player game**	F
35	deck	T	70	**Star**	F

Таблица A.5: Результаты работы предложенного подхода с фильтрацией по Википедии на наборе данных GENIA

1	transcription factor	T	36	receptor	T
2	cells	T	37	**TCF-1**	T
3	cell line	T	38	intercellular adhesion molecule-1	T
4	gene expression	T	39	gene transcription	T
5	glucocorticoid receptor	T	40	class II genes	T
6	gene	T	41	Granulocyte-macrophage colony-stimulating factor	T
7	human immunodeficiency virus	T	42	DNA binding domain	T
8	tumor necrosis factor-alpha	T	43	GM-CSF receptor	T
9	binding site	T	44	Th2 cells	T
10	NK cells	T	45	Th1 cells	T
11	DNA binding	T	46	**human immunodeficiency virus type**	F
12	proteins	T	47	factor	T
13	squirrel monkey	T	48	rheumatoid arthritis	T
14	expression	T	49	Jurkat cells	T
15	breast cancers	T	50	All-trans retinoic acid	T
16	tumor necrosis factor	T	51	Electrophoretic mobility shift assay	T
17	transcription	T	52	neutrophil granulocytes	T
18	DNA binding activity	T	53	A-myb	T
19	peripheral blood mononuclear cells	T	54	p21ras	T
20	RAR alpha	T	55	retinoblastoma protein	T
21	nuclear factor	T	56	dendritic cells	T
22	Bcl-6	T	57	IL-4R	T
23	peripheral blood	T	58	mononuclear leukocytes	T
24	protein kinase	T	59	T-lymphoid cells	T
25	TAL1	T	60	monocytes	T
26	peripheral blood lymphocytes	T	61	signal transduction	T
27	endothelial cell	T	62	natural killer cell	T
28	IL-12	T	63	CIITA	T
29	Epstein-Barr virus	T	64	SLP-76	T
30	human T-cell leukemia virus	T	65	erythroid cells	T
31	vascular cell adhesion molecule-1	T	66	**activation**	T
32	retinoic acid	T	67	adult T-cell leukemia	T
33	retinoic acid receptor-alpha	T	68	human immunodeficiency viruses	T
34	Interferon gamma	T	69	transcriptional activation	T
35	major histocompatibility complex	T	70	heat shock proteins	T

Таблица А.6: Результаты работы предложенного подхода без фильтрации по Википедии на наборе данных GENIA

1	transcription factor	T	36	**human immunodeficiency virus type**	F
2	cells	T	37	electrophoretic mobility shift	T
3	cell line	T	38	site	T
4	gene expression	T	39	immunodeficiency virus	T
5	gene	T	40	binding	T
6	expression	T	41	promoter activity	T
7	binding site	T	42	human monocytes	T
8	DNA binding	T	43	**human immunodeficiency**	F
9	proteins	T	44	mononuclear cells	T
10	transcription	T	45	cell cycle	T
11	factor	T	46	**virus type**	F
12	glucocorticoid receptor	T	47	breast cancers	T
13	nuclear factor	T	48	lymphocyte	T
14	peripheral blood	T	49	monocytes	T
15	activation	T	50	cell types	T
16	human immunodeficiency virus	T	51	monocytic cell	T
17	receptor	T	52	mobility shift	T
18	DNA binding activity	T	53	reporter gene	T
19	tumor necrosis factor	T	54	retinoic acid	T
20	protein kinase	T	55	transcriptional activity	T
21	tumor necrosis factor-alpha	T	56	Jurkat cells	T
22	promoter	T	57	Epstein-Barr virus	T
23	tumor necrosis	T	58	blood mononuclear cells	T
24	activities	T	59	human T-cell leukemia virus	T
25	NK cells	T	60	**necrosis factor**	F
26	binding activity	T	61	phorbol ester	T
27	**necrosis factor alpha**	F	62	cell activation	T
28	peripheral blood mononuclear cells	T	63	NF-kappaB	T
29	NF-kappa	T	64	**terminal repeat**	F
30	endothelial cell	T	65	differentiation	T
31	gene transcription	T	66	regulatory elements	T
32	transcriptional activation	T	67	effects	T
33	**nuclear factor kappa**	F	68	**MHC class**	F
34	signal transduction	T	69	levels	T
35	tyrosine phosphorylation	T	70	NF-kappaB activation	T

