МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА

На правах рукописи

Лунев Кирилл Владимирович

Теоретико-графовые алгоритмы выявления семантической близости между понятиями на основе анализа наборов ключевых слов взаимосвязанных объектов

Специальность 05.13.17 —
«Теоретические основы информатики»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук, профессор
Васенин Валерий Александрович

Москва — 2021

Оглавление

Стр.

Стр.

Стр.

Введение

Основными задачами современных информационных систем является эффективная организация сбора, хранения, систематизации, поиска и анализа данных. На настоящее время наиболее представительно и массово востребованные из таких систем способны хранить огромные объемы данных. Стремительный рост хранящейся в них информации приводит к необходимости исследования методов и инструментальных средств разработки программных комплексов, более эффективно решающих задачи организации сбора и хранения, поиска и анализа данных внутри таких больших систем.

Исследования, результаты которых представлены в настоящей диссертации, затрагивают важные и востребованные практикой задачи интеллектуального анализа объектов информационно-аналитической наукометрической системы. Автором предлагаются методы решения задачи определения семантической близости объектов, кластеризации объектов, поиска экспертов в различных областях научных знаний, определения тематической направленности объектов. Решение этих задач улучшает качество работы поисковых механизмов, упрощает работу конечного пользователя с системой, позволяет определять экспертные сообщества и находить коллекции похожих объектов в системе.

В рамках проведенных исследований основополагающей является задача определения семантической близости пары объектов. Постановка этой задачи требует определения понятия «семантически похожих объектов». Понятие семантической близости изучалось многими авторами (например, в работах [1; 2]). Далее под мерой (степенью) смысловой (семантической) близости и похожести (далее - «близость», «схожесть») будет подразумеваться показатель семантического сходства пары рассматриваемых слов или пары наборов слов естественного языка. Здесь следует также отметить, что в контексте рассматриваемой проблемной области мера не всегда является мерой в строгом математическом смысле. В этой сложно формулируемой проблемной области, как правило, активно используются интуитивно понятные эвристические соображения, понятия и основанные на них математические модели и алгоритмы.

Под мерой смысловой схожести в исследовании, результаты которого представлены в настоящей диссертации, будем понимать величину, которая сложно поддается формальному определению. Несмотря на это, интуиция позволяет дать

следущее определение паре семантически близких слов: если в речи или в письменном изложении присутствует возможность заменить одно слово на другое так, что смысл предложения не изменится, то два эти слова (заменяемое и замененное) семантически близки. Другими словами, у слушателя возникнет одинаковое представление о цитируемом объекте реального мира в обоих случаях.

Более того, легко дать определение семантически различным словам: после замены одного такого на другое предложение сильно изменяется по смыслу или даже становится абсурдным, то есть теряет какой-либо смысл, даже в том случае, если имеется возможность его «домыслить» до некоторого синтаксически корректного предложения (поставив, например, это слово в подходящую форму).

Рассмотрим, например, предложение «пошив мужского костюма» и последоватьльно будем заменять слово «костюм» на слова «одежда», «фрак», «обувь», «костюмер», «насекомые», «карнавал». Если в первых двух случаях заменя кажется разумной: «пошив мужской одежды», «пошив мужского фрака», то третий пример значительно изменяет смысл предложения: «пошив мужской обуви». При замене на четвертое и пятое слова, полученное предложение окончательно теряет смысл.

Исходя из этих соображений, можно всем парам семантически близких слов давать значение меры близости, равное 1, а всем различным - 0. Трудность возникает в случаях, когда пара слов не является в рамках данных определений ни парой близких по смыслу, ни парой различных по смыслу слов. Таким парам необходимо ставить некоторое промежуточное значение из интервала $(0, 1)$. В рамках примера, описанного выше, такой парой может являться пара «костюм-обувь». В этот момент возникает неоднозначность в определении того, какое именно значение должна получить данная пара и по какому принципу ранжировать близости различных пар. Что является более близкими понятиями: пара, связанная отношением гиперонимии («стол-мебель»), пара слов, часто встречающаяся в одних предложениях («уголовный-кодекс») или «слова-братья», имеющие общего предка-гиперонима («декабрь-ноябрь»)?

Ответ на этот вопрос кроется в постановке задачи, которую мера семантической близости призвана решить. Если рассматривать в качестве системы интернет-магазин и решать внутри этой системе задачу рекомендации товаров, то логично предложить пользователю такой товар, которые часто покупают с тем, что он приобрел. Другими словами, в качестве «близкого по смыслу» взять тот, который чаще всего встречается с заданным. Следует однако отметить, что в данном

примере предложение товара, абсолютно идентичного по смыслу с уже купленным (тот же самый товар), не имеет смысла.

Другим примером различной трактовки семантической близости может быть классическая поисковая система с программным модулем поисковых расширений. Поисковые расширения - это модуль, добавляющий в текст запроса пользователя новые слова, связанные с запросом. Это позволяет при ранжировании документов вывести на более высокие позиции документы, содержащие эти новые добавленные слова.

Если эти слова - действительно синонимы к словам запроса, то скорее всего выдача обогатится и это улучшит ранжирование в целом. Если же расширить слова запроса гиперонимами (например, по запросу «купить iphone Москва» для слова «iphone» добавить гипероним «смартфон», а для слова «Москва» - «Россия», то в выдачу попадут предложения о продаже различных смартфонов (не только iphone), которые, к тому же, будут продаваться не только в Москве, а по всей России. Тем не менее, в некоторых случаях, например, когда задано слово, по которому нет практически никаких документов в базе, добавление документов с гиперонимом к этому слову обычно является разумной идеей.

Однако, самый плохой из возможных сценариев - расширять словами-братьями. В этом случае iphone заменится на (в некотором смысле) близкое по смыслу samsung и пользователь будет весьма опечален: если в прошлой ситуации он попадал на сайты со *смартфонами*, среди которых мог быть нужный ему, то теперь ему целенаправленно показывают на выдаче нерелевантный для него товар.

Кроме того, можно заметить, что уровень близости зависит от тематики той системы, в которой слова употребляются. В системе самой общей тематики слова «школа-университет» должны иметь достаточно высокий уровень близости. Если же рассматривать некий образовательный портал, тематическая направленность которого узкоспециальна и относится к образованию, близость данной пары должна быть заметно ниже, потому что в данном контексте это два совершенно разных учебных заведения и различия в данной ситуации имееют принципиальное значение.

Помимо этого, слова могут быть многозначными и даже тривиальная пара «орган-орган» может иметь уровень схожести близкий к нулю, если считать, что первое слово - это музыкальный инструмент, а второе - часть тела или термин из юриспруденции (при этом совершенно не очевидно, какое из этих двух значений ближе по смыслу к музыкальному органу).

Описанные выше примеры показывают всю неоднозначность трактовки семантической близости на примере слов естественного языка. В рамках данной работы принимается следующее правило: чем слабее изменяется смысл при замене одного слова вторым в различных предложениях, содержащих первое слово, тем больше семантическая близость этой пары слов.

В этой связи еще раз подчеркивается, что для достижения поставленых в данной диссертации целей, были использованы модели, которые во многом опираются на эвристические, интуитивные понятия. Следует также отметить, что задача разрешения смысловой многозначности в рамках данной работы не рассматривается.

Высоким уровнем близости должны обладать синонимы в привычном значении из языкознания, правильные расшифровки аббревиатур, переводы слова на другие языки, формы одного слова, различные способы написания. В следующей далее таблице 1 приведены примеры семантически похожих пар ключевых слов из наукометрической системы:

Первое слово	Второе слово	Комментарий
умение	навык	Синонимия
полином	многочлен	Синонимия
β-адреноблокаторы	бета-адреноблокаторы	Различные способы написания одного слова
орви	острые респираторные вирусные инфекции	Расшифровка аббревиатуры
хехцир	khekhtsyr	Транслитерация
корень	корни	Разные формы одного слова
crisis	кризис	Перевод на другой язык
cu	медь	Другая форма названия

Таблица 1 — Примеры семантически близких ключевых слов

Одним из известных и широкоиспользуемых способов высокоуровневого описания данных, представленных в системе, является использование *ключевых слов*. Ключевые слова (или теги) - это набор слов естественного языка или терминов, которые коротко описывают отдельный документ, который хранится в информационной системе. Они используются в качестве метаинформации для публикаций (в том числе и научных) в средствах массовой информации и печатных изданиях. Такой подход позволяет читателю быстро понять основное направления изложения и концептуальные положения представленной информации, отметить некоторые понятия и сущности, с помощью которых решаются представленные в этих публикациях задачи.

Многие современные информационно-коммуникационные структуры, такие как социальные сети, блоговые и поисковые системы, используют ключевые слова для описания содержащихся в них сущностей (объектов). Такой подход значительно упрощает для пользователя поиск необходимых ему объектов системы, потому что позволяет сделать это с помощью запроса к системе на естественном языке. Кроме того, ключевые слова помогают поисковым системам по данному запросу выделять наиболее релевантные объекты системы. К числу таких объектов относятся, например, текстовые документы, изображения, видеозаписи и любой другой объект, которому был приписан набор ключевых слов. Многие исследователи активно занимались и продолжают заниматься анализом ключевых слов в целях кластеризации, визуализации, классификации, индексации и поиска целевых объектов.

Кроме того, ключевые слова можно рассматривать и как классификаторы контента, формирующие тезаурус предметной области, на основе которого этот контент описывается. Примером такого классификатора является *универсальная десятичная классификация (УДК)*, используемая для систематизации и группировки накопленных человечеством знаний по тематическим разделам. Данная классификация различным областям науки, литературы и искусства ставит в соответствие цифровые коды. Описание областей задается с помощью небольшого набора ключевых слов, характеризующих данное направление. Данные УДК построены по иерархическому принципу: более общие направления науки (а также соответствующием им коды) описываются общими по смыслу словами, например, «Общественные науки». При углублении и выборе определенной специализации внутри данного направления, описание приводится с помощью более конкретных понятий, таких как «Политика», «Право», «Экономика», «Народное

хозяйство» и т.д.. Таким образом, с помощью небольшого множества ключевых слов появляется возможность структуризации необходимого любого направления и соответствующего ему кода. Несмотря на все многообразие тематик, поиск необходимого кода не занимает много времени, что является возможным благодаря использованию ключевых слов.

Важным является то обстоятельство, что реальные информационно-аналитические системы во многих случаях не обладают достаточным объемом данных для анализа. Рассмотрим, например, научные публикации, как объекты наукометрической системы. Зачастую в таких данных отсутствует полнотекстовая информация. Доступной в этом случае является лишь метаинформация: авторы, название, ключевые слова. В связи с этим, в рамках проводимых исследований информация об объекте ограничивается набором ключевых слов на естественном языке, а также связями данного объекта с другими объектами системы. Другими словами, каждому объекту системы ставится в соответствии набор ключевых слов. Семантическая близость между объектами такой системы сводится к семантической близости между соответствующими им наборами ключевых слов. В свою очередь, семантическая близость между парой наборов ключевых слов опирается на семантическую близость между словами, входящими в эти наборы.

Кроме того, отмечается, что каждый объект описывается очень малым объемом данных (обычно это 5-6 слов). Это обстоятельство вносит существенные ограничения в допустимые методы решения обозначенных выше задач.

В качестве предмета исследования и анализа в диссертации выступают объекты наукометрической информационно-аналитической системы, которые описываются наборами ключевых слов. Кроме того, объекты такой системы связаны между собой различными отношениями, например, для научной публикации это может быть список соавторов, для научного работника - список проектов, в выполнении которых он принимал участие, или список конференций, в которых он принимал участие. Публикации, персоналии, научные проекты и конференции в данном примере являются объектами и, следовательно, могут иметь собственные наборы ключевых слов.

Побудительным мотивом и конечной целью исследований, результаты которых представлены в настоящей диссертации, является создание интеллектуального программного модуля, встраиваемого в наукометрическую информационно-аналитическую систему, способного по имеющимся в системе

ключевым словам и определенным связям между ними выявлять семантическую информацию и с ее помощью решать задачи информационного поиска и классификации. Следует также отметить то обстоятельство, что зачастую информационные системы не обладают большим объемом данных для анализа, что делает затруднительным качественное семантическое сравнение объектов. Однако и в таких системах необходимо уметь точно определять релевантную пользователю информацию. Как следствие, важным требованием к разрабатываемому модулю является его способность эффективно работать в условиях ограниченного объема входной информации.

Кроме того, отсутствие достаточного объема данных в реальных системах показывает актуальность и востребованность на практике исследования, результаты которого представлены в настоящей диссертации.

Целью диссертационной работы является исследование и разработка математических моделей, алгоритмов и программных средств интеллектуального анализа наборов ключевых слов, характеризующих объекты в наукометрических интеллектуальных системах, с использованием методов из теории графов и дополнительной информации онтологического характера об объектах в системе. Такая деятельность соответствует областям исследования, отмеченным в пп. 1, 2, 5, 9 Паспорта специальности 05.13.17 – теоретические основы информатики.

Требования к разрабатываемой системе. Согласно стандарту ГОСТ Р ИСО/МЭК 9126-93 к качеству разрабатываемой системы интеллектуального анализа объектов информационной системы предъявляются следующие требования:

– широкие функциональные возможности;
– надежность;
– эргономичность;
– эффективность;
– сопровождаемость;
– мобильность.

Более подробно изложенные выше пункты определены в приложении А. Описанные в данном приложении характеристики определяют отличительные стороны решаемой в настоящей диссертации задачи от известных работ, связанных с выделением семантической информации между объектами информационных систем. Существующие системы в большинстве своем опираются на обилие входных данных, к числу которых относятся:

– текстовая информация, а именно - аннотации, заголовки, полные тексты документов;
– общие объемы данных системы, которые характеризуются значительным количеством сущностей внутри системы и числом связей между ними.

В то же время, разрабатываемый программный комплекс является более гибким решением для систем, не обладающих большим объемом данных. Такие системы с одной стороны не содержат в себе огромного количества различных объектов. С другой стороны, о каждом из объектов известно минимальное количество информации - сущности таких систем должны лишь обладать описывающим их набором ключевых слов, либо быть соединены внутренними связями с сущностями, которым набор ключевых слов ассоциирован. Кроме того, разработанные подходы позволяют получать узконаправленные семантические модели для конкретной области знаний. Ручной труд при внедрении таких систем сводится к минимуму.

В разделе 1.1.4 содержится краткое изложение мотивации предъявленных требований к системе, разрабатываемой в данной работе. Кроме того, представлены недостатки существующих методов решения подобных задач. Описываются проблемные места, которые не позволяют применять эти подходы к некоторому классу систем. В конечном итоге выделяется **специфика** разрабатываемого комплекса, отличающего его от известных аналогов. На основе анализа перечисленных требований была разработана методология решения поставленной задачи.

В работе применяются методы анализа текстов на естественном языке, методы машинного обучения и программной инженерии. При изложении результатов диссертационной работы широко используется аппарат теории графов, а также математической логики и математической статистики.

Положения, выносимые на защиту. На защиту выносятся: обоснование актуальности, научная новизна, теоретическая и практическая значимость работы, а также следующие положения, которые подтверждаются результатами исследования, представленными далее в заключении диссертации.

1. Создание моделей и их программных реализаций вычисления уровня семантической близости между ключевыми словами интеллектуальной системы с учетом специфики этих систем.
2. Создание методов автоматической генерации обучающей выборки для определения семантически близких ключевых слов.

3. Создание модели и ее программной реализации для вычисления семантической близости между парой объектов информационно-аналитической системы по ассоциированным с ними наборами ключевых слов.
4. Решение востребованных практикой задач в рамках рассматриваемой информационно-аналитической системы. Среди таких задач выделяются следующие: поиск экспертов в различных областях научных знаний, кластеризация ключевых слов, определение тематической направленности объекта информационно-аналитической системы.

Научная новизна работы определяется тем, что автором разработаны новые алгоритмы определения семантической близости для пары ключевых слов, а также для пары наборов ключевых слов, описывающих объекты интеллектуальной наукометрической системы. Созданы уникальные методы автоматической генерации обучающей выборки, а также методы автоматической проверки качества работы программных реализаций алгоритмов определения семантически похожих ключевых слов и алгоритмов выявления кластеров близких понятий. Последнее обстоятельство важно, поскольку тестирование программ в данной предметной области требовательно к наличию специалистов, способных точно определить степень близости для пары объектов или понятий. Разработаны алгоритмы построения иерархических классификаторов научных направлений в автоматическом режиме, использующие исключительно наборы ключевых слов.

Важными особенностями указанных алгоритмов являются:

1. отсутствие необходимости больших объемов данных для обучения моделей с приемлемым уровнем качества;
2. возможность использования разработанных моделей для произвольных интеллектуальных систем, использующих ключевые слова для описания сущностей;
3. возможность применения к любым задачам, в которых объекты системы представляются в виде некоторого графа и имеется необходимость в классификации отношений между парой объектов;
4. небольшие человеческие трудозатраты для выставления экспертных оценок.

Проведена работа по уменьшению числа параметров системы, что делает разработанные модели и программные средства эргономичными и легкими для настройки.

Теоретическая значимость работы. Разработаны алгоритмы вычисления уровня семантической близости между ключевыми словами интеллектуальной системы, а также алгоритмы вычисления семантической близости между парой объектов информационно-аналитической системы. Доказана вычислительная сложность разработанных алгоритмов, подтверждающая их адекватность (соответствие) требованиям, предъявляемым к разрабатываемому программному комплексу.

Практическая значимость работы. Рассматриваемый в работе программный комплекс для анализа, обработки и поиска объектов интеллектуальных информационных систем по ключевым словам представляет собой самостоятельный инновационный продукт. Он может использоваться не только в системе, рассматриваемой в данной диссертации, но и в любой информационно-аналитической системе, объекты которой описываются наборами ключевых слов. Кроме того, разработанные автором методики обработки связей между объектами могут быть перенесены на другие задачи анализа взаимосвязанных объектов. Рассматриваемый программный модуль определения семантической близости между словами порождает словарь синонимов той области, на которой был обучен. Этот словарь может быть использован в самых разнообразных задачах информационного поиска и обработки естественного языка, и потенциально может привнести дополнительный полезный сигнал для моделей классификации, ранжирования и кластеризации текстовых или текстово-аннотированных объектов.

Методология исследования включает следующие характеризующие ее аспекты.

- **Концептуальные положения.**
  - Опора на наборы ключевых слов, ассоциированых с объектами информационной системы.
  - Возможность использования различных информационных объектов (источников) - НИР, публикации, патенты и т.п..
  - Наличие механизмов, позволяющих в автоматизированном режиме получать оценки адекватности полученных решений.
- **Модели, методы и средства достижения цели.**
  - Модели, реализующие концептуальные положения создания и развития на основе графовых представлений данных и эвристических алгоритмов над ними, работающие в условиях отсутствия строгого математического описания.

  - Методы машинного обучения, необходимые для улучшения качества определения семантических связей между объектами информационной системы;
- **Инструментальные средства.** Для разработки программных комплексов, решающих поставленные задачи, использованы открытые математические, графовые библиотеки, программные пакеты для обработки естественного языка, программные реализации моделей машинного обучения с отрытым исходным кодом.
- **Перечень и постановка задач, решение которых обеспечивает достижение цели.**
  1. Разработка графовой модели представления данных. Необходимо представить данные системы в виде множества графов, вершинами которых являются некоторые понятия (ключевые слова/наборы слов/сущности системы), а ребрами - отношения между ними. Такие графы необходимы для вычисления различных характеристик для пар понятий. Решение задачи приводится в главе 2.
  2. Разработка моделей определения семантической близости пары ключевых слов. Для этого используются построенные графы, разработанные подходы и технологии машинного обучения. Формируется набор количественных признаков и решающее правило, определяющее степень семантической близости по этому набору. Описание разработанных моделей приводится в главе 2.
  3. Разработка моделей определения семантической близости пары наборов ключевых слов. Разработанные модели используют различные графовые представления, подходы и модели, рассмотренные в предыдущих пунктах. Решению этой задачи посвящена глава 3.
  4. Апробация разработанных моделей. Используя функцию близости наборов ключевых слов и отношения между сущностями системы, решаются прикладные задачи определения семантической близости пары сущностей. Этой задаче посвящается глава 4.

Для достижения поставленных целей и удовлетворения описанных выше требований были рассмотрены различные методы решения, их преимущества и недостатки. По окончании поиска была составлена методология исследования, наиболее подходящая поставленным в настоящей диссертации задачам в рамках имеющихся особенностей и ограничений в наборов исходных данных. Подробная мотивация выбранной методологии описывается в разделе 1.2.

Соответствие диссертации паспорту научной специальности. Полученные в диссертации результаты соответствуют паспорту специальности 05.13.17 — теоретические основы информатики (физико-математические науки). Теоретические основы информатики – специальность, включающая исследования процессов создания, накопления и обработки информации; исследования методов преобразования информации в данные и знания; создание и исследование информационных моделей, моделей данных и знаний, методов работы со знаниями, методов машинного обучения и обнаружения новых знаний; исследования принципов создания и функционирования аппаратных и программных средств автоматизации указанных процессов. Области исследования:

1. Исследование, в том числе с помощью средств вычислительной техники, информационных процессов, информационных потребностей коллективных и индивидуальных пользователей.
2. Исследование информационных структур, разработка и анализ моделей информационных процессов и структур.
5. Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечениях разработка и исследование методов и алгоритмов анализа текста, устной речи и изображений.
9. Разработка новых интернет-технологий, включая средства поиска, анализа и фильтрации информации, средства приобретения знаний и создания онтологии, средства интеллектуализации бизнес-процессов.

Апробация работы. Результаты, изложенные в диссертации, докладывались на следующих семинарах и конференциях:

1. на всероссийской конференции с международным участием «Знания–Онтологии–Теории (ЗОНТ-2015)»,
2. на международной конференции «Ломоносовские чтения» (2014, 2016, 2018, 2019),

3. на механико-математическом факультете МГУ имени М.В. Ломоносова на семинаре «Проблемы современных информационно-вычислительных систем» под руководством д.ф.-м.н., проф. В.А. Васенина (2013, 2015, 2017, 2018),
4. на семинаре в ВЦ РАН под руководством д.ф.-м.н., проф. В.А.Серебрякова (2018).

Публикации. По теме диссертации опубликовано 7 работ, в том числе одна в зарубежном издании. Из них 5 научных статей опубликованы в журналах Scopus и RSCI, что соответствует требованиям, предусмотренным пунктом 2.3 Положения о присуждении учёных степеней в МГУ.

По результатам апробации модели и реализующих её программных средств получен акт о внедрении.

Объем и структура работы. Диссертация состоит из введения, четырёх глав, заключения и двух приложений. Полный объём диссертации составляет 195 страниц, включая 16 рисунков и 16 таблиц. Список литературы содержит 130 наименований.

Глава 1. Методы и средства анализа информации с использованием ключевых слов

В данной главе рассматриваются известные подходы к решению задач, направленных на изучение семантической информации, проводится их анализ, выявляются недостатки. После чего автором описывается использованная в настоящей диссертации методология. Эта методология призвана решить недостатки в существующих моделях определения смысловой близости объектов информационных систем по их ключевым словам.

В первом разделе проводится библиографический обзор существующих методов, выделяются сильные и слабые стороны моделей. Анализу подвергаются работы, в которых исследуются семантическая близость для пар слов и пар коротких предложений естественного языка. В частности, обозреваются работы, связанные с анализом ключевых слов.

Второй раздел содержит в себе выводы из обзора и указывает на недостатки рассмотренных методов и на причины сложностей применения этих методов к решаемым в данной работе задачам.

В заключающем разделе главы вводится методология, используемая автором в последующих главах для разработки моделей семантической близости. Показывается целесообразность и необходимость рассмотренной методологии, ее преимущества в сравнении с уже существующими. Следует также отметить, что отдельные пункты методологии представляют собой самостоятельный интерес.

1.1 Библиографический обзор

С целью анализа эффективности уже существующих и поиска новых подходов к решению рассматриваемой задачи автором проведены библиографические поисковые исследования, результаты которых представлены в настоящей разделе.

1.1.1 Методы определения близости между парой слов естественного языка

Существует большое число общих методов определения похожести пары слов естественного языка. Их можно разделить на методы, не использующие или использующие дополнительные источники информации.

Методы, использующие только информацию о написании слов. К числу таких относятся исторически наиболее ранние и наивные подходы, которые вычисляют меру близости, не используя никакой дополнительной информации о словах, кроме их непосредственного написания. Одной из основных метрик данного типа является расстояние Левенштейна (редакторское расстояние, [3]). Эта метрика подсчитывает количество необходимых операций добавления, удаления или замены одного символа на другой, чтобы из одной строки получить вторую. Традиционно этот алгоритм используется для исправления опечаток: для введенного слова можно найти ближайшие по этой метрике слова из фиксированного словаря.

Существует ряд более сложных версий алгоритма, в числе которых алгоритм Демерау-Левенштейна [4]. Усовершенствование этого алгоритма заключается в том, что дополнительно используется четвертая операция транспозиции двух соседних символов.

Несмотря на простоту описанных выше методов, исследования в этом направлении ведутся до сих пор. Данный класс методов может использоваться в более сложных и совершенных моделях в качестве дополнительных источников для определения близости. Примером более сложной модели в данном направлении является модель редакторского расстояния с настроенными стоимостями для операций вставки, удаления и замены символов. Авторы [5] с помощью разработанных алгоритмов и обучающей выборки определяют стоимость замены одного символа на другой (а также добавления и удаления каждого символа). Авторы высказывают гипотезу о том, что различные замены символов не должны иметь один и тот же вес: если человек опечатался, то весьма вероятно, что он ввел символ, который находится близко к правильному символу на клавиатуре. Такая замена не должна сильно влиять на общее расстояние метрике. Другим примером важности индивидуального подбора весов замен под каждый символ могут являться безударные гласные: люди чаще путают при написании пару букв «а» и «о», чем, например, пару букв «а» и «е». Важным достоинством такого

алгоритма является то, что слово и его транслитерированная версия (например, «компьютер»-«computer») становятся близки по данному расстоянию. В то же время, недостатком является необходимость обучающей коллекцию различных написаний одного слова.

Следующий важный этап развития идеи редакторского расстояния заключается в использовании контекста. В работе [6] авторы настраивают стоимости переходов между символами с учетом контекста. Выдвигается гипотеза о том, что стоимость замены одного символа на другого может сильно зависеть от символов, которые стоят рядом с заменяемым символов и символом-заменителем. Например, удаление символа «ь» более обоснованно в конце глаголов, так как ошибки «ться»/«тся» частотны и вероятнее всего подразумевал одно из слов, написав другое. Как и в предыдущей работе, данный алгоритм требует обучающую выборку, но в данном случае ее размер должен быть значительно больше, поскольку число параметров растет экспоненциально с увеличением размера рассматриваемого контекста.

Еще одной разновидностью метрик на строках является расстояние Джаро — Винклера ([7; 8]). Эта метрика подсчитывает минимальное число односимвольных преобразований, которое необходимо для того, чтобы изменить одно слово в другое и использовалась для сравнения написаний имен в Бюро переписи населения США.

Ряд авторов рассматривают слово как множество символов (или как множество символьных n-грамм - последовательностей из n подряд идущих символов) и далее определяют близость между словами, как близость между соответствующими множествами, либо как близость между соответсвующими векторами, на i-ой позиции в которой стоит единица, если данная n-грамма присутствует в слове и ноль - в противном случае. Примером метрики на векторах может являться алгоритм q-grams ([9]), в ходе работы которого подсчитывается число совпавших n-грамм. Другими метриками могут служить, расстояние Жаккара или косинусное расстоения, приведенное, например, в [10].

Другим направлением исследований в области определения смысловой близости пары слов является использование фонетической информации рассматриваемых слов. Примерами таких работ могут служить [11; 12]. Работы опираются на гипотезу о том, что похожие слова могут звучать одинаково. Авторы первой работы по слову строят его короткий код таким образом, чтобы различные

слова с одним кодом звучали похоже. Авторы второй работы предлагают различные алгоритмы определения фонетической близости, в том числе они используют расстояние Левенштейна на фонемах для рассматриваемых слов.

Таким образом, данное направление позволяет строить метрики близости, основанные исключительно на написании конкретных слов. Данный класс методов позволяет достаточно эффективно решать задачу исправления опечаток, но имеет множество недостатков. Основной из них заключается в том, что при его использовании не учитывается семантика слов. Это обстоятельство существенно сужает круг прикладных задач, для решения которых эти методы можно было бы применить.

Методы, использующие дополнительные знания о словах. Существенного улучшения качества определения близости между парой ключевых слов можно добиться, используя дополнительные знания о словах. Это могут быть тексты, в которых слова употреблены, коллекции ключевых слов, информация об объектах, к которым эти слова приписаны, вручную составленные тезаурусы и словари. Каждое из этих направлений имеет свои преимущества и недостатки, анализ которых приведен далее.

Одно из направлений определения близости пар ключевых слов с использованием вспомогательных данных - **изучение частот использования** рассматриваемых слов в различных коллекциях текстовых документов. Базовым методом определения близости пары ключевых слов в рамках таких исследований является сбор информации о совместной встречаемости слов внутри одного набора. Факт появления пары слов в одном предложении или тексте может быть важным сигналом для определения смысловой близости. Методы, основанные на этой идее, разбираются в [13—15]. Использование таких алгоритмов позволяет получить решение об уровне близости пар слов не только по их совместной встречаемости, но и путем учета частоты встречаемости каждого из слов в коллекции. Согласно метрике, введенной в [15], высокое значение семантической близости имеют пары слов, которые часто встречаются вместе и редко поодиночке.

Недостатком описанных выше статистических методов является необходимость сбора коллекции данных большого размера, поскольку значения PMI и подобных ему статистических метрик сильно неустойчивы ([16]). Зачастую эта особенность приводит к тому, что наиболее близкими парами в смысле этой меры близости являются те, которые встретились единственный раз в одном общем документе. Для того, чтобы уменьшить негативный эффект на практике,

принято исключать пары слов, которые встретились в корпусе меньше некоторого порогового значения. Другой способ обойти сложившуюся трудность в ином определении вероятностей появления каждого из слов, а также вероятности их совместного появления. Для этого служат методики сглаживания вероятности, принятые в области построения языковых моделей. Основные способы сглаживания представлены в работе [16]. Также существуют усовершенствования PMI меры различными эвристическими предположениями: усредненная и средневзвешенная взаимная информация (average and weighted average mutual information), рассмотренные, соответственно, в [17] и [18]; контекстная усредненная взаимная информация (contextual average mutual information), введенная в [19]; нормированная взаимная информация (normalized mutual information), введенная в [20], квадратичная и кубическая взаимная информация (PMI2 и PMI3), рассмотренные в [21]. В этих публикациях отмечается, что сам контекст, в котором употребляются слова, используется в самом примитивном виде, а именно, в данном контексте проверяется факт наличия обоих рассматриваемых слов. Остальные слова контекста никак не учитываются, что является существенным недостатком описанных выше подходов.

В другой работе [22] вопрос вычисления семантической близости решается с помощью поисковых систем. Программа запрашивает пару сравниваемых слов через открытый API и получает совместную и индивидуальные частоты встречаемостей слов в интернете. На основе этой информации подсчитывается уровень похожести слов друг на друга. Очевидным недостатком такого подхода является ограниченная поисковой системой пропускная способность (количество запросов в единицу времени) и общее количество запрос.

Различные вариации PMI-метрик являются, по сути, вероятностными методами, поскольку подразумевают вычисления оценки вероятности встретить каждое из понятий, а также эту пару понятий совместно внутри одного текста. Существуют и другие вероятностные методы сравнения пары слов естественного языка. Например, может быть использован χ^2 критерий и тест отношения правдоподобия. Способ применения данных методов описан в [23]

Важной особенностью в применении PMI-подобных метрик к набору текстов является то, что они не показывают смысловую близость между понятиями в явном виде, а скорее определяют коллокации: «Российская Федерация», «крейсер Аврора», «завод имени Кирова», «средний класс», «пластическая операция» и другие. Тем не менее, знание того, что совместная встречаемость пары понятий

внутри одного множества слов (предложение, документ, набор ключевых слов, короткое описание объекта и т.д.) статистически значимо превосходит случаи их отдельных появлений, является важным фактором для определения в том числе и семантической близости между этими понятиями.

Многие подходы к решению задачи определения близости пары слов используют понятие n-граммы. Символьной/пословной n-граммой называют последовательность фиксированной длины из определенного числа подряд идущих символов/слов. Символьные и пословные n-граммы широко используются в различных задачах из области обработки естественного языка таких как построение языковых моделей [24—27] и моделей машинного перевода [28—30]. Недостатком n-граммных моделей является так называемое проклятие размерности, которое в данном случае свидетельствует о том, что при увеличении длины n-граммы катастрофически быстро растет число возможных n-грамм данного размера, а также параметров системы, что делает затруднительным их применение во многих случаях. Также при работе с n-граммами необходимы объемы текстов огромных размеров. Если предметная область, в которой решается задача, является узкоспециальной, то получение данных достаточного объема зачастую является невозможным.

В работе [31] авторами предложены различные метрики близости на основе n-граммного представления слов. В работе [32] авторы используют меру Жаккара для определение близости пары слов: каждому слову ставится в соответствии множество понятий и для определения близости вычисляются размеры объединения и пересечения этих множеств. Отмечается, что как только словам в однозначное соответствие поставлены некоторые множества, сразу становится возможным вычисление близости на основании различных мер близости пары множеств. Помимо меры Жаккара, существует чуть менее популярная мера Серенсена ([33]). Эти и другие меры близости на множествах подробно описаны в [34]. В работе [35] пара слов считается близкой, если существует много документов сети Веб, в которых присутствуют оба слова.

Авторы [36] для определения близости используют комбинацию из трех моделей определения близости: n-граммную модель, модель близости жаккара, а также модель векторного пространства. Последняя подразумевает представление слов в виде вектора определенной длины. Близость в свою очередь сводится к величине скалярного произведения векторов для двух слов. Эта модель детально описывается в [37]. Имея три функции близости, авторы вычисляют среднее

по их значениям. Это приводит к тому, что такая композиция уменьшает влияние слабых сторон каждого отдельного алгоритма. Тем не менее такой наивный метод комбинирования может даже ухудшать результат, если входящие в нее модели демонстрируют низкое качество.

Важной особенностью алгоритмов, основанных на вычислении символьной n-граммной близости, расстояния левенштейна и наибольшей общей подпоследовательности, является то, что такие методы не дают представления о смысловой близости между словами и способны определять близкие слова только по похожести написания. Это является серьезным недостатком для задач, в которых важна смысловая близость между понятиями. Примером такой задачи может быть разработка классической поисковой системы, где удачное добавление синонимов для слов запроса, сформулированного пользователем, может вылиться в более релевантную выдачу для этого запроса. Несмотря на этот недостаток, данные методы могут быть применены внутри более сложных алгоритмов для повышения их качества.

Существует класс методов, решающих задачу семантической близости пар слов с помощью **готовых тезаурусов, словарей или других семантических сетей**. Важным источником знаний об отношениях между словами английского языка является семантическая сеть WordNet ([38]). Слова в данной сети могут быть связаны одним из нескольких отношений: гипероним, гипоним, «имеет участника» (факультет-профессор), «является участником» (пилот-экипаж), мероним, антоним. Также имеются лексические, антонимические, контекстные связи между словами. Для русского языка существует несколько аналогов: RussNet ([39]), YARN ([40; 41]), RuThes ([42]), Russian WordNet ([43]). Чтобы посчитать близость по таким тезаурусам, строится дерево, в вершинах которого стоят слова (вершина в WordNet является синсетом - множеством слов, не отличимых по смыслу), а ребра указывают на отношение гиперонимии между парой вершин. Таким образом, в листьях дерева лежат узкоспециальные понятия, которые обобщаются их предками в дереве, в корне же лежит слово наиболее общего значения. Имея такое дерево, появляется возможность вычислять смысловую близость понятий по их взаимному расположению внутри этого дерева. Так авторы [44] в своей формуле близости используют глубину наиболее конкретного по значению предка, а авторы [45] в дополнении к этому считают расстояние между вершинами. Чем больше расстояние между словами и чем глубже находится общий предок, тем

меньше уровень смысловой похожести. В работах [46; 47] используются гибридные методы определения близости: расстояния и глубина в дереве, вероятности встречаемостей в корпусах, признаки, основанные на свойствах слов в рассматриваемых тезаурусах. Недостаток тезаурусных подходов в их неполноте, а также в том, что некоторые из них не являются публично доступными. Кроме того, тезаурусы обычно охватывают общий домен и существует мало словарей для специфических областей.

Еще одним открытым источником отношений между понятиями является интернет-энциклопедия wikipedia (www.wikipedia.org). Данная база позволяет эффективно использовать категории, ссылки, полные тексты и мета-данные статей для извлечения семантической информации о словах. Авторы [48] строят различные меры близости, опираясь на тексты статей и представляя сравниваемые понятия в виде векторов определенной длины. В [49] автор использует данные википедии (в частности используются информация о ссылках между статьями) и разработанные им метрики близости слов для решения задачи снятия лексической неоднозначности.

С развитием вычислительной техники большой популярностью начинают пользоваться методы, основанные на обучении **нейронный сетей**. Одними из самых известных методов определения семантической близости слов является модели word2vec ([50]), GloVe ([51]) и StarSpace ([52]) . Модель word2vec представляет собой нейронную сеть, на вход которой подаются большие корпуса текстовых данных. Задачей обучения является построение такого векторного представления для текущего слова (word embeddings), которое максимально точно способно предсказать рядом стоящие в тексте слова. Обученная модель строит векторное пространство, обладающее рядом полезных свойств, которые в наше время широко используются для решения многих задач естественного языка, связанных с семантической информацией. Одним из таких свойств - семантическая близость понятий, векторные представления которых похожи. Таким образом любую пару слов из словаря можно сравнить, использую, например, косинусное расстояние между векторами.

Мощной моделью построения векторных представлений для слов является модель GloVe, которая строит матрицу частотностей встречаемостей слов во всех возможных контекстах. Далее используются методы уменьшения размерности пространства, которые оставляют только наиболее значимые компоненты в

разложении. В то время, как Word2Vec является предиктивной моделью, GloVe представляет собой модель на основе подсчета статистики.

Модель StarSpace является более современной моделью построения векторных представлений и обобщает идеи, заложенные в word2vec. Использование StarSpace дает возможность построения векторных представлений для слов, предложений или целых документов. Кроме того, возможно построение представлений для графовых данных.

Рассматривая известные подходы построения векторных представлений, нельзя не упомянуть про языковую модель BERT ([53]). Предобученные текстовые представления, полученные этой моделью, могут быть дообучены для решения конкретных задач обработки естественного языка. Примерами таких задач служат задачи разработки вопросно-ответной системы, построения автоматических переводчиков, создания модели анализа тональности текста, распознавание именованных сущностей. В ряде задач обработки естественного языка с помощью получены наилучшие из известных результаты.

Еще один способ построения представлений показан в работе [54]. Подход основан на предварительном вычислении меры *PMI* ([15]) и построением матрицы, в ячейках которой лежит значение PMI для пары рассматриваемых слов. После чего к этой матрице применяется SVD-разложение ([55]), что значительно понижает ее размерность. Строки новой матрицы пониженной размерности являются векторными представлениями для слов. Близость считается с помощью косинусного расстояния.