Таблица А.7: Результаты работы предложенного подхода с фильтрацией по Википедии на наборе данных Krapivin

1	**exe- cution**	F	36	Finite Element Method	T
2	algorithm	T	37	Computer Vision	T
3	linear programming	T	38	semidefinite programming	T
4	support vector machines	T	39	compiler optimizations	T
5	machine learning	T	40	**et al**	F
6	interior-point method	T	41	on-line algorithm	T
7	Temporal Logics	T	42	Preconditioner	T
8	data mining	T	43	**matrix**	F
9	polynomial time	T	44	linear algebra	T
10	model-checking	T	45	scheduling algorithms	T
11	sparse matrix	T	46	data structure	T
12	Fortran	T	47	binary decision diagrams	T
13	decision tree	T	48	parallel algorithm	T
14	Gaussian elimination	T	49	**systems**	F
15	Bayesian network	T	50	linear program	T
16	programming languages	T	51	branch prediction	T
17	computational complexity	T	52	Nonlinear Programming	T
18	Markov chain	T	53	edge	T
19	**Java**	F	54	dynamic programming	T
20	QUADRATIC PROGRAMMING	T	55	optimization	T
21	neural networks	T	56	**processors**	F
22	data dependence	T	57	linear systems	T
23	**control flow**	F	58	hypercubes	T
24	Speculative execution	T	59	partial evaluation	T
25	logic programming	T	60	polynomials	T
26	node	T	61	parallel computing	T
27	graphs	T	62	graph coloring	T
28	iterative method	T	63	**vector**	F
29	register allocation	T	64	Krylov subspace	T
30	abstract interpretation	T	65	mutual exclusion	T
31	model	T	66	VLSI	T
32	genetic algorithm	T	67	Graph Theory	T
33	FORMAL VERIFICATION	T	68	programming	T
34	**time**	F	69	load balancing	T
35	approximation algorithms	T	70	Prolog	T

Таблица А.8: Результаты работы предложенного подхода без фильтрации по Википедии на наборе данных Krapivin

1	linear programming	T	36	**matrix**	F
2	algorithm	T	37	node	T
3	machine learning	T	38	VLSI	T
4	support vector machines	T	39	branch prediction	T
5	polynomial time	T	40	optimization	T
6	**exe- cution**	F	41	**et al**	F
7	Temporal Logics	T	42	load balancing	T
8	model-checking	T	43	edge	T
9	computational complexity	T	44	approximation algorithms	T
10	sparse matrix	T	45	data structure	T
11	Markov chain	T	46	dynamic programming	T
12	data mining	T	47	real-time systems	T
13	interior-point method	T	48	code generation	T
14	programming languages	T	49	QUADRATIC PROGRAMMING	T
15	**Java**	F	50	model	T
16	Fortran	T	51	MPI.	T
17	Bayesian network	T	52	caching	T
18	decision tree	T	53	**processors**	F
19	logic programming	T	54	linear systems	T
20	polynomials	T	55	programming	T
21	neural networks	T	56	**vector**	F
22	scheduling algorithms	T	57	partial evaluation	T
23	parallel algorithm	T	58	matrices	T
24	Preconditioner	T	59	FORMAL VERIFICATION	T
25	iterative method	T	60	abstract interpretation	T
26	linear program	T	61	Parallelization	T
27	**control flow**	F	62	Prolog	T
28	classifiers	T	63	**linear**	F
29	graphs	T	64	complexity	T
30	Gaussian elimination	T	65	Finite Element Method	T
31	hypercubes	T	66	Speculative execution	T
32	genetic algorithm	T	67	compiler	T
33	data dependence	T	68	compiler optimizations	T
34	register allocation	T	69	TCP	T
35	multicast	T	70	cache	T

Таблица А.9: Результаты работы предложенного подхода с фильтрацией по Википедии на наборе данных FAO