Различные методы векторного представления описаны и протестированы на открытых источниках в работе [56]. Данные методы являются очень эффективными, но требуют огромные наборы данных для обучения моделей. Это обстоятельство делает их неприменимыми к задачам, в которых полные тексты документов недоступны. К их числу которых принадлежит задача определения семантической близости пары ключевых слов научных публикаций. Эти методы зачастую определяют также контекстную близость, по определению которой два слова близки, если они встречаются в похожих контекстах. Во многих практических задачах подобного эффекта использования метрики близости хочется избежать, поскольку, например, слова “Математика” и “Физика” могут встречаться в одних и тех же контекстах, но как пара ключевых слова для научных публикаций эти слова явно не являются семантически близкими.

Несмотря на высокое качество определения семантической близости моделей, использование полнотекстовой информации существенно ограничивает область применения данных методов, посколько для многих прикладных задач не имеется достаточного количества текстовой информации. Возникает трудность при работе в узкоспециальных областях: модели, обученные на корпусах общего назначения, не могут улавливать особенности таких областей. Использование текстовых данных из рассматриваемой области для обучения ведет к неправильной настройке параметров модели и недообучению, по причине недостатка этих самых данных. Это приводит к низкому уровню качества моделей.

Существует группа При исследовании области семантической близости пары ключевых слов возникает дополнительная информация о наборах, в которые входят рассматриваемые слова. Помимо этого зачастую для набора ключевых слов известен также объект, к которому этот набор приписан. Авторы [57] вводят понятие **фолксономии**. Фолксономией называется кортеж (U, T, R, Y), где U,T и R - конечные множества, элементами которых служат, соответственно, пользователи, ключевые слова и ресурсы. Y - тернарное отношение между ними, т.е. $Y \subseteq U \times T \times R$. Постом пользователя u в ресурсе r называется тройка (u, T_{ur}, r), где $u \in U, r \in R$, T_{ur} - непустое множество ключевых слов такое, что $T_{ur} := \{t \in T | (u, t, r) \in Y\}$. Авторы [58] считают близость между ключевыми словами несколькими способами. Первый способ заключается в построении меры близости по статистике совместной встречаемости пары ключевых слов. Второй способ предполагает построение векторного пространства для каждого слова. На $i-$ой позиции стоит количество документов, в которые одновременно входит рассматриваемое ключевое слово и $i-$ое. Далее мера близости вводится как косинусное расстояние между векторами в этом пространстве. Последний способ, который описывается в [57] подсчитывает меру, подобную мере PageRank ([59]) для документов в сети Веб.

Недостатки существующих решений

Основные недостатки наиболее востребованных на практике существующих подходов для определения семантической близости пары слов естественного языка заключаются в следующем.

– Сложные модели определения близости требуют больших объемов полнотекстовой информации. Информационно-аналитические системы зачастую имеют на порядки меньшие объемы данных, что делает невозможным обучение таких моделей.
– Использование готовых моделей не позволяет в должной мере учитывать семантическую специфику конкретной информационной системы. Другими словами, пара семантически близких понятий одной системы может не являться таковой во второй. Например, пара слов «вычислительная математика» и «теория чисел» могут считаться похожими в системах общего назначения, как два направления в математике. В то же время, для более узкоспециальной наукометрической системы эти понятия не должны быть слишком близки семантически.
– Простые модели не способны восстанавливать сложные семантические связи в данных.
– Существующие модели не используют в достаточной мере дополнительную информацию об отношениях между сущностями системы.

1.1.2 Методы определения близости между объектами в графах знаний

Переходя на более высокий и целостностный уровень, можно отметить, что некоторые исследования в области семантической близости были проделаны и на этапе определения близости объектов информационной системы. Более того, исследователями зачастую решалась общая задача восстановления отношений между объектами, что не всегда равносильно наличию смысловой близости между ними.

Возможны различные способы хранения данных информационной системы. Одним из самых современных на момент написания является граф знаний. Графы знаний - ориентированные графы с подписанными ребрами (отношениями) между вершинами (сущностями). При этом количество возможных типов отношений может быть огромным и исчисляться тысячами. Существует другое название для графов знаний - *гетерогенная информационная сеть (heterogeneous information network, HIN)*. Этот термин представляется удобным в ситуациях, когда необходимо отличать системы с многочисленными типами отношений между

сущностями (гетерогенные) от систем, в которых сущности связаны единственным типом отношения (гомогенные информационные сети).

Примерами моделей в области гомогенных информационных сетей являются модели *SimRank* ([60]), *персонализированный PageRank* ([61]). По заложенным идеям данные работы похожи на уже описанную ранее работу [59]. Однако, в отличие от модели *PageRank*, числовая характеристика важности считается не для вершины графа, а для пары вершин, то есть для ребра. Тем самым, определяется близость между сущностями системы.

Во многих случаях в графы знаний восстановить пропущенные сущности и отношения, отсутствующие в данных. Например, если предположить, что отношение *БылРожденВСтране* не может быть прописано для каждой сущности, но при этом для всех них известно отношение *БылРожденВГороде*, то отношение *БылРожденВСтране* может быть восстановлено. Для этого также используется отношение *ГородПринадлежитСтране*.

Хранение информации в виде графов знаний является эффективным способом, поскольку объединяет в себе преимущества тезаурусов, таксономий и онтологий, оставаясь при этом удобным для использования и человеком, и машиной. Примерами различных графов знаний могут служить DBpedia [62], FreeBase [63], YAGO3 [64], Google Knowledge Graph [65].

В контексте задач, рассматриваемых в настоящей диссертационной работе, сущностями могут быть, например, ключевые слова, а восстанавливаемым отношениям - отношение синонимии пары слов. Аналогично можно составить графы знаний объектов наукометрической системы с ассоциированными словами, между которыми могут быть отношения *ЯвляетсяУчастником*, *ЯвляютсяСоАвторами*, *ЯвляетсяСотрудником* и т.д..

Далее приводится обзор современных методов восстановления связей в описанных выше графах знаний.

Существует два подхода к решению задачи восстановления отношений в графе знаний, каждое из которых детально описано в [66]. Авторы демонстрируют работу программных реализаций различных моделей. Эти модели восстанавливают ребра в графах знаний (например, в Google Knowledge Graph [65]). Первый из подходов к решению - графовые признаковые модели. Для решения задачи выбирается и подсчитывается некоторая метрика близости между вершинами (такими метриками могут являться локальные меры такие, как число общих соседей, индекс Адамика-Адара и глобальные меры такие, как индекс

Катца и другие). Модели, использующий подходы данного семейства, рассматриваются в [67—70]. Второй подход заключается в использовании матричных разложений. Применение таких подходов описывается в работах ([71; 72]).

Еще одним классом моделей, использующие графовые признаковые представления, являются модели, вычисляющие числовые характеристики по различным путям между парой вершин в графе. Например, в работе [73] рассматривается множество $\Pi_L(i, j, k)$ всевозможных путей длины L между сущностями $e_i, e_j \subset E$ системы. Переменная k означает, что данные сущностями связаны отношением $r_k \in R$. Далее с помощью случайного блуждания появляется возможность оценить вероятность существования каждого из путей между заданными вершинами. Ключевой идеей метода является использование вероятностей различных путей в качестве признакового описания для модели машинного обучения. Далее авторы обучают линейную модель (логистическую регрессию), позволяющую предсказывать вероятности существования отношения между любой парой сущностей. Важное преимущество такой модели заключается в ее интерпретируемости: после того, как программная реализация модели предсказала вероятность существования отношения между сущностями, появляется возможность узнать, на какие пути модель опиралась в большей степени в при подсчете вероятности.

Авторы модели *PathSim* в работах ([74; 75]) используют пути по различным отношениям в графе и на их основе определяет меру близости пары вершин. Эта мера демонстрирует высокий уровень качества на различных исходных данных. Вторая из цитируемых работ дополняет первую. В ней рассматриваются дополнительные внешние источники данных и способы привнести дополнительный сигнал в основную модель, тем самым, улучшив её качество.

Другим расширением идеи, заложенной в *PathSim*, является модель *W-PathSim* ([76]), которая для улучшения качества определения близости между сущностями использует методы тематического моделирования (латентное размещение Дирихле, [77]).

Второй подход к решению данной задачи заключается в построение латентных векторов для вершин и ребер графа. Как и в задаче определения семантической близости пары слов, в области восстановления отношений между сущностями в последнее время преобладают методы, строящие векторные представления для сущностей и отношений.

Примером построения векторного пространства на базе графа знаний может являться модель, построенная в [78]. Автором строится билинейная форма

$$f_{ijk}^{RESCAL} := e_i^T W_k e_j = \sum_{a=1}^{H_e} \sum_{b=1}^{H_e} w_{abk} e_{ia} e_{jb},$$

где f_{ijk}^{RESCAL} - значение близости между i-ым и j-ым сущностями системы по k-ому отношению, e_i - векторное представление размерности H_e для i-ой сущности, W_k - матрица параметров для k-ого отношения размерности $H_e * H_e$, w_{abk},e_{ia},e_{ib} - компоненты соответсвующих матриц и векторов. При такой постановке видно, что требуется настроить огромное число параметров, даже если размерность векторного представления выбрана относительно небольшого размера.

Некоторые исследователи в своих работах за основу модели берут описанную ранее модель *Word2Vec*. Примерами таких работ могут служить [79; 80] Графы, используемые в работах, в своих вершинах содержат слова естественного языка. В обоих работах целью является решение некоторой задачи регрессии или классификации на определенных исходных данных, частично или полностью представленных графами.

В первой работе для вершин графа проводится процедура *GraphWalk*. В рамках нее выбирается одна из вершин графа и выполняется переход по случайному его ребру к следующей вершине. Такой процесс проходит некоторое фиксированное число раз, после чего появляется последовательность вершин. Учитывая тот факт, что в вершинах лежит текстовая информация, то фактически получается некоторое предложение естественного языка. Собрав достаточное число предложений, обучаются разные виды модели *Word2Vec*. Далее векторное представление слов принимается за признаковое описание объектов, после чего появляется возможность обучать известные модели машинного обучения. А именно, в работе были использованы *Support Vector Machines, Linear Regression, Naive Bayes, k-NearestNeighbors*. С помощью этих моделей были решены поставленные задачи регрессии и классификации.

Вторая работа по принятым в ней идеям схожа с первой. Однако здесь отношения между вершинами также задаются словами естественного языка и рассматриваются тройки: исходная вершина, тип отношения, конечная вершина. Для этой тройки создается короткое текстовое предложение, после чего происходит аналогичное обучение модели.

В работе [81] была построена двухмерная сверточная нейронная сеть с добавлением полносвязных слоев для определения вероятности связи пары сущностей e_i, e_j отношением r_k. Другие подходы к построению представлений для пар сущностей можно найти в работах [82; 83].

Авторы популярной работы [84] предлагают следующую функцию расстояния: $f_r(h,t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2^2$, где h,t - субъект и объект, r - отношение, **h**,**r**,**t** - их векторные представления. Таким образом, авторы пытаются построить такое векторное пространство, в котором под действием отношения субъект будет переходить в точку, максимально близкую к истинному объекту.

Усовершенствованием предыдущей идее занимались авторы [85], которые показали, что пространство отношений не обязано быть таким же, как пространство сущностей и может иметь другую размерность. Для более эффективного вычисления функций близости были введены матрицы проекции, по которым можно было перевести векторное представления из пространства сущностей в пространство отношений. Отмечается, что такой подход усложняет модель [84], что требует настройки большего числа параметров и, соответственно, данных для обучения.

В работе [86] описывается созданный авторами программно-аппаратный комплекс интеллектуального поиска и анализа больших массивов текстов, именуемый *TextAppliance*. Основной функцией комплекса является семантический поиск текстовых документов в текстовых коллекциях большого объема. Кроме того, реализованы функции реферирования, семантического поиска заимствований, описанного в [87], кросс-языкового поиска, кластеризации документов системы. Перечисленный функции реализован с использованием методов лингвистического анализа, описание которых приводится в [88]. Одной из особенностей *TextAppliance* является способ представления текста в виде неоднородной семантической сети. На этом представлении основываются авторские алгоритмы нечеткого сравнения текстов. Является важным отметить тот факт, что комплекс разработан для обработки больших объемов полнотекстовой информации и многие инженерные решения направлены на эффективную распределенную работу различных модулей сбора, индексации и поиска информации. Следует однако отметить, что в рамках настоящей диссертационной работы не подразумевается наличие полнотекстовой информации, а также интеллектуальных систем большого размера, что делает затруднительным эффективное применение описанных в [86—88] подходов для решения рассматриваемых задач.

Важным направлением исследований в области извлечение знаний и интеллектуального анализа данных являются подходы, основанные на теории анализа формальных понятий (АФП) (англ. Formal Concept Analysis, FCA). Анализ формальных понятий является ветвью прикладной алгебраической теории решеток и нашел свое применение в различных областях современной науки о данных. Среди таких областей выделяются направление построения онтологий, поиск ассоциативных правил, машинное обучение, а также классификация, категоризация и другой интеллектуальный анализ текстов. Основы теории анализа формальных понятий подробно описаны в [89].

Фундаментальным в теории анализа понятий является определение *формального контекста*. Формальным контекстом называют тройку множеств $K = (G, M, I)$ такую, в которой G - множество объектов, M - множество признаков, а $I \subseteq G \times M$ - бинарное отношение между G и M. Выражение $(g, m) \in I$ (эквивалентная форма записи - gIm) означает, что объект g обладает признаком m. Например, можно рассмотреть контекст, в котором объектами являются научные публикации, признаками - ключевые слова, а бинарное отношение определяет существование данного ключевого слова в списке ключевых слов данной статьи. Контекст может быть представлен в виде таблицы, строками которой являются объекты, столбцами - признаки, а в ячейках проставляется единица, если данный объект обладает данным признаком, и ноль в ином случае.

Для произвольных $A \subseteq G$ и $B \subseteq M$ определены операторы Галуа:

$$A' = \{m \in M | \forall g \in A(gIm)\},$$

$$B' = \{g \in G | \forall m \in B(gIm)\}.$$

Другими словами, множество A' содержит в себе все признаки, общие для объектов множества A, а B' - все объекты, которые обладают признаками из B. Пара множеств (A, B) таких, что $A \subseteq G, B \subseteq M, A' = B, B' = A$, называется формальным понятием контекста K. Множество A называется объемом понятия, а B - содержанием понятия (A, B).

Для заданного контекста K вводится естественное отношение частичного порядка на формальных понятиях:

$$(A_1, B_1) \leqslant (A_2, B_2) \iff (A_1 \subseteq A_2 \iff B_2 \subseteq B_1).$$

Такой частичный порядок позволяет составить полную решетку на понятиях данного контекста. Для небольших выборок данных такую решетку представляется удобным визуализировать с помощью следующей диаграммы. Для примера рассмотрим изображенный в таблице 3 набор данных о ключевых статьях к научным публикациям. Определив все формальные понятия, можно построить диаграмму, представленную на рисунке 1.1. Читать эту диаграмму следует следующим образом. Метки понятий записаны в сокращенной нотации. Содержание понятия получается путем объединения содержаний всех понятий по всем путям от данной вершины вверх, а объем понятия определяется через объединение объемов всех понятий по всем путям от данной вершины вниз. Алгоритмы построения решеток и анализ вычислительной эффективности подробно описывается в [90].

-	Browsing	Mining	Software	Web Services	FCA	Information Retrieval
Paper 1	X	X	X		X	
Paper 2			X		X	X
Paper 3		X		X	X	
Paper 4	X		X		X	
Paper 5				X	X	X

Таблица 3: Пример набора данных о ключевых словах к публикациям

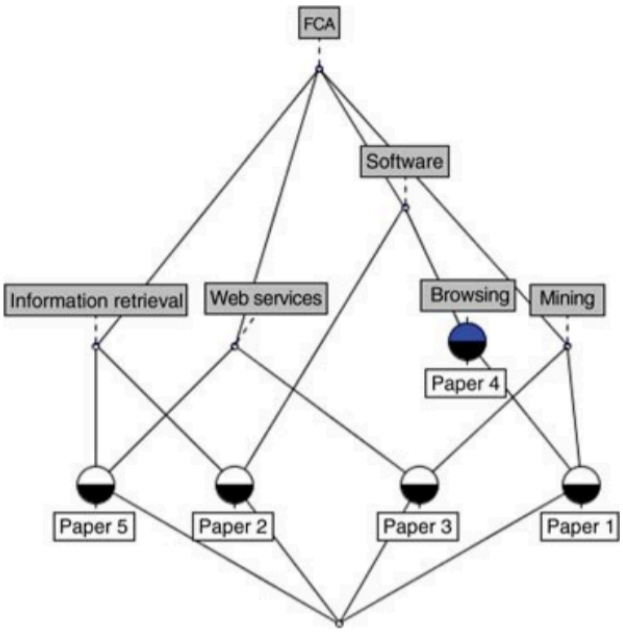


Рисунок 1.1 — Диаграмма набора данных о ключевых словах к публикациям

Методы анализа формальных понятий позволяет решать широкий спектр задач интеллектуального анализа. Одним из важных направлений исследований является разработка методов автоматического построения онтологий. Авторами

[91] разработана рекомендательная система, помогающая студентам систематизировать знания из определенных отраслей знаний. Система собирает, концептуализирует и классифицирует информацию по различным темам, используя данные из релевантных источников. Авторы [92] используют методы формального анализа понятий и на примере наукометрических данных строят онтологии в полуавтоматическом режиме.

В работе [93] авторами предложены методы автоматического построения таксономии по данным информационной системы определенной области знаний (авторы экспериментируют на корпусе данных туристической направленности). Перед построением решетки понятий авторы используют лексические парсеры естественного языка, чтобы определить глаголы (например, ride, drive, join) и образуют из них признаки (соответственно, rideable, driveable, joinable) для объектов.

Теория анализа формальных понятий находит свое применения в направлении исследования моделей машинного обучения. Для этого целевая переменная рассматривается в качестве отдельного признака. Объекты при этом разделяются на множества положительных, отрицательных и неопределенных примеров, после чего возникают соответсвующие этим множествам контексты. В работах [94; 95] авторами был представлен ряд известных моделей машинного обучения (среди которых, например, модель решающего дерева), описанных в терминах решеток понятий. Авторами [96] представлен алгоритм, получивший название IGLUE. Этот алгоритм комбинирует в себе идеи обучения на примерах с методами, основанными на построении решеток. Авторы [97] используют АФП для построение нейронной сети.

Многочисленные работы посвящены различным подходам к обобщению идей анализа формальных понятий. В работе [98] приводится обзор методов работы с нечеткими (fuzzy) признаками. Нечеткими называются небинарные признаки, т.е. признаки, принимающие многочисленные значения. Кроме того, существуют подходы, учитывающие не только описание объектов, но и различные связи между ними. Авторы [99] разработали программный модуль, который позволяет итеративно строить множество решеток понятий по набору данных взаимосвязанных объектов. Методы, исследованные авторами [100; 101], позволяют выявлять понятия из данных, имеющих графовое представление. Авторами [102] был предложен способ обобщения методов анализа формальных понятий

для графов знаний. Для этого вводятся n-арные формальные понятие, то есть формальные понятия, объемы которых представляются не множеством объектов, а множеством кортежей объектов.

Ряд работ посвящен задачам определения семантической близости между понятиями. В работах [103; 104] авторами были исследованы меры близости, основанные на взвешенном расстоянии Жакара на содержаниях и объемах. Такого рода определения связаны с пониманием сходства формальных понятий, определяемым как длина пути в диаграмме Хассе решетки понятий от одного понятия до другого.

Таким образом, анализ формальных понятий позволяет решать практические задачи интеллектуального анализа данных, близкие к рассматриваемым в настоящей диссертации. Однако, описанные выше методы во многом опираются на качественное признаковое описание объектов системы. В то время как объекты систем, используемых в настоящей работе, зачастую обладают лишь ассоциированным множеством ключевых слов. В этой связи для эффективного применения методов АФП потребуется анализ и разработка признаков более сложных, чем факт принадлежности данного ключевого слова данному объекту, что, в свою очередь, является нетривиальной задачей.

1.1.3 Графовые методы кластеризации слов естественного языка

Одной из прикладных задач настоящей диссертации является задача кластеризации ключевых слов больших и сложно организованных аналитических систем. По этой причине были проведены исследования существующих решений в данной области. Обзор основных методов и подходов к решению этой задачи представлен далее.

В работе [105] авторами предлагается графовый подход к кластеризации слов естественного языка. Для построения такого графа используется ресурсы поисковой системы. В процессе работы программной реализации выполняется множество запросов к поисковой системе и фиксируется число найденных документов по каждому запросу. Для пары слов w_a, w_b задается 3 запроса: один из них содержит оба этих слова, а второй и третий только слово w_a и только слово w_b, соответственно. Имея информацию о количестве документов, релевантных

каждому из этих запросов, появляется возможность подсчета меры близости PMI, которая уже описывалась ранее. Если значение меры выше порогового значения, то в графе слов, который строится авторами, между вершинами w_a, w_b будет проходить ребро.

Далее построенный граф кластеризуется алгоритмом, описанным в [106]. В ходе работы алгоритм оптимизирует следующий функционал:

$$Q = \sum_i \left(e_{ii} - \left(\sum_j e_{ij} \right)^2 \right),$$

где суммирование проводится по уже полученным кластерам, e_{ii} обозначает долю ребер в графе, соединяющих вершины i-го кластера, а e_{ij} - долю ребер, соединяющих между собой вершины кластеров i и j. Данный алгоритм является *агломеративным*. Это означает, что процесс кластеризации начинается с максимального числа кластера (каждая вершина - кластер), после чего кластера объединяются из условия минимизации выписанного выше функционала.

Другой метод, решающий вспомогательную оптимизационную задачу для решения задачи кластеризации, представлен в работе [107]. В этом подходе максимизируется функционал:

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j),$$

где δ - дельта функция, A_{ij} - вес ребра между вершинами i и j, $k_i = \sum_j A_{ij}$, $m = \frac{1}{2} \sum_{i,j} A_{ij}$.

Кроме описанных выше агломеративных подходов, существуют *дивизивные* или *дивизионные* методы кластеризации, которые в отличие от агломеративных строят новые кластера путем деления более крупных. Примером дивизивного метода кластеризации является [108]. В ходе работы алгоритма происходит случайное блуждание между вершинами, результатом которого является матрица вероятностей переходов от вершины к вершине. Кроме этого, авторами этой работы были добавлены эвристики, нормирующие нужным образом эту матрицу. По достижении сходимости происходит интерпретация результатов и граф делится на кластера.

Существует ряд дивизивных алгоритмов кластеризации, представленных в работе [109]. Подходы, используемые в работе основываются на построении *минимального остовного дерева* по графу, а именно дерева, полученного по графу, обладающего свойством минимальной суммы весов ребер.

Следует отметить, что в области графовой кластеризации слов естественного языка важную (а возможно даже решающую) роль играет именно метод построения графа. Различные подходы к построению представлены в работах [110—113]. После этого задача сводится к более общей задаче кластеризации графа. В ходе работы обычно не возникает необходимости в конструировании специфического алгоритма кластеризации. Напротив, гораздо более перспективной выглядит работа в области построения графа, наилучшим образом отражающего семантику стоящих у него в вершинах слов посредством определения ребер между этими вершинами. Когда такой граф построен, естественным выглядит использование одного из существующих и прошедших опробацию алгоритмов кластеризации. Правильный выбор необходимого алгоритма кластеризации также является подзадачей, которую необходимо решить.

Недостатки существующих решений

Если рассматривать информационно-аналитическую наукометрическую систему с точки зрения некоторого графа знаний, где сущностями являются, например, научные сотрудники, лаборатории, конференции и между сущностями возникают различные отношения, то существующие и традиционно используемые методы анализа имеют перечисленные далее недостатки.

– Такой граф знаний имеет объем, недостаточный для применения существующих латентных и признаковых графовых моделей, описанных в 1.1.2. Данные, используемые в этих работах оперируют графами на миллионы и десятки миллионов вершин и ребер.
– Для проведения аналитической работы нет необходимости все имеющиеся объекты использовать внутри одного графа. Например, такие сущности, как ключевые слова выглядит разумным вынести в отдельный граф, ребра которого будут иметь только одно отношение «семантической близости». В ином случае может возникнуть ситуация при которой

модель хуже обрабатывает более редкие понятия (лаборатории, кафедры, департаменты), потому что большая часть графа «забита» ключевыми словами и на них делается основной акцент в обучении.

Наличие описанных выше недостатков свидетельствует о необходимости создания методов и подходов, способных определять семантическую связь между объектами по небольшому объему входных данных и различными связями между ними.

1.1.4 Выводы из библиографического обзора

Подводя итог анализу существующих решений на рассматриваемом направлении, можно выделить тот класс информационно-аналитических систем, для которых такие решения не являются эффективными. Этот класс обладает следующими характеристиками и свойствами.

- **Малые объемы имеющейся информации**. Сложных модели информационно-аналитических систем обладают большим числом внутренних параметров, которые необходимо оценить по имеющимся данных. Если наблюдается дифицит данных, то эффективно настроить такие модели невозможно. В то же время простые и наивные модели не могут выявить в достаточной мере семантическую информацию из данных.
- **Сущности описываются небольшим объемом текста.** Для лучшего качества желателен именно набор ключевых слов.
- **Системы узкоспециальной направленности.** Для таких систем не подойдут предобученные модели на объемах данных общего назначения.
- **Отсутствие достаточных человекоресурсов для ручного сбора необходимой информации.** Обычно трудозатраты человека для разметки или подготовки данных - очень дорогой ресурс, которым не обладают владельцы небольших информационно-аналитических систем. Более того, зачастую такие люди должны обладать экспертными знаниями. Это факт уменьшает количество такого ресурса и значительно увеличивает его стоимость.

– **Существование дополнительных связей различной природы между сущностями системы.** Кроме того, эти связи могут быть достаточно разреженными.

Программный комплекс, разрабатываемый в рамках данной диссертационной работы, призван эффективно решать аналитические задачи в системах данного класса. Для этого к нему предъявляется ряд требований, описанных в приложении А, а также разрабатывается специальная методология исследования, подробно описанная далее.

1.2 Методология

В данном разделе описывается основные положения методологии построения решений задач, поставленных в настоящей диссертации. Методология строится исходя из недостатков существующих решений подобных задач, а также требований, предъявленных к создаваемому программному комплексу. Далее по пунктам излагаются важные в выборе общей методологии решения.

1. Во многих прикладных областях, связанных с анализом данных, появляется необходимость решения задачи определения семантически близких понятий. При этом объекты, подлежащие сравнению, не хранят в себе информации достаточно для качественного смыслового анализа. Самих объектов, при этом, может быть также не много.

   Для восстановления неявных семантических связей в имеющихся данных естественным выглядит использование **аппарата теории графов**.

   Рассмотрим, например, граф, вершинами которого являются объекты системы, а ребрами - некоторые семантические отношения между объектами. Некоторые из этих ребер могут отсутствовать ввиду недостаточности данных. Восстановить такую связь в некоторых случаях возможно, по связям между другими вершинами. В простейшем случае между парой вершин может существовать путь в графе, что может свидетельствовать о некоторой связи между соответствующими объектами.
2. Для решения рассматриваемых в настоящей диссертации задач используется анализ ключевых слов. Следует отметить, что обычно ключевые слова не имеют никаких ограничений в написании или использовании:

по факту, это могут быть произвольные тексты на естественном языке. Пользователи системы могут по-разному вписывать одно и то же слово, употреблять аббревиатуры, переводить термины на другие языки, транслитерировать, допускать опечатки, менять формы слов.

Использовав аналитические инструменты, можно сделать пространство ключевых слов менее разреженным: все различные написания одного слова, а также его синонимы связать в общую смысловую единицу и работать сразу с группой слов, а не с каждым словом отдельно. Кроме того, для анализа необходимо уметь определять, что пара ключевых слов является близкой по смыслу.

В условиях недостаточности исходных данных, дополнительная семантическая информация о ключевых словах может в значительной степени увеличить точность, а главное полноту моделей определения семантической близости понятий системы.

Все перечисленные выше доводы наталкивают на мысль, что перед началом решения основной задачи, **необходимо решить более низкоуровневую задачу определения семантической близости пары ключевых слов**. Таким образом, решение данной задачи становится **базовым** шагом для решения основных задач диссертации

3. Согласно идеям, изложенным в п.1, решение задачи определения семантической близости пары ключевых слов также может быть выполнено с применением **графовых алгоритмов**.

   Построив граф, в вершинах которого лежат ключевые слова, а ребра задают некоторые отношения, появляется возможность определять семантическую близость не только для тех пар, для которых в графе присутствует ребро, но и для произвольной пары ключевых слов.

   Для такой пары анализируются различные характеристики, такие как длина кратчайшего расстояния в графе, поток между парой вершин, меры центральности и другие. Другими словами, в условиях сильной ограниченности в объемах данных появляется возможность восстанавливать семантические связи между двумя вершинами по имеющимся данным о других вершинах. Важной задачей, возникающей при применении таких методов, является задача построения графов, наилучшим образом отображающим семантические отношения между словами.

4. Исходя из постановки основной задачи, и атрибуты системы, и ключевые слова их характеризующие, могут быть связаны дополнительными отношениями. Это обстоятельство предоставляет возможность построить несколько различных графов на одних и тех же наборах вершин. Например, по наукометрической коллекции наборов ключевых слов можно построить графы, в вершинах которых будут сами слова, а ребро будет ставиться в зависимости от выполнения различных условий, к числу которых относятся следующие.
   - Пара слов входит в один набор.
   - Пара слов использовалась одним пользователем во время ввода поискового запроса.
   - Пара слов, согласно некоторому классификатору, принадлежит общему классу. Классификатор уже разработан и внедрен в рассматриваемую систему. Например, таким классификатором может быть научный рубрикатор.

   Каждая такая идея порождает дополнительный граф, обладающий индивидуальным сигналом, который может помочь в определении семантической близости пары ключевых слов (пары вершин каждого из графов) Интеллектуальный анализ набора собранных графов позволяет выделить максимальный объем семантической информации из данных.
5. Определение методов семантической близости пары слов **является базовым средством определения близости** более общих понятий - **наборов ключевых слов**. Здесь принимается естественная гипотеза, что если пара наборов состоит из попарно похожих элементов, то и сами наборы похожи.

   Помимо использования низкоуровневых пословных моделей, имеется возможность построения нового графового представления данных. В этих графах вершинами являются наборы ключевых слов, а ребра указывают на отношения между наборами. Два таких подхода позволяют добиться улучшения качества определения близости наборов ключевых слов.
6. Как и ранее, за счет **дополнительных связей** в данных существует возможность создавать **различные вариации графов**, в вершинах которых

уже наборы ключевых слов, а ребра отражают некоторые различные отношения между наборами. Например, пара наборов может иметь связь по следующим причинам:

- оба набора содержат одно и то же ключевое слово;
- наборы являются наборами ключевых слов к различным публикациям одного автора;
- наборы являются наборами ключевых слов двух конференций, на которых выступал один автор.

7. Поскольку смысловая близость между парой понятий использует множество различных графовых характеристик, которые к тому же подсчитаны по различным графам, то получение искомой формулы близости вручную является очень трудоемкой задачей. Кроме того, необходимость решения такой задачи будет возникать каждый раз при обновлении данных или смене информационной системы.

   В этой связи для преодоления отмеченной трудности задействуется **аппарат машинного обучения**, который позволяет подбирать необходимую формулу в автоматическом режиме.

8. Следующим этапом является построение классификаторов и моделей, непосредственно решающие прикладные задачи, которые лежат в основе данного исследования. Благодаря разработанным на предыдущих шагах моделях представления информации и методам определения семантической близости, появляется возможность определения близости между объектами информационных систем. Кроме того, различные связи между объектами позволяют определить близость более точно.

Таким образом, для решения поставленных задач выбраны представленные далее положения, определяющие методологию.

1. Данные системы представляются в виде множества графов, вершинами которых являются некоторые понятия (ключевые слова/наборы слов/сущности системы), а ребрами - отношения между ними. По построенным графам подсчитываются различные характеристики для пар вершин.
2. Решается задача определения семантической близости пары ключевых слов. Для этого используются построенные графы, разработанные подходы и технологии машинного обучения. Решению этой задачи посвящена глава 2.

3. Решается задача определения семантической близости пары наборов ключевых слов. Разработанные модели используют различные графовые представления, подходы и модели, рассмотренные в предыдущих пунктах. Моделям, решающие указанную задачу описаны в главе 3.
4. Используя функцию близости наборов ключевых слов и отношения между сущностями системы, решаются прикладные задачи определения семантической близости пары сущностей. Этой задаче посвящается глава 4.

В дальнейших главах, согласно указанной выше методологии, детально и последовательно описываются разработанные в ходе работы модели, приводятся мотивация выбора того или иного подхода, проводятся тестовые испытания программных реализаций, делаются выводы и рассматриваются планы на будущее по улучшению отдельных компонент.

1.3 Экспертное оценивание качества результатов программных реализаций

Для проверки качества полученных результатов в ходе тестовых испытаний в рамках представленного в диссертации исследования зачастую необходима экспертная оценка. Основной причиной такой необходимости является тот факт, что исследования затрагивают сложную и неоднозначную области семантической отношений между объектами. Осложняет процесс оценивания также то обстоятельство, что значительная часть исследования проводилась на данных из определенной специфичной области знания - данных наукометрических систем. Эта область имеет сложную иерархическую структуру, каждое ответвление в которой является узкоспециальным направлением науки. Для таких направлений в открытых источниках не существует достаточного количества знаний об уровне семантической близости от компетентных в данной области людей.

По отмеченным выше причинам в исследованиях, результаты которых описаны в настоящей диссертации, используются специально полученные оценки от группы экспертов, состоящая из кандидатов и докторов физико-математических наук в количестве 8 человек. Экспертам было предложено оценить уровень сементической близости для различных задач, решаемых в рамках работы. Среди задач, качество решения которых оценивалось экспертами, выделяются следующие:

- определение уровня семантической близости между парой ключевых слов;
- определение уровня семантической близости между парой наборов ключевых слов;
- определение уровня семантической близости между парой объктов информационной системы;
- определение уровня абстрактности ключевого слова;
- релевантность экспертов, определенных поисковым модулем поиска эксперта, пользовательскому запросу;
- определение кластеров ключевых слов;
- определение тематических ключевых слов;
- качество построяния тезауруса ключевых слов.

Однако, важно отметить то, что для некоторых задач, которые решаются в рамках работы, могут быть частично или полностью протестированы в автоматическом режиме с использованием кажущихся разумными эвристиками, а также открытых наборов данных. Схема проведения таких тестовых испытаний и обоснования их адекватности будут расписана в подразделах «тестовые испытания» соответствующих разделов, посвященных обозначенным выше задачам.

Следующие главы детально описывают разработанную методологию, программную реализацию алгоритмов и соответствующие тестовые испытания. Для проведения значительной части таких испытаний были использованы силы экспертной группы, описанной в данном разделе.

Глава 2. Определение смысловой близости пары ключевых слов

В настоящей главе подробно описываются разработанные автором эвристические модели, алгоритмы и реализующее их программное обеспечение для определения семантической близости пары ключевых слов. Разработанные автором на основе проведенных исследований с использованием полученных в их ходе эвристических соображений модели определения семантической близости являются важным результатом деятельности в рамках подготовки настоящей диссертации.

Задача, решение которое представлено в данной главе, имеет следующую математическую постановку. Дано множество объектов системы D и множество ключевых слов W, множество наборов ключевых слов T. Каждый элемент $d_i \in D, i = 1 \dots N$ представлен *набором ключевых слов* $t_i \in T$, состоящим из k_i ключевых слов из множества W:

$$d_i \to t_i = (w_{i_1}, w_{i_2}, ..., w_{i_{k_i}}).$$

Множество T состоит из *уникальных наборов ключевых слов*. Наборы ключевых слов совпадают, если существует перестановка, отождествляющая эту пару наборов. Кроме того, между объектами системы заданы отношения различных типов, определенные направленным графом $G_{rel} = (D, R)$, в котором D - множество вершин (объекты системы), а R - ребра, определяющие отношения. Ребра этого графа помечены типом отношения: $l : R \to T$, где T - множество возможных типов. Множества (D, T, W, G_{rel}) задают модель информационной аналитической системы.

В рамках данной главы будет представлена модель определения семантической близости и в ее рамках такая функция близости $f : W \times W \to [0, 1]$, бóльшие значения которой означают сильный уровень смысловой близости между ключевыми словами.

Для определения семантической близости вводятся вспомогательные графы, вершинами которых являются ключевые слова, а ребра указывают на некоторые свойства пары ключевых слов. В самом простом виде таким свойством может являться факт причастности пары ключевых слов одному набору. Кроме этого существуют более сложные и неявные связи между парами ключевых слов. Примером такой связи может служить *контекстная близость*, определенная в главе

2.1.2. Согласно введенной модели, такая близость устанавливается между парой ключевых слов w_a и w_b в том случае, если существует большое число ключевых слов, которые входят одновременно и в наборы, содержащие слово w_a, и в наборы, содержащие слово w_b. Этот факт означает, что если для пары слов существует достаточное количество общих «соседей по набору», то между ними проявляется такая контекстная связь.

Следует отметить, что *контекстная близость* по результатам тестовых испытаний, приведенных в разделе 2.1.4, оказалась весьма точным способом приближения близости семантической. Этот факт означает, что данная мера близости может заменять разработанную в настоящей главе семантическую близость в ряде практических задач, в которых по каким-либо причинам нет возможности применеть более сложные моделей. Такими причинами могут являться ограниченность объема исходных данных, необходимость в использовании вычислительно более простой модели, отсутствие какой-либо информации, кроме коллекции наборов ключевых слов и другие.

На основе разработанной в данной главе модели контекстной близости *WordContSim*, автором был разработан алгоритм кластеризации множества ключевых слов информационной системы. Результаты по данному направлению описываются далее в разделе 4.1.

Еще одним способом получать связи между ключевыми словами является использование новых источников данных, поставляемых рассматриваемой информационно-аналитической системой. В рамках наукометрических систем такими источниками могут служить данные о конференциях, публикациях, проектах и других сущностях системы, если с ними ассоциируются ключевые слова. Связи такого рода не обязаны напрямую влиять на семантическую близость пары слов, для которой они проявляются. Например, пара ключевых слов, используемая одновременно двумя сотрудниками одного факультета в своих научных работах, не должна обладать семантической близостью. Тем не менее, такая информация при учете других факторов может нести дополнительный сигнал для модели определения семантической близости.

Большое количество различных характеристик, связей и источников данных различной природы затрудняет процесс определения близости. Эвристические подходы для выбора окончательных формул семантической близости являются менее эффективными и сложнореализуемыми. Причиной этому является необходимость вручную подбирать математическое выражения, учитывающие большое

число различных аспектов близости. Для преодоления этой трудности автором настоящей диссертации предлагаются подходы, основанные на применении методов машинного обучения. Подробное их описание представлено в разделе 2.2. По результатам исследований и тестирования программных реализаций, такие подходы в значительной мере улучшают качество определения смысловой близости между ключевыми словами.

С помощью известных алгоритмов теории графов, среди которых присутствуют поиски кратчайших путей, нахождения потока между вершинами, определения мер центральностей вершин и другие теоретико-графовые подходы, появляется возможность количественно охарактеризовать рассматриваемую пару вершин внутри построенных графов. Кроме этого, используется новая разработанная автором числовая характеристика меры абстрактности ключевого слова, описанная в 4.2.1. Эти числовые характеристики служат описанием (так называемым признаковым описанием объектов). Полученная информация используется для более точного определения семантической близости пары слов.

В связи с использованием методов машинного обучения с учителем для лучшего определения уровня смысловой близости, возникает необходимость сбора обучающей информации для подлежащих обучению моделей. В настоящей работе описывается новый алгоритм автоматического формирования обучающей выборки для проведения обучения. Важность такого алгоритма, описанного в разделе 2.2.1, заключается в том, что он избавляют от необходимости ручной разметки данных, которая обычно является ресурсозатратной.

В заключительной части главы представлены результаты и выводы относительно разработанных автором моделей определения семантической близости для пары ключевых слов.

2.1 Модель семантической близости *WordContSim*

В данном разделе описывается модель близости, названная *WordContSim*. Начало раздела посвящается введению вспомогательного графа ключевых слов, после чего приводится описание и аргументация эвристических идей, заложенных в данную модель. Следующим этапом изложения является представление

реализающих данную модель алгоритмов, асимптотических оценок вычислительной сложности и затрат по памяти для их программных реализаций.

2.1.1 Построение графа ключевых слов

В основе модели вычисления семантической близости *WordContSim* лежит построение специального взвешенного *графа ключевых слов* $G_{kw}(W, E_{mut})$, вершинами W которого являются ключевые слова системы, а ребра E_{mut} - отражают факт вхождения пары ключевых слов в один набор. Вершину, соответствующую ключевому слов w обозначим за v_w. Вес ребра отражает в простейшем виде смысловую связь между соответствующими ключевыми словами и задается формулой:

$$\omega(v_{w_i}, v_{w_j}) = \sum_{\{t \in T : w_i \in t, w_j \in t\}} \frac{1}{|t|}, \tag{2.1}$$

где $|\cdot|$ обозначает количество ключевых слов набора.

Выбор формулы, определяющей вес ребра, объясняется следующими двумя соображениями.

1. Чем чаще пара слов встречается в одном наборе, тем более вероятна их семантическая связь и тем больший вес ставится соответствующему ребру.
2. Чем больше слов содержит набор, тем менее вероятно, что два рассматриваемых слова семантически связаны непосредственно друг с другом, а не с другими словами набора. Поэтому набор, состоящий из большего числа слов, вносит меньший вклад в вес ребра.

Кроме этого, помечаются вершины описанного графа. Каждой вершине приписывается количество упоминаний в коллекции ассоциированного с этой вершиной ключевого слова:

$$\upsilon(v_w) = |\{t : w \in t, t \in T\}|$$

Характеристика совместной встречаемости внутри одного набора является важной для определения семантической близости, но не определяющей: слова одного набора не обязаны быть похожими друг на друга по смыслу. Напротив, нередко они служат для того, чтобы более точно описать общую тему документа,

к которому относятся, а добавление точного синонима не добавляет информации о тематике документа.

Использование формулы 2.1 в качестве меры семантической близости протестировано вместе с другими моделями вычислений. Результаты тестирования программной реализации модели, основанной на данной формуле, представлены в 2.1.4. Качество такой модели заметно уступает более сложным моделям, существенно использующим семантику. Несмотря на этот факт, введенный выше *граф ключевых слов* является необходимой структурой данных для определения контекстной меры близости, описание которой следует разделе 2.1.2. Здесь и далее граф ключевых слов будет обозначен через G_{kw}

2.1.2 Контекстная модель определения семантической близости для пары ключевых слов

Для введения понятия *контекстной близости* рассмотрим подграфы введенного ранее графа ключевых слов, изображенных далее.

Проанализиров изображение, можно отметить следующий факт: пара вершин, ассоциированных со словами «многочлен» и «полином» не входят ни в один набор ключевых слов коллекции (поскольку в графе отсутствует ребро между этими вершинами), но имеет при этом большое число общих соседей. Поскольку эти два слова действительно семантически близки (являются синонимами), то это наводит на идею использования такой характеристики для пары вершин в качестве одной из компонент в формуле, позволяющей вычислять меру семантической близости.

Другими словами, отмечается следующее наблюдение: уровень семантической близости слов w_x и w_y тем больше, чем больше у вершин v_{w_x}, v_{w_y} общих соседей в *графе ключевых слов*. Соответствующее множество слов $\mathcal{C}_{T;w_x,w_y}$ для этих вершин записывается в следующем виде: $\mathcal{C}_{T;w_x,w_y} = \{w|(w,w_x) \subset t_i, (w,w_y) \subset t_j, (t_i,t_j) \in T, i \neq j\}$ и обозначается *контекстным множеством для ключевых слов* w_x *и* w_y. Название «контекстное» выбрано по той причине, что такое множество определяет в каком общем контексте употребляются два ключевых слова в рамках одной коллекции. Чем больше мощность этого множества, тем более вероятна семантическая связь между словами w_x и w_y.

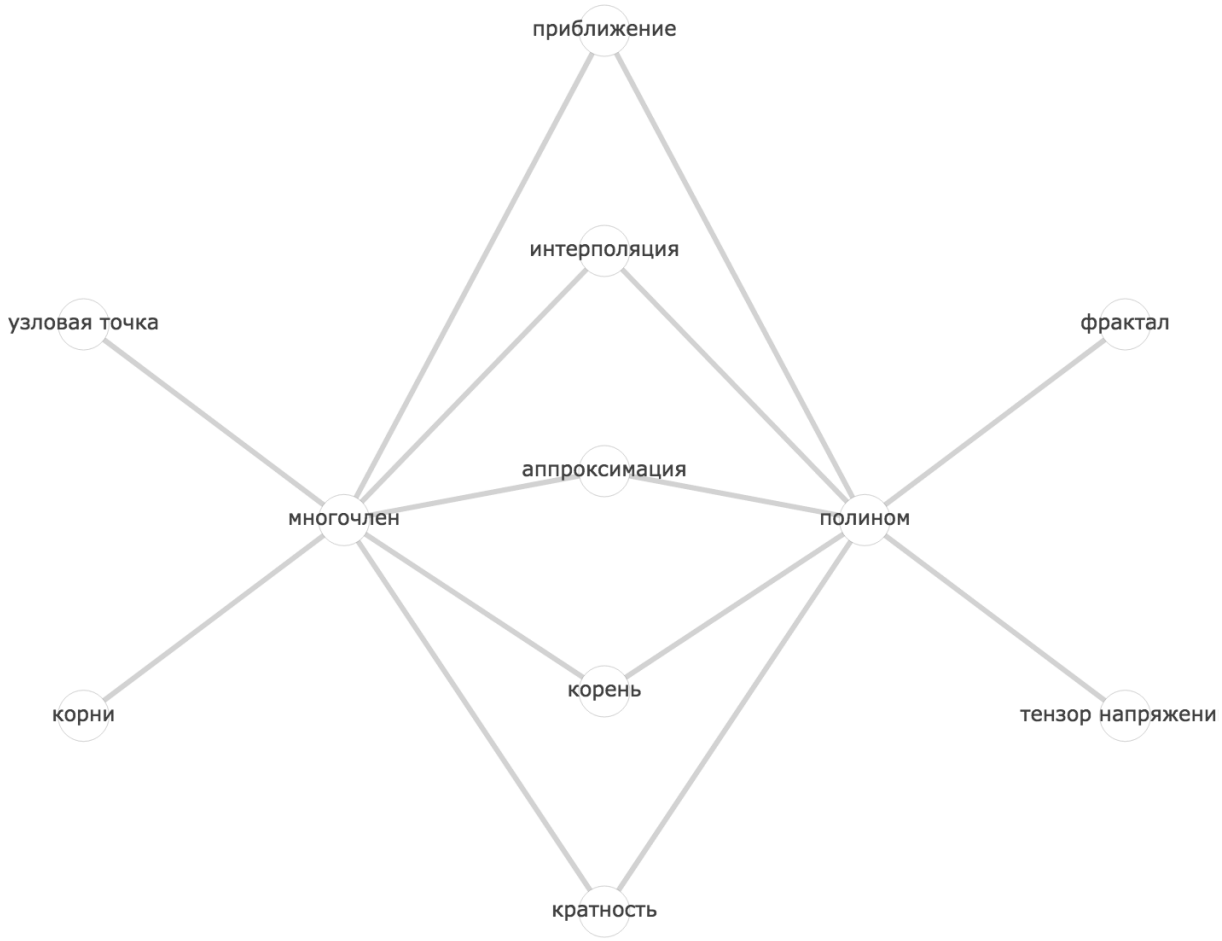


Рисунок 2.1 — Соседи вершин «полином» и «многочлен» в графе ключевых слов

После того, как контекстное множество определено, необходимо уметь получать по этому множеству численную характеристику близости. Для этих целей в рамках принятых ранее обозначений определена следующая формула:

$$\hat{C}_T(w_i, w_j) = \sum_{w_i \in \mathcal{C}_{T;w_x,w_y}} \min(\omega(v_{w_x}, v_{w_i}), \omega(v_{w_i}, v_{w_y})). \tag{2.2}$$

Согласно этой формуле, больший вклад в итоговый уровень близости дают те вершины графа ключевых слов, которые сильнее связаны с оцениваемой парой ключевых слов с точки зрения веса соответствующих ребер.

Следующим шагом в построении контекстной семантической модели близости является нормировка величины $\hat{C}_T(w_i, w_j)$. Дело в том, что формула 2.2 никак не учитывает частотности встречаемости оцениваемых слов в коллекции. Это обстоятельство приводит к тому, что значение, вычисленное по этой формуле, может оказаться большим, если рассматриваемые слова являются достаточно

часто встречающимися (далее для краткости изложение - частотными) в рамках коллекции. Причина этому в том, что частотные слова имеют больше различных контекстов употребления и эти контексты с большей вероятностью будут сильнее пересекаться. При этом очевидно, что на семантическую близость такое свойство формулы влиять не должно. Поэтому уровень семантической близости должен быть обратно зависим от частотностей встречаемости рассматриваемых ключевых слов.

Для пары ключевых слов w_i, w_j *контекстная близость* определяется по формулам:

$$C_T(w_i, w_j) = \frac{\hat{C}_T(w_i, w_j)}{f_T(w_i) + f_T(w_j)}, \tag{2.3}$$

где $f_T(w_i)$ - частота встречаемости слова w_i в коллекции T.

Формула контекстной близости, введенная выше, показывает высокий уровень качества определения семантически близких пар ключевых слов. Кроме того, для оптимизации вычислений по формуле контекстной близости разработан алгоритм, эффективно вычисляющий значения для всех пар рассматриваемой коллекции. Описание этого алгоритма представлно далее. Вычислительная эффективность и высокий уровень качества получаемых результатов подтверждаются тестовыми испытаниями программной реализации модели, которые описаны в 2.1.4.

2.1.3 Алгоритм вычисления значения контекстной близости по коллекции ключевых слов

Кроме того обстоятельства, что вычисление контекстной близости имеет самостоятельный интерес, эта мера близости используется также для решения ряда других задач, рассматриваемы в рамках настоящей диссертации. Во-первых, формула контекстной близости необходима при расчете более сложной модели семантической близости, использующей методы машинного обучения (раздел 2.2). Во-вторых, она используется для решения задачи кластеризации ключевых слов (раздел 4.1).

В этой связи важным является умение эффективно предрассчитать (рассчитать в упреждающем режиме) значения меры близости для большого числа пар (например, для всех пар ключевых слов коллекции). Для этих целей автором разработан алгоритм, вычисляющий значения по формуле контекстной близости для пар рассматриваемой коллекции ключевых слов. Автором доказываются оценки на вычислительную сложность этого алгоритма и необходимые ресурсозатраты для его программной реализации.

Первым необходимым шагом для вычисления всех значений контекстной близости по коллекции T является построение *графа ключевых слов* G_{kw}, введенного ранее. Пусть граф ключевых слов хранится в виде массива хеш-таблиц. Индексы массива соответствуют номерам вершин, а хеш-таблица содержит отображение соседей задаваемой индексом вершины в значение веса соответствующего ребра. Тогда следующее утверждение дает оценку вычислительной сложности построения графа ключевых слов.

Утверждение 1. Построение графа ключевых слов G_{kw} по коллекции ключевых слов T амортизационно требует $O(|T|p^2)$ элементарных операций, где p - максимальное число ключевых слов в одном наборе коллекции, $|T|$ - количество наборов в коллекции T.

Под амортизационным анализом здесь и далее понимается анализ средней производительности в худшем случае при усреднении по всем проводимым операциям.

Доказательство. В вершинах графа будем хранить не ключевые слова, а их уникальные числовые идентификаторы. Хранение графа организовано с помощью отображения идентификатора ключевого слова x_{id} в другое отображение. Это внутреннее отображение, в свою очередь, ставит идентификаторам вершин соседей вершины x_{id} в соответствии веса образованных ребер. Для этих целей наиболее подходящим является использование словаря, ключи которого являются целыми числами, а значениями - другой словарь. Ключами второго словаря также являются целые числа, а значениями - действительные числа. Внутренней структурой в реализации словаря и множества является хеш-таблица, что позволяет добиться амортизационно константного времени работы ($O(1)$) основных операций с этими структурами данных. Обозначим словарь верхнего уровня за g. Для хранения отображения ключевого слова в его уникальный идентификатор используется дополнительный словарь (хеш-таблица). Обозначим его за w.

Кроме этого необходим дополнительный массив-счетчик c для хранения количества упоминаний каждого ключевого слова в коллекции (по определению, в вершинах графа ключевых слов хранится эта характеристика). В ячейке с индексом i в таком массиве будем хранить количество упоминаний в коллекции T ключевого слова с идентификатором i. Идентификаторами служат целые числа от 0 до $n - 1$, где n - число уникальных ключевых слов коллекции.

Для заполнения структур w и c необходимо один раз рассмотреть каждое слово каждого набора коллекции. Если текущее слово уже имеет идентификатор (то есть существует запись в w), то нужно инкрементировать соответствующий счетчик в массиве-счетчике c. В ином случае нужно сначала добавить слово в словарь w, присвоив ему новый уникальный номер (номера считаются от нуля и каждый раз увеличиваются на единицу), а после этого добавить в массив c новое значение, равное единице (что соответствует первому появлению слова в коллекции).

Добавление в динамический массив и в хеш-таблицу, а также поиск в хеш-таблице имеют амортизационную сложность $O(1)$. Доступ к элементу массива и инкрементация его значения имеют сложность $O(1)$. Эти операции проводятся для каждого слова каждого набора, которая в худшем случае оценивается, как $|T|p$ слов. Поэтому построение этих структур амортизационно имеет сложность $O(|T|p)$.

Для построения графа необходимо рассмотреть все наборы коллекции и в каждом наборе рассмотреть все комбинации ключевых слов. Для каждой такой пары ключевых слов (x, y) необходимо найти их идентификаторы (x_{id}, y_{id}), а затем найти в словаре w текущее значение веса ребра $w[x_{id}][y_{id}]$. Это значение необходимо инкрементировать на величину $\frac{1}{n_t}$, где n_t - число ключевых слов в данном наборе. Если изначально такое значение в w отсутствовало, то считать его равным нулю. Сложность всех этих операций амортизационно равна $O(1)$. Количество пар ключевых слов в худшем случае достигает $\frac{|T|(p-1)p}{2}$, если в каждом наборе оказалось ровно p ключевых слов. Таким образом, сложность построения самой структуры графа амортизационно достигает $O(|T|p^2)$.

Итак, общая вычислительная сложность амортизационно равна $O(|T|p^2 + |T|p) = O(|T|p^2)$. Утверждение доказано.

Наивный подход, который рассчитывает меру близости для всех вершин графа ключевых слов заключается в переборе всех пар ключевых слов и вычислении общих соседей для каждой такой пары. По построенному пересечению и

информации о частотностях слов вычисляется мера контекстной близости. Под частотностями, как и ранее, следует понимать количество упоминаний слова в коллекции T. Более подробно наивный алгоритм представлен далее.

1. Построение графа ключевых слов G_{kw}.
2. Инициализация разреженной матрицы C размером $n * n$.
3. Цикл по всем парам вершин (v_{w_i}, v_{w_j}) графа G_{kw}:
    а) определение множества соседей для вершины v_{w_i},
    б) определение множества соседей для вершины v_{w_j},
    в) определение пересечения множеств соседей $\mathcal{C}_{T;w_x,w_y}$,
    г) вычисление $\hat{C}_T(w_i, w_j)$ по формуле 2.2,
    д) вычисление $C_T(w_i, w_j)$,
    е) сохранение значения в соответствующей ячейке разреженной матрицы C.
4. Матрица C - возвращаемое алгоритмом значение.

Следующее утверждение доказывает оценку вычислительной сложности наивного алгоритма вычисления контекстной близости для пар ключевых слов коллекции.

Утверждение 2. При построенном графе ключевых слов расчет весов всех пар ключевых слов наивным алгоритмом $C_T(w_i, w_j)$ имеет амортизированную сложность $O(n^2m)$, где n - число вершин графа ключевых слов, m - максимальное число ребер у вершины в графе ключевых слов.

Доказательство.

Рассмотрим пару вершин графа. Поскольку сложность операции доступа к элементу хеш-таблицы в худшем случае равно $O(1)$, то сложность получения списков соседей в графе для каждой из этих вершин также амортизированно равна $O(1)$. Пересечение двух множеств занимает амортизированно $O(l)$, где l - число элементов в меньшем из двух списков. В худшем случае сложность операции взятия пересечения амортизированно равна $O(m)$, если каждая вершина имеет m соседей. Вычисление значений формул $C_T(w_i, w_j)$ также занимает $O(m)$. Поскольку число пар вершин квадратично по числу вершин и для каждой пары необходимо произвести $O(m)$ операций, то вычислительные затраты на подсчет значений оказываются равными $O(n^2m)$ Этот факт завершает доказательство.

Описанный выше алгоритм вычисления контекстной близости недостаточно производителен и не способен провести необходимые вычисления в графах,

состоящих из десятков тысяч вершин и более. При этом следует отметить, что даже в небольших аналитико-информационных системах количество уникальных ключевых слов легко может достигать сотен тысяч. Очень большие системы содержат порядка миллиона уникальных ключевых слов.

Представленные доводы свидетельствуют о том, что необходим более совершенный алгоритм вычисления контекстной близости пар ключевых слов коллекции. Автором был разработан такой алгоритм. Его основная идея состоит в том, чтобы проходить по всем вершинам графа и для каждой рассматривать всевозможные комбинации пар соседей. Такой подход позволяет снизить асимптотическую сложность алгоритма до $O(nm^2)$. Более детальное изложение данного алгоритма представлено далее.

1. Построение графа ключевых слов G_{kw}.
2. Инициализация *нулями* разреженной матрицы C размером $n * n$.
3. Для каждой вершины v_{w_i} графа ключевых слов G_{kw} выполняется:

   Для каждой пары соседей (v_{w_j}, v_{w_k}) вершины v_{w_i} выполняется:

   Если пара (v_{w_j}, v_{w_k}) не соединена ребром в графе G_{kw}, то вычисляется:

$$C[j,k] \mathrel{+}= \min(\omega(v_{w_i}, v_{w_j}), \omega(v_{w_k}, v_{w_i})),$$

   где $\omega(v_{w_a}, v_{w_b})$ - вес ребра между вершинами v_{w_a}, v_{w_b}.
4. Для каждой ненулевой ячейки (i,j) матрицы C выполняется:

$$C[i,j] \mathrel{/}= f_T(w_i) + f_T(w_j).$$

5. Матрица C - возвращаемое алгоритмом значение.

Утверждение 3. При построенном графе ключевых слов расчет значений матрицы C имеет вычислительную сложность $O(nm^2)$, где n - число вершин графа ключевых слов, m - максимальное число ребер у вершины в графе ключевых слов.

Доказательство. Вычислим сложность подсчета формул $C[i,j]$. Для вычисления требуется рассмотреть все n вершин графа ключевых слов и для каждой вершины рассмотреть все пары ее соседей, не смежных друг с другом. Поскольку у вершины не более, чем m соседей, то обработка одной вершины занимает $O(m^2)$, а обработка всех вершин, соответственно, $O(nm^2)$. Таким образом, и общее время работы алгоритма составляет $O(nm^2)$. Что и требовалось доказать.

В целях большей минимизации времени рассчетов на построение матрицы C, разумным является ограничение числа рассматриваемых соседей для текущей вершины графа. Данная оптимизация приводит к следующему алгоритму.

1. Пусть k - входной параметр алгоритма.
2. Построение графа ключевых слов G_{kw}.
3. Инициализация *нулями* разреженной матрицы C размером $n * n$.
4. Для каждой вершины v_{w_i} графа ключевых слов G_{kw} выполняется:
   а) Множество соседей соседей $N(v_{w_i}) = (v|\omega(v, v_{w_i}) > 0)$ сортируется по убыванию весов $\omega(v, v_{w_i})$.
   б) Из упорядоченное множества $N(v_{w_i})$ выбираются первые k вершин. Полученное подмножество обозначается $N_k(v_{w_i})$.
   в) Для каждой пары соседей (v_{w_j}, v_{w_k}) вершины v_{w_i} **из множества** $N_k(v_{w_i})$ выполняется:

      Если пара (v_{w_j}, v_{w_k}) не соединена ребром в графе G_{kw}, то вычисляется:

$$C[j, k] \mathrel{+}= \min(\omega(v_{w_i}, v_{w_j}), \omega(v_{w_k}, v_{w_i})),$$

      где $\omega(v_{w_a}, v_{w_b})$ - вес ребра между вершинами v_{w_a}, v_{w_b}.
5. Для каждой ненулевой ячейки (i, j) матрицы C выполняется:

$$C[i, j] \mathrel{/}= f_T(w_i) + f_T(w_j).$$

6. Матрица C - возвращаемое алгоритмом значение.

Значение параметра k выбирается исходя из компромисса между скоростью выполнения и точностью алгоритма. Чем меньше k, тем меньше пар соседей для каждой вершины будет обработано, но тем меньше модель сможет использовать информации из входной коллекции и тем ниже ожидаемое качество работы программной реализации алгоритма.

Следует однако отметить, что слишком большое число рассматриваемых пар может также повлечь ухудшение качества модели. Это происходит по той причине, что начинают использоваться *ненадежные ребра*. Под такими ребрами следует понимать ребра с настолько низким значением веса, что факт связи между соотвествующими ключевыми словами с высокой долей вероятности случаен. Поэтому необходимо настраивать параметр k для достижения лучшего результата.

Вычисление матрицы C с описанным выше эвристическим соображением имеет вычислительную сложность $O(nk^2 + nm\log(m))$, о чем свидетельствует следующее утверждение.

Утверждение 4. При построенном графе ключевых слов расчет значений матрицы C для случая ограниченного числа рассматриваемых соседей для текущей вершины имеет сложность $O(nk^2 + nm\log(m))$, где n - число вершин графа ключевых слов, k - количество рассматриваемых соседей, m - максимальное число соседей у вершины.

Доказательство. Суть доказательсва аналогична представленному в доказательстве утверждения 3, за исключением того, что теперь для текущей вершины будет рассмотрено порядка $O(k^2)$ пар соседей. Это делает оценку сложности вычисления значений матрицы C равной $O(nk^2)$. Кроме того, появляются дополнительные затраты на сортировку соседей каждой из n вершин графа. Каждая такая сортировка занимает $O(m\log(m))$ времени в худшем случае. Таким образом, общая сложность алгоритма - $O(nk^2 + nm\log(m))$. Что и требовалось доказать.

При $k = \sqrt{m}$, например, достигается оценка $O(nm\log(m))$ времени работы, что является значительным ускорением работы алгоритма.

Следует также отметить, что отношение контекстной близости коммутативно, поэтому хранить достаточно только верхний правый угол матрицы $C(p,q)$.

Таким образом, разработанный алгоритм показывает конкурентноспособное время работы в сравнении с известными моделями определения семантической близости, выигрывая при этом у них в качестве классификации семантически близких пар. Этот факт подтверждается тестовыми испытаниями, которые представлены в следующем далее разделе 2.1.4.

2.1.4 Тестовые испытания

Опишем сценарий проведения тестовых испытаний программных реализаций разработанных автором моделей. Для проведения процедуры тестирования моделей необходимы примеры семантически близких пар ключевых слов. Сбор

качественного тестового множества требует больших ресурсозатрат для получения достаточного количества экспертных оценок. В качестве замены тестирования на экспертных оценках была разработана следующая методика тестирования в автоматическом режиме.

Выделяется случайное множество слов W_{test}, на котором впоследствии будут протестированы модели. Далее один за другим просматриваются каждое ключевое слово w каждого набора t исходной коллекции T. Если текущее слово является тестовым, то есть $w \in W_{test}$, то с вероятностью p это слово заменяется на синтетически созданное слово w^*, отличающееся от w наличием спецсимвола на конце, что делает его уникальным в рамках коллекции T. Таким образом, коллекция T преобразуется в коллекцию T^*. Эта коллекция содержит для слов из W_{test} соответствующие парные синтетические слова.

Важным является то обстоятельство, что вследствие такой процедуры семантические свойства синтезированных слов не были изменены. Это позволяет утверждать, что пары слов (w, w^*) являются семантически близкими и, более точно, семантически идентичными - то есть представляют собой по сути одно и то же слово. Следовательно, качественная модель определения семантической близости должна таким парам сопоставлять сильный уровень близости.

Следует однако заметить, что «сильный» уровень близости является относительной величиной и изменяется в зависимости от используемой модели и ее параметров. По этой причине представляется более правильным проверять уровень семантической близости в сравнении с некоторыми случайными словами из коллекции T^* и ожидать, что предсказания эффективной модели на тестовых примерах будут выше, чем на случайных. Таким образом, чем выше модель ранжирует искусственно созданные тестовые пары, тем более качественной можно считать эту модель. Описанное случайное множество обозначается за W_{cand}.

Полный алгоритм подготовки данных для тестирования изложен далее:

1. входные параметры алгоритма:
   - исходная коллекция ключевых слов T;
   - предполагаемое количество примеров семантически идентичных синтезированных пар - $N_{positive}$;
   - вероятность модификации слова p;
   - количество случайных пар-кандидатов, для сравнения их с тестовыми парами - $N_{negative}$.
2. для слов w коллекции T подсчитываются частоты встречаемости $f_T(w)$;

3. с учетом частотностей слов случайным образом выделяется множество из $N_{positive}$ ключевых слов из T. Учет частотностей означает, что каждое слово w может быть взять в данную выборку с вероятностью $\frac{f_T(w)}{\sum_{w_i \in W} f_T(w_i)}$. Таким образом фиксируется множество W_{test};
4. формируется модифицированная коллекция T^*:
   - цикл по всем наборам $t \in T$;
     - цикл по всем ключевым словам $w \in t$ набора:
       * если $w \in W_{test}$, то с вероятностью p произвести замену $w \to w^*$;
       * иначе оставить исходную версию слова.
5. формируется тестовые пары семантически идентичных пар ключевых слов:
   - цикл по всем уникальным словам w множества ключевых слов W_{test}:
     - если для w в коллекции T^* присутствует парное слово w^*, то пара (w, w^*) добавляется в множество тестовых пар.
6. с учетом вероятностей выбрать случайное множество W_{cand} из $N_{negative}$ слов-кандидатов коллекции T^* для сравнения с тестовыми парами;
7. возвратить модифицированную коллекцию ключевых слов, множество тестовых пар и множество кандидатов.

Заметим, что такой алгоритм набирает число тестовых пар чуть меньшее, чем $N_{positive}$. Это происходит по той причине, что при случайной замене слова $w \to w^*$ могло произойти так, что во всех упоминаниях в коллекции слово либо было изменено, либо, наоборот, не было изменено ни в одном. Вследствие этого обстоятельства для некоторых слов из W_{test} могло не оказаться парного. Тем не менее, это не является существенным вопросом, потому что таким образом теряется $3 - 5\%$ от размера тестового множества. Для решения достаточно увеличить значение параметра $N_{positive}$ и получить примерно столько же тестовых пар, сколько планировалось изначально.

Процесс тестирования происходит следующим образом:

- выбирается очередная пара (w, w^*) тестового множества;
- для каждой модели, подлежащей тестированию, выполняется:
  - модель предсказывает значения семантической близости для всех пар w, w_c, где $w_c \in W_{cand}$, а также для пары w, w^*.

- вычисляется позиция, на которую данная модель отранжировала истинную пару (w, w^*);
- по позиции вычисляются метрики качества данной модели на данном тестовом примере.

– значения метрик усредняются по всем тестовым примерам.

При проведении экспериментов были использованы следующие метрики качества:

– **Полнота@N**. Для подсчета метрики необходимо отсортировать пары по уменьшению уровня близости и выбрать первые N пар. Далее подсчитывается доля настоящих семантически близких пар, попавших в первые N наиболее релевантных;

– **DCG**. В рамках данного эксперимента используется формула $\sum_{i=1}^{N_{negative}+N_{positive}} \frac{rel_i}{\log_2(i+1)}$. Значение переменной rel_i равно 1 только в случае истинной пары (w, w^*), в остальных случаях $rel_i = 0$. Таким образом, значение метрики будет выше, если семантически близкая пара попала на первые позиции. Данная метрика более чувствительна к перестановкам на первых позициях. Это означает, что если позиция релевантной пары поднялась с 10000 на 9999 позицию, то это в меньшей степени увеличит значение метрики, чем в случае, когда позиция изменилась с 10 на 9.

Модель, разработанная в рамках диссертационных исследований, была протестирована и проанализирована в сравнении с современными эффективными моделями в области семантической близости, описании которых представлено далее.

– **Word2Vec [50]**. Обучение модели *Word2Vec* проводилось на коллекции ключевых слов T^*. Каждому ключевого слову $w \in T^*$ модель ставит в соответствии вектор, определенной размерности. Эффективность таких представлений заключается в том, что векторы, близкие по косинусному расстоянию, принадлежат семантически близким словам;

– **SPMI[54]**. Предобработка представляет собой двухэтапный процесс. На первом этапе для всех пар ключевых слов вычисляется мера близости PMI ([15]). В результате этого действия образуется матрица размерности $|T^*| \times |T^*|$, в ячейках которой лежат значения меры PMI для соответствующих слов. На втором этапе происходит понижение размерности данной матрицы до размерности $|T^*| \times k$ с помощью процедуры сингулярного

разложения (Singular Value Decomposition, SVD, [55]). Параметр k подбирается таким образом, чтобы качество модели на отложенной части тестовой выборки было максимальным. В итоге, как и в случае с моделью $Word2Vec$, получено представление слов в векторном пространстве. В данном случае представлением для i-го ключевого слова коллекции является i-ая строка полученной матрицы. Семантическая близость между словами определяется через косинусное расстояние.

Параметры моделей подбирались по отложенной части тестовой выборки.

Тестирование проводилось на коллекции из 329037 наборов ключевых слов. Учитывая тот факт, что каждый набор состоит лишь из нескольких слов, то такой набор данных является набором небольшого размера. В некоторых современных системах число сущностей, для которых известны наборы ключевых слов, могут достигать десятков и сотен миллионов наборов. Кроме того, в подобных системах обычно присутствуют дополнительные текстовые источники данных, например, полнотекстовые документы.

Параметры, используемые для подготовки тестового набора данных:

– количество наборов в коллекции - 329037;
– $N_{positive} = 2500$;
– $p = 0.5$;
– $N_{negative} = 100000$;

Другими словами, необходимо определить для каждого из 2500 тестовых слов нужное парное слово, выбирая его из 100000 возможных слов-кандидатов.

Результаты тестовых испытаний представлены в следующей далее таблице 5:

Модель	DCG	Полнота@1	Полнота@10	Полнота@30	Время предобработки, с	Время подсчета, с
TagGraph	0.0912	0.0000	0.0026	0.0056	17	664
SPMI	0.4176	0.000	0.3920	0.4591	147	763
Word2Vec	0.4436	0.1664	0.2986	0.3624	46	7510
WordContSim	**0.9175**	**0.5514**	**0.6551**	**0.6883**	1509	2091

Таблица 5: Результаты тестирования модели WordContSim

В дополнение к представленным выше, такие же тестовые испытания были проведены для *подвыборки* данной коллекции, состоящей лишь из 50000 наборов ключевых слов. Такой эксперимент был необходим для подтверждения способности разработанных программных реализаций моделей эффективно действовать в

условиях сильно ограниченного количества исходных данных. Отметим, что такой подход отвечает требованию к эффективности, указанному в приложении A.

Параметры, которые были использованы для построения уменьшенного набора тестовых данных:

– количество наборов в коллекции - 50000;
– $N_{positive} = 500$;
– $p = 0.5$;
– $N_{negative} = 100000$.

Результаты тестирования моделей на уменьшенном объеме данных представлены в следующей далее таблице 7:

Модель	DCG	Полнота@1	Полнота@10	Полнота@30	Время предобработки (сек)	Время подсчета (сек)
TagGraph	0.0895	0.0000	0.0040	0.0040	2	108
Word2Vec	0.1743	0.0325	0.0630	0.0752	7	485
SPMI	0.2419	0.000	0.1504	0.2337	19	113
WordContSim	**0.9055**	**0.5548**	**0.6280**	**0.6686**	23	304

Таблица 7: Результаты тестирования модели WordContSim на уменьшенном объеме данных

В результате тестовых испытаний программных реализаций моделей были получены значения метрик, подтверждающие эффективность разработанной модели *WordContSim*. С помощью этой модели удалось восстановить правильную пару для тестовых слов в более, чем 50% случаев. Другими словами, среди 100000 кандидатов для заданного слова, истинный находится на первой позиции по мере семантической близости, используемой в *WordContSim*. Такого результата не удалось добиться используя существующие эффективные решения в области определения семантической близости слов естественного языка: реализации других моделей серьезно отстают по имеющимся метрикам качества.

Кроме того, результаты работы реализации *WordContSim* практически не ухудшились при значительном уменьшении объемов исходных данных до размеров небольшой коллекции. Показатели качества других моделей, подвергшихся тестированию, резко понизились до уровня, который неприемлем для реальных аналитических информационных систем.

В дополнение к представленным выше выводам следует отметить, что модель $Word2Vec$, несмотря на свою эффективность в задачах определения семантической близости, в большей степени нуждается в достаточных объемах данных для обучения, чем модель $SPMI$.

Применение программных реализаций моделей к реальным коллекциям ключевых слов. В заключающей части настоящего подраздела описаны результаты тестовых испытаний программных реализаций на реальных коллекциях ключевых слов. В качестве тестовых данных были использованы корпусы ключевых слов для научных публикаций, собранных из сети Интернет. Использовалась также информация из социальной сети Вконтакте, а именно - были выкачаны посты (публичные сообщения из групп и страниц пользователей), часть из которых помечены хеш-тегами. В рамках проведенных тестовых испытаний хеш-теги отождествляются с обычными ключевыми словами.

В процессе сбора данных проводился парсинг текстовых данных на предмет наличия в них наборов ключевых слов. Точное решение этой задачи не является предметом исследования данной работы, поэтому для парсинга данных были использованы наивные подходы, которые, тем не менее, позволяют собрать корпус достаточного размера и качества для проведения дальнейшего анализа. В итоге собрано два объемных набора данных:

– 329.000 наборов ключевых слов для русского языка;

– 3.069.000 наборов ключевых слов из сети Вконтакте.

Далее приведены примеры наборов ключевых слов из обоих источников. Наборы ключевых слов научных публикаций:

[топонимический концепт, языковое сознание, когнитивная база, прецедентность, апеллятивация]

[вариабельность сердечного ритма, гребля на каноэ, вегетативный тонус]

[архитектуры, деформации, геологическая среда, сфера взаимодействия].

Наборы ключевых слов из социальной сети:

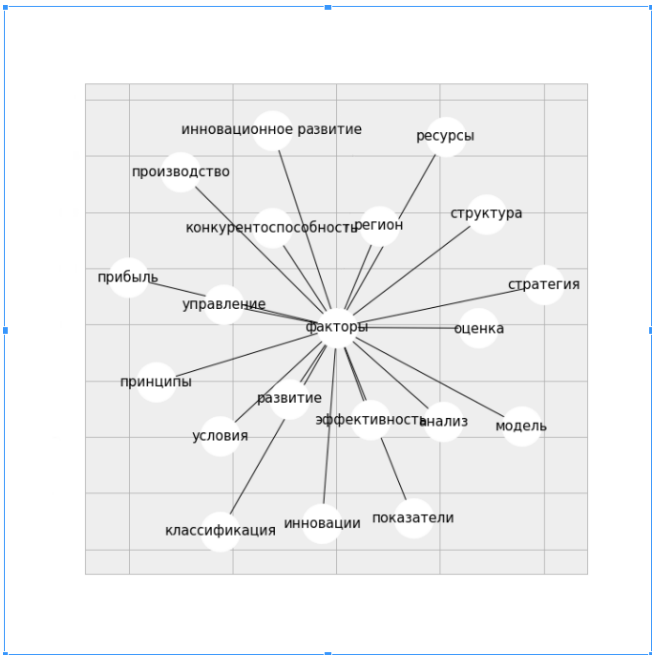
[electro_pop, dance, fresh, music, new_zealand]

[vitaminhealth, oxygenwater, waterhealth]

[bodyfan, питание, bodyfanпитание, bodyfanmotivation, motivation, bodybuilding, фитнес, gym, спорт, мотивация, зож].

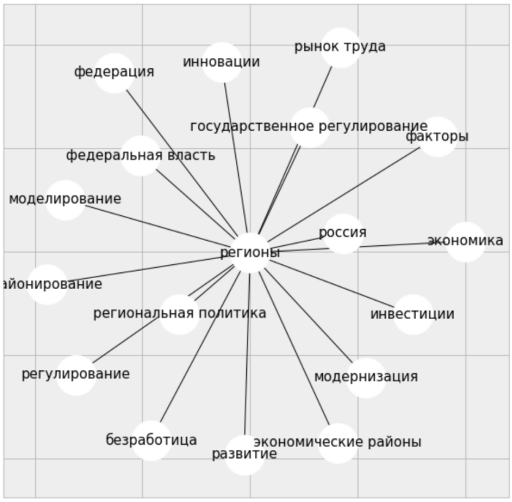
Собранные из социальной сети наборы ключевых слов включают в себя различные способы написания одного и того же слова естественного языка. Кроме того, в данных могут присутствовать переводы слов на другие языки, варианты транслитерации, морфологические словоформы, а также вариации слитного и раздельного написания многословных ключевых слов.

Программные реализации описанных в предыдущих главах алгоритмов были применены к собранным данным. Далее на рисунке 2.1.4 представлены ближайшие соседи для слов «федерация» и «регионы» в графе ключевых слов.



а)

Рисунок 2.2 — Соседи вершины «факторы» в графе ключевых слов



б)

Рисунок 2.3 — Соседи вершины «регионы» в графе ключевых слов

На рисунках, представленных выше, видно, что графа ключевых слов не хватает для определения семантической близости пары слов. Легко заметить, что существуют связи, такие как «факторы-модель», «регионы-экономика», которые не обладают явной смысловой связью. Применение методов построения усеченного контекстного графа дает значительное улучшение качества классификации пар ключевых слов на семантически близкие и далекие. Далее указаны примеры найденных пар ключевых слов, близких по смыслу:

β-адреноблокаторы - бета-адреноблокаторы

новые виды - новый вид

орви - острые респираторные вирусные инфекции

текущий уровень информационной безопасности - политика информационной безопасности

умения - навыки

образное мышление - художественный вкус

хехцир - khekhtsyr

рынок банковских услуг - банковский рынок

тромболизис - тромболитическая терапия

параллельные алгоритмы - параллельное программирование

феминность - фемининность

полином - многочлен

корень - корни

primerun - примерун

fvk - fotovideoclub

еврореволюция - єврореволюція

silk_plaster - шелковая_штукатурка.