1	Forests	T	36	sustainable development	T
2	**country**	F	37	forestry	T
3	**United Nations**	F	38	**Woodfuel**	F
4	forest products	T	39	**fish farming**	F
5	**et al**	F	40	**project**	F
6	forest management	T	41	farmers	T
7	**WORLD BANK**	F	42	products	T
8	food security	T	43	fresh water	T
9	**water**	F	44	Burkina Faso	T
10	**development**	F	45	animal	T
11	FAO	T	46	**forest industries**	F
12	**RIL**	F	47	private sector	T
13	**K[cb**	F	48	**fishing**	F
14	food	T	49	tree	T
15	natural resources	T	50	**levels**	F
16	fish	T	51	fishing vessel	T
17	aquaculture	T	52	**regions**	F
18	production	T	53	research	T
19	Fisheries	T	54	maize	T
20	soils	T	55	**activities**	F
21	rural development	T	56	food production	T
22	forest resources	T	57	SOIL CONSERVATION	T
23	**resources**	F	58	**aquaculture development**	F
24	managements	T	59	**information**	F
25	**systems**	F	60	species	T
26	water resources	T	61	resources management	T
27	women	T	62	irrigation	T
28	land	T	63	Sri Lanka	T
29	Tropical Forest	T	64	energy	T
30	agriculture	T	65	wood	T
31	**sustainable forest management**	F	66	**natural forests**	F
32	fisheries management	T	67	**agricultural production**	F
33	plants	T	68	**soil erosion**	F
34	National parks	T	69	**time**	F
35	crops	T	70	raw materials	T

Таблица A.10: Результаты работы предложенного подхода без фильтрации по Википедии на наборе данных FAO

1	**K[cb**	F	36	non-wood forest products	T
2	Forests	T	37	water resources	T
3	**RIL**	F	38	**levels**	F
4	**country**	F	39	animal	T
5	**et al**	F	40	research	T
6	forest products	T	41	plants	T
7	**water**	F	42	**activities**	F
8	forest management	T	43	Burkina Faso	T
9	**development**	F	44	fresh water	T
10	**United Nations**	F	45	**information**	F
11	**Pulau Payar**	F	46	**non-wood forest**	F
12	food	T	47	crops	T
13	food security	T	48	**forest industries**	F
14	production	T	49	fishing vessel	T
15	**WORLD BANK**	F	50	agriculture	T
16	natural resources	T	51	**fish farming**	F
17	forest resources	T	52	forestry	T
18	aquaculture	T	53	sustainable development	T
19	fish	T	54	**rural women**	F
20	FAO	T	55	**time**	F
21	**resources**	F	56	farmers	T
22	managements	T	57	species	T
23	rural development	T	58	resources management	T
24	women	T	59	fisheries management	T
25	**systems**	F	60	**programmes**	F
26	land	T	61	Sri Lanka	T
27	Fisheries	T	62	tree	T
28	**aquaculture development**	F	63	**studies**	F
29	**sustainable forest management**	F	64	private sector	T
30	**Woodfuel**	F	65	**regions**	F
31	soils	T	66	raw materials	T
32	Tropical Forest	T	67	policy	T
33	**project**	F	68	cost	T
34	National parks	T	69	**natural forests**	F
35	products	T	70	market	T

Приложение В

Зависимость точности от числа извлекаемых терминов

В данном приложении приводится сравнение по точности разработанного подхода с тремя лучшими из существующих методов для каждого набора данных. В нижеприведенных графиках по оси абсцисс отмечен размер учитываемой части списка кандидатов (число лучших кандидатов, или число извлекаемых терминов), по оси ординат — значения точности.