Интересным в плане проводимых исследований является решение задачи определения похожих слов для заданного многозначного слова. Рассмотрим семантически близкие понятия для слова «орган». В то время как в графе ключевых слов соседями для слова «орган» являются слова «государство», «сибирь», «контроль», «циркуляция», «управление», в контекстном графе ближайшими являются слова «музыковедение», «организм», «отклонение», «делегирование полномочий», «объект контроля». Легко заметить, что соседи в графе ключевых слов часто употребляются совместно со словом «орган», однако определение таких отношений не вызывает затруднений. В случае контекстного графа были определены близкие по смыслу слова, при этом удалось восстановить не только значение слова, связанное с юриспруденцией, но и близкие слова для значения из области музыки («музыковедение») и биологии («организм»).

На рисунке 2.4 изображены несколько ближайших контекстно близких слов для слова «студенты» (отмечается, что полное множество соседей вершины слишком велико, чтобы его изобразить).

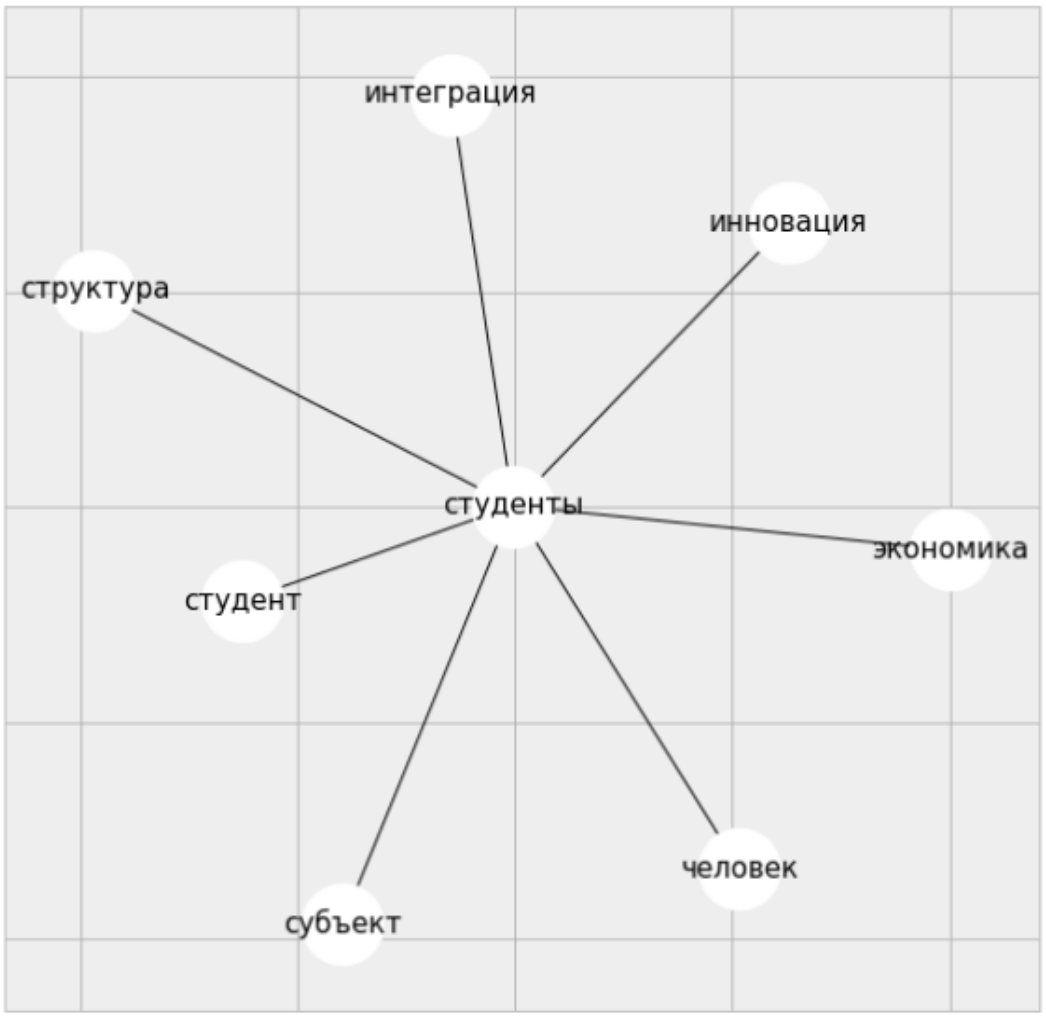


Рисунок 2.4 — наиболее близкие слова для слова «студенты» в контекстном графе

2.1.5 Выводы

В настоящем разделе представлена разработанная автором модель семантической близости *WordContSim*. Эта модель основывается на построении *графа ключевых слов* и на идее о контекстной близости, детально описанной в разделе 2.1.2. Наряду с реализациями известных современных моделями семантической

близости, программная реализация модели протестирована и по результатам сравнительного анализа продемонстрировано значительное улучшение уровня качества определения с ее помощью семантически близких пар на коллекциях наборов ключевых слов небольшого и среднего размеров. Процедура тестовой апробации описывается в разделе 2.1.4. Кроме этого, представлены примеры пар семантически близких ключевых слов для реальных коллекций ключевых слов. Эти примеры демонстрируют адекватность полученных результатов.

К недостаткам разработанной модели можно отнести необходимость подбора правильных нормирующих множителей в формуле. Подбор формулы в общем случае может быть трудозатратной процедурой, поэтому были проведены исследования в направлении автоматического подбора формулы семантической близости по имеющейся графовой информации о ключевых словах и информационной системе. Результатом этих исследований стала модель, которая описывается далее в главе 2.2.

Отмечается практическая значимость полученных результатов в определении семантической близости ключевых слов. Кроме того, важной является возможность применения новой модели в других задачах информационного поиска и анализа данных, необходимых для функционирования аналитическо-информационных систем. Примером такой задачи является кластеризация множества ключевых слов, авторское решение которой описывается в разделе 4.1.

2.2 Использование методов машинного обучения для улучшения модели близости слов. Модель *WordMLSim*

В настоящем разделе рассматриваются методы улучшения качества определения семантической близости для пары ключевых слов с помощью методов машинного обучения с учителем. В тексте подробно описывается подход, который позволяет свести рассматриваемую задачу к задаче классификации, то есть к задаче определения целевой метки заданного объекта из заранее сформированного множества меток. Обучение с учителем подразумевают использование обучающей выборки - множества объектов, для которой известны истинные целевые метки. Эта выборка необходима для тренировки модели машинного обучения.

С помощью обученной модели машинного обучения вычисляется предсказание целевой метки для произвольного объекта системы.

Сбор обучающего набора данных является обычно трудоемким процессом. Как правило, для его реализации требуется, как минимум, участие нескольких экспертов в предметной области, для которой собираются тренировочное множество. Более того, для некоторых задач, в число которых входит и задача определения семантической близости пар ключевых слов, является сложно даже правильно составить инструкцию для экспертов для адекватной оценки тестируемых примеров. Причина таких затруднений заключается в том, что семантическая близость - субъективная величина и сильно зависит от области применения, контекста, человека и задачи. Например, слово «стол» может быть близким по смыслу для слова «стул» из бытовых соображений. В то же время, если бы эти два слова были синонимами (а следовательно взаимозаменяемыми) в информационной системе, представляющей интернет-магазин, продающий мебель, имела бы место ситуация, когда пользователь ищет один из этих предметов, а в поисковой выдаче получает другой, что является недопустимым. Рассмотренные в следующих далее разделах алгоритмы сбора обучающего множества призваны разрешить обозначенные вопросы.

Автором настоящей диссертации были разработаны два автоматизированных метода формирования обучающих примеров без привлечения к этому процессу экспертов. Первый из них включает в себя набор эвристических алгоритмов создания обучающей выборки. Второй метод является полностью автоматическим и строится исключительно при помощи описанных в предыдущих главах графовых моделей представления данных. Важнейшим преимуществом этого метода является его универсальность и возможность его применимости не только к задачам определения семантической близости, но и к любым другим задачам, в которых объекты системы представляются в виде некоторого графа и имеется необходимость в классификации отношений между парой объектов. Универсальность проявляется также в том, что выборка строится непосредственно по данным и не использует никакие внешние источники. Такой подход позволяет определить отношения близости, специфичные для конкретной системы. Другое преимущество метода состоит в возможности использовать эффективные модели машинного обучения с учителем, вместо более слабых моделей без учителя, которые, к тому же представляют сложности при валидации без обучающих примеров.

Задача классификации для определения семантической близости пары ключевых слов формулируется следующим образом. Пусть $X = \{x_i\}_{i=1}^{N}$ - множество пар ключевых слов. $x_i = (x^l,x^r)_i$ - пара ключевых слов, где x_l,x_r - левое и правое слова из пары. $Y = 0,1$ - множество меток. Нулевое значение соответствует отсутствию семантической близости для пары ключевых слов, а единичное, напротив, сильной смысловой связи. Поскольку Y состоит только из двух элементов, то такая задача является задачей бинарной классификации. $X^l = (x_i,y_i)_{i=1}^{l}$, где $x_i \in X, y_i \in Y$ - обучающая выборка. Далее по обучающей выборке строится классификатор $a : X \to Y$

Процесс сбора обучающей информации является одним из важнейших этапов обучения эффективной модели определения семантической близости. Сложность этого процесса заключается в отсутствии возможности точно формализовать для пары ключевых слов отношения «являются семантически близкими» и «не являются семантически близкими». Во многих случаях определение смысловой близости зависит от решаемой задачи, поэтому обучающие выборки для одной задачи могут не подходить для обучения моделей из другой. Например, пары ключевых слов «математика» и «математическая статистика» связаны отношением гиперонимии («математическая статистика» является разделом «математики»), что влечет некоторую смысловую близость между понятиями. С другой стороны, если рассматривать задачу поиска документов системы по ключевым словам, то при заданном пользователем запросе «информатизация, математика, вектор информатизации, информационные технологии, mathematica». Для данного запроса ключевое слово «математическая статистика» не подходит по контексту данного запроса, кроме того, это ключевое слово является более узким по смыслу, чем «математика» и поэтому шансы пользователя получить релевантные запросу документы минимальны: если даже пользователь предполагал какую-то более узкую область, нет никаких оснований полагать, что это именно «математическая статистика», а не, например, «вычислительная математика». Обратная ситуация, когда пользователь ввел запрос «математическая статистика, статистические тесты», то поиск в документах предположительно синонимичного слова «математика» также может привести к ухудшению выдачи. В результате этого действия, в выдачу вероятно попадут документы слишком общего смысла, такие как статьи про математику как науку.

Другим примером неоднозначности в определении семантической близости может служить различие в тематической направленности систем, в которых

используются модели определения смысловой близости. Для наукометрических систем пара ключевых слов «математическая статистика» и «вычислительная математика» вряд ли должны иметь большое значение метрики смысловой близости, поскольку в рамках такой системы эти два понятия представляют два совершенно различных направления математики. В то же время, в системах более общей направленности, где могут присутствовать документы любого рода (а не только научные публикации), данная пара ключевых слов должна иметь более высокий уровень семантической похожести. Причина заключается в том, что для такой системы все термины относительно небольшого раздела «математика» могут считаться близкими по смыслу.

Качество модели определения семантической близости в данном случае целиком определяется двумя составляющими: параметрами обучающей выборки (качеством, разнообразием и количеством обучающих примеров) и эффективностью выбранного алгоритма машинного обучения.

Лучшим алгоритмом для решения поставленной задачи бинарной классификации является модель градиентного бустинга на решающих деревьях XGBoost [114]. В ходе обучения происходит последовательное построение композиции решающих деревьев. Каждое следующее дерево стремится максимально уменьшить ошибку уже построенной части ансамбля. Анализ эффективности выбранной модели и результаты тестовых испытаний представлены в следующих далее подразделах.

2.2.1 Методы формирования обучающей выборки

В данном подразделе описывается два разработанных способа получения обучающего набора данных. Первый из них использует различные эвристические идеи. Их суть состоит в том, как с помощью простых детерминированных процедур а, в некоторых случаях, внешних открытых наборов данных, получать примеры близких по смыслу пар ключевых. Такие методы имеют высокий уровень точности, однако слабое покрытие пространства пар ключевых слов системы. Второй способ заключается в использовании теоретико-графовых алгоритмов, способных выдавать примеры семантически близких и далеких пар объектов, основываясь на графовой структуре представления данных.

Эвристические методы

С позиции методологии различаются два типа алгоритмов сбора обучающей выборки. Первый из них заключается в использовании некоторых внешних словарей и дальнейшей фильтрации этих словарей по тем словам, которые присутствуют в информационной системе. Второй способ сначала использует некоторый генеративный алгоритм, а именно алгоритм, который для заданного слова генерирует различные слова-кандидаты, потенциально близкие по смыслу к заданному. В качестве такого алгоритма может выступать, например, алгоритм, который в ходе работы для ключевого слова выбирает все слова, которые встретились с заданным внутри одного набора. Далее полученное множество фильтруется с помощью некоторого фильтрующего алгоритма.

В результате работы программной реализации такого алгоритма определяются пары ключевых слов, которые принимаются за семантически близкие. Этот алгоритм должен обладать высокой точностью, хотя и возможно низкой полнотой определения семантической близости. Другими словами, пары ключевых слов, объявленные семантически близкими, должны действительно являться похожими с вероятностью, близкой к 100%. Следует однако отметить, что при этом количество таких пар может быть относительно невелико.

Примером такого алгоритма может быть алгоритм, считающий расстояние Левенштейна. Если редакторское расстояние не превосходит 1, то слова имеют очень схожее написание и весьма вероятно, что они близки по смыслу. Такая процедура фильтрует значительную часть пар. Однако, те пары, которые прошли фильтрацию, почти всегда имеют высокую степень смысловой близости. Таким образом каждый такой фильтр получает примеры близких по смыслу пар слов с высокой точностью, но низкой полнотой. Собранные с помощью такого подхода пары ключевых слов являются положительными примерами обучающей выборки для дальнейшего обучения.

В рамках исследований по теме настоящей диссертации автором были разработаны перечисленные далее эвристические методы сбора обучающей выборки.

– Поиск простых аббревиатур. Ключевые слова системы разделяются на понятия, содержащие ровно одно слово - аббревиатуры, и понятия, содержащие более одного слова - расшифровки аббревиатур. Далее для

каждого слова из множества аббревиатур и каждого слова из множества расшифровок проверяется, действительно ли данная расшифровка является расшифровкой для данной аббревиатуры. Другими словами проверяется выполнение условия, что существуют такие префиксы слов расшифровки, которые могут полностью покрыть аббревиатуру. Если соответствие установлено, то пара аббревиатура-расшифровка добавляется в список пар-кандидатов. Далее к парам-кандидатам приписываются частоты входящих в них ключевых слов. Для каждой аббревиатуры берется не более 5 наиболее частотных вариантов расшифровок. Пары, прошедшие данную фильтрацию, формируют окончательное обучающее множество аббревиатур.

– Поиск скобочных аббревиатур. Иногда при написании ключевых слов-аббревиатур в скобках указывается правильная расшифровка для данной аббревиатуры. Такие действия позволяют собрать дополнительные пары аббревиатура-расшифровка для обогащения множества обучающих примеров.
– Поиск разных форм одного слова. С помощью пакета обработки естественного языка NLTK для каждого слова в системе рассматриваются различные его формы. Если одна из форм также присутствует в множестве всех ключевых слов, то пара слово-форма добавляется в обучающее множество.
– Поиск похожих по написанию слов. Рассматриваются все пары слов, расстояние Левенштейна между которыми равно 1. Эти пары добавляются в обучающее множество.
– Поиск переводов с одного языка на другой. Данный метод использует API сервиса Яндекс.Переводчик и собирает варианты переводов ключевых слов с русского языка на английский.
– Поиск синонимов в тезаурусе WordNet. Были использованы пары синонимов и пары гипоним-гипероним. Кроме поиска по англоязычным синсетам был использован двухступенчатый подход определения синонимов: русскип слова переводились средствами сервиса Яндекс.Переводчик на английский язык, а затем поиск проводился уже по английским версиям слов.

– Использование открытых источников синонимов. Из словарей синонимов, доступных в сети Интернет, выделяются те пары, слова которых присутствуют в множестве ключевых слов.

Помеченные методы сбора данных позволяют собрать положительные примеры для обучения, то есть примеры пар ключевых слов, являющиеся семантически похожими. Однако для правильного обучения модели необходимы также и отрицательные примеры. С помощью небольших модификаций эвристических алгоритмов сбора положительных примеров можно получить алгоритмы для сбора отрицательных примеров. На этом направлении использованы следующие идеи.

– Поиск антонимов в тезаурусе WordNet. Аналогично поиску синонимов был проведен поиск антонимов в базе WordNet. Также были использованы пары слов, расстояние в дереве WordNet между которыми превышает 2.
– Неправильные расшифровки для аббревиатур. Рассматриваются те аббревиатуры, для которых были найдены правильные расшифровки. Этим аббревиатурам ставятся в пару случайные многословные ключевые слова, которые точно не являются правильными расшифровками.
– Использование открытых источников антонимов.
– Случайные пары ключевых слов. Для каждого ключевого слова, для которого в выборке присутствуют положительные примеры, были взяты случайные ключевые слова в пару. На одно слово было сгенерировано не более, чем 10 пар. Данный метод был использован по следующим соображениям. Предполагается, что для каждого ключевого слова существует константное число близких по смыслу ключевых слов. То есть если в достаточно большой информационной системе начинает расти количество ключевых слов, то кол-во слов, похожих на данное слово, не будет расти линейно по числу уникальных слов системы. В то же время количество пар растет квадратично, а количество пар для заданного слова - линейно, а это означает, что если взять для заданного слова в пару случайное слово, то эта пара не будет связана семантически. При этом важным является подбор доли случайных пар. Если эта доля мала, то в выборке не будет достаточного разнообразия отрицательных примеров. Если же доля слишком велика, то случайные пары будут вносить слишком большой

вклад в обучение модели. Это не является положительным фактором, потому что случайные отрицательные пары не настолько качественны, как, например, антонимы из словаря. Оба этих случая приводят к ухудшению качества модели определения семантической близости.

В результате работы программных реализаций перечисленных выше алгоритмов было собрано 234974 положительных и 1175610 отрицательных примеров.

Обучение моделей на эвристически подобранных выборках имеет представленные далее заметные недостатки.

– Смещение в данных. Для наилучшего обучения классификатора необходимо, чтобы объекты обучающей выборки выбирались из распределения тех объектов, на которых классификатор будет работать в реальной системе. Рассмотрим данную особенность на следующем тривиальном примере. Допустим, что все пары ключевых слов системы делятся на Аббревиатуры (то есть пара аббревиатура - расшифровка аббревиатуры) и Переводы (слово на русском языке - его перевод на английский). Если Аббревиатур в системе 10%, то и в обучающем подмножестве Аббревиатур должно быть 10%. Если же аббревиатур при обучении будет значительно больше (например, 70%), то модель начнет усиленно использовать те факторы, которые повышают ее качество на аббревиатурной части выборки, потому что это лучшим образом оптимизирует функцию потерь. Однако, в момент применения модели, к ней на вход будет поступать только 10% аббревиатур и 90% переводов и очень вероятно, что факторы, используемые моделью не будут оптимальны для классификации переводов и, тем самым, это ухудшит качество реальных решаемых данным классификатором задач.

  Другой недостаток смещения заключается в том, что если перегрузить обучающую выборку примерами неправильных переводов, то это негативно отражается на предсказанных уровнях близости во время этапа применения модели. Как следствие, это понизит средний уровень близости между переводами в системе. Отмеченное понижение приведет к тому, что если для данного ключевого слова («MSU») есть и правильная аббревиатура («Moscow State University»), и правильный перевод («МГУ»), то из-за рассмотренной манипуляции с обучающей выборкой,

значимость перевода будет понижена. В конечном счете может оказаться, что согласно модели, все аббревиатуры лучше всех переводов. Такое поведение не очевидно и не основывается на реальной работе информационной системы. Такой сценарий возможен, поскольку положительные и отрицательные примеры берутся из разных источников. Поэтому степень покрытия ими всего разнообразия пар ключевых слов, а также возможность контролировать нужную долю положительных примеров остается под вопросом;

– Модель настраивается не под конкретную задачу и не под конкретные данные системы. Если, например, слова «объединять» и «интегрировать» попали в один внешний словарь синонимов общего назначения, то внутри информационной системы, направленной на изучение технических наук, эти слова определяют две различные математические операции. Может также случиться, что в узкоспециальных областях понятия могут являться слишком общими по смыслу, чтобы быть похожими. Примером такой пары может быть «ураган» и «тайфун». Для пользователя, например, социальной сети эти слова действительно похожи. Однако, в рамках системы, изучающей различные природные явления, слова имеют важные смысловые отличия. Чем большей спецификой обладает тематика системы, тем четче прослеживаются различия между словами, которые с точки зрения словарей общей направленности, очень похожи.
– Сложность процесса сбора обучающей выборки. Для каждой системы нужны свои наборы эвристик, что затрудняет внедрение таких классификаторов повсеместно.
– Ошибки в выборке. По причине того, что частично выборка состоит из случайных примеров, весьма вероятно, что существует небольшое число пар, которые ошибочно промаркированы отрицательными. Это может понизить качество обученной модели. Помимо этого могут возникать ошибки второго рода. Например, если внешний сервис, предоставляющий переводы для слов и фраз допустил ошибку, то в множество положительных примеров попадет неправильный перевод для слова.

В связи с описанными выше недостатками, автором были проведены исследования возможности генерации обучающей выборки в автоматическом или полуавтоматическом режиме. Результаты этих исследований детально описаны в следующем далее подразделе.

Алгоритм автоматизированной генерации обучающей выборки

Для устранения недостатков, описанных в предыдущем подразделе, был разработан метод автоматической генерации обучающей выборки. Целью, как и ранее, является определение множества пар ключевых слов с соответствующими оценками $((w_{i1}, w_{i2}), y_i)_{i=1}^{N}$, где $w_{ij} \in W$ - слова из множества всех ключевых слов информационно-аналитической системы, число $y_i \in 0, 1$ указывает на то, близки ли семантически соответствующие слова (позитивные или положительные примеры) или нет (негативные или отрицательные примеры). Суть данного алгоритма заключается в использовании графовой структуры данных для получения обучающих примеров.

Рассмотрим алгоритм генерации обучающей выборки на упрощенном примере. Допустим, коллекция наборов ключевых слов T состоит из приведенных далее наборов:

[X, Y, W, A, C]
[A, B, V, W]
[A, C, Z, W]
[C, Y, Z, W].

Здесь каждая заглавная буква подразумевает одно ключевое слово. Различным буквам соответствуют различные ключевые слова. По этим данным имеется возможность построить граф ключевых слов G_{kw}, описание построение которого описывается в 2.1.1. Изображение этого графа приводится на рисунке 2.5.

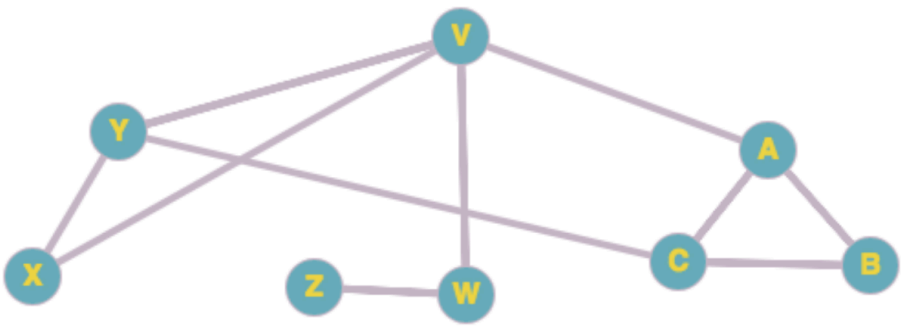


Рисунок 2.5 — Граф ключевых слов на упрощенной коллекции ключевых слов

Для простоты изложения веса ребер на приведенной выше иллюстрации опускаются. Отмечается также, что структура графа выбрана исключительно для наглядности примера и предполагается, что существование ребра между двумя

вершинами говорит о некоторой «похожести» соответствующих ключевых слов. Следующим шагом алгоритма является искусственная модификация исходной коллекции T. Для этого производятся следующие операции:

– фиксируется дискретное распределение с неотрицательными значениями P;
– для каждого вхождения каждого слова коллекции T из распределения P выбирается случайное число r;
– если $r > 0$, то рассматриваемое вхождение слова w заменяется на w^{*r}.

В программной реализации алгоритма генерации обучающей выборки для модификации ключевого слова к нему в конец добавляется специальный символ, который не используется в коллекции, с последующим добавлением номера r, выбранным случайно. Данный способ замены гарантирует уникальность нового ключевого слова среди множества ключевых слов W изначальной коллекции T.

Рассматривая для упрощения в качестве дискретного распределения равномерное распределение из двух значений, исходная коллекция может быть модифицирована, например, следующим образом:

[X, Y, W, A_1, C]
[A, B_1, V, W]
[A_1, C_1, Z, W]
[C, Y, Z, W].

Обозначим модифицированную коллекцию за T^m. После этого построим граф ключевых слов G^m_{kw}. Вид этого графа изображен на рисунке 2.6.

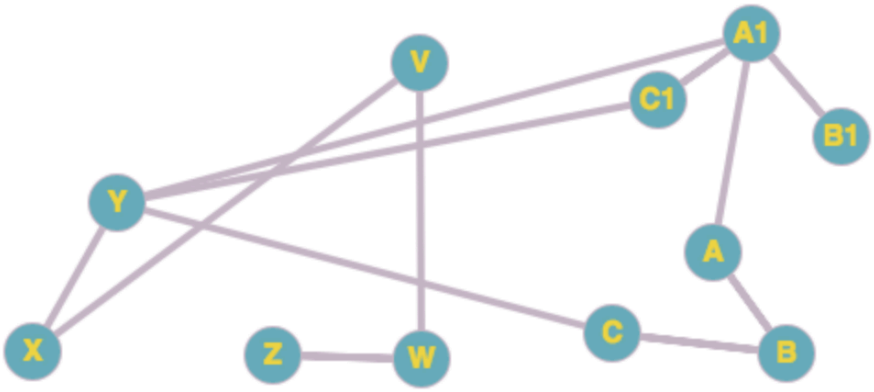


Рисунок 2.6 — Граф ключевых слов на модифицированной упрощенной коллекции ключевых слов

Можно заметить, что в ходе работы алгоритма были модифицированы вершины A, B и C. После такого перестроения, в графе появились соответствующие

им вершины A_1, B_1 и C_1. Однако, эти пары не обязаны соединяться ребром в этом графе.

В процессе модификации семантическому изменению слова не подверглись, поскольку к ним, фактически, были приписаны служебные символы в конец. Другими словами, пара слов (A, A_1) является, по сути, одним и тем же словом, и, следовательно, может быть положительным обучающим примеров для выборки, потому что для этих слов существует две вершины в графе. Таким образом, для рассмотренного примера определены положительные обучающие примеры $(A, A_1), (B, B_1), (C, C_1)$.

Задачей машинного обучения в данном случае является восстановление факта идентичности различных версий одного слова путем выявления различных графовых связей между парами обучающих примеров. Под графовыми связями здесь и далее подразумеваются различные меры и характеристики, которые можно вычислить для пары вершин одного графа. Такими характеристиками являются длины кратчайших путей, количество общих соседей между вершинами, факт принадлежности одной компоненте связности и прочее. Далее используется предположение о том, что найденные закономерности, влияющие на значение уровня семантической близости в графе G^m_{kw}, также присутствуют в графе G_{kw}. Это предположение имеет место по той причине, что графы G^m_{kw} и G_{kw} строятся при помощи одного и того же алгоритма.

Важную роль в разработанном алгоритме генерации выборки играет распределение, из которого берутся индексы модифицированных объектов. Использование бернуллиевского распределения с вероятностью успеха $p = 0.5$ имеет существенный недостаток. Этот недостаток состоит в том, что ключевые слова w и w_1 будут встречаться примерно одинаковое количество раз в коллекции T^m. Таким образом, обученная модель будет предполагать, что пара хороших синонимов в исходной коллекции T также обладает этим свойством, что неверно. Проиллюстрировать это можно следующим примером. Пара слов «жалованье», «зарплата» являются синонимами, однако очевидно, что слово «жалованье» является устаревшим и в наше время употребляется значительно реже.

В связи с приведенными замечаниями, для исследований было выбрано более сложное распределение: геометрическое распределение. Такое распределение помогает преодолеть недостаток, описанный ранее. Кроме того, засчет бесконечного носителя это распределение позволяет генерировать множество различных вариаций для одного слова (количество различных вариаций для одного слова

ограничено числом употреблений исходного слова в коллекции T). Этот факт является важным достоинством этого распределения по двум причинам. Во-первых, такое положение дел более приближено к реальности, потому что ключевое слово может иметь более, чем одно близкое по смыслу слово. Во-вторых, это распределение позволяет существенно обогатить выборку, поскольку любую пару модификаций исходного слова можно считать положительным примером для обучения. Плотность геометрического распределения в зависимости от вероятности успеха p изображена на рисунке2.7. На рисунке 2.8 показана гистограмма индексов при генерации модифицированных версий некоторого ключевого слова.

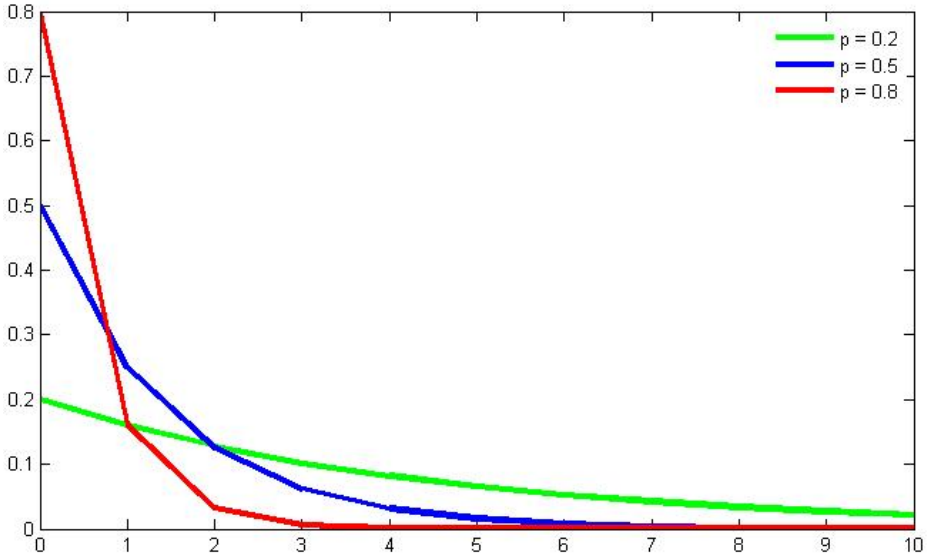


Рисунок 2.7 — Плотность геометрического распределения при разных значениях

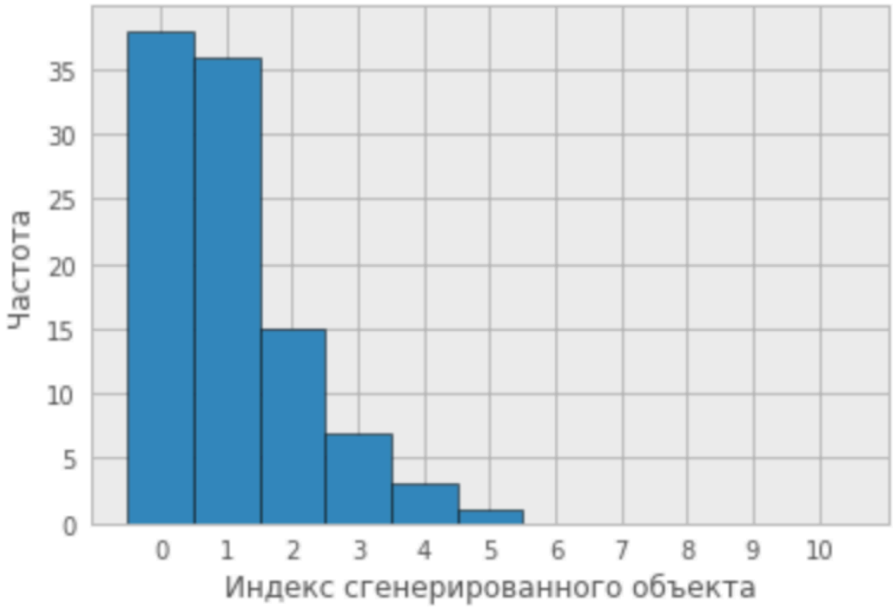


Рисунок 2.8 — Зависимость количества объектов от сгенерированного индекса

После того, как положительные примеры обучающей выборки собраны, необходимо создать также и отрицательные примеры. Для отрицательных примеров используется следующий алгоритм:

1. построение графа ключевых слов $\hat{G}$ по модифицированному множеству ключевых слов;
2. для каждого слова w, попавшего в множество пар положительных примеров, рассматриваются случайные k соседей $v_1, v_2, ..., v_k$ на расстоянии 2 в графе $\hat{G}$;
3. все пары $(w, v_i)_{i=1}^{k}$ принимаются за отрицательные примеры.

Мотивация для использования этого алгоритма заключается в следующем. Для обучения модели отрицательные примеры должны быть содержательными. Если рассмотреть абсолютно случайные пары ключевых слов, то большинство полученных примеров не будут представлять особой значимости для модели. Причина в том, что большинство статистических характеристик в этом случае будут иметь нулевое или близкое к нему значение. Между случайными вершинами в графе может не быть путей, они могут не иметь общих соседей, могут быть расположены на слишком большом удалении друг от друга и т.д.. Такие случайные пары не дают никакой информации для модели для более сложных случаев, когда слова находятся достаточно близко друг к другу в графе, но при это не являются семантически близкими. Однако, именно такие случаи и представляют интерес в рамках решаемой задачи.

С другой стороны, если для заданного ключевого слова в качестве отрицательных примеров брать непосредственных соседей в графе (то есть максимально близкие вершины), то слишком часто будут появляться пары, которые в действительности близки по смыслу. Это происходит потому, что среди слов одного набора могут встречаться похожие слова. Вследствие этого, среди отрицательных примеров будет большое число примеров, которые ошибочно были промаркированы отрицательными, что не позволит качественно обучить модель.

Поэтому соседи вершины на расстоянии два представляются наиболее подходящим компромиссом: среди них не так часто встречаются семантически близкие пары, однако при этом они несут в себе содержательный сигнал, на котором можно эффективно обучаться. С учетом изложенных выше соображений, алгоритм автоматизированной генерации обучающей выборки выглядит следующим образом.

– Входные параметры: N, p, k, D.

- Для каждого набора ключевых слов $t \in T$ системы D:
  - для каждого *вхождения* ключевого слова w:
    - если количество вхождений в коллекцию превосходит пороговое значение N:
      - из геометрического распределения с параметром p выбирается число;
      - вхождение ключевого слова w заменяется w_p;
      - слово w_p заносится в список модификаций для слова w.
    - иначе
      - вхождение ключевого слова w заменяется w_0.
  - модифицированная версия набора ключевых слов добавляется в множество D^m.
- Для каждого уникального ключевого слова $w \in W$:
  - каждую пару модификаций (w_i, w_j) ключевого слова w добавить в обучающее множество X_{train} в качестве положительного примера.
- Для модифицированной версии коллекции наборов ключевых слов D^m произвести построение графа ключевых слов G^m_{kw}.
- Для каждой вершины w_i из множества модифицированных вершин графа G^m_{kw}:
  - выделить множество $G^m_{kw,2}(w_i) = neigbors(G^m_{kw}, w_i, 2)$ соседей вершины w_i в графе G^m_{kw} на расстоянии 2;
  - из множества $G^m_{kw,2}(w_i)$ выбрать случайно k вершин $v_1, ...v_k$ и для каждой вершины добавить пару (w_i, v_j) в X_{train} в качестве отрицательного примера.
- Вернуть обучающее множество X_{train}, граф G^m_{kw}.

Значения параметров N, p, k были определены как 10, 0.9, 5, соответственно. Необходимость выбора параметра N обусловлена тем наблюдением, что если слово встречается в коллекции D слишком редко, то всего его модификации будут встречаться в коллекции D^m еще реже. Для этих слов в графах будет мало связей. Поэтому те слова w, которые встретились в исходной коллекции меньше, чем N раз, игнорируются алгоритмом, а точнее, всегда заменяются на w_0.

Вхождением ключевого слова в описанном выше алгоритме является появление ключевого слова в одном конкретном наборе ключевых слов. Поскольку

одно слово может присутствовать во многих наборах, то число вхождений равно количеству наборов, содержащих данное ключевое слово.

Следующее далее утверждение дает оценку вычислительной сложности алгоритма генерации обучающей выборки.

Утверждение 5. Расчет искусственной обучающей выборки имеет сложность $O(f^2k + w^2k^2)$, где f - максимальная частотность слова в коллекции, k - количество уникальных слов в коллекции, w - максимальный размер набора в коллекции ключевых слов.

Лемма 1. Построение положительных примеров искусственной обучающей выборки имеет сложность $O(f^2k)$, где f - максимальная частотность слова в коллекции, k - количество уникальных слов в коллекции.

Доказательство. Число всех ключевых слов системы с повторениями не превышает значение $f \cdot k$. Необходимо пройти один раз по всем ключевым словам коллекции, чтобы сгенерировать для них модифицированную версию - это требует $O(fk)$ операций. При этом полагается, что выбор случайного числа из выбранного распределения занимает константное время. Следующим шагом необходимо из модифицированных версий каждого слова получить всевозможные пары. Для каждого из k слов может быть получено не более, чем f модификаций слова, поскольку каждая модификация была получена из некоторого исходного слова, которое в свою очередь было использовано не более, чем f раз в коллекции. Для генерации всех пар для данного исходного слова необходимо $O(f^2)$ времени, а для всех уникальных слов - $O(kf^2)$.

Лемма 2. Построение отрицательных примеров искусственной обучающей выборки имеет сложность $O(w^2k^2)$, k - количество уникальных слов в коллекции, w - максимальный размер набора в коллекции ключевых слов.

Доказательство. . Построение графа ключевых слов занимает не более $O(k^2)$ времени. Далее необходимо рассчитать соседей для каждой вершины на расстоянии 2. Для этого необходимо возвести в квадрат матрицу смежности графа. При хранении графа в виде разреженной матрицы, операция умножения матрицы размера $(k \times k)$ на себя требует $O(nnz \cdot k)$ операций, где nnz - количество ненулевых элементов. Поскольку в графе ключевых слов ребрами являются те пары ключевых слов, которые входят в один набор, то количество ребер в графе, как и количество ненулевых элементов в матрице смежности, не превышает $O(w^2k)$, то сложность поиска соседей второго порядка будет равна $O(w^2k^2)$. Предполагая, что операция взятия случайного числа из заданного распределения занимает

константное время, получаем, что генерация отрицательных примеров для искусственной обучающей выборки требует $O(w^2k^2)$ операций.