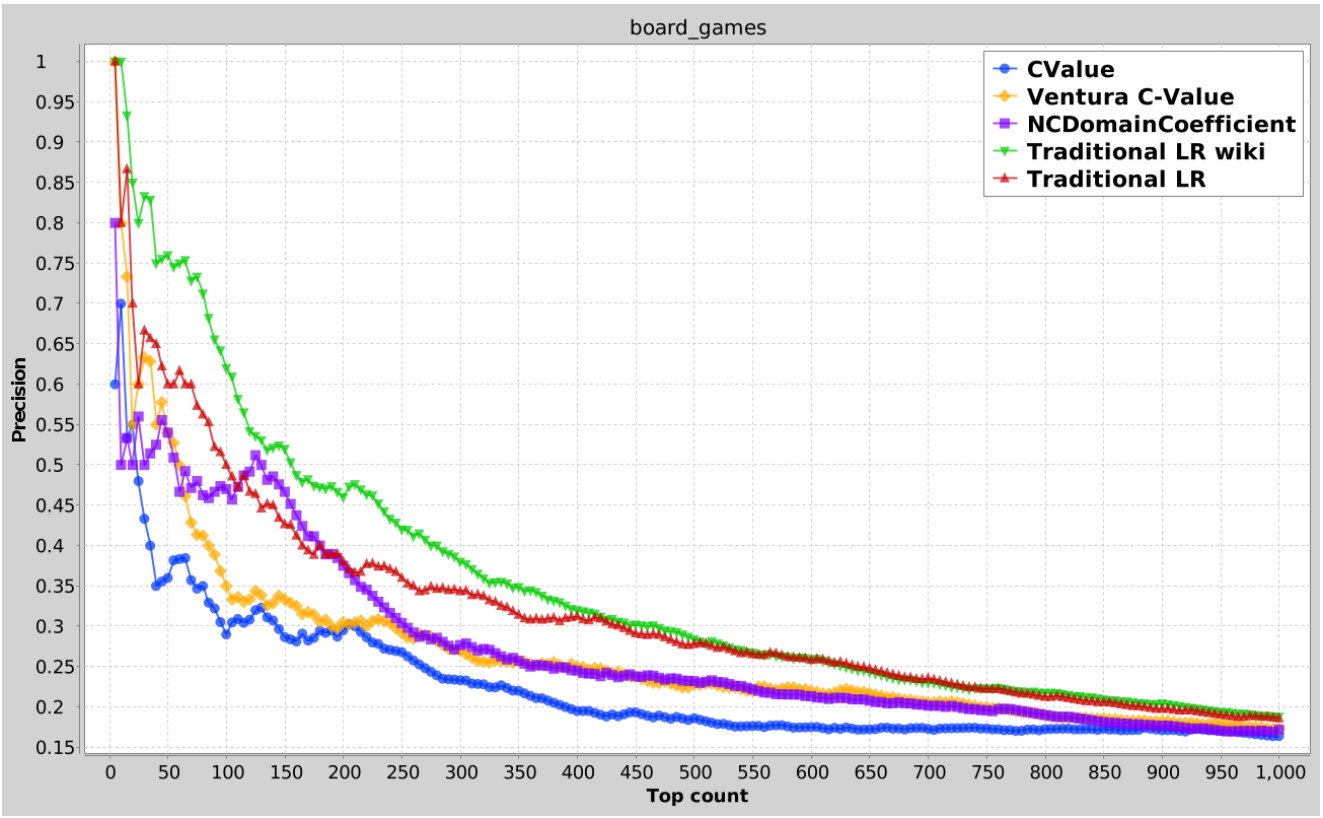

Рисунок В.1: Зависимость точности от размера учитываемой части списка кандидатов для набора данных Board games

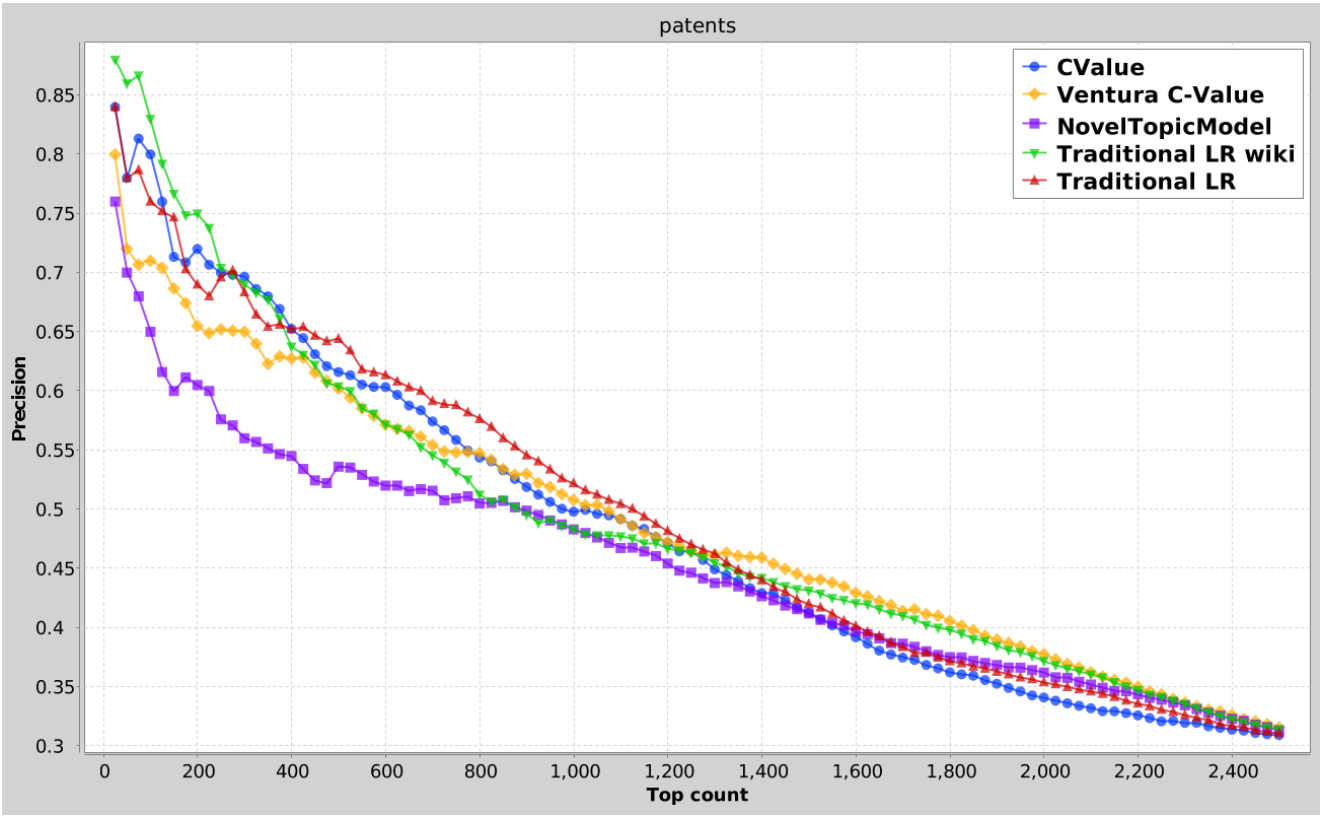

Рисунок В.2: Зависимость точности от размера учитываемой части списка кандидатов для набора данных Patents