Из двух доказанных выше лемм следует утверждение 5. После того, как обучающие выборки созданы, необходимо для обучающих объектов посчитать различные графовые характеристики, которые будут описывать данный объект. Вместе с целевой переменной эти характеристики необходимы для обучения классифицирующей модели.

Разработанный подход к определению смысловой близости с помощью машинного обучения и автоматизированной обучающей выборки обладает следующей особенностью. Обученная на графе G^m_{kw} модель впоследствии используется для предсказания семантически близких пар уже на графе G_{kw}. Это накладывает некоторые ограничения на признаки, которыми описываются объекты машинного обучения. В следующем разделе приводится описание этих признаков и объясняется отмеченная особенность.

2.2.2 Признаковое описание модели машинного обучения

Под признаковым описанием пары ключевых слов понимается определение набора функций, вычисляющих некоторые числовые характеристики для рассматриваемых слов. Такие функции (факторы) могут использовать оба слова в явном виде (расстояние Левенштейна, количество общих слов и т.д.) или только одно из слов (частотность данного слова в корпусе, количество символов в слове и т.д.), что порождает пару значений признака для левого и правого слов из пары.

Результатом применения программных реализаций функций к паре ключевых слов является фиксированный по длине массив чисел. Массивы чисел, посчитанные для всех пар ключевых слов обучающей выборки, образуют так называемую матрицу *объектов-признаков*. Строки этой матрицы представляют собой признаковое описание обучающих объектов выборки, а столбцы соответствуют признакам.

Признаком может являться любая функция, способная вычислить значение для поданного на вход объекта. Следует однако понимать, что не каждая функция может быть полезна в процессе обучения. Формирование списка признаков, наилучшим образом характеризующих объекты, является важным шагом

при решении задачи машинного обучения с использованием модели градиентного бустинга. Массив признаков представляет собой то, как «видит» объект модель машинного обучения. В связи с этим, признаки должны в полной мере описывать объект. В рамках поставленной задачи это означает, что для пары ключевых слов полезными могут быть признаки, описывающие различные синтаксические, морфологические, семантические, статистические и графовые характеристики пары рассматриваемых слов.

Таким образом, в ходе работы алгоритма для всех рассматриваемых пар ключевых слов происходит вычисление всего набора факторов. В результате появляется матрица объект-признак. Строка этой матрицы является описанием конкретной пары ключевых слов, а столбец определяет значения некоторого признака для всех пар. Для обучающей выборки кроме матрицы объект-признак также имеется вектор-столбец истинных значений. Это позволяет обучать алгоритмы машинного обучения, после чего применять обученные модели для предсказания на новых данных (для которых также можно получить признаковое описание). Является важным отметить тот факт, что описанные в предыдущем подразделе методы формирования обучающей выборки вносят существенные ограничения на разработку признаков. В рамках данной работы выделяется два принципиально различных класса признаков: *графовые* и *языковые*.

Графовыми факторами являются те, которые возможно получить по графам, описанным в предыдущих главах. Графовые признаки ни в каком виде не учитывают содержимое вершин графа, то есть не учитывают особенности ключевого слова с точки зрения естественного языка. Вместо этого, такие признаки используют только связи данной вершины с другими вершинами.

Языковые признаки, напротив, используют знания из области обработки естественного языка: морфологические, синтаксические и семантические. Кроме того, к классу языковых были отнесены признаки, вычисленные при помощи внешних по отношению к рассматриваемой информационно-аналитической системе наборов данных.

Необходимость выбранного разделения признаков на классы обусловлена следующим обстоятельством. В связи с особенностями процесса формирования автоматизированной обучающей выборки, для моделей, обученных на этом типе выборки, не представляется возможным использование языковых факторов. Причиной этому является принцип построения автоматизированной обучающей

выборки. Для генерации нового позитивного примера к рассматриваемому слову приписывается служебный, семантически незначащий суффикс. Это означает, что языковые различия между словами в такой паре является искусственными и не отражают действительную природу вещей. Использование эвристической обучающей выборки не накладывает ограничений на выбор признаков, однако обладает недостатками, описанными в предыдущем подразделе.

После вычисления значений всех признаков начинается процесс обучения модели. Для его запуска необходимо подать матрицу объектов-признаков и обучающую выборку (эвристическую или автоматизированную) на вход модели обучения. С учетом изложенных выше соображений, для решения поставленной задачи необходимо определить множество факторов, которое позволит достичь наилучшего показателя качества модели машинного обучения. Для автоматизированной выборки выделяются графовые факторы, а для эвристической - графовые и языковые. Особый интерес представляет автоматизированная выборка, поскольку ее использование позволяет полностью автоматизировать весь процесс решения задачи семантической близости пары ключевых слов.

Полный список рассмотренных признаков приводится в следующей далее таблице 8.

Признак	Описание	Тип
Расстояние в графе ключевых слов		Парный
Расстояние в усеченном контекстном графе ключевых слов		Парный/графовый
Контекстная близость	Вариации контекстной близости, описанные в подразделе 2.1.2	Парный/графовый
Ранк контекстной близости		Парный/графовый

Мера абстрактности	Мера общности понятия, описанная в разделе 4.2.1	Одиночный/графовый
Количество вершин в кластере	Количество вершин кластера для рассматриваемой вершины	Одиночный/графовый
PageRank вершины графа	Значение PageRank меры для вершин графов	Одиночный/графовый
EdgeRank ребра графа	Значение EdgeRank меры для ребер графов	Парный/графовый
Вес ребра в графе ключевых слов		Парный/графовый
Частотность левого слова		Одиночный/графовый
Частотность правого слова		Одиночный/графовый
Совместная частотность в корпусе		Парный/графовый
Pointwise mutual information		Парный/графовый
Мера Жаккара	Считается по вершинам соседям	Парный/графовый
Число простых путей в графах между вершинами	Считается по обоим графам	Парный/графовый
Betweenness centrality		Одиночный/графовый
Closeness centrality		Одиночный/графовый
Eigen vector centrality		Одиночный/графовый

Значение потока между вершинами		Парный/графовый
Количество соседей	На расстоянии 1 и 2 (два фактора)	Одиночный/графовый
Количество общих соседей	На расстоянии 1 и 2 (два фактора)	Парный/графовый
Расстояние Левенштейна		Парный/языковой
Часть речи слова	Категориальный признак	Одиночный/языковой
Количество слов		Одиночный/языковой
Количество букв		Одиночный/языковой
Количество общих слов		Парный/языковой
Пословная мера Жаккара		Парный/языковой
Нграммная мера Жаккара	Используется биграмное и триграмное представление слов	Парный/языковой
Пара слов на разных языках	Бинарный признак	Парный/языковой
Одно слово является аббревиатурой/сокращением для другого	Бинарный признак	Парный/языковой
Одно слово является формой другого	Бинарный признак	Парный/языковой
Одно слово является транслитерацией другого	Бинарный признак, использован наивный алгоритм транслитерации	Парный/языковой
Расстояние по дереву WordNet		Парный/языковой

Глубина слова в дереве WordNet		Одиночный/языковой
Косинусное расстояние между word2vec представлениями слов		Парный/языковой
Средняя частотность слов в корпусе	В качестве слов рассматриваются не ключевые слова, а отдельные слова естественного языка	Одиночный/языковой
Косинусное расстояние пословных TFIDF представлений	В качестве слов рассматриваются не ключевые слова, а отдельные слова естественного языка	Парный/языковой
Косинусное расстояние нграммных TFIDF представлений	Подсчитано на символьных триграммах	Парный/языковой

Таблица 8 — Факторы, используемые моделью для обучения

В таблице 8 представлен набор факторов, который был использован в работе для обучения модели.

Заключительным этапом в построении классификатора является процесс настройки его параметров. Данная процедура реализуется с помощью стандартной схемы подбора. Суть ее заключается в следующем.

1. **Выбирается набор параметров модели.**
2. **Производится разбиение обучающей выборки на несколько подмножеств.** В исследованиях использовалось 5 подмножеств. В каждое подмножество входило приблизительно 20%, причем для всех пар ключевых слов обучающей выборки выполняется, что если слово w_i находится в одном из подмножеств, то и все пары $\{(w_i, w_j)|(w_i, w_j) \in X_{train}\}$ также лежат в этом подмножестве.

3. **Производится обучение модели.** Модель обучается на 4 подмножествах из 5, используя набор входных параметров.
4. **С помощью обученной модели вычисляются предсказания уровня семантической близости.** Предсказание происходит на оставшемся пятом подмножестве обучающей выборки. Такая схема тестирование носит название перекрестной валидации.
5. **Вычисляется значение оптимизируемой метрики**. Предсказание происходит на оставшемся пятом подмножестве обучающей выборки. Такая схема тестирование носит название перекрестной валидации.
6. **Шаги 3-5 повторяются для всех возможных сочетаний подмножеств по 4.** Значение метрики усредняется.
7. **Шаги 1-6 повторяются для всех рассматриваемых наборов параметров.** В ходе работы запоминается тот набор параметров, на котором достигается наилучшее значение метрики.
8. **Обучается финальная модель.** Для этого используются оптимальные параметры, найденные на шаге 7. Для обучения используется вся обучающая выборка.

Обучив модель, появляется возможность с ее помощью предсказывать значения уровня близости для любой пары ключевых слов коллекции D. В следующем разделе описываются тестовые испытания программных реализаций представленных моделей.

2.2.3 Тестовые испытания

Далее представлены результаты тестирования программных реализаций описанных алгоритмов определения близости с использованием методов машинного обучения. Рассмотрены два способа проведения тестовых испытаний: на отложенной части обучающей выборки и на множестве пар одинаковых слов.

Тестирование разработанных программных реализаций моделей необходимо в связи с их практической важностью для интеллектуально-аналитических систем. Описанные ранее модели используются в дальнейшем для определения уровня семантической близости наборов ключевых слов, а также для решения прикладных задач, описание которых приводится в главе 4.

Тестирование обучения модели с автоматизированной выборкой. Тестирование программных реализаций алгоритмов проводилось на следующих наборах данных.

– **Aminer**[1]. Открытая коллекция наукометрических данных, созданная авторами [115]. Коллекция располагает англоязычной информацией о научных публикациях, авторах, конференциях и ключевых словах к научным статьям. Для тестирования была выбрана коллекция ключевых слов из категории *обработки естественного языка* (*natural language processing*).
– **KeywordsRu**. Коллекция наборов русскоязычных ключевых слов, собранных из сети Интернет. Используемая выборка была использована ранее в разделе 2.1.4.
– **KeywordsEn**. Коллекция наборов англоязычных ключевых слов, собранных из сети Интернет. Ключевые слова для данной выборки были собраны аналогично процедуре сбора русскоязычного набора данных *KeywordsRu*.

Валидация качества работы программных реализаций алгоритмов была проведена с помощью эвристической обучающей выборки, представленной в предыдущем подразделе. Модели на вход подается одна из рассматриваемых коллекций ключевых слов. Программная реализация строит все необходимые графы ключевых слов, определяет на их основе автоматизированную обучающую выборку и вычисляет факторы для каждого элемента выборки. Далее запускается процесс обучения модели. После чего происходит процесс применения обученной модели к объектом эвристической обучающей выборки. Последним этапом является вычисление значения метрик, характеризующих качество работы программной реализации.

Кроме того, для рассматриваемых коллекций ключевых слов используются известные модели вычисления семантической близости. В рамках этого эксперимента были выбраны уже представленная ранее модель *Word2Vec*, а также модель *Node2Vec* ([116]). В то время, как Word2Vec использует знания о языке для решения задачи семантической близости, модель Node2Vec определяет близость между вершинами графа. Для этого исследователя определяют векторное представление для вершин графа. Меньшее расстояние между векторами влечет более

[1] https://www.aminer.cn/oag2019

выраженную структурную связь между соответствующими вершинами. Следует также отметить, что модель Node2Vec переносит идеи, заложенные в модель Word2Vec, с текстовых данных на графовые.

Выбор моделей Word2Vec и Node2Vec для сравнения с моделью, представленной в данном разделе, обусловлен следующими соображениями. В первую очередь, было необходимо продемонстрировать высокий уровень определения семантической близости между словами естественного языка. Для этой цели авторская модель сравнивается с зарекомендовавшим себя в индустрии решением - моделью Word2Vec. Кроме этого, поскольку в ходе работы используются графовые методы определения близости, является важным сравнить реализацию модели с известными решениями вычисления близости между вершинами графа. В качестве такого решения выступает модель Node2Vec.

В дополнение к этому, представляет интерес следующее рассуждение. Как было показано ранее в подразделе 2.2.2, разработанные признаки являются функциями от пары вершин графа и, следовательно, каждый признак может представлять некоторую меру близости пары ключевых слов. Подобно этому модель *WordContSim*, с одной стороны, является признаком для модели *WordMLSim*, а с другой, оказывается самодостаточным подходом к определению смысловой близости, как было показано ранее в разделе 2.1. В этой связи важно проверить, что модель, обученная на всем множестве факторов, качественно превосходит любой из признаков, входящих в ее состав. Этот факт докажет целесообразность применения машинного обучения в рамках поставленной задачи.

В следующей далее таблице 16 представлены результаты тестирования программной реализации модели *WordMLSim*.

	Коллекция					
	Aminer		KeywordsRu		KeywordsEn	
Модель	AUC	NDCG	AUC	NDCG	AUC	NDCG
WordMLSim	0.6306	0.1666	0.7718	0.1073	0.8825	0.3336
Word2Vec	0.9475	0.2326	0.8822	0.1402	0.8506	0.3431
Node2Vec	0.6015	0.1658	0.7511	0.1015	0.8254	0.3109
Лучший признак	0.6275	0.1667	0.7549	0.1064	0.8606	0.3286
WordContSim	0.5979	0.1668	0.7279	0.1066	0.8103	0.3272

Таблица 9: Результаты тестирования модели WordMLSim

Наилучший результат, как и ожидалось, достигается применением модели Word2Vec, поскольку только эта модель опирается на заложенные в нее знания о языке. Остальные протестированные модели используют только информацию о коллекции, представленную в виде графа. Таким образом, модель Word2Vec получает неоспоримое преимущество в рамках данного эксперимента. Следует однако отметить, что результаты тестовых испытаний на коллекции *KeywordsEn* демонстрируют конкурентноспособность разработанной модели WordMLSim, даже без наличия каких-либо знаний о языке. По метрике AUC разработанная автором модель превзошла известный в индустрии аналог.

Среди моделей, использующих исключительно знания о тестируемой коллекции, модель WordMLSim превосходит модель Node2Vec. Данный факт свидетельствует о высоком уровне качества разработанной модели определения семантической близости. Кроме этого, WordMLSim показывает лучшие результаты в сравнении как с разработанной в разделе 2.1 моделью WordContSim, так и с отдельными признаками, использованными для обучения. Данное обстоятельство еще раз подтверждает целесообразность использования методов машинного обучения для определения уровня семантической близости.

Отмечается, однако, что для возможности применения представленных моделей в индустрии, необходим эксперимент, демонстрирующий преимущества представленного в данной разделе подхода. Для решения этой задачи был проведен следующий эксперимент.

Тестирование обучения модели с эвристической выборкой. Тестирование по следующему сценарию. Из обучающей выборки, собранной с помощью эвристических методов, была выделена тестовая часть, которая не использовалась в обучении моделей. После этого производилось обучение нескольких моделей. К их числу относятся:

- модель, обученная на эвристической обучающей выборке с использованием графовых признаков (*Graph*);
- модель, обученная на эвристической обучающей выборке с использованием языковых и графовых признаков (*Lang+Graph*);

Далее качество тестировалось на тестовой части обучающей выборки. Результаты приведены в таблице 10

Наилучший результат был достигнут при совместном использовании языковых и графовых признаков. Зафиксированное увеличение значений метрик NDCG и AUC свидетельствует об эффективности разработанных подходов к построению

	Коллекция					
	Aminer		KeywordsRu		KeywordsEn	
Модель	AUC	NDCG	AUC	NDCG	AUC	NDCG
WordMLSim(Lang+Graph)	0.9570	0.2322	0.9502	0.1710	0.8837	0.3540
WordMLSim(Graph)	0.9407	0.2257	0.9210	0.1589	0.8730	0.3515
Word2Vec	0.9512	0.2302	0.8877	0.1412	0.8520	0.3421

Таблица 10: Результаты тестирования модели WordMLSim с эвристической выборкой

модели семантической близости. Кроме того отмечается, что модель, обученная исключительно на графовых признаках показала сопоставимый уровень качества с моделью *Word2Vec*. Этот факт также показывает эффективность разработанной модели в сравнении с известными подходами в индустрии.

В дополнение к эксперименту, описанному выше, были проведены тестовые испытания по определению тождественно равных пар ключевых слов. Описание этого эксперимента приводится далее.

Тестирование тождественно равных пар. Отправным мотивом эксперимента, направленного на выявление тождественно равных пар слов, является тот факт, что самым ближайшим по смыслу словом является оно само. Поэтому необходимо, что модель могла определять, что для слова w среди всех возможных кандидатов, самым ближайшим по смыслу будет само слово w. В исследуемую модель явным образом не закладывается информация о таком свойстве метрики, поскольку среди обучающих примеров не было таких пар, в которых слова не отличались. Это делает эксперимент осмысленным и дает представление об адекватности разработанной модели.

Важность подобного эксперимента для информационно-аналитических систем заключается в том, что такой тест проверяет разработанные реализации модели на адекватность. Если существует большое число слов, семантически более близких к заданному, чем оно само, то это явный повод задуматься о качестве исследуемых моделей. Результаты работы программных реализаций, непроходящих такое тестирование, недопустимо показывать конечным пользователям системы.

Эксперимент был проведен с использованием лучшей из моделей, описанных в предыдущем подразделе. Для каждого из 10000 случайных ключевых слов были подсчитаны меры близости для 1000 других случайных слов (в том числе

близость и до рассматриваемого слова). В более, чем 99.9% случаев наиболее близким для рассматриваемого слово было определено оно само же. Таким образом, модель с высокой степенью уверенности можно считать качественной.

2.3 Выводы

В настоящей главе были подробно описаны разработанные автором модели, алгоритмы и реализующее их программное обеспечение для определения семантической близости пары ключевых слов. Разработанные модели определения семантической близости являются важным результатом деятельности в рамках подготовки настоящей диссертации.

В результате выполненных автором исследований были разработаны два вида моделей семантической близости на основе интеллектуального анализа коллекции ключевых слов: *WordContSim* и *WordMLSim*. Первая из них вычисляет контекстную близость для вершин специального графа, построенного по коллекции. Вторая модель с помощью методов машинного обучения в автоматическом режиме определяет семантическую близость через множество графовых характеристик. Программная реализации этих моделей прошла апробацию в системе ИАС «ИСТИНА».

В ходе тестовых испытаний программных реализаций были получены следующие основные результаты.

- Разработаны и апробированы новые графовые алгоритмы определения уровня семантической близости пары ключевых слов. Тестовые испытания демонстрируют превосходство авторских моделей над известными графовыми моделями семантической близости.
- Разработана модель WordContSim, определяющая семантическую близость через понятие контекстной близости. Высокий уровень качества подтверждается тестовыми испытаниями.
- Разработана модель WordMLSim, улучшающая качество работы известных в индустрии моделей семантической близости. Использованные методы машинного обучения помогают избавиться от необходимости разработки эвристических моделей.

– Разработан и апробирован алгоритм сбора автоматизированной обучающей выборки. Авторский алгоритм позволяет избежать трудоемкого процесса сбора данных для обучения моделей. Модели, обученные на данных автоматизированной выборки, показывают высокий уровень качества. Процесс подготовки модели полностью автоматизирован.

Глава 3. Определение смысловой близости пары наборов ключевых слов

В настоящей главе представлены разработанные автором настоящей работы модели и алгоритмы, позволяющие определять уровень смысловой близости между двумя наборами ключевых слов. С помощью таких алгоритмов можно определять уровень близости пары объектов информационно-аналитической системы, с которыми ассоциированы рассматриваемые наборы ключевых слов. Семантическая близость между наборами ключевых слов определяется с использованием введенных в предыдущей главе методах определения семантической близости между парой ключевых слов.

Семантическая близость, рассматриваемая в рамках данной главы, является важной и востребованной на практике мерой. Она используется для решения различных задач информационного поиска. В качестве примеров таких задач могут выступать:

- поиск экспертов информационно-аналитической системы, схожих по области интересов с заданным;
- определение множества статей и публикаций схожей тематики;
- рекомендации контента для пользователей блоговых систем и социальных сетей.

Для краткости изложения здесь и далее под наборами будут пониматься наборы ключевых слов.

Первый раздел настоящей главы посвящается введению разработанной автором модели семантической близости наборов ключевых слов. В следующих за ним разделах 3.2 и 3.3 приводится описание алгоритма, представляющего такую модель, а также оптимизационные улучшения для сокращения времени работы программной реализации алгоритма. Наличие таких оптимизаций связано с необходимостью многочисленных вызовов функции определения семантической близости пары слов. В свою очередь, каждый такой вызов является достаточно сложной вычислительной операцией.

В заключительной части главы представлены результаты тестовых испытаний программных реализаций алгоритмов, сформулированы выводы по результатам выполненной работы, а также идеи по дальнейшему улучшению качества определения семантической близости наборов ключевых слов.

3.1 Модель определения смысловой близости наборов ключевых слов

Решаемая задача имеет следующую постановку. Дано множество D объектов (текстовых, графических, видео документов или физических объектов другой природы) и множество ключевых слов W. Каждый элемент $d_i \in D$ представлен конечным набором из k_i ассоциированных с ним ключевых слов из множества W : $d_i = (w_{i,1}, w_{i,2}, ..., w_{i,ki})$.

Необходимо разработать функцию близости $TupleSim : 2^W \times 2^W \to [0, 1]$, высокие значения которой означали бы высокий уровень смысловой близости между наборами и, следовательно, соответствующими объектами системы.

К разрабатываемой метрике близости предъявляются следующие требования.

- Программная реализация должна быть способна вычислять значения метрики в режиме реального времени. Значительные задержки между отправленным пользователем запросом и полученным ответом не могут быть слишком долгими, иначе использование такой системы будет трудновыполнимым или неэффективным.
- Необходимо уметь вычислять уровень близости, в том числе, и для наборов, которые не имеют общих слов. Нулевое число общих слов еще не гарантирует отсутствие семантической близости между наборами. Например, слова одного набора могут быть синонимами, транслитерациями или переводами для слов другого набора. Для таких примеров уровень семантической близости должен быть высоким.

Основой для модели определения уровня близости являются модели определения семантической близости пары ключевых слов, описанные в предыдущей главе. В качестве базовых блоков определения близости наборов ключевых слов естественно положить идеи о близости между словами, из которых состоят эти наборы. Формально, согласно принятым в гл.2 определениям, это можно описать как $TupleSim(d_i, d_j) = TupleSim(WordSim(w_{i,1}, w_{j,1}), ..., WordSim(w_{i,ki}, w_{j,kj}))$.

Для вычисления уровня близости используется простое эвристическое соображение. Суть его заключается в следующем. Два набора d_i и d_j обладают большим значением семантической близости, если каждому из слов набора d_i можно сопоставить хотя бы одно семантически близкое слово из набора d_j. Таким образом, исследуемая модель в значительной мере опирается на пословные

модели, описанные в предыдущей главе, перенимая как их достоинства, так и недостатки.

Преимущество такой модели заключается в эффективном использовании информации, пришедшей от моделей пословного уровня. Они, как было описано в гл.2, демонстрируют хороший уровень качества даже в условиях недостатка данных. Основным недостатком является время вычисления меры близости между наборами в рамках такой модели. Причина заключается в том, что расчет уровня пословной близости является трудоемкой задачей. Рассматриваемая модель, в свою очередь, подразумевает многократные вызовы фукнции пословной близости.

В следующих далее разделах 3.2 и 3.3 подробно описываются алгоритмы вычисления семантической близости пары наборов ключевых слов. В то время, как в разделе 3.2 дается описание основных шагов для вычисления семантической близости, раздел 3.3 представляет дополнительный набор эвристических и инженерных идей по оптимизации вычислений. Такие улучшения дают возможность вычислять меру семантической близости более эффективно.

3.2 Алгоритм определения уровня близости пары наборов, основанный на переборе всех пар ключевых слов

Первая версия алгоритма определения семантической близости наборов ключевых слов представлена следующим образом.

- Входные данными алгоритма является пара наборов $d_i = (w_{i,1}, w_{i,2}, ..., w_{i,ki})$, $d_j = (w_{j,1}, w_{j,2}, ..., w_{j,kj})$.
- Для каждой пары ключевых слов $w_{i,p}, w_{j,q}$ с индексами $p \in [1, k_i], q \in [1, k_j]$:
  - вычисляется признаковые характеристики, описанные в 2.2.2;
  - посчитанный массив значений передается предобученной модели XGBoost, описанной в 2.2;
  - происходит предсказание пословной близости моделью;
  - результаты сохраняются в ячейке (p, q) матрицы m_{sim} размером k_i, k_j.
- Из каждой строки матрицы m_{sim} выбирается наибольшее значение.

– По этим наибольшим значениям вычисляется среднее, которое возвращается в качестве меры близости для пары рассматриваемых наборов.

Далее приведены примеры близких наборов ключевых слов, согласно описанной модели. При построении примеров рассматривались наборы, имеющие пустое пересечение по словам. Следующие далее примеры представляют существенно больший интерес.

Первый набор	Второй набор	Значение функции близости
мультиферроики, магнитные структуры, фазовые переходы, магнитоэлектрики, сильные магнитные поля	магниторезонансная томография, гигантское комбинационное рассеяние, суперпарамагнетизм	0.82
мембранные белки, молекулярное моделирование, фоточувствительные белки, ретиналь, молекулярная динамика, membrane proteins, molecular modeling, retinal, molecular dynamics, photosensitive proteins	малые белки теплового шока, универсальный белковый адаптер 14-3-3, фосфорилирование	0.68
социальная теория, социальная философия, понятие общества, философия истории	идея, социальная практика, научно-технический прогресс, наука, парадигма	0.57

мультиферроики, низкоразмерные и фрустрированные магнитные системы, термодинамические и резонансные свойства, зарядовое и орбитальное упорядочение	магниторезонансная томография, гигантское комбинационное рассеяние, суперпарамагнетизм	0.8

Пары наборов ключевых слов, имеющие в своем составе общие слова, представляют существенно меньший интерес, чем пословно различные пары. Причина заключается в том, что определение высокого уровня близости для наборов, имеющих в составе одинаковые слова, является простой задачей. Другими словами, факт наличия общего слова в паре наборов в значительной мере повышает уровень семантической близости. Для определения близости в случае повторяющихся слов можно воспользоваться, например, описанной ранее мерой Жаккара. Напротив, сложной задачей является построение модели, способной определять уровень семантической близости для различающихся пословно наборов.

Представленные выше примеры демонстрируют способность предлагаемой модели выявлять близость между наборами, не имеющими общих слов, что свидетельствует о ее практической ценности. Отмечается, однако, что такой алгоритм не является вычислительно эффективным. Такая его реализация не позволит вычислять уровень близости для большого числа пар наборов. Этот факт, в свою очередь, ставит под сомнение возможность использования программное реализации в существующих информационно-аналитических системах. Способы преодоления описанных выше трудностей представлены в следующем разделе.

3.3 Оптимизированный алгоритм определения близости пары наборов

Описанный в разделе 3.2 алгоритм определения уровня близости пары наборов ключевых слов с полным перебором всех пар ключевых слов по экспертной оценке имеет достаточно высокий уровень качества. Однако, он имеет

и существенный недостаток. Этот недостаток заключается в низком уровне быстродействия его программной реализации.

Существует несколько составляющих алгоритма, которые главным образом влияют на скорость вычисления уровня близости. Первый из них заключается в рассчете попарных уровней близости между словами разных наборов. Эта процедура имеет асимптотическую сложность $O(mn)$, где m, n - размеры наборов. Таким образом, для пары наборов необходимо рассчитать 30-50 уровней близости между словами этих наборов.

Отметим, что значение рассчитанных значений функций близости можно сохранять и переиспользовать. В предствленной выше наивной версии алгоритм опирается на факт необходимости вычисления уровней близостей для всех возможных пар. Однако, представляется логичным следующее предположение: если для слова из первого набора уже найдено близкое слово из второго, то нет необходимости продолжать сравнивать это слово с другими словами второго набора. Например, если в одном наборе присутствует слово «белок», а в другом слово «протеин», то установив факт близости, можно остановиться и не вычислять близость слова «белок» до других слов другого набора («мембранные белки», «молекулярное моделирование», «фоточувствительные белки», «ретиналь»).

Следующим узким местом с точки зрения вычислительной производительности является непосредственно вычисление значений функции близости, которая определена в главе 2. Для вычисления уровня близости необходимо применить обученную модель из раздела 2.2. В наивном варианте алгоритма, описанного в предыдущем разделе, функция применения модели выполняется для каждой пары рассматриваемых слов. Однако, более эффективным является применение модели сразу для всего множества рассматриваемых пар.

На более низком уровне сложность вычисления зависит от скорости вычисления признаков для пары ключевых слов, описанных в 2.2.2. Естественно, что все необходимые графы хранятся в оперативной памяти, их загрузка и построение происходит один раз до вычисления значений функций уровня близости. При этом вычисления различных графовых характеристик (таких как длины путей, степени абстрактностей вершин и другие) выполняются каждый раз для каждой пары ключевых слов. Это значительно замедляет вычислительный процесс. Поэтому была проведена оптимизация вычисления признаков, которая будет описана далее. Исходя из соображений, описанных выше, алгоритм был оптимизирован при помощи следующих перечисленных далее действий.

– В качестве значения близости между одинаковыми словами $WordSim(w, w)$ автоматически ставится уровень близости 1. Как было показано в разделе 2.2.3, с помощью модели вычисления близости пары слов можно правильно определять наивысший уровень близости между тождественно равных слов. Поэтому в целях оптимизации необходимо избегать вызовов функций от одинаковых слов и в этом случае за уровень близости принимать максимально возможное значение.
– Если для слова уже найдено похожее к нему по уровню близости, то вычисления близости до других слов не происходит. Процесс вычисления останавливается, если для данного слова найдено близкое по смыслу слово в другом наборе и уровень близости превышает значение 0.4.
– Выполнение предрассчета значений близости для самых частотных пар ключевых слов. Вычислив заранее необходимые уровни близости для пар самых популярных в системе ключевых слов, можно добиться ускорения вычисления значений функции близости пары наборов ключевых слов.
– Выполнение предрасчета значений некоторых признаков модели машинного обучения.
– Применение модели машинного обучения в момент, когда все признаки для всех необходимых пар посчитаны. Это позволяет оптимально использовать матричные операции при вычислении значений формулы.

Данные средства позволяют значительно уменьшить среднее время выполнения вычисления близости. Тестированию производительности, а также качества определения близости программной реализации представленного выше алгоритма посвящен раздел 3.4.

3.4 Тестовые испытания

В качестве примера, который, с одной стороны, иллюстрирует практическую востребованность представленного алгоритма в информационно-аналитических системах, а с другой, показывает его работоспобоность, рассмотрим тестовые данные, содержащиеся в ИАС «ИСТИНА». Для проведения тестовых испытаний программных реализаций функции определения близости наборов ключевых слов были взяты данные о научных проектах в МГУ,

аккумулированные в ИАС «ИСТИНА». Этот набор данных содержит в себе информацию о выполненных и занесенных в систему научных проектах, а также следующую сопутствующую информацию:

- название проекта;
- название проекта на английском языке (может отсутствовать).
- идентификационный номер проекта;
- короткое текстовое описание проекта (может отсуствовать);
- набор ключевых слов (может отсутствовать);
- руководители участники проекта (заданы идентификационными номерами, может отстутствовать);
- факультет, институт которого выполняет проект (задан идентификационными номерами, может отсутствовать).

Всего в наборе данных присутствует информация о 12350 проектах.

Для тестирования программной реализации моделей определения уровня близости было проведено два эксперимента над рассмотренным набором данных. Кроме этого, в ходе экспериментов было проведено измерение производительности реализаций моделей. Рассмотренная в настоящей главе модель определения близости наборов сравнивается с классической мерой близости Жаккара для наборов и моделью Word2Vec, обученной на большом объеме (12.9 млрд. словоупотреблений) данных в рамках проекта Russian Distributional Thesaurus. Близость наборов с помощью модельи Word2Vec вычислялась по следующему алгоритму:

- для каждого слова каждого набора рассматривались его векторные представления моделью;
- векторное представление набора определяется как среднее по векторам входящим в него слов;
- вычисляется косинусное расстояние между векторами наборов - оно возвращается в качестве близости наборов ключевых слов.

Далее описывается процедура каждого из экспериментов и приводятся результаты тестирования.

Тестирование близости научных проектов по факультетам. Тестирование качества разработанного алгоритма близости проведено с использованием имеющихся данных о выполняемых в вузах и НИИ научных проектах на примере МГУ и основывается на следующей идее. Естественно предположить, что внутри одного факультета, проекты не так сильно отличаются друг от друга

и, следовательно, ключевые слова проектов одного факультета должны быть в среднем более похожи друг на друга, чем слова разных факультетов. Исходя из этих рассуждений, был подготовлен тестовый набор данных. В качестве примеров близких по смыслу наборов ключевых были взяты пары наборов, проекты которых принадлежат одному факультету. Для каждого набора W, для которого определены положительные примеры $\overline{W}_1, ...\overline{W}_k$, случайным образом из множества проектов других факультетов выбирается 1000 проектов. Наборы $\hat{W}_1, ..., \hat{W}_{1000}$, соответсвующие этим проектам, объявляются семантически непохожими на набор W. Таким образом, для набора W присутствует k наборов, объявленных похожими по смыслу к W, и 1000 наборов, объявленных непохожими.

Далее для каждого набора W моделями $TupleSim$, $Word2Vec$ и $Jaccard$ вычисляются меры близости. Но основе правильных ответов и ответов, определенных моделями, для каждого из подходов подсчитывается метрика $ROC - AUC$. Значение этой метрики усредняется по всем W.

Результаты тестирования приведены в следующей далее таблице 13.

Отмечается, что абсолютные значения в данном эксперименте не играют существенной роли: набрать стопроцентный результат и даже близкий к нему не представляется возможным в силу построения тестирущего множества. Дело в том, что в действительности проекты одного факультета не обязаны быть строго похожими друг на друга. Напротив, многие из них могут сильно различаться, но согласно эксперименту, в этом случае они все равно будут объявлены как близкие и если модели факт близости не установят, то значение метрики уменьшится.

Однако, связь между близостью наборов и принадлежностью их к одному факультету присутствует. Разработанная автором настоящей диссертации модель лучше других известных моделей установила эту связь, что является показателем ее качества. Этому факту свидетельствуют результаты тестовых испытаний, приведенные в таблице 13.

Тестирование наборов с общими словами. В качестве дополнительного способа проверки качества было проведено следующее исследование. Как и прежде, в качестве входных данных выступает информация о научных проектах в МГУ, аккумулированная в ИАС «ИСТИНА». Данный эксперимент аналогичен предыдущему, но вместо факта принадлежности пары проектов одному факультету используется факт существования общего ключевого слова у ключевых

наборов двух проектов. Предполагается, что вероятность семантической близости пары наборов, имеющих общее слово, выше, чем случайной пары наборов. Более сильное предположение, что даже если удалить это общее слово, все равно близость между такими наборами должна быть в среднем выше, чем у случайных.

Множества $\overline{W}_1, ...\overline{W}_k$ и $\hat{W}_1, ..., \hat{W}_{1000}$ собирались таким же способом, как в предыдущем эксперименте. Результаты тестовых испытаний приведены в таблице 13

Отмечается, что и в этом испытании разработанная автором модель показала лучший результат. Важно также отметить, что модель $Jaccard$ отстает более значительно от двух других. Причиной этому является то, что удаление общего слова в рассматриваемых парах в подавляющем большинстве случаев (порядка 80%) означает, что больше общих слов в наборах не осталось. По определению меры, близость для таких наборов будет равна нулю. А это значит, что у таких примеров нет возможности сделать метрику качества выше 0.5.

Тестирование производительности программной реализации модели близости. Для тестирования производительности программной реализации модели близости наборов ключевых слов был проведен эксперимент. Необходимость этого эксперимента обусловлена тем фактом, что исследуемая функция близости должна обладать достаточным уровнем быстродействия для возможности ее использования в реальных системах. Для тестирования было измерено время расчета функции близости для трех моделей на обоих качественных экспериментах, описание которых приводится ранее в настоящем разделе. Результаты также приведены в таблице 13.

Модель	Тип тестирования	Метрика ROC-AUC	Время расчета в сек.
Jaccard	Факультеты	0.564	8
Word2Vec	Факультеты	0.672	12
TupleSim	Факультеты	**0.692**	15
Jaccard	Общее слово	0.601	3
Word2Vec	Общее слово	0.720	4
TupleSim	Общее слово	**0.764**	8

Таблица 13: Результаты тестирования

Тип тестирования «Факультеты» соответствует тестированию близости научных проектов по факультетам. В свою очередь, тип «Общее слово» обозначает тестирование наборов с общими словами. По метрике ROC-AUC разработанная автором модель превосходит известные модели определения близости в рамках рассмотренных экспериментов.

Следует отметить, что модель $Word2Vec$ обучена на огромных объемах данных. Время обучения такой модели требует недель или месяцев процессорного времени. Однако это дает возможность более эффективно использовать обученные векторные представления и быстро вычислять функцию близости. Модель $Jaccard$ способна вычислять степень близости более быстро за счет своей простоты. Несмотря на это, разработанная автором модель $TupleSim$ показывает лучшее качество, с относительно небольшим отставанием по времени.

3.5 Выводы

По результатам исследований, проведенных в рамках настоящей главы, были разработаны модели определения уровня семантической близости пары наборов ключевых слов. Для построения таких моделей используются пословные модели семантической близости, подробно описанные в гл.2. Для разработанных моделей представлены алгоритмы и программные реализации этих алгоритмов. Тестовые испытания, проведенные в разделе 3.4, демонстрируют улучшения качества определения уровня семантической близости в сравнении с известными моделями.

Следующим шагом в исследованиях может стать разработка модели представления набора ключевых слов. Суть такого представления заключается в следующем. По аналогии с идеями, заложенными авторами [50] в модель *Word2Vec*, представляется возможность построить векторное представление для каждого набора ключевых слов системы. Близость пары таких векторов будет означать близость соответствующих им наборов ключевых слов.

Существующие модели построения представлений обучаются на данных широкого профиля, что не позволяет передать специфику рассматриваемой интеллектуально-аналитической системы. Другой подход, заключающийся в обучении таких моделей на имеющихся в системе данных, не способен показать

высокий уровень качества, если система не обладает большими объемами накопленной информации. По этим причинам важным в построении новой модели представлений является использование пословных моделей близости, описанных в гл.2. Как и модели, описанные в настоящей главе, новые модели смогут обучаться на небольших объемах данных, эффективно используя модели близости пары слов. В то же время, процесс вычисления семантической близости сведется в этом случае к прозрачной и вычислительно эффективной процедуре определения близости пары векторов. Такой подход полностью разрешит вопрос долгого вычисления функции близости наборов ключевых слов.

Глава 4. Приложения моделей близости ключевых слов

В настоящей главе рассматриваются решения востребованных на практике задач, необходимые для этого модели и алгоритмы, их программные реализации. Представляется авторское решение задачи кластеризации ключевых слов информационной системы. Рассматривается задача поиска экспертов в области, определяемой запросом из заданного набора ключевых слов. Приводится решение задачи определения тематики объекта информационно-аналитической системы. Описывается также процедура использования тезауруса ключевых слов для решения задачи улучшения результатов ранжирования в поисковой составляющей информационно-аналитической системы «ИСТИНА». В заключении главы представлены выводы, а также дальнейшие направления применения разработанных автором программных средств в приложениях, которые диктуются практикой использования большой наукометической системы.

4.1 Модель семантической кластеризации ключевых слов

Механизмы кластеризация ключевых слов любой большой информационно-аналитической системы являются важным аналитическим инструментарием анализа данный. В первую очередь, они важны для систем, не обладающих необходимыми для получения надежного результата объемами данных. В таких системах, как правило, нет достаточного количества информации о ее объектах и о соответствующих этим объектам ключевых словах. Как следствие, значительная доля ключевых слов будет иметь низкие частоты встречаемости в системе, что делает затруднительным любые аналитические процедуры для таких слов. Преодолеть эти трудности можно путем решения задачи кластеризации. При таком подходе появляется возможность разбить все множество ключевых слов на кластера из семантически близких слов. После подобной предобработки можно отождествить исходные слова с соответствующими им кластерами, что предоставит более богатую статистику встречаемости для каждого ключевого слова.

В настоящем разделе описывается разработанный автором подход к решению данной задачи. В первых подразделах описывается новая модель кластеризации ключевых слов информационно-аналитической системы, основанная на мере *контекстной близости*, которая была введена в 2.1.2. С помощью такой меры конструируется граф, именуемый далее *контекстным графом*. После этого используются дополнительные эвристические соображения, позволяющие отбросить наименее значимые связи в этом графе. Такой граф получил название *усеченный контекстный граф*. Далее описывается метод кластеризации этого грава. Заключительная часть раздела посвящена апробации программной реализации разработанной модели и выводам.

Задача кластеризации формулируется следующим образом. Задано множество ключевых слов рассматриваемой системы $W = w_1, \cdots, w_N$. Необходимо построить такое отображение $c : W \to \mathbb{N}$, ставящее в соответствие каждому ключевому слову системы номер кластера, которому это слово принадлежит. Условие, которое накладывается на эту функцию c, заключается в том, что все слова, попавшие в один кластер, должны быть семантически близкими понятиями, а слова из разных кластеров должны семантически различаться.

4.1.1 Модель полного контекстного графа ключевых слов

Ненулевое значение введенной в разделе 2.1.2 величины $C_T(w_i,w_j)$ означает контекстную связь между ключевыми словами. На основе контекстной близости естественным выглядит построение графа, вершинами которого являются ключевые слова коллекции T. Ребро между парами вершин обозначается в случае, если соответсвующая пара имеет ненулевой уровень контекстной близости. Обозначим данный граф за G_{cont}. Таким образом, модель данных, лежащая в основе исследуемого подхода к кластеризации ключевых слов, является графовой моделью. Ключевые слова хранятся в вершинах, а для определения факта существования ребра используется модель контекстной близости, описанная в 2.1.2. Представляется целесообразным использовать данный граф для решения поставленной задачи кластеризации ключевых слов. Однако сразу отметим, что такой граф имеет ряд недостатков.

Несмотря на то, что для значительной части пар ключевых слов значение $C_T(w_i, w_j)$ равно нулю, существует большое число ненулевых связей, что затрудняет их дальнейший анализ с технической точки зрения. Каждая вершина в графе может быть связана с тысячами других вершин. Это обстоятельство значительно увеличивает вычислительную нагрузку на любые используемые графовые алгоритмы.

В дополнении к изложенному выше, низкие значения контекстной близости пары слов зачастую являются ненадежными. Если значение контекстной близости мало, то этот факт означает, что у пары слов мало общих соседей в графе ключевых слов и связь с этими общими соседями имеет небольшой вес. Низкие ненулевые значения контекстной близости могут иметь даже семантически различные пары, поскольку связь в этом случае не является статистически надежной. Такие данные не привносят полезной информации, а только ухудшают качество работы реализованных алгоритмов.

По изложенным причинам возникает необходимость в усечении графа, собранного согласно описанной выше модели. Под усечением понимается удаление ребер, которые представляют наименее качественные и статистически проверенные связи. Описание модели усеченного контекстного графа приводится в следующем далее подразделе.

4.1.2 Модель и алгоритм построения усеченного контекстного графа ключевых слов

В качестве модели данных усеченного контекстного графа принята модель, схожая с введенной в предыдущем подразделе моделью G_{cont}. Основное отличие новой модели состоит в более сильных условиях, определяющих факт существования ребра между парой вершин, что ведет к значительно меньшему числу ребер в графе. Эти условия нацелены на удаление семантически незначащих ребер из графа G_{cont}.

Автором определены следующие эвристические соображения, определяющие модель усеченного контекстного графа.

1. **Для всех ключевых слов w_i фильтруются связи со слишком низким уровнем близости $C_T(w_i,w_j)$.**

Этот шаг необходим для того, чтобы удалить слабые связи и ненадежные связи из рассмотрения и оставить только наилучшие. Следует также отметить, что одного этого соображения недостаточно для качественного удаления лишних ребер. Причиной этому является тот факт, что сами значения близости $C(w_i,w_j)$ не так важны, как порядок, который они задают на множестве соседей вершины w_i. Другими словами, данная задачу фильтрации стоит рассматривать как задачу ранжирования соседей вершины i , а не как задачу классификации ребер (w_i, w_j) на классы полезных и бесполезных ребер.

2. **От оставшихся выбирается некоторая доля связей (например, 20%) с наибольшими значениями** $C_T(w_i,w_j)$ **.**
   Такое условие представляется естественным, потому что слова, которые встречаются со многими другими в одинаковых контекстах, должны иметь больше ребер в графе, чем те слова, которые контекстно близки только с небольшим числом слов.
3. **Количество отобранных связей должно находиться в некоторых рамках (например, не менее 3 и не более 10 соседей на вершину).**
   Верхняя граница является преимущественно техническим ограничением: если было взято 20% от числа всех соседей, но это число оставшихся связей по-прежнему достаточно велико, то хранение таких вершин потребует значительных ресурсов. Нижняя граница берется для того, чтобы вершина, для которой имеется мало кандидатов, получила хотя бы их в качестве ребер. С учетом ограничения из п.1. можно ожидать, что эти связи будут достаточно качественными для дальнейшего анализа.
4. **Общее количество оставшихся связей должно быть ограничено сверху (например, не более 1000000 ребер в графе)**
   Слишком большое число ребер в графе негативно сказывается на вычислительной производительности системы. В то же время, сильное ограничение на максимальное количество ребер не позволяет модели использовать в полной мере полезный сигнал из подлежащих анализу данных.

Граф, ребра которого отфильтрованы по указанным выше правилам, обозначим за G_{cont_trunc} и назовем *усеченным контекстным графом*. Далее приведено более формальное описание алгоритма, реализующего введенную выше модель.

– Входные параметры:

  - пороговое значение C_{thr};
  - минимальное и максимальное число ребер для одной вершины в усеченном графе n_{min}, n_{max};
  - максимальная доля ребер вершины, которая переносится из графа G_{cont} в граф G_{cont_trunc} - $ratio$;
  - максимальная число ребер в графе G_{cont_trunc} - n_{edge_max}.
- Удаляются все ребра (w_i, w_j), для которых значение $C_T(w_i, w_j) < C_{thr}$.
- Пусть $rank_j$ - порядковый номер соседа w_j в отстортированном по убыванию значения $C_T(w_i,w_j)$ списке всех соседей вершины w_i. Тогда, если $rank_j > \max(n_{min}, \min(n_{max},n * ratio))$, то связь (w_i,w_j) должна быть отфильтрована. Здесь n- число соседей вершины w_i в полном контекстном графе.
- Если число ребер, оставшихся в результате фильтрации, превышает n_{edge_max}, то выбирается выбирается n_{edge_max} ребер (w_i, w_j) с наибольшими значениями $C_T(w_i, w_j)$.

Можно отметить, что в описанном выше алгоритме присутствует значительное число параметров, а именно - $C_{thr}, n_{min}, n_{max}, ratio, n_{edge\_max}$. Значение входных параметром преимущественно зависят от специфики исследуемой системы, а также от объема входных данных. Несмотря на это, выбор конкретных значений не является сложной задачей, поколько значения параметров $n_{min}, n_{max}, ratio$ слабо варьируются в зависимости от входных данных. Для построения графа в программной реализацией алгоритма были использованы входные значения параметров $n_{min} = 3, n_{max} = 10, ratio = 0.2$. Отмачается, что данные значения являются адекватным приближением для этих параметров и могут быть использованы при внедрении в другие, отличные от используемой в эксперименте интеллектуальные системы. Значение параметра n_{edge_max} зависит от объема вычислительных ресурсов, имеющихся на поддержание работоспособности системы.

Единственным параметром, требующим дополнительного анализа, является параметр C_{thr}. Причиной является тот факт, что введенная ранее контекстная мера близости не устойчива к изменению входных данных. Это означает, что для разных коллекций ключевых слов T и T' оказываются разными значения контекстной блиости $C_T(w_i, w_j)$ и $C'_T(w_i, w_j)$. Более того, значения, при которых с

высокой долей вероятности можно заключать сильную семантическую связь между парой ключевых слов в разных коллекциях ключевых слова могут значительно разниться.

Для определения оптимальных значений параметров автором разработан описанный далее в разделе 4.1.5 метод, преимущество которого состоит в полной автоматизации процесса настройки.

Отмечается также, что ребра построенного графа могут быть помечены числами и представляется возможным ввести функцию расстояния между вершинами. Например, такой функцией может служить величина $C_T(w_i,w_j)$. В рамках рассматриваемого подхода, ребра графа остаются непомеченными, что существенно понижает сложность и не ухудшает качество разработанных моделей.

В следующих далее подразделах описывается процедура кластеризации построенного усеченного графа G_{cont_trunc}. Реализация этого алгоритма демонстрирует высокий уровень качества в тестовых испытаниях. В большинстве случаев выделенные кластера ключевых слов состоят из попарно близких по смыслу ключевых слов. Данный факт подтверждеются тестовыми испытаниями, описанию которых посвящен подраздел 4.1.5.

4.1.3 Модель кластеризации усеченного контекстного графа

Ребра усеченного контекстного графа, который был введен в предыдущем подразделе, в большей мере показывают семантическую близость между парой вершин, чем ребра полного контекстного графа и, тем более, графа ключевых слов. При этом отметим, что усеченный граф значительно уменьшает вычислительные затраты. Он повышает точность выявленных семантических связей между ключевыми словами. По этой причине рассмотрение длинных путей в графе становится более оправданным, поскольку семантическая близость между парой вершин лучше сохраняется с увеличением расстояния в графе. Поясним данное утверждение на следующем примере.

Предположим, что заданы 2 графа: полный контекстный граф G_{cont} и усеченный контекстный граф G_{cont_trunc}. Для ясности допустим, что пара слов либо

является семантически близкой, либо таковой не является. Другими словами, уберем все возможные «градации» семантической близости и не будем рассматривать случаи, когда пара слов *сильно* или *слабо* связана по смыслу.

Далее предположим, что ребро в графе с некоторой вероятностью отражает факт семантической близости между соответствующими вершинами. Обозначим эти вероятности за p_{cont} и p_{cont_trunc} для графов G_{cont}, G_{cont_trunc}. По построению очевидно, что $p_{cont\_trunc} > p_{cont}$, поскольку из графа G_{cont_trunc} исключены те ребра, которые с меньшей долей уверенности соединяют семантически близкие ключевые слова.

Далее рассмотрим тройку ключевых слов w_i, w_j, w_k графа G_{cont} таких, что в графе присутствуют ребра (v_{w_i}, v_{w_j}) и (v_{w_j}, v_{w_k}), но отсутствует ребро (v_{w_i}, v_{w_k}). Необходимость кластеризации заключается в том, что по имеющимся семантическим связям появляется возможность восстановить отсутствующие связи. Такие отсутствующие связи имеет место быть, например, по причине недостатка исходных данных.

Допустим, что некоторый алгоритм кластеризации отнес слова w_i, w_j, w_k в один и тот же кластер. Это факт означает, что появилась гипотеза о семантической близости ключевых слов w_i, w_k, которой не было изначально. Исходя из определения введенной модели, можно оценить вероятность того, это действительно пара семантически близких слов. С вероятностью p_{cont} семантически близки слова w_i и w_j, с той же вероятностью семантически близкими являются слова w_j и w_k. Тогда вероятность близости пары ключевых слов w_i и w_k можно определить как p^2_{cont}, предположив транзитивность отношения семантической близости в рамках данной модели. Другими словами, для того, чтобы w_i и w_k были семантически близки, достаточно, чтобы были семантически близки и слова пары (w_i, w_j), и слова пары (w_j, w_k). Вероятность одновременного выполнения этих условий равна p^2_{cont}.

Аналогично можно показать, что если в графе существует путь между вершинами w_i, w_k длины n, то есть существует последовательность ребер $(v_{w_i}, v_1), (v_1, v_2), \cdots, (v_{n-3}, v_{n-2}), (v_{n-2}, v_{w_k})$, то при отсутствии других путей вероятность семантической близости пары (w_i, w_k) равна p^n_{cont}. Таким образом, с увеличением расстояния между словами в графе, вероятность семантической близости экспоненциально уменьшается.

Теперь заметим, что все описанные выше рассуждения в той же мере верны для графа G_{cont_trunc}. Однако, в связи с тем, что $p_{cont\_trunc} > p_{cont}$, то скорость убывания вероятности семантической близости с увеличением расстояния в графе

будет ниже в графе G_{cont_trunc}. При этом, если значение p_{cont_trunc} близко к 1, то вероятность семантической близости для вершин, расположенных на расстоянии, например, 3 или 4, в графе будет достаточно высокой. Отмечается, чем больше ребер было удалено при построении графа G_{cont_trunc}, тем сильнее оценка уровня семантической связи между вершинами, ребра которых прошли фильтрацию, и тем больше значение вероятности p_{cont_trunc}.

Следует однако отметить тот факт, что для пары слов (w_i, w_k) в любом из графов может присутствовать множество различных путей. Наличие большего числа путей, очевидно, должно положительно отражаться на вероятности того, что ключевые слова (w_i, w_k) являются семантически похожими. Кроме того, заметим, что по построению графа G_{cont_trunc}, между любой парой вершин в этом графе *не больше* различных путей, чем в графе G_{cont}.

Данное замечание приводит к следующим двум возможным стратегиям поиска неявных семантических связей:

1. использование большого числа коротких путей в графе G_{cont};
2. использование небольшого числа длинных путей в графе G_{cont_trunc}.

Фактически, коротким путем для графа G_{cont} можно считать путь длины 2, а длинным путем для G_{cont_trunc} пути, длины 3 и более.

Использование графа G_{cont_trunc} имеет неоспоримое преимущество. Оно заключается в возможности гибкой настройки структуры этого графа посредством параметров алгоритма построения усеченного контекстного графа. Таким образом, данный факт делает возможным *компромиссную стратегию выявления неявных связей*:

- чем больше ребер осталось в графе G_{cont_trunc}, тем меньше необходимо использовать длинные пути и тем большее число коротких путей, доступных для анализа;
- чем меньше ребер осталось в графе G_{cont_trunc}, тем больше возможность использования длинных путей, но тем меньше их число.

При этом следует отметить, что преимущество графовых подходов к задаче кластеризации состоит в возможности учитывать длинные пути в процессе построения кластеров. По этой причине в основе предложенного автором метода кластеризации лежит известный графовый алгоритм кластеризации, прошедший апробацию в различных практических задачах.

В следующем далее подразделе описывается алгоритм кластеризации, разработанный в рамках представленной модели.

4.1.4 Алгоритм кластеризации усеченного контекстного графа

За основу кластеризующего алгоритма, соответствующего модели из подраздела 4.1.2, взят алгоритм Louvain Modularity [107]. Подход, описанный в этой статье, решает задачу разбиения входного графа на кластеры таким образом, что количество ребер между вершинами одного кластера велико, а количество ребер между вершинами разных кластеров, напротив, мало. Обозначим за плотность подграфа отношение числа ребер к числу вершин этого подграфа. Тогда в рамках данной задачи необходимо максимизировать плотность внутри внутри одного кластера, одновременно с этим минимизируя плотность подграфов, образованных вершинами из разных кластеров.

Качество разбиения принято измерять с помощью так называемой *модулярности разбиения*. Модулярность - скалярная величина, значение которой лежит в интервале $[0, 1]$. В случае графа, ребрам которого проставлены веса, модулярность определяется следующим образом:

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j),$$

где A_{ij} - вес ребра между вершинами i и j, $k_i = \sum_j A_{ij}, m = \frac{1}{2} \sum_{i,j} A_{ij}$, c_i - номер кластера, в котором лежит вершина i, δ- дельта-функция (функция, принимающая значение 1, если $c_i = c_j$ и 0 в другом случае).

Если пара вершин i, j принадлежат одному кластеру, то значение слагаемого в функционале для этой пары будет тем больше, чем выше ее вес и чем меньше эти вершины имеют связей с вершинами других кластеров. Вершины одного кластера, не связанные ребром друг с другом оказывают негативное влияние на значение Q, поскольку вес ребер в этом случае равен нулю. Если вершины располагаются в разных кластерах, то за счет дельта-функции, которая принимает нулевое значение в этом случае, явного воздействия в значение функционала Q пара вершин не оказывает. Термин «явное» при этом означает, что соответсвующее слагаемое функционала Q равно нулю. Однако отмечается, что вершины из разных кластеров уменьшают значение функциона из-за члена $\frac{k_i k_j}{2m}$ в других слагаемых.

Как показано авторами [117], задача максимизации представленного выше функционала является NP-сложной, поэтому для её решения используются аппроксимационный агломеративный алгоритм. Агломеративность означает, что при инициализации такой алгоритм создает отдельный кластер для каждой вершины, а затем объединяет кластера таким образом, чтобы максимально увеличить значение функционала Q. Основные шаги алгоритма описаны далее.

1. Входные параметры алгоритма: взвешенный граф G.
2. Для каждой вершины графа $v_i \in G$ создается свой кластер $c_i = i$.
3. Для каждой вершины v_i и для каждого соседа v_j вершины v_i в графе G выполняется:
   а) временное добавление вершины v_i в кластер вершины v_j;
   б) подсчет изменения оптимизируемого функционала при перемещении вершины v_i в кластер вершины v_j: $\Delta Q_{i,j}$.
4. Окончательное добавление вершины v_i в кластер того соседа, на котором достигается максимальное увеличение значения Q, то есть $c_i := c_k$, где $k = \text{argmax}_j \Delta Q_{i,j}$. Если функционал нельзя увеличить, то есть $\Delta Q_{i,j} < 0$ для всех j, то кластер вершины остается без изменения.
5. Построение нового графа G', вершинами которого являются кластера, а веса ребер отражают связи между кластерами. Вес ребра равен сумме весов всех пар ребер, вершины которых лежат в соответствующих кластерах.
6. Повторение п.2-5 для графа G' до полной сходимости алгоритма.

Преимущество данного алгоритма заключается в его масштабируемости на графы больших размеров. Возможность эффективного пересчета кластеров достигается засчет быстрого пересчета изменения значения функционала качества в п.3.б представленного выше алгоритма:

$$\Delta Q_{i,j} = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right],$$

где $\sum_{in}$ - сумма весов ребер внутри кластера вершины v_j, $\sum_{tot}$ - сумма весов ребер, инцидентных вершинам кластера вершины v_j, k_i - сумма весов ребер, инцидентных вершине v_i, $k_{i,in}$ - сумма весов ребер, инцидентных вершине v_i внутри кластера вершины v_j, m - сумма весов всех ребер в графе.

Следует отметить, что количество кластеров не является параметром данного алгоритма. На практике кластера, получающиеся в результате работы программной реализации алгоритма, оказываются слишком большого размера. В некоторые кластера могут попасть тысячи или десятки тысяч слов. Однако, очевидно, что не существует такого огромного множества попарно похожих по смыслу слов. Алгоритм является общим графовым алгоритмом и никаким образом не использует информацию о семантической близости. Даже точное решение оптимизационной задачи не гарантирует качественного разбиения вершин графа на подмножества, элементы которого семантически близки друг к другу. Как следствие, необходимы дополнительные действия, связывающие процессы кластеризации графа и определения семантической близости. Далее представлен алгоритм кластеризации усеченного контекстного графа, разрешающая отмеченную трудность.

1. Входные данные алгоритма: усеченный контекстный граф G_{cont_trunc}, параметр k - максимальный размер кластера.
2. Создается выходное множество кластеров $Clusters$
3. Создается очередь $Queue$ для подграфов исходного графа.
4. Исходный граф добавляется в нее.
5. Пока очередь $Queue$ не пуста выполняется:
    а) кластеризация подграфа, взятого из очереди, описанным ранее алгоритмом кластеризации;
    б) для каждого полученного в результате кластеризации подграфа-кластера c:
        1) если кластер c содержит меньше, чем k вершин, или c совпадает с кластеризуемым подграфом (не существует разбиения данного подграфа на хотя бы 2 кластера меньшего размера), то c добавляется в выходное множество кластеров $Clusters$;
        2) иначе кластер c добавляется в очередь подграфов.
6. Возвращается множество определенных кластеров $Clusters$.

Данный алгоритм позволяет проводить кластеризацию до тех пор, пока не будут получены кластеры необходимого размера. Следует отметить, что в некоторых ситуациях подграф, к которому применяется алгоритм кластеризации, может иметь размер больший, чем k, но быть при этом оптимальным с точки зрения максимизируемого функционала Q. Такой граф не может быть разделен на несколько

подграфов меньшего размера, поэтому он добавляется ко множеству определенных кластеров, несмотря на свой размер. Примером такого подграфа может быть *клика* (полный подграф в графе), размера большего, чем k.

Резюмируя представленные в данной главе соображения, автором получен алгоритм кластеризации ключевых слов коллекции T, описание которого следует далее.

– Для ключевых слов коллекции T вычисляется степень контекстной близости.
– Выполняется построение контекстного графа G_{cont}.
– Для графа G_{cont} выполняется процедура построения усеченного контекстного графа G_{cont_trunc}.
– С помощью алгоритма кластеризации усеченного контекстного графа, вычисляются необходимые кластеры ключевых слов коллекции.

Таким образом, автором разработан алгоритм кластеризации ключевых слов по входной коллекции T. Данный алгоритм получил название *ContGraphClustering*. Эффективность реализации данного алгоритма, а также качество производимой им кластеризации ключевых слов обосновываются в следующем далее подразделе 4.1.5.

4.1.5 Тестовые испытания

Описанный ранее алгоритм кластеризации контекстного графа позволяет удалять недостаточно надежные связи между словами, полученные с помощью алгоритма определения близости по контекстному графу, и, наоборот, добавлять новые ребра между семантически похожими парами слов.

Рассмотрим результат работы алгоритма на конкретном примере. На рисунке 4.1 изображены несколько ближайших контекстно близких слов для слова «студенты», полученных с помощью модели *WordContSim*. Описание этой модели можно найти в разделе 2.1. Пример со словом «студенты» взят из тестовых испытаний, описанных в 2.1.4. Отметим, как и ранее, что полное множество соседей вершины слишком велико, чтобы его изобразить.

Кластер, полученный в ходе работы программной реализации алгоритма кластеризации для того же слова «студенты» изображен на рисунке 4.2.

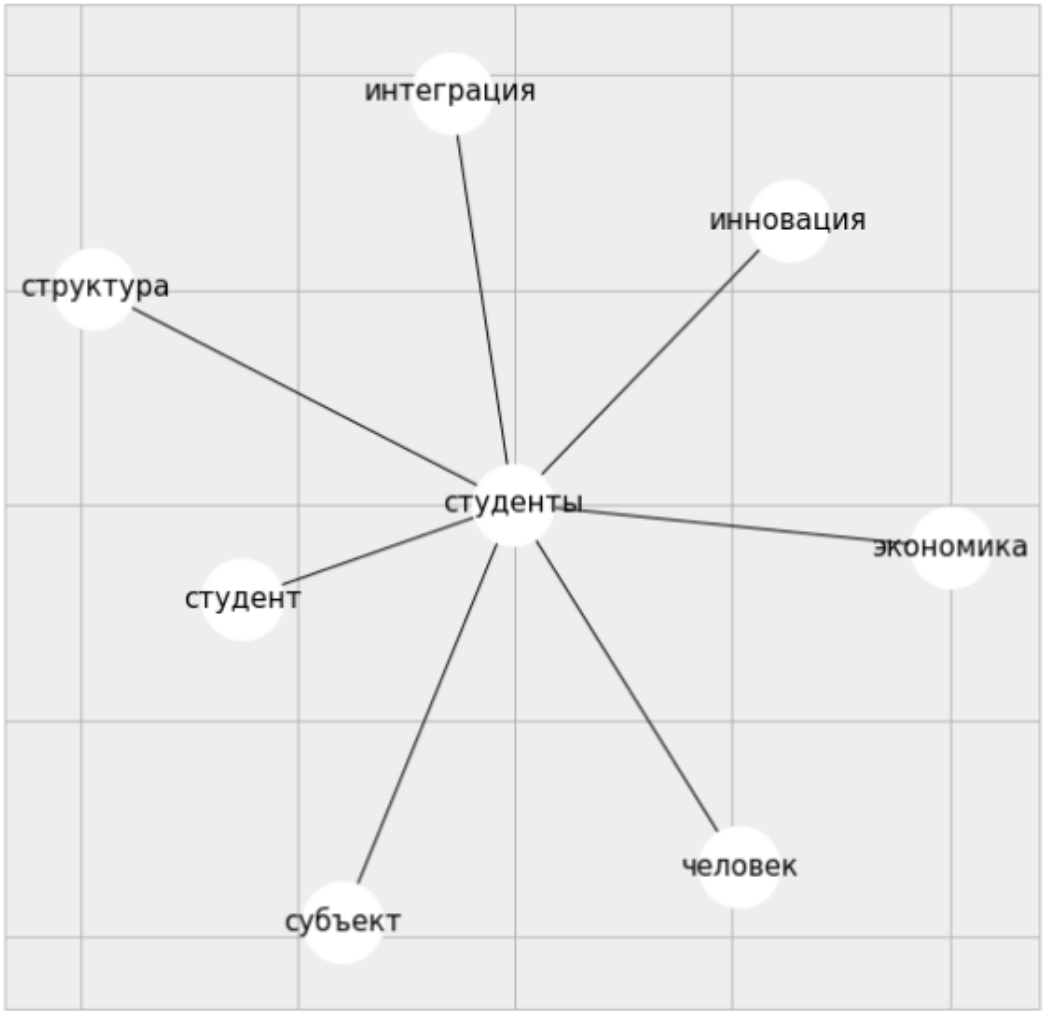


Рисунок 4.1 — наиболее близкие слова для слова «студенты» в контекстном графе

Можно заметить, как в результате кластеризации были разорваны связи (т.е. удалены ребра в графе) «студенты-экономика», «студенты-инновации», вместо которых на первый план вышли связи «студенты-подростки». Отмечается также, что выбранный метод кластеризации графа может допускать ошибки. Например, в случае с парой «студенты-структура», которая была как в усеченном контекстном графе, так и в кластере слова студент.

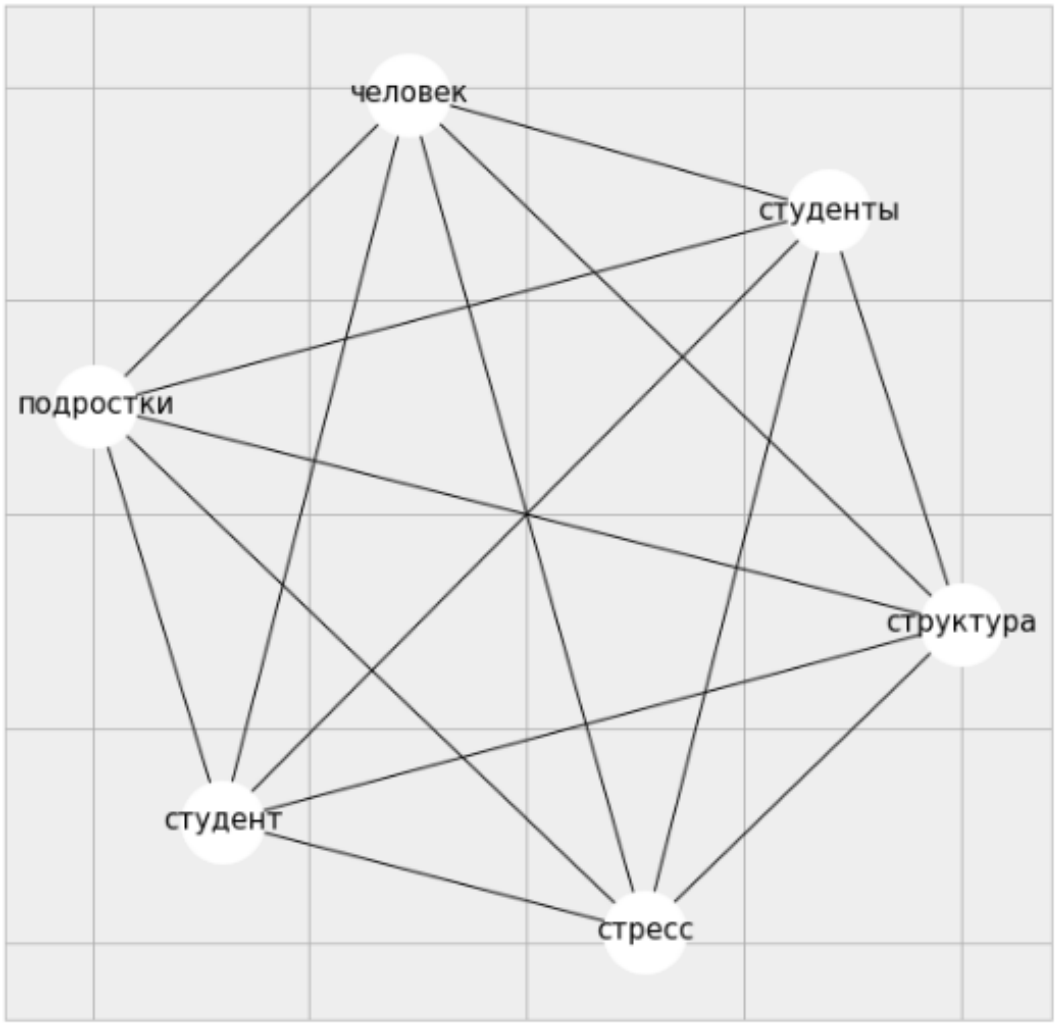


Рисунок 4.2 — кластер, содержащий слово «студенты»

Для интегральной проверки качества выбрана процедура, аналогичная описанной в разделе 2.1.4. Однако, для тестирования программной реализации алгоритма кластеризации была внесены модификации для учета специфики данной задачи. Причина ее необходимости заключается в том, что ранее алгоритм генерации тестовых примеров создавал тестовые пары ключевых слов. Однако, в рамках задачи кластеризации необходимо иметь возможность проверки реализаций алгоритмов на тестовых *кластерах* - множествах семантически близких ключевых слов.

Для достижения этой цели разработана следующая программа эксперимента. Из множества всех ключевых слов W выделяется случайное множество слов W_{test}, на котором впоследствии будут протестированы модели. Как и ранее, последовательно просматриваются наборы t коллекции T и каждое ключевое слово $w \in t$. Если текущее слово является тестовым, то есть $w \in W_{test}$, то это слово заменяется на синтетически созданное слово w^{*n*}, где «$*$» - спецсимвол, а n - число, взятое случайно из геометрического распределения $Geom(p)$, где p - вероятность успеха в бернуллиевском эксперименте. Наличие спецсимвола в слове делает его уникальным в рамках коллекции T.

Таким образом, коллекция T преобразуется в коллекцию T^*. Эта коллекция содержит для слов $w \in W_{test}$ одно или несколько слов вида w^{*n*}. Это множество слов определяет один тестовый кластер. Кластер для данного слова w может содержать от одного до $f_T(w)$ слов, где $f_T(w)$, как и ранее, частота встречаемости слова w в коллекции. Кластеры, содержащие слишком малое количество объектов, удаляются. В программной реализации алгоритма удаляются кластеры, содержащие менее, чем 4 слова. За множество W^* обозначим множество ключевых слов созданной коллекции T^*. Множество ключевых слов из W^*, попавших в тестовые кластера, обозначим $W_{clusters}$. Перенумеровав полученные тестовые кластеры, определим отображение $C_{test} : W_{clusters} \to \mathbb{N}$, которое ставит слову из $W_{clusters}$ номер его кластера.

В отличие от решения задачи определения семантической близости пар слов, данная задача не требует определения отрицательных тестовых примеров. Для тестирования достаточно указать множество тестовых кластеров. Задачей алгоритмов кластеризации в этом случае является определение такого отображения $C : W_{clusters} \to \mathbb{N}$, которое определяет кластер для каждого тестового слова. После этого с помощью метрик кластеризации определяется наиболее качественный алгоритм. Более точные алгоритмы должны построить отображение C совпадающее с отображением C_{test} с точностью до перенумерации кластеров. Другими словами, важно, чтобы слова из одного тестового кластера были помещены алгоритмом в один кластер, а слова из разных тестовых кластеров - в разные. Метрики, выбранные для проверки качества, описаны далее. В рамках данного подраздела будем называть кластеризацией отображение, которое ставит словам из $W_{clusters}$ номера кластеров.

– **Adjusted Rand Index (ARI).** Пусть a - количество пар ключевых слов из $W_{clusters}$, которым присвоен один и тот же номер тестовой кластеризацией

C_{test}, а также один и тот же номер испытуемым алгоритмом кластеризации C.

Аналогично, пусть b - количество пар, слова из которых присвоены различные кластера как в C_{test}, так и в C.

Величина $RI = \frac{a+b}{C^2_{|W_{clusters}|}}$ - нескорректированная метрика $RandIndex$. Чем больше значение данной метрики, тем более правильной оказалась кластеризация, полученная испытуемым алгоритмом. Для того, чтобы значения метрики было близко к нулю в случае случайной кластеризации (кластеризации таким алгоритмом, который ставит каждому слову случайный номер кластера), вычисляется следующее скорректированное значение метрики:

$$ARI = \frac{RI - E[\hat{RI}]}{max(\hat{RI}) - E[\hat{RI}]},$$

где $E[\hat{RI}], max(\hat{RI})$ - математическое ожидание и максимум случайной величины RI, определенной алгоритмом случайной кластеризации.

– **V-Measure.** Для подсчета этой меры вычисляются две вспомогательные величины.

1. Однородность (Homogeneity):

$$h = 1 - \frac{H(C|C_{test})}{H(C)}.$$

2. Полнота (Completeness):

$$c = 1 - \frac{H(C_{test}|C)}{H(C_{test})}.$$

В данных формулах используется понятие условной энтропии, которая для пары кластеризаций C_1, C_2 задается следующим образом:

$$H(C_1|C_2) = -\sum_{c_1=1}^{|C_1|}\sum_{c_2=1}^{|C_2|}\frac{n_{c_1,c_2}}{n}\log\left(\frac{n_{c_1,c_2}}{n_{c_2}}\right),$$

а также понятие энтропии для кластеризации C, которая вычисляется по следующей формуле:

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n}\log\left(\frac{n_c}{n}\right).$$

В двух последних формулах $n = |W_{clusters}|$, n_{c_1}, n_{c_2}, n_c - количество слов, попавших в кластер с номером c_1, c_2, c соответственно, n_{c_1,c_2} - количество слов, попавших в кластер c_1 согласно кластеризации C_1 и в кластер c_2 согласно кластеризации C_2.

Максимальное значение однородности показывает, что в кластера, определенные тестируемым алгоритмом, состоят из объектов одного тестового кластера. В свою очередь, максимальное значение полноты говорит о том, что все объекты одного тестового класса отнесены проверяемым алгоритмом к одному кластеру.

V-Measure определяется как геометрическое среднее между значениями однородности и полноты:

$$v = 2 \cdot \frac{h \cdot c}{h + c}.$$

Полный алгоритм подготовки данных для тестирования изложен далее.

1. Входные параметры алгоритма:
   - исходная коллекция ключевых слов T;
   - предполагаемое количество кластеров - $N_{clusters}$;
   - вероятность успеха бернуллиевского эксперимента в геометрической прогрессии $Geom(p)$ - p;
   - минимальный размер кластера s.
2. Для слов w коллекции T подсчитываются частоты встречаемости $f_T(w)$;
3. С учетом частотностей слов случайным образом выделяется множество из $N_{clusters}$ ключевых слов из T. Учет частотностей означает, что каждое слово w может быть взять в данную выборку с вероятностью $\frac{f_T(w)}{\sum_{w_i \in W} f_T(w_i)}$. Таким образом фиксируется множество W_{test};
4. Формируется модифицированная коллекция T^*:
   - цикл по всем наборам $t \in T$;
     - цикл по всем ключевым словам $w \in t$ набора:
       * если $w \in W_{test}$, то выбрать случайное число k из геометрического распределения $Geom(p)$ произвести замену $w \to w^{*k*}$;
       * иначе оставить исходную версию слова.
5. Формируется тестовые кластеры семантически идентичных пар ключевых слов:

- цикл по всем уникальным словам w множества ключевых слов W_{test}:
  - если для w в коллекции T^* присутствует хотя бы s *различных* слов вида w^{*i*}, то из всех возможных слов вида w^{*i*} формируется новый тестовый кластер и добавляется в выходное множество кластеров.

6. Возвращается модифицированная коллекцию ключевых слов, множество кластеров для тестовых слов.

Таким образом, в результате работы программной реализации алгоритма происходит создание тестовых кластеров, каждый из которых состоит из различных синтетических вариаций написания одного слова. Под синтетическими вариациями следует понимать дописывание к слову специальных символов и некоторого числа, что не меняет смысловую составляющую слова, а меняет лишь форму написания этого слова. Очевидно, что все такие вариации должны попадать в один и тот же кластер, поскольку требуется, чтобы кластера состояли из семантически близких понятий.

Помимо алгоритма, разработанного автором настоящей диссертации, для тестовых испытаний были использованы известные алгоритмы кластеризации, прошедшую апробацию на многих востребованных практикой задачах. Для апробации были выбраны алгоритмы кластеризации DBSCAN([118]) и K-means([119]). Для применения данных алгоритмов необходимо уметь рассчитывать расстояние между объектами. В качестве расстояния может быть использована величина, обратная мере контекстной близости. Такой подход позволяет проверить качество работы непосредственно кластеризующей части алгоритма. Кроме того, в качестве расстояния было использовано евклидово расстояние в пространстве, построенном с помощью модели Word2Vec([50]). Такой подход позволяет проверить одновременно и качество работы алгоритма построения усеченного контекстного графа, и качество работы кластеризации этого графа.