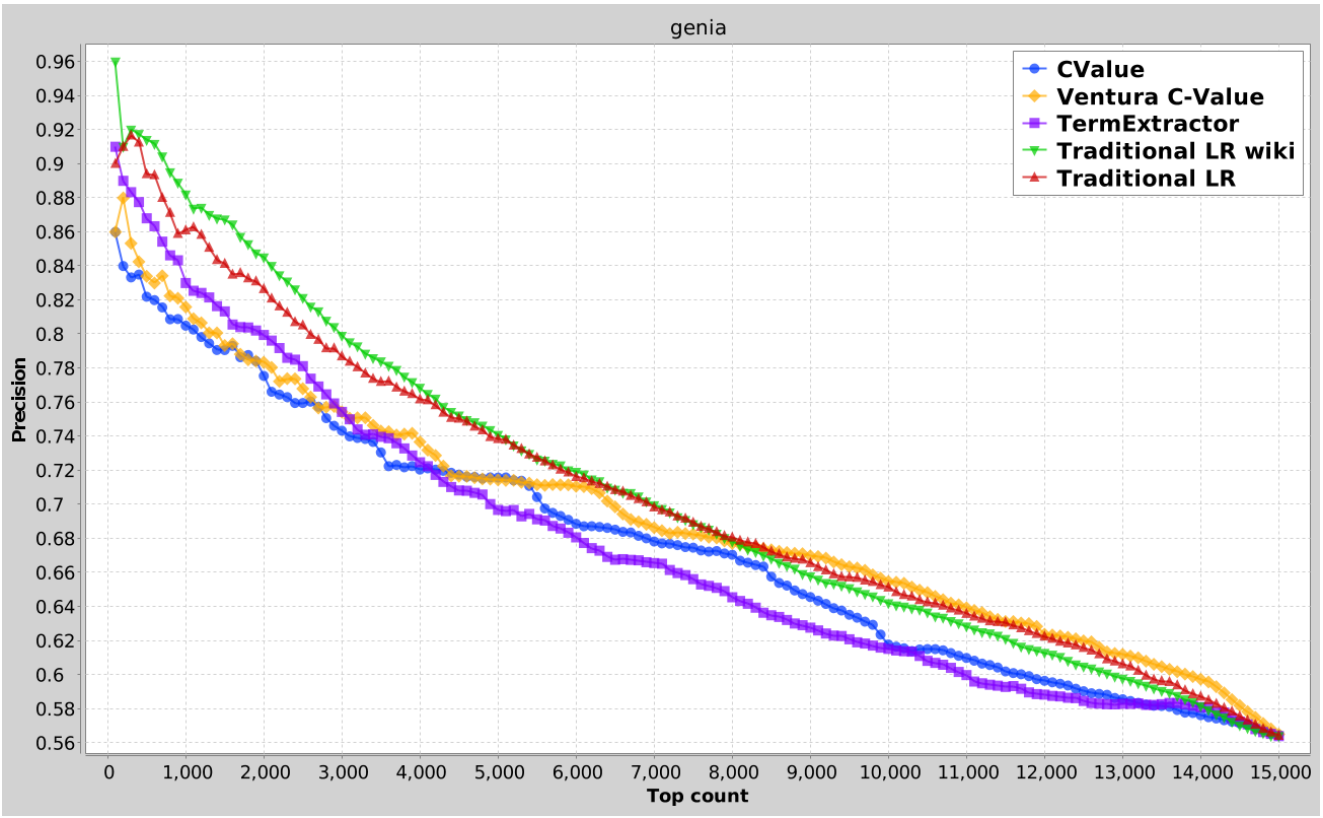

Рисунок В.3: Зависимость точности от размера учитываемой части списка кандидатов для набора данных GENIA

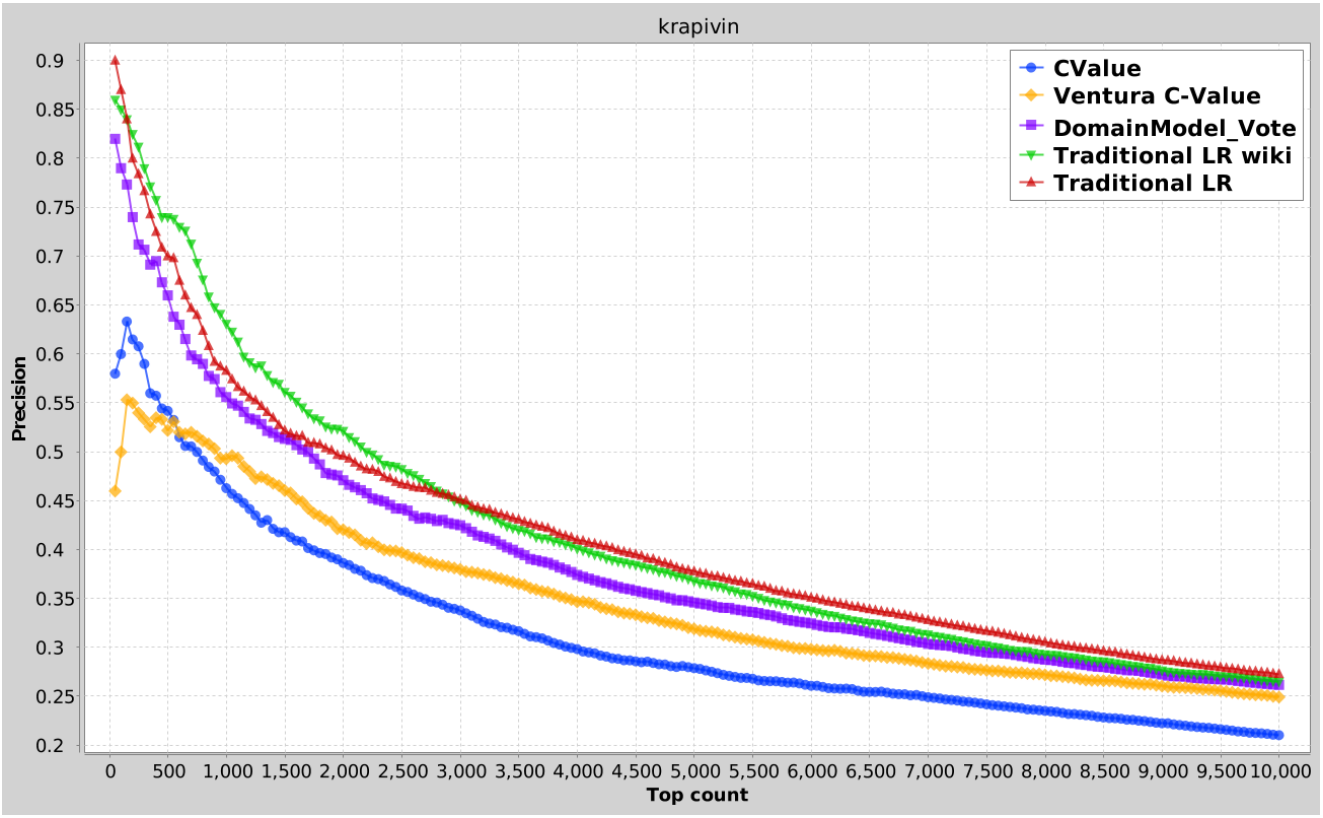

Рисунок В.4: Зависимость точности от размера учитываемой части списка кандидатов для набора данных Krapivin