Необходимо также отметить, что качество работы алгоритмов зависит от параметров, с которыми были запущены их программные реализации. В этой связи реализован тестирующий программный комплекс, который запускается на одних и тех же тестовых данных и настраивает параметры таким образом, чтобы максимизировать значение той или иной метрики качества. Параметры, необходимые алгоритмам, можно разделить на два описанных далее типа. Первый из них - это

параметры *модели пространства*, в рамках которого считается расстояние между ключевыми словами. В ходе экспериментов рассмотрены две различные модели пространства: модель векторного представления слов *Word2Vec*, а также разработанная автором модель, основанная на построении усеченного контекстного графа, который описан в подразделе 4.1.2. Построение этого графа, в свою очередь, опирается на меру контекстной близости *WordContSim*, описание которой представлено в подразделе 2.1.2.

Второй тип параметров - параметры кластеризующего алгоритма. Для алгоритма *K-Means* параметром является требуемое количество кластеров, для алгоритма *DBSCAN* - максимально возможное расстояние между парой точек, при котором они еще считаются соседями, а также минимальное количество точек, необходимое для создания ядра кластеризации. Подробно параметры описываются в [118]. Алгоритм кластеризации, предложенный автором настоящей диссертации, имеет только один параметр - максимальный размер кластера. Во всех экспериментах этот размер зафиксирован и положен равным 10.

Эти параметры необходимы для построения усеченного контекстного графа и описываются в подразделе 4.1.2. Непосредственно алгоритм кластеризации графа имеет только один параметр - максимальный размер кластера. Во всех экспериментах этот размер указан равным 10. Таким образом, общая схема тестирования выглядит следующим образом.

- По коллекции ключевых слов T строится коллекция T^*, фиксируются тестовые кластеры.
- Для каждого набора параметров $dist\_params$ модели пространства, в котором считается расстояние между объектами, выполняется:
  - построение модели пространства D с параметрами $dist\_params$;
  - для каждого набора параметров $cluster\_params$ алгоритма кластеризации выполняется:
    - определение кластеров для тестовых объектов с помощью алгоритма кластеризации с параметрами $cluster\_params$;
    - вычисление метрик качества кластеризации для тестовых объектов.
- Из рассмотренных наборов параметров $dist\_params$, $cluster\_params$ выбираются те из них, которые демонстрируют наилучший уровень качества по выбранным метрикам.

Описанный выше алгоритм позволяет сравнить пару подходов к решению задачи кластеризации. Кроме того, с помощью данного алгоритма для заданной тестовой выборки в автоматическом режиме определяется оптимальный с точки зрения метрик качества набор параметров. По этой причине применение данного алгоритма не ограничивается лишь задачей апробации и тестирования. Такой подход позволяет определять необходимые параметры кластеризации для программных реализаций алгоритмов кластеризации, внедряемых в реальные системы. Следует также отметить, что в рамках эксперимента учитываются такие разбиения множества ключевых слов, которые порождают не более 100000 кластеров.

Наилучшие результаты работы программных реализаций алгоритмов по метрикам *Adjusted Rand Index* и *V-measure* приводятся таблице 14.

Модель пространства	Алгоритм кластеризации	V-Measure	ARI
Word2Vec	DBSCAN	0.4280	0.0766
Усеченный контекстный граф	DBSCAN	0.5438	0.1225
Усеченный контекстный граф	**ContGraphClustering**	**0.5869**	**0.1496**

Таблица 14 — Результаты тестирования алгоритмов кластеризации

В результате тестовых испытаний программных реализаций алгоритмов было определено, что разработанный автором алгоритм демонстрирует лучшее качество кластеризации на коллекциях ключевых слов научных публикаций. Важно также отметить, что усеченный контекстный граф является лучшей моделью пространства, чем модель *Word2Vec*, поскольку использование этого графа позволяет повысить уровень качества для фиксированного алгоритма кластеризации. Кроме того отмечается, что наилучшие значения по различным метрикам для одного алгоритма достигаются на различных значениях входных параметров. Другими словами, оптимальные параметры для максимизации метрики *V-Measure* не являются таковыми для метрики *ARI*.

Таким образом, в рамках данного раздела была решена задача кластеризации ключевых слов системы. Тестовые испытания подтверждают эффективность разработанного алгоритма. Кроме того, в данном подразделе описан метод подбора необходимых алгоритму значений параметров. Этот факт значительно упрощает этап внедрения реализованного модуля кластеризации в существующие информационно-аналитические системы. В следующем далее разделе представлено решение другой востребованной практикой задачи поиска тематической

направленности объекта информационной системы по набору ключевых слов. Для ее решения также используются результаты, описанные в предыдущих главах.

4.2 Определение тематической направленности объекта информационной системы по набору ключевых слов

В данном разделе описывается решение задачи определения тематической направленности набора ключевых слов и соответствующего ему объекта информационно-аналитической системы. Под тематической направленностью или тематикой следует понимать название некоторой области знаний, дисциплины, специальности или направления.

Необходимость в решении задачи определения тематики документа или объекта возникает в информационно-аналитических системах. Например, умение определить тематическую направленность научной статьи по набору ключевых слов позволяет в автоматическом режиме построить тематический рубрикатор системы. Такой рубрикатор полезен в задачах улучшения качества поискового модуля внутри рассматриваемой информационно-аналитической системы.

Перед формальной постановкой задачи введем важное в рамках данного раздела понятие *степени абстрактности ключевого слова* (далее для краткости - абстрактности). Под абстрактностью ключевого слова понимается степень общности значения этого слова. Подробное описание авторского алгоритма вычисления меры абстрактности представлено в следующем далее подразделе 4.2.1.

Другим необходимым понятием раздела является понятие *тематического ключевого слова*. Тематическое ключевое слово - это такое ключевое слово, которое с высокой степенью достоверности определяет тематическую направленность или предметную область объекта. Для наукометрической системы тематическими ключевыми словами могут являться, например, названия дисциплин: *вычислительная математика, теория чисел, теория вероятности*. С одной стороны эти слова не являются слишком абстрактными и представляется возможным определить предметную область документа по ним. С другой стороны, смысл этих слов не слишком специфичен. Эти слова понятны любому пользователю системы, а не только узкопрофильному специалисту.

Умение определять тематические ключевые слова полезно в том числе и тем, что позволяет строить автоматические классификаторы и рубрикаторы для заданной системы.

Определение тематических ключевых слов напрямую связано с понятием абстрактности. В наборах ключевых слов присутствуют слова, которые указывают на общую тематическую направленность документа, а также слова, отражающие более узкую специализацию (термины). Проиллюстрируем это на примере наукометрической системы и следующих наборов ключевых слов, ассоциированных с научными публикациями (подчёркнуты слова более общего значения):

[рынок труда, профессиональная ориентация, прогнозирование, модель, алгоритм]

[внеаудиторная работа, учебный проект, целеустремленность, творчество]

[компетентностный подход, система компетенций, межпредметная интеграция, моделирование, математика, естественно-научные дисциплины]

По ключевым словам, выделенным в каждом из наборов, трудной является процедура, позволяющая понять специфику научной работы, соответствующей этим ключевым словам. Такие слова, как *математика, естественно-научные дисциплины* могут определить тематическую направленность документа, но конкретику добавляют остальные слова (не подчеркнутые слова наборов). Таким образом, для дальнейшего определения тематической направленности объекта системы необходимо уметь отличать слова более общего значения от узкоспециальных по смыслу слов.

Все множество ключевых слов можно разбить на перечисленные далее три уровня в зависимости от степени абстрактности этих слов.

- **Термины**. Наиболее узкоспециальные по смыслу ключевые слова. Примерами терминов являются: *цитовир-3, титаносиликаты, двухфазные струи*.
- **Тематические ключевые слова**. Ключевые слова, определяющие тематическую направленность документа. Примерами таких слов являются: *психология,физика, педагогическая деятельность*.
- **Абстрактные ключевые слова**. Слова, наиболее абстрактные по смыслу. Примеры абстрактных слов - *задача, моделирование, концепт, система, технология, методология*.

Дадим теперь формальное описание задачи определения тематической направленности объекта. Пусть, как и ранее, дано множество объектов D информационно-аналитической системы и множество ключевых слов W. Каждый объект $d_i \in D$ представлен набором из k_i ключевых слов из множества $W : di = (w_{i_1},w_{i_2},...,w_{i_{ki}})$. Подмножество тематических ключевых обозначим за $T \subset W$. Необходимо разработать метод определения тематики для каждого объекта коллекции. Тематикой объекта d_i назовем подмножество тематических ключевых слов $T_i \subset T$, которое определяет предметную область этого объекта. Таким образом, задача заключается в определении для каждого объекта коллекции соответствующего ему множества тематических ключевых слов, т.е. в построении отображения $r_{theme} : D \to 2^T$. Определив такое отображение для объекта, можно с высокой вероятностью сказать, о чем этот объект.

Для решения поставленной задачи использована введенная в подразделе 4.2.1 модель определения степени абстрактности ключевого слова, а также представленная в подразделе 4.2.2 модель, определяющая тематические ключевые слова в коллекции. В дополнении к этим моделям, для достижения лучших результатов задействована графовая модель данных, введенная в 2.1.1. Кратко общий подход к решению задачи можно описать следующим образом:

1. по коллекции наборов ключевых строится графовое представление данных;
2. для каждого ключевого слова вычисляется степень его абстрактности;
3. по степени абстрактности определяется является ли данное ключевое слово тематическим или нет;
4. тематическая направленность набора ключевых слов вычисляется по тематическим ключевым словам, входящим в данный набор (либо по тематически словам, семантически близким к словам из данного набора).

В следующем далее подразделе представлено описание алгоритма, определяющего степень абстрактости ключевого слова по набору ключевых слов согласно предложенной модели.

4.2.1 Определение степени абстрактности слова

Настоящий подраздел посвящен понятию степени абстрактности ключевого слова. Под абстрактностью ключевого слова понимается степень общности значения этого слова. Абстрактность будет использована в дальнейшем для решения задачи определения тематики набора ключевых слов, поставленной в подразделе 4.2.2. Кроме того, она нашла применение в модели семантической близости пары ключевых слов, описанной в разделе 2.2.

Задача, авторское решение которой представлено в настоящем подразделе, формулируется следующим образом. Рассматривается модель информационно-аналитической системы, введенная в начале 2. По коллекции ключевых слов T и множества W ключевых слов данной коллекции необходимо разработать такую меру абстрактности

$$a_T : W \to \mathbb{R} + ,$$

в соответствии с которой бо̀льшим значениям меры соответствовали слова более широкого значения, а меньшим - слова более конкретного, обладающего определённой спецификой значения.

Следует однако отметить, что абстрактность ключевого слова зависит от коллекции, по которой она считается. Например, ключевое слово *морфология* обладает достаточно высоким уровнем абстрактности относительно других слов этой системы и может считаться тематическим ключевым словом. Причиной этому является то обстоятельство, что в наукометрических системах присутствует большое множество различных научных работ, посвященных некоторым вопросам из области морфологии. Соответственно, большое число ключевых слов-терминов этого направления, имеют прямое отношение к морфологии. Однако, рассматривая не связанную с наукометрией аналитическую систему общего назначения, абстрактность ключевого слова *морфология* уменьшается. Причина в том, что для такой системы названия дисциплин, как и любые связанные с наукой ключевые слова, являются узкопрофильными по смыслу словами.

Таким образом, для решения данной задачи не имеет смысла пользоваться внешними по отношению к рассматриваемой системе данными: они не могут отразить специфику конкретной области. Это обстоятельство усложняет как

процесс нахождения решения задачи, так и процесс апробации программных реализаций разработанных моделей.

Кроме того, дополнительная трудность решения поставленной задачи обусловлена тем обстоятельством, что необходимо уметь отделять действительно абстрактные по значению слова от слов популярных, и, как следствие, часто используемых в наборах ключевых слов.

Определение степени абстрактности слов на основе свойства центральности. В теории графов существует характеристика важности вершины графа, которая по-русски именуется центральность. Для того, чтобы определить влияние вершины внутри графа, существует несколько различных характеристик. Далее вводятся основные из них.

- Центральность по посредничеству (betweenness centrality) — характеристика, которая определяется через количество кратчайших путей в графе, проходящих между всеми парами вершин в графе через данную вершину. Подсчет величины данной характеристики производится с помощью формулы: $bc(i) = \sum_{s,t \in V \wedge s \neq i \wedge t \neq i} \frac{n^i_{s,t}}{n_{s,t}}$, где $n_{s,t}$ - количество кратчайших путей через вершины s и t. $n^i_{s,t}$ - количество кратчайших путей через вершины s и t, проходящих через i, V — множество вершин графа.
- Центральность по близости (closeness centrality) - мера, основанная на средней длине кратчайшего пути между исходной вершиной и всеми другими вершинами графа. $c_i = \frac{k}{\sum_{j \in V_i} dist(i,j)}$, где $dist(i,j)$ - длина кратчайшего пути между вершинами i и j. k- количество вершин из компоненты связности вершины i. V_i - множество вершин из этой компоненты связности.
- Центральность по степени (degree centrality) - мера, в которой важность вершины равна её степени (числу инцидентных ребер).
- Центральность собственного вектора (eigen vector centrality) - мера, описанная в [120], вычисляется по формуле $x(i) = \frac{1}{\lambda} \sum_{j \in V} a(i,j)x(j)$, где a - матрица смежности графа, λ - константа. Это выражение быть быть переписано в векторной форме следующим образом: $Ax = \lambda x$. Большее значение меры ставится той вершине, которая соединена с большим числом вершин, меры которых высоки.
- PageRank-центральность (PageRank centrality) - алгоритм, используемый Google для ранжирования страниц (метод определения важности или популярности документов) [59]. Согласно этому алгоритму, веб-страница

имеет больший вес, если на неё много ссылок из других веб-страниц, также имеющих большой вес. Заметим, что в данном виде алгоритм применим не только для веб-страниц, но и для любых графов. Общая идея этого алгоритма близка к идее, которая реализуется алгоритмом вычисления eigen vector centrality. Однако в уравнениях, которые используются алгоритмом PageRank, вместо собственных значений присутствует коэффициент затухания q. Физический смысл этого коэффициента в том, что пользователь имеет некоторую вероятность перехода с одной веб-страницы на другую по ссылке. При этом, он может никуда не переходить и закончить случайное блуждание по сети. Таким образом, q — вероятность того, что пользователь перейдёт по ссылке. Получаемое уравнение имеет следующий вид: $PageRank(i) = (1 - q) + q \sum_{j \in V_i} \frac{PageRank(j)}{C_j}$, где C_j - количество ссылок в документе j. В программной реализации параметр q определен как 0.9.

Автором в его исследованиях, направленных на решение задачи определения степени абстрактности, использвалась идея, основанная на том, что слишком длинный путь между вершинами в графе обычно не означает наличия хотя бы какой-то связи между словами. По этой причине программная реализация алгоритмов Betweenness Centrality и Closeness Centrality автором выполнена таким образом, чтобы длинные пути не учитывались. Для этого соответствующие меры для вершины считаются не во всем графе, а в подграфе соседей на расстоянии 3 от вершины. Для каждого из алгоритмов определения центральности вершины, описанных выше, за меру абстрактности ключевого слова принимается значение, вычисленное реализацией этого алгоритма. Результаты работы программной реализации алгоритмов представлены в приложении Б.

В качестве окончательного алгоритма, решающего задачу определения степени абстрактности ключевого слова, автором принят алгоритм, объединяющий идеи описанных выше алгоритмов определения центральности вершины. Следует однако отметить, что указанные алгоритмы возвращают значения различных порядков. Этот факт означает, что, например, центральность по степени возвращает целое неотрицательное число, которое не ограничено сверху. В то же время значения центральности по посредничеству лежат в интервале $[0, 1]$. По этой причине, для того чтобы использовать все значения различных мер центральности, необходимо их предварительно единообразно отнормировать.

Нормировка значений каждого алгоритма центральности происходит следующим образом:

- алгоритмом определения центральности вершины вычисляются значения центральности для всех уникальных ключевых слов коллекции;
- полученные значения хранятся в массиве $Centrality(i)$, где i - порядковый номер ключевого слова в коллекции;
- для массивов значений центральности вычисляются две величины:
    1. минимум значений $MIN = \min\{Centrality(i)|w_i \in W\}$;
    2. минимум значений $MAX = \min\{Centrality(i)|w_i \in W\}$;
- исходные значения центральности нормируются по следующей формуле: $NormCentrality(i) = \frac{Centrality(i)-MIN}{MAX-MIN}$.

Преимущества выбранного метода нормировки состоит в том, что массивы значений для каждого алгоритма центральности теперь лежат в отрезке $[0, 1]$, независимо от того, в каком диапазоне они были до проведения данной процедуры. Изначальный порядок значений центральностей для каждого отдельного алгоритма при этом сохраняется. Финальным этапом вычисления степени абстрактности является усреднение всех нормированных оценок для каждого слова коллекции T:

$$a_T(w_i) = \frac{\sum_j NormCentrality_j(i)}{n_{alg}},$$

где $NormCentrality_j$ - нормированное значение центральности $NormCentrality$ для j-го алгоритма определения центральности, n_{alg} - общее числов используемых алгоритмов определения центральности. Для вычисления по принятой формуле используются все описанные выше алгоритмы, определяющие степень центральности вершин.

Резюмируя представленные выше соображения, необходимо отметить, что степень абстрактности слова является достаточно субъективной величиной. По этой причине проведение объективной количественной оценки результатов работы алгоритмов на больших объёмах данных затруднительно. В качестве основного метода проверки адекватности предлагаемого метода используется выборочная экспертная оценка результатов.

4.2.2 Алгоритм определения тематических ключевых слов

Ранее была описана модель определения степени абстрактности ключевого слова w внутри коллекции ключевых слов T. Кроме этого, во вводной части этого раздела ставилась задача по определенному уровню абстрактности уметь выделять так называемые *тематические ключевые слова* - такие ключевые слова аналитической системы, которые определяют тематическую направленность или предметную область документа системы. Ранее было показано, что решение этой задачи напрямую зависит от решения задачи об определении степени абстрактности ключевого слова. Зная значения абстрактностей для ключевых слов системы, задача сводится к задаче определения для слова с данным уровнем абстрактности одной из трех выделенных категорий: ключевые слова-термины; тематические ключевы слова; абстрактные ключевые слова.

Таким образом, необходимо построить функцию: $f_{theme} : \mathbb{R} \rightarrow A$, отображающую степень абстрактности во множество категорий ключевых слов по общности значений $A = \{$*Ключевые слова-термины, тематические ключевые слова, абстрактные слова*$\}$ Множество A здесь и далее будем называть классами абстрактности.

Трудность решения поставленной задачи заключается в том, чтобы определить границы, где заканчивается один класс и начинается другой. Необходимо подчеркнуть, что выбор этих границ субъективен: ключевое слово «динамика ударных волн» является, с одной стороны, названием некоторой области научного знания, а с другой стороны, по объективным причинам складывается представление, что это ключевое слово имеет достаточно узкоспециальный характер. Помечать такое ключевое слово тематическим или нет, зависит от конкретной решаемой задачи.

Очевидным является тот факт, что подавляющее большинство ключевых слов должно являться терминами, а меньше всего должно быть абстрактных ключевых слов. При этом термины, в отличие от абстрактных слов, редко повторяются. Для определения степени абстрактности слова будет использоваться алгоритм, описанный в разделе 4.2.1.

Для решения поставленной задачи рассмотрим типичные наборы ключевых слов. Далее приведены примеры наборов, содержащих тематическое ключевое слово (предполагаемое ключевое слово подчеркнуто):

[экологические процессы, социальная история, природные ресурсы]

[беспризорность, политика государства, детский дом, адаптация]

[экономика, модернизация экономики, подготовка кадров, система повышения квалификации]

[удар, динамика, летательный аппарат, экран, модель, физическое моделирование]

Идея разработанного алгоритма заключается в следующем. Рассматривается набор ключевых слов $t \in T$ коллекции T. Для каждого слова $w \in t$ рассматривается значение уровня абстрактности. Далее слова набора необходимо разделить в соответствии с тремя выделенными категориями: ключевые слова-термины; тематические ключевые слова; абстрактные ключевые слова. Предполагая, что в каждом наборе присутствуют представители всех трех видов слов, происходит процесс одномерной кластеризации множества слов по значениям уровней абстрактностей на три множества.

В качестве алгоритма кластеризации был выбран алгоритм *KMeans* ([119]). Этот алгоритм позволяет задать число требуемых кластеров, что является необходимым, поскольку необходимо разбить множество значений ровно на три категории.

Поскольку задача кластеризации в данном случае одномерна, центр каждого кластера задается единственным действительным числом. Таким образом, для каждого набора ключевых слов с помощью кластеризации определяется 3 числа: меньшее из них определяет центр кластера ключевых слов-терминов; среднее - кластер тематических ключевых слов; наибольшее - кластер абстрактных слов. Центры кластеров для всей коллекции T получаются усреднением соответсвующих значений центров для всех наборов $t \in T$.

Следует заметить, что зачастую наборы представляются без тематических ключевых слов. Можно предположить, что авторы соответствующих статей считают такие ключевые слова бесполезными для определения их содержания. Если человек хочет прочитать статью о схемах Рунге-Кутта, то он должен и без явного указания понимать, что речь идёт о численных методах.

Для того, чтобы снизить влияние наборов, состоящих целиком из ключевых слов-терминов, до начала кластеризации исходная коллекция подвергается фильтрации, а именно - из рассмотрения удаляются те наборы t, для которых уровни абстрактности всех слов лежат ниже уровня 90-го перцентиля всех значений абстрактностей. Другими словами, рассматриваются только те наборы, в которых

есть хотя бы одно ключевое слово со значением уровня абстрактности, входящим в первые 10% среди всех посчитанных значений для слов коллекции. Обозначим получившуюся таким образом подколлекцию ключевых слов за $T_{90} \subset T$.

Финальным этапом в процессе определения тематических ключевых слов является выбор таких слов, уровни абстрактности которых наиболее приближены к соответсвующему центру. Количество тематических ключевых слов зависит от специфики коллекции, на которой проводятся расчеты, и оно является параметром алгоритма.

Важно отметить, что лучшего качества определения тематических ключевых слов можно добиться, если кластеризовать не сами значения абстрактностей, а величину $\frac{a_T(w)}{1-a_T(w)}$. Такое преобразование увеличивает разницу между парами значений, лежащих внутри отрезка $[0, 1]$. Однако, более значительно разница увеличивается для пар с большими значениями. По этой причине в результате преобразования абстрактные ключевые слова станут находиться относительно дальше друг от друга, а ключевые слова-термины - ближе. Данное обстоятельство приводит к тому, что кластер ключевых слов терминов будет иметь большее число слов, а кластер абстрактных ключевых слов, напротив, становится более разреженным. Это, в свою очередь, соответсвует распределению ключевых слов-терминов и абстрактных слов в реальном мире.

Исходя из соображений, представленных выше, был разработан следующий алгоритм классификации ключевых слов по степени общности значения.

1. Входные параметры: n - количество необходимых тематических ключевых слов, коллекция ключевых слов T, значения уровня абстрактности $a_T(w)$ для слов коллекции T.
2. Создается пустое множество $theme\_centers$.
3. Подсчитывается 90-ый перцентиль $a_{T,90}$ по множеству всех значений абстрактностей $a_T(w)$.
4. Создается подколлекция $T_{90} \subset T$, содержащая наборы t, в которых множество $\{w|a_T(w) > a_{T,90}, w \in t\}$ не пусто.
5. Для каждого набора ключевых слов $t \in T_{90}$, содержащего более, чем n_{thr} ключевых слов, выполняется:
   - рассматривается множество значений преобразованных абстрактностей для слов набора t: $A_{odd,T,t} = \{\frac{A_T(w)}{1-A_T(w)}|w \in t\}$;
   - для множества $A_{odd,T,t}$ решается задача одномерной кластеризации алгоритмом *KMeans* с количеством кластеров $n_{clusters} = 3$;

- из трех полученных центров кластеров выделяется медианный по значению $cluster_{median}$;
- значение центра кластера $cluster_{median}$ добавляется в множество A_{theme}.

6. Вычисляется среднее значения элементов множества $theme\_centers$: $A_{theme\_avg} = E(x|x \in theme\_centers)$.
7. Для каждого слова коллекции вычисляется его удаленность от среднего центров тематических кластеров: $A_{dist,T}(w) = |A_t(w) - A_{thema\_avg}|$.
8. Множество n наиболее близких ключевых слов определяются как тематические ключевые слова.
9. Каждое из оставшихся слов w помечается ключевым словом-термином, если $A_T(w) < A_{theme\_avg}$, или абстрактным ключевым словом, если $A_T(w) > Atheme\_avg$.

Таким образом, представленный алгоритм позволяет классифицировать каждое ключевое слово системы и определить его как ключевое слово-терми, тематическое ключевое слово или абстрактное ключевое слово. Алгоритм основывается на понятии степени абстрактности ключевого слова, которая была введена ранее. В следующей разделе приводятся результаты тестовых испытаний программной реализации данного алгоритма.

В следующем подразделе представлено описание алгоритма, определяющего тематику объекта по набору ключевых слов согласно предложенной модели.

4.2.3 Алгоритм выбора тематики объекта

В разделе 4.2.1 представлена модель классификации ключевых слов на W на три уровня в зависимости от степени абстрактности этих слов. С ее помощью для каждое ключевое слово систему можно отнести к *абстрактным ключевым словам*, *тематическим ключевым словам* или *ключевым словам-терминам*. Множество тематических ключевых слов представляет наибольший интерес, поскольку именно среди них находятся слова, определяющие тематическую направленность объекта. По этой причине для ключевых слов набора вычисляются степени абстрактности и определяются классы абстрактности, к которым принадлежат слова.

Если в наборе присутствует несколько тематических ключевых слов, то в качестве ответа выдается наиболее абстрактное из них. Следует однако отметить, что во многих наборах тематические ключевые слова отсутствуют. В этом случае для определения тематики объекта используется граф ключевых слов G_{kw}, введенный в разделе 2.1.1. Алгоритм проходит по всем словам набора, начиная с наиболее абстрактного слова и заканчивая словом, обладающим минимальной величиной абстрактности. Для каждого такого слова просматриваются все его соседи в графе ключевых слова G_{kw}. Если среди них присутствует хотя бы одно тематическое ключевое слово, то все они возвращается в качестве ответа. В другом случае алгоритм переходит к следующему слову исходного набора.

В том случае, если ни слова исходного набора, ни соседи этих слов в графе G_{kw} не являются тематическими, алгоритм возвращает пустой ответ. Отмечается также, что поиск тематических соседей на расстоянии 2 и более от заданной вершине в графе G_{kw} не приводит к улучшению качества решения задачи. Дело в том, что исходя из соображений из раздела 2.1.1, семантическая связь между парой вершин очень быстро уменьшается при увеличении расстояния между этими вершинами.

В целях оптимизации вычислений, для множества ключевых слов $w \in W$ коллекции предварительно вычисляются значения степени абстрактности $a(w)$, а также классы абстрактности $b_w = f_{theme}(a(w))$.

Резюмируя представленные выше соображения, алгоритм выбора тематики представляется в следующем виде.

1. Входные параметры алгоритма: набор ключевых слов $t \in T$.
2. Ключевые слова набора t сортируются по убыванию уровня абстрактности.
3. Цикл по словам w набора t:
   - если $b_w =$ *тематическое ключевое слово*, то вернуть его в качестве результата;
   - иначе, продолжить цикл.
4. Цикл по словам w набора t:
   - по графу G_{kw} определяются множество соседей $N_G(w)$ для слова w;
   - если существует такое слово v, что $v \in N_G(w)$ и $a_b =$ *тематическое ключевое слово*, то вернуть все такие слова v;
   - иначе, продолжить цикл.

5. Вернуть пустое множество

Вычислительные затраты на работу программной реализации такого алгоритма оказыва.тся равными $O(nm)$, где n - максимальное число слов в наборе, а m - максимальное число соседей для вершины в графе G_{kw}. Первый множитель этой оценки получен в следствии необходимости просмотра всех слов заданного набора. Второй множитель появляется из-за просмотра всех соседей в графе ключевых слов G_{kw}.

В следующем разделе 4.2.4 демонстрируются результаты работы программной реализации алгоритма на реальных наборах ключевых слов.

4.2.4 Тестовые испытания

В настоящем подразделе результаты тестовых испытаний программной реализации алгоритмов определения степени абстрактности ключевого слова, тематических ключевых слов и тематики документа, а также описываются методы предварительной обработки исходных данных и исправления ошибок. В качестве таких тестовых данных в работе использован корпус ключевых слов научных публикаций. Коллекция представляла собой вручную составленные списки ключевых слов для публикаций технического и гуманитарного профиля, полученные из различных источников, включая сеть Интернет. По этой причине в этих данных присутствовали ошибки и неточности, полученные при их формировании.

По причине того, что данные для тестовых испытаний вводились людьми вручную, было необходимо провести предварительную обработку данных. Предобработка данных - необходимая мера для увеличения точности работы алгоритмов. К мерам, которые применялись для улучшения качества тестовых данных относятся следующие:

- все ключевые слова переводились в нижний регистр;
- самые популярные из аббревиатур вручную сопоставлялись со своими развёрнутыми формами;
- использовалось несколько разделительных символов;
- длинные строки без разделителей разделялись по символам пробелов;
- в ключевом слове убирался дефис, если уже существует такое же слово без дефиса.

При этом замечается, что длинные строки без разделителей в действительности могут представлять собой единственное ключевое слово:

[оценка экономического косвенного эффекта от проекта информатизации].

Или же набор ключевых слов, разделенных по пробелу

[шахта метан утилизация газогенераторная станция].

По причине того, что длинные ключевые слова встречаются в данных пренебрежимо мало и в соответствующих вершинах построенных графов мало связей, такие длинные одиночные ключевые слова расценивались как множество однословных ключевых слов.

Далее приведен список самых популярных ключевых слов:

наночастицы, инновации, метод конечных элементов, механические свойства, динамика, наноструктуры, прочность, научный потенциал, удар, структура, остаточные напряжения, компьютерное моделирование, управление, модель, оптимизация, мониторинг, образование, математическое моделирование, математическая модель, моделирование

Популярность некоторых из этих слов является следствием высокой степени абстрактности (к примеру, слово «моделирование»). Однако некоторые из них попали в список, потому что некоторая тема может быть популярной (по крайней мере, в рамках рассматриваемой коллекции). По этой причине ключевые слова, относящиеся к этой тематике, могут быть часто использованы в списке ключевых слов (наноструктуры, метод конечных элементов). Однако, по уже представленным ранее данным видно, что наивный алгоритм подсчёта количества вхождений ключевого слова в коллекцию не даёт необходимого результата.

Далее на Рис. 4.3 показано количество наборов, обладающих соответствующей долей неуникальных ключевых слов (ключевых слов, которые упоминаются хотя бы в двух наборах из коллекции). Другими словами, для каждого набора подсчитан процент неуникальных ключевых слов и по этим данным построена гистограмма. Важно отметить, что именно неуникальные ключевые слова дают возможность дальнейшего анализа. Если бы наборы в основном состояли из уникальных ключевых слов, то использование графов и статистического анализа не привело бы к достижению каких-либо значимых результатов. Самое популярное значение доли неуникальных ключевых слов - 2.0. Этот факт означает, что ключевые слова всех наборов этой категории не являются уникальными, несмотря

на то, что коллекция состоит по большей части из уникальных ключевых слов. Примером такого набора является:

[образование, наука, высшая школа, идеология, математика, филология, история, педагогика, биология].

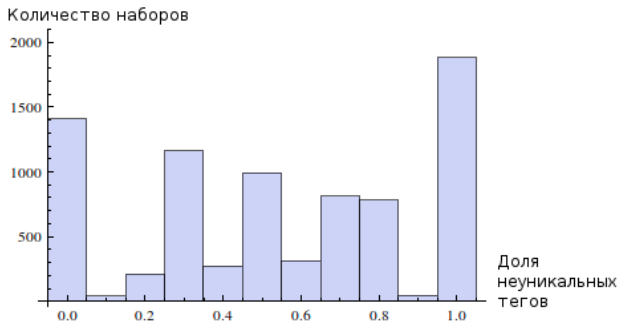


Рисунок 4.3 — Кол-во наборов, с соответсвующей долей неуникальных ключевых слов

Зачастую ключевые слова наборов такого типа являются очень абстрактными понятиями, что в редких случаях позволяет понять тематику документа. Второе по популярности значение - 0.0. Оно показывает, что существует много наборов, целиком состоящих из уникальных ключевых слов. Причины появления таких наборов состоят в следующем:

– ключевые слова являются слишком узкоспециализированными, например - **фемтосекундная спектроскопия, лазерные солитоны, дипирролилметаны, подповерхностный радиолокатор**;
– ключевые слова представлены на другом языке - **control, sensitivity, equilibria, chaos**.

Следует однако отметить, что основная причина именно в узкой специализации ключевых слов.

Между значениями 0.0 и 1.0 находится более половины всех наборов. Средняя доля неуникальных ключевых слов по всей коллекции равна 0.53, то есть, в среднем половина ключевых слов набора встречается в некотором другом объекте коллекции. Типичный набор состоит из нескольких абстрактных ключевых слов, указывающих на общее направление работы и дисциплины, и нескольких ключевых слов, позволяющих понять, о чем конкретно представленный документ.

Исходя из перечисленных выше факторов, представляется возможным изучать методы автоматического определения тематики документа и особенности алгоритмов ассоциативного поиска. При этом логичным инструментарием для

решения поставленных задач являются графы, которые были введены в предыдущем разделе. Такие графы будут иметь достаточную связность для дальнейшего анализа и возможности применения алгоритмов, которые представлены в предыдущих разделах.

Для графа ключевых слов, построенного по данным, были определены компоненты связности и вычислены их размеры. Общее число компонент связности - 1856, что, очевидно, очень много для графа из 17428 вершин. Однако, как показывает график на рис.4.4 зависимости номера компоненты и её размера (ось ординат логарифмическая), наибольшая компонента связности содержит более половины всех ключевых слов (11558), а вторая по величине имеет лишь 92 вершины. Начиная с 38 позиции, в компонентах содержится менее 10 вершин.

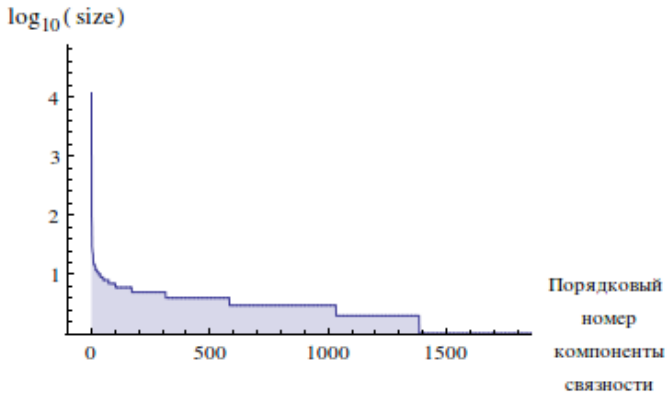


Рисунок 4.4 — Распределение размеров компонент связности

Дополнительный тестовый набор данных По причине того, что имеющийся набор данных недостаточно велик, был разработан алгоритм сбора информации о ключевых словах научных статей из сети Интернет. Для этого был использован API поисковой системы Яндекс, представляющей функциональные возможности получения поисковой выдачи по запросу. Суть алгоритма в следующем.

На вход алгоритма подается одно ключевое слово <KEYWORD>. В поисковую систему отправляется запрос вида «mime:pdf keywords: /5 <KEYWORD> ». Этот запрос означает, что необходимо найти документы формата pdf, в которых после слова «keywords:» на расстоянии не более 5 слов находится заданное слово <KEYWORD>. Поисковая система возвращает сниппеты релевантных документов. Ожидается, что значительная часть сниппетов будет содержать список ключевых слов некоторых научных статей. Производится парсинг сниппетов, выделяются кортежи ключевых слов. Собранные ключевые слова добавляются в

множество ключевых слов. Ключевое слово-запрос добавляется в список использованных ключевых слов. Новое ключевое слово <KEYWORD> выбирается из разности множества всех ключевых слов и множества использованных. Действия алгоритма повторяются до тех пор, пока не будет собрана база ключевых слов достаточного размера.

Недостатком программной реализации представленного алгоритма является то обстоятельство, что внешняя поисковая система ограничивает количество запросов в день. По этой причине сбор необходимых данных занимает продолжительное время. Чтобы ускорить процесс, для каждого запроса выкачивается максимально возможное число документов. Как следствие, ключевые слова, по которым строился запрос, встречаются гораздо чаще в собранном множестве ключевых слов. Такое смещение в данных негативно влияет на статистические параметры выборки и ухудшает качество работы реализаций алгоритмов. Другой недостаток состоит в том, что если брать слишком большое число документов, то хвост выдачи становится менее релевантен и в выборку добавляются «мусорные» данные.

Тем не менее, алгоритм решает важную задачу - восполняет недостаток данных. С помощью программной реализации было собрано более 380000 наборов ключевых слов. Для них были проведены методы предобработки данных, описанные в предыдущем пункте. Кроме того, удалялись наборы без разделителей. Вероятнее всего, такие наборы - это обычные предложения со словом keywords, а также наборы, в которых ключевые слова имеют слишком малую длину (обычно в такие сниппеты попадали инициалы авторов статей). Далее это множество данных обозначается как данные из Веб, а первая коллекция именуется чистыми данными.

Результаты тестовых испытаний модели определения абстрактности слова.

Самые абстрактные слова, полученные при использования программной реализации описанного выше алгоритма на чистых данных, следующие:

моделирование, модель, образование, оптимизация, управление, структура, математическая модель, математическое моделирование, мониторинг, прогнозирование, инновации, эффективность, методика, личность, прочность, эксперимент, оценка, история, методы, развитие, анализ, здоровье, инновационная деятельность, культура, качество, свойства, модернизация, синтез, надежность, самоорганизация, адаптация, конкурентоспособность,

интеграция, студенты, безопасность, компетенции, взаимодействие, технологии, диагностика, наука, государство, компьютерное моделирование, инновационное развитие, устойчивость, компетентностный подход, динамика, технология, высшая школа, нано- частицы, метод конечных элементов.

В целом получены неплохие по оценке квалифицированных экспертов результаты. Из выделенных слов значительную часть можно назвать абстрактными в некотором смысле. Смешивание помогло избавиться от явных выбросов в каждом из алгоритмов и несколько усреднить результат. Тем не менее, добиться заметного повышения качества за счет таких действий не удалось, поскольку представленные алгоритмы имеют одну природу и решают схожие задачи. По этой причине они зачастую ошибаются на некоторых данных одновременно, что влечет за собой ошибку в результатах работы общего алгоритма.

Результаты работы программной реализации алгоритма на дополнительном наборе данных из Веб, описание которого приведено ранее, следующие:

development, data mining, environment, evaluation, model, management, machine learning, modelling, growth, reliability, neural networks, design, stability, learning, security, uncertainty, clustering, education, performance, modeling, optimization, simulation.

Для этого набора получены схожие по качеству результату. Однако можно заметить, что некоторые слова определяются абстрактными по причине того, что они были использованы в поисковый запросах, с помощью которых был собран дополнительный тестовый набор данных. Таким словом, например, является термин «machine learning» , который не должен был войти в множество абстрактных слов.

Результаты тестовых испытаний модели определения тематических ключевых слов. Для тестирования было выбрано 65 ключевых слов на чистых данных. Ключевые слова, которые имеют степень абстрактности в отрезке $[0.012, 0.013]$, определены как тематические. В приложении В приводится полный список определенных тематических ключевых слов. Таким образом, алгоритм определил 65 ключевых слов, из которых 19 при любых обстоятельствах являются тематическими потому, что это название дисциплин и направлений науки. Еще 13 ключевых слов субъективно можно считать тематическими. Таким образом точность результата составляет 49.2%.