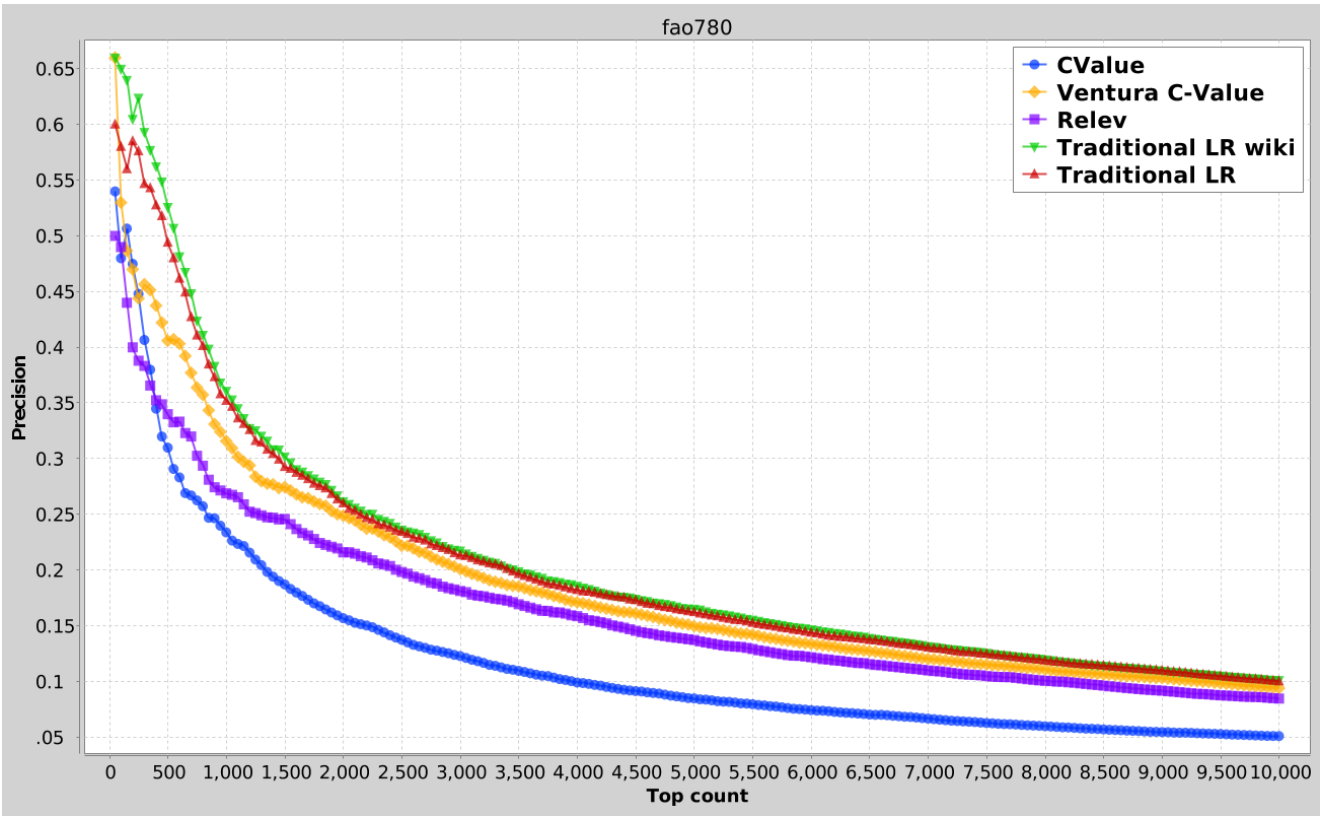

Рисунок В.5: Зависимость точности от размера учитываемой части списка кандидатов для набора данных FAO