Такой уровень точности можно считать удовлетворительным, принимая во внимание тот факт, что разработанная модель не использует никаких априорных

знаний о тематике ключевых слов. Другими словами, в ходе построения модели не были использованы внешние источники данных, такие как тезаурусы, содержащие в себе информацию о степенях абстрактности слов и их тематической направленности. Это обстоятельство позволяет утверждать об универсальности исследуемого подхода с точки зрения применимости к произвольным наборам данных. Становится возможным определение тематических ключевых слов для коллекций различной природы (наукометрия, данные социальных сетей и пр.), предметной направленности (технические, гуманитарные науки) и степени специализированности (узкопрофильные системы и системы общего назначения).

Результаты тестовых испытаний модели определения тематики документа. Тестирование программное реализации алгоритма проводилось на данных, описанных ранее. Некоторые из результатов приведены в таблице 15.

В первом примере показано верное определение тематики. При этом можно заметить отсутствие тематического ключевого слова в исходном наборе. Последние два примера демонстрируют случаи, когда тематическое ключевое слово не определено или определено слишком большое число таких слов.

Отмечается также, что рассмотренные примеры подбирались из условия, что в самом наборе отсутствуют тематические ключевые слова. Такая мера делает тестовое испытание более содержательным, поскольку в этом случае требуется найти наиболее подходящие тематические ключевые слова во всей рассматриваемой коллекции слов. По экспертным оценкам получен адекватный уровень качества работы программной реализации алгоритмов.

В следующем далее подразделе сформулированы основные выводы и предложены идеи для дальнейшего развития данного подхода.

Набор ключевых слов	Возможная тематика набора
выпуклое программирование, принцип лагранжа, теорема куна-таккера в недифференциальной форме, параметрическая задача, минимизирующая последовательность, двойственность, регуляризация	оптимальное управление
состояние метакультуры, культуры среды, границы, этика творчества, личность	биометрия, массовая культура, педагогическая деятельность
окружающая среда, биосферная совместимость, система жизнеобеспечения	гидродинамика, факторный анализ
персонализированное обучение, подготовка учителя информатики, синергетический подход	конструирование, педагогическая деятельность
жидкие кристаллы, ориентирующие слои, аморфный гидрогенизированный улерод	жидкие кристаллы
прогноз, формы предвидения, интерпретация темпоральных модусов, аксиология, социальное противоречие	конструирование, массовая культура
металлические материалы, термическая обработка, структура, свойства	конструирование, физическое моделирование
инклюзия, инвалидность, сопровождение, студенты-инвалиды	массовая культура
модуль выпуклости, равномерно выпуклая функция, равномерно выпуклое множество	[не определено]
урбанизированная территория, гуманитарный баланс, корреляция регрессия математическая модель	биометрия, кинематика, конструирование, нелинейные колебания, физическое моделирование

Таблица 15 — Наборы и предсказанные им тематические направления

4.2.5 Выводы

В данном представлено описание моделей, алгоритмов, а также реализующих их программных комплексов, решающих задачу определения тематической направленности объекта информационно-аналитической системы.

В целом, следует отметить, что алгоритм достаточно часто ошибается. Во многих случаях ошибки возникают на наборах, состоящих только из редких слов-терминов. Причина в том, что слова в подобных наборах не соединены напрямую с тематическими, а увеличение пути сильно уменьшает качество результатов. Не определяется тематика и у тех наборов, ключевые слова которых попали в малые компоненты связности без тематических ключевы слов. Однако основная трудность - это накопление ошибок разных алгоритмов. Сначала слово ошибочно получило высокий уровень абстрактности, затем попал в список тематических и, в итоге, неправильно определена тематика набора.

Несмотря на отмеченные недостатки, предложенный подход потенциально может быть улучшен. Для этого можно использовать все ключевые слова из набора для определения тематики, проверять абстрактности вершин на пути от данного ключевого слова к тематическому и использовать многие другие эвристики. Однако уже сейчас можно констатировать, что алгоритм способен с некоторой адекватной точностью решать поставленную задачу.

4.3 Решение задачи поиска экспертов

В данном разделе представлено описание моделей, алгоритмов и программных комплексов, решающих востребованную практикой задачу поиска эксперта. В рамках этой задачи необходимо по запросу, состоящему из набора ключевых слов, находить наиболее релевантные объекты информационно-аналитической системы, наилучшим образом удовлетворяющие исходному запросу. Задача поиска эксперта является важной для наукометрических систем, поскольку позволяют существенно улучшить качество поиска необходимой пользователю информации. Модели и их программные реализации, описанные в настоящем разделе, прошли апробацию в системе ИАС «ИСТИНА».

4.3.1 Постановка задачи

Рассмотрим математически формализованную постановку задачи. Дано множество экспертов системы E и множество ключевых слов W. Каждый эксперт $e_i \in E$ представлен набором ключевых слов $t_i \in T$, состоящим из k_i ключевых слов из множества W. наборов ключевых слов W_X и объектов информационной системы X, а также множество Q запросов к системе. Обозначим за W множество всех уникальных ключевых слов из всех наборов W_X. Каждый элемент $x_i \in X$ множества объектов ассоциирован с набором ключевых слов $W_i = \{w_{i_0}, w_{i_1}, ..., w_{i_{n_i}}\} \in W_X \in 2^W$. Точно также каждый запрос $q_j \in Q$ связан с некоторым набором ключевых слов $W_j = \{w_{j_0}, w_{j_1}, ..., w_{j_{n_j}}\} \in 2^W$. Необходимо определить меру близости пары запрособъект для каждого объекта и каждого запроса, т.е. функцию $f : Q \times X \to R$. Поскольку запросам и объектам единственным образом сопоставляются наборы ключевых слов, то задача сводится к определению меры близости на наборах: $f_w : 2^W \times 2^W \to R$. Кроме того, необходимо разработать эффективный алгоритм, который, используя меру близости и некоторые дополнительные идеи, мог бы по запросу выдавать множество объектов, наиболее релевантных данному запросу.

4.3.2 Процедура поиска экспертов

По данному множеству наборов ключевых слов (множеству экспертов) строится граф ключевых слов. Далее необходимо для каждого ключевого слова x найти ближайшие по смыслу слова. Мера близости слов вычисляется сначала между ключевым словом x и его соседями в графе, после чего просматриваются соседи соседей x и так до тех пор, пока не наберется фиксированное число кандидат. Часть наиболее релевантных ключевых слов сохраняются, как наиболее близкие к x. В дополнение к этому, строится инвертированный индекс, который позволяет по слову восстановить наборы, содержащие это слово. После того, как в систему приходит запрос, для каждого слова из запроса выгружаются ближайшие по смыслу слова и первоначальный запрос расширяется. Затем по словам из

расширенного запроса восстанавливаются наборыкандидаты. Для каждого из них считается мера близости с исходным запросом $TupleSim_{expert}$:

$$TupleSim_{expert}(X,Y) = \frac{\sum_{i=1}^{|X|}\sum_{j=1}^{|Y|}\sigma(WordMLSim(X_i,Y_j))}{|X\bigcup Y|},$$

где $|\cdot|$ количество слов в наборе, X_i, Y_j - i-ое и j-ое ключевые слова наборов X, Y соответственно, *WordMLSim* - мера близости, введенная ранее в 2.2, а $\sigma(x) = \frac{1}{1+\exp(-x)}$ - сигмоидальная функция. Числитель формулы $TupleSim_{expert}$ аккумулирует близость всех пар слов из разных наборов. Сигмоидальное преобразование позволяет заключить значения близости, возвращаемое функцией *WordMLSim*, внутри интервала $[0, 1]$. Наиболее близкие по смыслу слова будут при этом иметь значение, близкое к единице. Если положить $WordMLSim(x, y) = \mathbb{1}_{\{x=y\}}$, то числитель будет равен числу общих ключевых слов, что приведет к более простой модели вычисления близости по мере Жаккара. Без нормировки длинные пары наборов были бы сильнее похожи друг на друга, чем короткие пары. В конце своей работы алгоритм возвращает наиболее релевантные наборы-кандидаты.

4.4 Построение тезауруса ключевых слов по коллекции наборов

В данном разделе описывается построение тезауруса ключевых слов по коллекции наборов ключевых слов. Схема построения построения приводится на рисунке 4.5.

В качестве внешних данных используется коллекция наборов ключевых слов научных публикаций, собранная из сети Веб. Это коллекция, насчитывающая сотни тысяч наборов ключевых слов для русскоязычных статей и миллиона наборов на английском языке. С помощью этих наборов поочередно строится несколько графов близости. Одним из таких графов является, например, граф, в вершинах которого находятся ключевые слова, а в ребро означает причастность двух ключевых слов одному набору. Вес ребра тем больше, чем чаще пара слов встречается в одном наборе. Другим примером контекстного семантического графа является граф, в вершинах которого как и прежде стоят ключевые слова, а

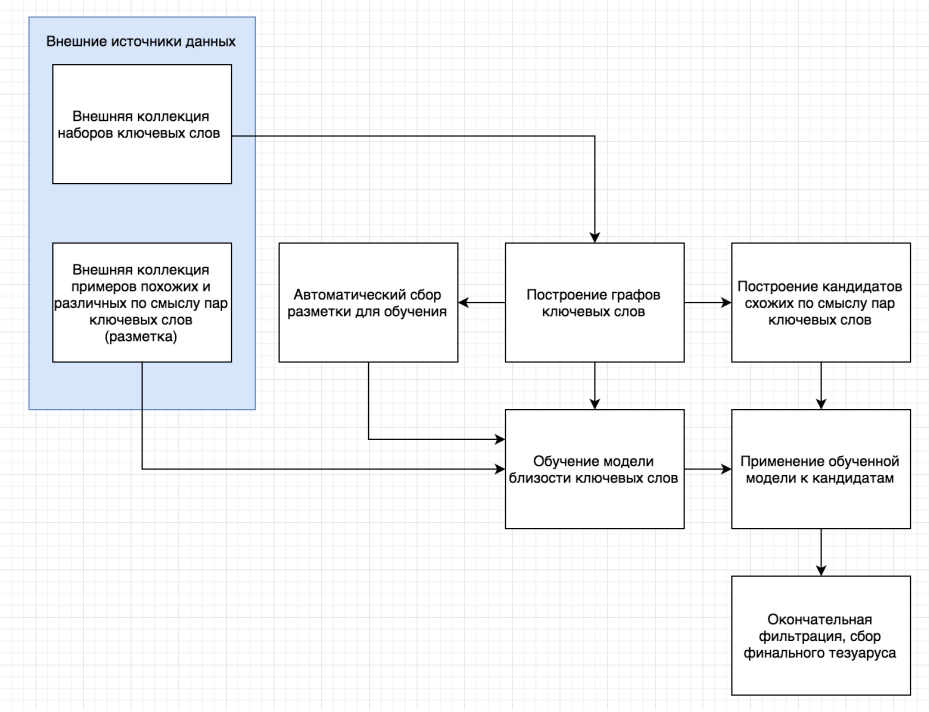


Рисунок 4.5 — Схема построения словаря тезауруса ключевых слов

ребра указывают на наличие у этой пары общего ключевого слова, которое входит в некоторые наборы вместе и с первым словом из пары, и со вторым (но не с двумя сразу). Было показано, что ребра такого графа гораздо сильнее отражают семантическую близость, в то время как ребра первого графа служат для улучшения механизмов поисковых подсказок, потому что предлагают связанные, но более разнообразные слова для заданного.

Также внешними данными является коллекция достоверно похожих и различных пар ключевых слов (разметка). Такая разметка необходима для настройки алгоритмов и тестирования качества их программных реализаций. В эту разметку входят открытые базы синонимов, антонимов, аббревиатур, а также некоторые переводы на другие языки.

Помимо словарной разметки, при построении тезауруса на базе графов ключевых слов генерируется автоматическая разметка. Преимущество этой разметки в том, что она полностью строится по данным, что означает отсутствие необходимости иметь внешнюю собранную вручную разметку (обычно это дорогой и трудоемкий процесс). Вместе с этим, автоматическая разметка не вносит смещение в данные. Например, если в базах синонимов встречается много математических терминов, но мало биологических, то модель настроится на то, чтобы давать парам математических терминов большее значение близости.

В то же время автоматическая разметка будет использовать обучающие примеры из разных областей ровно в той пропорции, в которой они представлены в конкретных данных. С другой стороны, такая разметка может быть менее точной, поэтому в финальной версии алгоритма используется обе разметки.

Поскольку ключевых слов достаточно много, то рассмотреть все пары не представляется возможным. Как следствие, по построенным графам строятся кандидаты близких по смыслу пар ключевых слов. Их количество достаточно велико, но значительно меньше всех возможных пар.

После этого для пар-кандидатов подсчитываются числовые факторы: различные расстояния и характеристики в графе, статистические показатели и другие. По этим факторам строится модель, которая настраивается с помощью разметок, описанных выше. Затем модель применяется к парам-кандидатам и те пары, которым модель дала наибольший вес, проходят в окончательный словарь ключевых слов.

4.5 Реализация поиска по ключевым словам на базе собранного тезауруса синонимов

Программные механизмы использования ключевых слов позволяют существенно улучшить поиск необходимой пользователю информации в системе. Информацию о большинстве объектов системы (публикации, конференции, сведения о пользователях и др.) можно дополнить произвольным набором ключевых на естественном языке. После внесения информации, происходит индексирование и обработка ключевых слов, что позволяет проводить дальнейший интеллектуальный анализ данных с целью получения релевантного ответа на запрос.

Основные задачи, которые могут быть решены при помощи ключевых слов - улучшение качества поисковых алгоритмов ранжирования подлежащих анализу объектов, а также поисковые подсказки рекомендательного характера для пользователя.

Для решения обозначенных выше задач был реализован поисковый модуль на базе фреймворка Django. Он представляет собой поисковую строку, в которую вводятся ключевые слова, разделенные запятой, а также таблицы поисковой выдачи, в которой указаны объекты, удовлетворяющие критериям поиска, в порядке

убывания релевантности. В момент ввода пользователю предлагается расширить свой запрос некоторыми связанными ключевыми словами, что впоследствии помогает найти более релевантные запросу объекты. Интерфейс системы представлен на рис.4.5

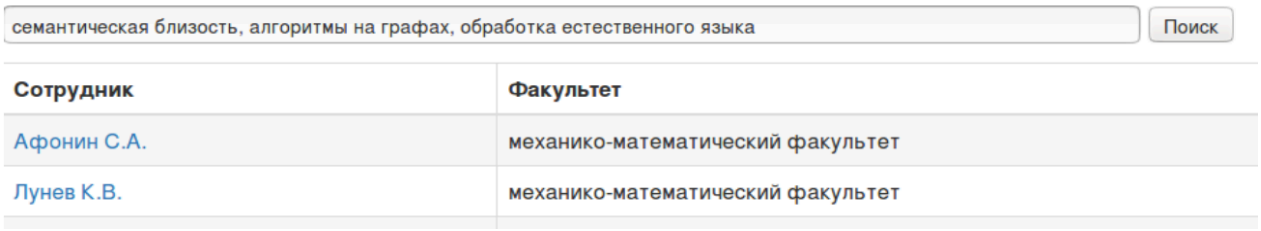


Рисунок 4.6 — Интерфейс модуля поиска по ключевым словам

Помимо этого реализовано ядро подсистемы обработки ключевых слов, которое проводит основной анализ, определение семантической близости пары слов, подготовку тезауруса ключевых слов. Методы определения сематнической близости пары ключевых слов описаны в главе 2, а описание алгоритма построения тезауруса дано в разделе 4.4. Код ядра представляет собой модули, процедуры и скрипты на языке Python с использованием открытых математических пакетов (Numpy, Pandas, Scipy), а также пакетов для анализа данных и машинного обучения (Scikit-learn, XGBoost). Данный модуль может использоваться отдельно от основного кода системы.

На рис.4.5 представлена схема выполнения и обработки запроса. Основной процесс работы с ключевыми словами состоит из трех шагов. На первом из них в ядре модели рассчитывается тезаурус ключевых слов. В этом тезаурусе для каждого ключевого слова хранятся ключевые слова, близкие по смыслу к данному, с указанием значения меры близости. Далее полученный словарь загружаются в базу данных системы, после чего появляется возможность по введенному пользователем слову быстро восстанавливать множество ключевых слов, похожих на заданное. Данный этап сбора словаря стоит обособленно от процесса поиска и может быть перезапущен в любой момент времени.

Второй шаг - обогащение объектов системы ключевыми словами из собранного на прошлом шаге тезауруса. Если для объекта указан набор ключевых слов, то для к каждому из этих ключевых слов добавляется информация о близких словах из тезауруса. Данный этап также является шагом предобработки данных.

Последний этап - непосредственно поиск по ключевым словам. Введенный пользователем запрос расширяется словами тезауруса, далее происходит поиск

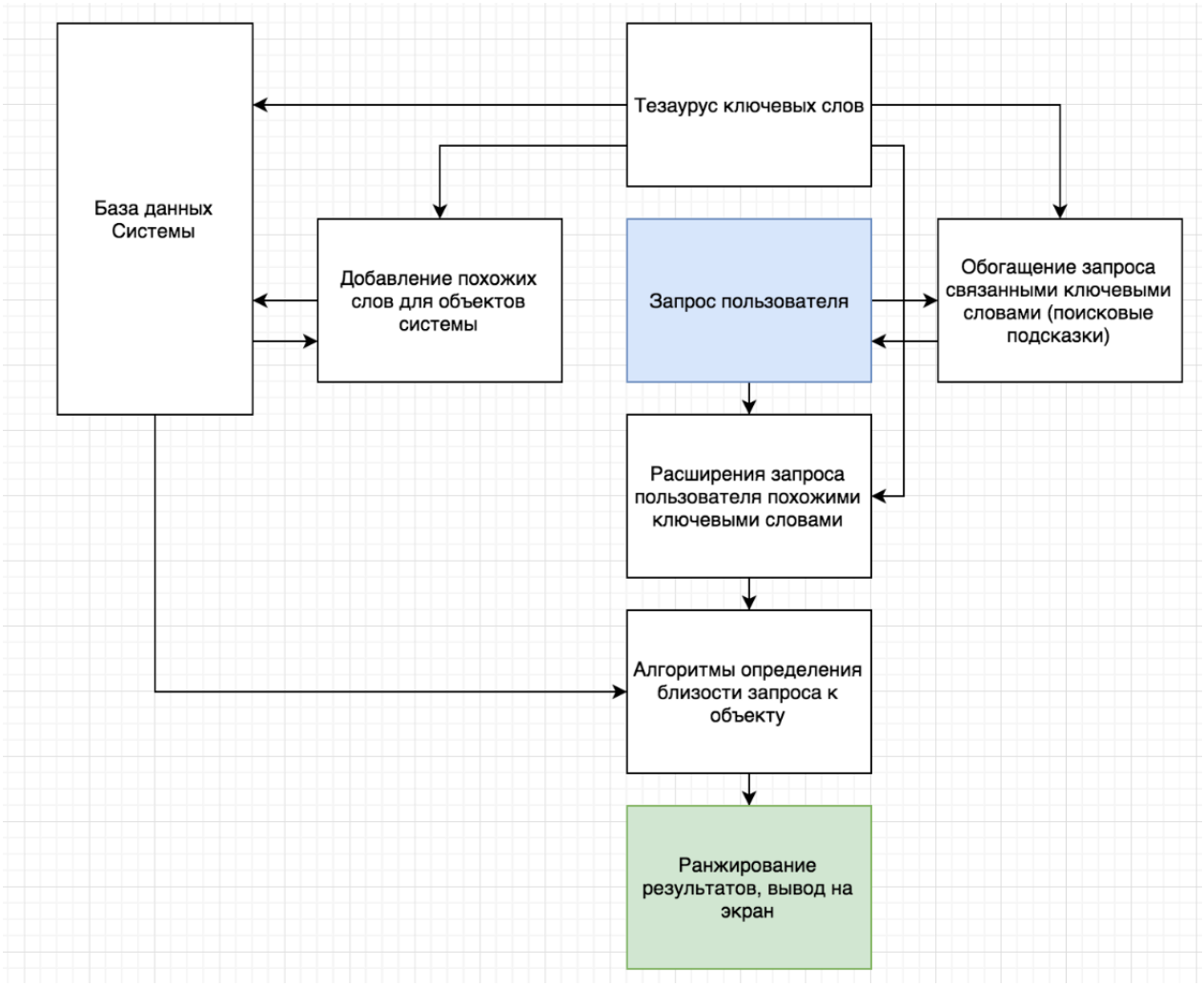


Рисунок 4.7 — Схема обработки запроса на поиск по ключевым словам

сущностей по расширенным наборам ключевых слов (как в запросе, так и в описании объекта). Для найденных объектов вычисляется релевантность (функция, возвращающая действительное число по паре запрос-объект, большие значения которой соответствуют более релевантным объектам), результаты сортируются по убыванию релевантности и выводятся на экран пользователя. Использование поискового запроса пользователя как набора ключевых слов, позволяет применять упомянутые выше алгоритмы для разработки поисковых подсказок. Такие подсказки предлагают пользователю дополнить свой запрос ключевыми словами, связанными по смыслу с теми, которые он уже ввел.

Трудностью в решении задачи реализации поиска по ключевым словам является тот факт, что для значительной части слов в системе нет достаточной статистики их использования. Как следствие, возникает ситуация, когда про слово, добавленное к описанию объекта или про слово, заданное пользователем в поисковую строку, нет достаточной информации, что не позволяет получить

релевантную запросу поисковую выдачу. Для преодоления этой трудности в системе реализованы интеллектуальные алгоритмы анализа ключевых слов, а также используются внешние корпусы данных на естественном языке, описанные в разделе 2.2. Основными направлениями работы по использованию ключевых слов для выполнения поисковых запросов являются следующие:

– определение семантической близости между парой ключевых слов с помощью алгоритмов машинного обучения и внешних наборов данных;
– определение семантической между парами наборов ключевых слов;
– использование связей между объектами системы (например, списки публикаций одного автора, таблицы соавторства, списки участников конференции, работников лаборатории и др.)

Таким образом, поиск по ключевым словам осуществляет не только определение точного вхождения слов запроса в слова объектов, по которым ведется поиск, но также происходит расширение как запроса, так и подлежащих анализу документов семантически близкими словами. В дополнении к этому, наборы ключевых слов, ассоциированные с объектами системы, обогащаются словами, от связанных с ними объектов. Все это позволяет увеличить описание к имеющимся данным и, следовательно, улучшить возможности поиска.

4.6 Решение задачи поиска экспертов для графов знаний

В данном разделе приводится важное с практической точки зрения приложение реализаций моделей, описанных в главе 2. Разработанные автором подходы позволяют в значительной мере улучшить качество решения задачи поиска эксперта в системах, объекты которых задаются графами знаний. Определение графов знаний представлено ранее в 1.1.2. Методы хранения и обработки информации, основанные на использовании графов знаний, часто встречаются в современных информационно-аналитических системах. В этой связи важной является возможность применения разработанных в рамках диссертации подходов к системам такого типа.

Далее приводится описание данных, на которых были проведены тестовые испытания, процедуры применения и тестирования программных реализаций моделей, а также результаты тестовой апробации и выводы.

4.6.1 Выборка данных

Для проведения тестовых испытаний был использован открытый набор данных *Science Knowledge Graph (SciKG)*[1]. Указанный набор данных содержит информацию о научных публикациях, ключевых словах для них, а также об экспертах, которым ассоциированы ключевые слова. В дополнении к этому, для эксперта представлена информация о его научных интересах и должности в научной организации.. Подготовка исходного набора данных сводится к построению троек (s, r, o), где s - субъект отношения, o - объект отношения, а r - тип отношения, связывающий субъект с объектом. Субъектом и объектом могут выступать любые сущности из описанных данных, например, *персоналия, ключевое слово, научное направление, публикация*. По имеющимся данным были построены отношения следующих типов:

1. человек X *является экспертом* в области Y;
2. человек X *является автором* публикации Y;
3. человек X *использовал* ключевое слово Y в одной из своих работ;
4. человек X *интересуется* научным направлением Y;
5. ключевое слово X *было использовано* в публикации Y.

Тестовые испытания были проведены на случайной подвыборке вышеупомянутого набора данных. Использование подвыборки данных в ходе эксперимента обусловлено необходимостью апробации разработанной модели на небольших объемах данных. Кроме того, такая мера позволяет существенно уменьшить время проведения тестового испытания, описание которого представлено далее.

4.6.2 Тестовые испытания

Тестовое испытание проводилось по следующему сценарию. Часть троек субъект-отношение-объект, представленных в выборке данных, была скрыта до обучения. Оставшиеся данные используются тестовыми моделями для обучения. Задачей моделей является восстановление скрытых отношений. Особый интерес

[1] *Более подробное описание данных и ссылки для скачивания доступны по адресу https://www.aminer.cn/scikg*

представляет способность моделей восстанавливать отношения типа «является экспертом». Умение предсказывать отношения данного типа дает возможности качественно решать задачу поиска эксперта. В связи с этим уровень качества программной реализации модели определяется по следующему алгоритму:

- рассматривается множество скрытых троек H;
- среди скрытых троек $(s, r, o) \in H$ выбирается множество троек $H_e = \{(s, r, o) | r =$ «является экспертом»$\}$;
- для каждого субъекта s множества H_e создаются всевозможные тройки $(s, r, \hat{o})$, где r - отношение *«является экспертом»*, а $\hat{o}$ - объекты данного отношения из исходной выборки;
- для всех троек $(s, r, \hat{o})$ моделью предсказывается вероятность существования такой тройки;
- вычисляется доля субъектов s относительно всех субъектов множества H_e, в которой истинная тройка (s, r, o) оказывается среди первых n наиболее вероятных троек из $(s, r, \hat{0})$, где n - параметр алгоритма.

В программной реализации алгоритма параметр n выбирался равным 1, 10 и 30. Соответствующие метрики далее обозначаются $hits@1$, $hits@10$, $hits@30$. Таким образом, в ходе тестирования проверяется, попадает ли скрытый объект в 1, 10 или 30 наиболее вероятных объектов для данной пары субъект-отношение.

Для проведения эксперимента были использованы программные реализации известных в индустрии моделей *ComplEx* ([83]), *TransE* ([121]), *ConvE* ([122]), *ConvKB* ([123]). Наилучшего качества удалось добиться с помощью модели *ComplEx*.

Следующим шагом эксперимента является применение программных реализаций моделей, описанных в главе 2, для улучшения качества определения скрытых связей. Для этого дополнительно к имеющимся вводится отношение «является близким по смыслу». Это отношение определяется для двух ключевых слов из набора данных. Таким образом, граф знаний обогащается дополнительными связями между вершинами. Например, если эксперт A использовал ключевое слово X в своих публикациях, эксперт B использовал ключевое слово Y и с помощью модели семантической близости *WordMLSim* была выявлена семантическая связь между словами X и Y, то этот вносит дополнительную информацию о близости между экспертами A и B. Эта информация помогает модели *ComplEx* лучше восстановить искусственно скрытые связи в графе знаний.

Далее представлены результаты тестовых испытаний программных реализаций моделей *ComplEx* и *WordMLSim*.

	hits@1	hits@10	hits@30
ComplEx	0.308	0.562	0.640
ComplEx+WordMLSim	0.310	0.618	0.688

Таблица 16: Результаты тестирования модели WordMLSim

Как видно из таблицы, значения всех трех рассматриваемых метрик выше у подхода *ComplEx+WordMLSim*, использующего разработанные автором модели. Основная из рассматриваемых метрик - *hits@10* - была улучшена более, чем на 5%. В результате совместного использования моделей *ComplEx* и *WordMLSim* позволило заметно улучшить качество определения скрытых связей и, как следствие, качество определения экспертов на рассматриваемом наборе данных. Установленное улучшение достигнуто засчет разработанных автором моделей семантической близости, что позволяет утверждать о практической значимости предложенных методов.

4.6.3 Выводы

В рамках данного раздела была решена задача поиска эксперта на открытом наборе данных *SciKG*. Тестовые испытания показали, что разработанные автором методы позволяют добиться значительного улучшения качестваопределения экспертов научных направлений. При этом используется информация о взаимосвязях объектов рассматриваемой информационно-аналитической системы.

Успешно пройденные испытания свидетельствуют о высоком уровне качества авторских моделей определения семантической близости. Этот факт, в свою очередь, позволяет использовать разработанные подходы в реальных информационно-аналитических системах. Реализация описанного подхода внесена в программный код информационно-аналитической системы «ИСТИНА» для дальнейшего использования.

4.7 Соответствие программного модуля интеллектуального анализа на основе ключевых слов предъявляемым требованиям

В настоящем разделе показаны результаты анализа разработанного автором программного модуля на соответствие требованиям, заявляенным в приложении А. Здесь и далее верхнеуровневыми моделями называются модели, решающий практические задачи, описание которых предложено в данной главе. Нижнеуровневыми являются те модели, описанные в главах 2, 3. Далее приводятся формулировки требований и комментарии о степени их выполнения.

1. **Функциональные требования.**
   - 1.1. **Наличие для каждого из используемых в модуле интеллектуального анализа на основе ключевых слов программного средства строго описанных алгоритмов, на которых они реализованы и моделей, в рамках которых эти алгоритмы построены.**
     Описание всех необходимых моделей было приведено в данной и предыдущих главах настоящей диссертации. Для разработанных программных реализаций проведены тестовые испытания, для основных алгоритмов получены оценки производительности и потребления памяти. Таким образом данное требование является выполненным.
   - 1.2. **Эффективное обновление имеющихся и добавление новых данных в систему.**
     Наиболее сложным для обновления является процесс добавления новых ключевых слов в систему. Под «новыми» в данном случае понимаются те слова, которые не встречались ранее ни в одном из наборов ключевых слов. Причиной этому является тот факт, что ключевые слова - базовые единицы всех разработанных моделей, поэтому при добавлении новых ключевых слов в систему необходимо пересчитать все верхнеуровневые модели. В программном комплексе имеется две способа обновления данных. Первый из них заключается в пересборке всего набора моделей с самого начала. Такой способ применим для систем небольшого размера (до 25000 объектов). Второй способ - проведение частичного обновления по уже построенной системе. В этом случае новые ключевые

слова добавляются в существующие графы, после чего для этих слов вычисляются необходимые меры семантической близости. Отмечается, что такой подход менее эффективен с точки зрения временных затрат на одно ключевое слово, поскольку использует более наивные методы вычисления. Однако, в случае больших по объему имеющихся данных систем этот способ позволяет быстрее провести обновление системы.

Добавление новых наборов ключевых слов не требует дополнительной ресурсозатратной по объему операций предобработки, если в новых наборах нет новых ключевых слов. Добавление новых отношений в систему также является эффективной операцией.

Следует также отметить, что перерасчет всех моделей даже для больших систем занимает не более суток. Время выполнения подготовительных процедур составляет 7 часов на входном объеме данных в 300000 наборов. Таким образом, при ежедневном пересборе моделей, изменения в семантических моделях дойдут до пользователей системы на следующий день. Для сложных интеллектуальных моделей и систем, содержащих сотни тысячи и миллионы сущностей, данный период обновления можно считать допустимым. Исходя из вышеизложенного, требование является выполненным.

1.3. **Эффективная процедура кластеризации ключевых слов системы и поиск необходимого кластера.**

Процесс кластеризации занимает не более получаса даже на больших наборах данных (миллионы наборов ключевых слов). Следует отметить, что необходимость в перекластеризации данных не возникает слишком часто, поэтому время работы алгоритма является удовлетворительным. Поиск необходимого кластера эффективен за-счет хранения ключевых слов в системе. Таким образом, требование является выполненным.

1.4. **Эффективный поиск похожих объектов с помощью обученных моделей.**

Вся необходимая семантическая информация подсчитывается на этапах предобработки, что позволяет сделать этап поиска быстрым. Данное требование является выполненным.

1.5. **Реализация подмодулей, решающих практически значимые задачи информационного поиска в рамках аналитической системы.**

Описание необходимых модулей представлено в данной главе, поэтому требование является выполненным.

1.6. **Сбор пользовательской информации в ходе взаимодействия с комплексом.**

Для решения задачи используется готовый подмодуль логирования фреймворка Django. Отмечается при этом, что разрабатываемые модули не реализуют базу данных для системы, в которую они внедряются. Другими словами, эффективное хранение ключевых слов и наборов ключевых слов должно быть реализовано на стороне интеллектуальной системы, а разрабатываемые автором модули хранят в себе вычисленную семантическую информацию, вспомогательные данные, собранные тезаурусы и специфическ представления для взаимосвязанных объектов.

Кроме того, для последующего улучшения качества разработанных моделей проводится сбор данных о пользовательских действий. Такие действия дают важную обратную связь от пользователя, помогают понять насколько правильным было решение показать пользователю ту или иную информацию. Примером таких действий может быть факт клика пользователем на конкретный результат поиска по его запросу. Если пользователи часто выбирают первую строчку в результатах поиска, то это означает, что программный модуль нашел необходимые объекты по данному запросу. Исходы из вышесказанного, требование считается выполненным.

1.7. **Система должна функционировать под управлением ОС с открытым исходным кодом.**

Разработанный программный модуль функционирует под управлением ОС семейства Linux, поэтому требование считается выполненным.

2. **Надежность.**

2.1. **Качество обученных моделей должно валидироваться на отложенных выборках после каждого изменения моделей.**

Для моделей вычисления семантической близости ключевых слов были разработаны обучающие выборки, описание которых приводится в 2.2.1. Кроме того, для моделей выполняется ряд тестовых испытаний, описанных в соответствующих разделах главы 2. Тестирование при этом не применяется к тем данным, на которых обучались модели, другими словами, для валидации используется отложенная выборка данных.

Валидация остальных моделей проводится по зафиксированным примерам и правильным ответам для них, которые набирались группой экспертов.

Процесс обучения прекращается, если хотя бы в одном из тестов качество модели упало хотя бы на 10%.

2.2. **Стабильная работа в условиях одновременного использования сотрудниками крупной организации.**

Использование разработанных высокоуровневых семантических моделей не требует больших вычислительных мощностей, поскольку все необходимые данные, а также модели нижнего уровня являются подготовленными. Это обстоятельство позволяет получать нужные результаты применения программных реализаций одновременно большому количеству сотрудников. Следовательно, данное требование выполнено.

2.3. **Устойчивость к программным ошибкам и ошибкам интерфейса.**

Программные реализации моделей устроены таким образом, чтобы при возникновении ошибок программа выдает некоторые значения по умолчанию. Факт ошибки при этом логируется программным модулем. Такого же поведения придерживается модуль, если нужная модель не выдает ответ слишком долгое время. Таким образом, реализованное поведение по умолчанию выполняет требование.

3. **Практичность.**

3.1. **Комплекс должен иметь простой интуитивный интерфейс для пользователя.**

На данном этапе разработки модуля пользователь обладает минимальным интерфейсов для задания запроса. Данное требование выполнено.

3.2. **Комплекс должен быть легко читаемым и понимаемым для разработчиков.**

Комплекс разработан на языке Python, читаемость кода которого была заложена разработчиками языка. Код разработан в объектно-ориентированном стиле, что позваляет инкапсулировать внутренние методы и переменные. Для математических вычислений используются удобные библиотеки с открытым исходным кодом и обширной документацией. Код реализован согласно стандарту *PEP8*. Данные стандарт дает рекомендации по написанию грамотного и легкочитаемого кода. Основные неоднозначные моменты в коде имеют поясняющий их комментарий. В этой связи, требование является выполненным.

3.3. **Комплекс должен включать средства обратной связи пользователя с разработчиками.** Все пользовательские действия логируются разработанным модулем, задача явной обратной связи пользователя с разработчиком решается основной интеллектуальной системой, в которую внедряется реализованный модуль. Таким образом, требование выполнено частично.

4. **Эффективность.**

4.1. **Удовлетворительные показатели качества работы моделей на сильно ограниченных по объему данных.**

Эффективность работы моделей на небольших объемах данных продемонстрирована в подразделах «Тестовые испытания» соответствующих разделов. Получены удовлетворительные показатели качества, поэтому требование считается выполненным.

4.2. **Этап предподготовки комплекса.**

Этапы предподготовки данных, обучения и настройки моделей, а также подготовки ресурсов, используемых непосредственно верхнеуровневыми моделями на объемах данных, для систем, включающих в себя сотни тысяч сущностей, занимает порядка 10 часов на ЭВМ, обладающей 24-ядерным процессором и 64Гб оперативной памяти. Таким образом, и добавление новых сущностей, и перевычисление всех элементов комплекса укладывается в период в несколько часов, что выполняет предъявляемое требование.

4.3. **Этап использования моделей.**

В то время, как выдача нужного кластера ключевых слов и вычисление поисковых подсказок занимают доли секунды, построение полной поисковой выдачи имеет точки для роста производительности. Тем не менее, благодаря предрасчету данных и моделей, поисковые вычисления происходят с удовлетворительной скоростью. Поэтому требование считается выполненным частично.

5. **Сопровождаемость.**

5.1. **Весь комплекс архитектурно должен разбиваться на ряд отдельных модулей. Логика и параметры этих модулей системы должны быть инкапсулированы друг от друга.**

Код разработан в объектно-ориентированном стиле, что позваляет инкапсулировать внутренние методы и переменные. Реализована иерархическая структура использования моделей: верхнеуровневые классы, решающие заявленные практические задачи, используют модели семантической близости объектов системы, которые, в свою очередь, используют модели семантической близости пары наборов ключевых слов. На самом низком уровне модели для пар наборов ключевых слов используют модели семантической близости пары ключевых слов. При этом прямого доступа из моделей верхнего уровня к моделям нижнего уровня разработчикам не дается. Это упрощает понимание кода и делает разработку более эффективной. Обозначенные выше доводы позволяют считать требование выполненным.

5.2. **Иметь возможность быстрого и эффективного способа расширения функционала комплекса.**

Ввиду обособленности реализаций различных моделей, а также объектной ориентированности кода, расширение имеющогося функционала не должно вызывать проблем у разработчиков. По этой причине требование считается выполненным.

5.3. **Быть документированной.**

Основные классы и функции задокументированны. Кроме того, описан формат входных и выходных данных для каждой модели. Тем не менее, не все места в программном коде имеют исчерпывающую документацию, поэтому требование принимается выполненным частично.

6. **Мобильность.**

6.1. **Возможность внедрения в различные информационно-аналитические системы произвольной направленности с допустимым уровнем качества моделей. Модели должны иметь возможность обучаться на данных новой системы.**

Единственным необходимым требованием к системе является наличие ассоциированных ключевых слов к сущностям этой системы. В дополнении к этому, обогащение моделей с помощью информации о взаимосвязанных объектах может дать значительный рост эффективности верхнеуровневых моделей.

6.2. **Возможность обучения специфических моделей семантической близости, автоматически подстраиваемых к предметной области системы, в которой разворачивается комплекс.**

Модели разрабатывались из условий максимальной независимости от внешних данных. Семантическая информация, которую определяют эти модели, целиком берется из тех данных, в которых они обучаются, поэтому требование подстраиваемости под конкретную область выполнено.

6.3. **Возможность обучения моделей семантической близости без имеющихся обучающих примеров.**

Модели вычисления семантической близости для пары ключевых слов, а также модель кластеризации ключевых слов способны обучаться в полностью автоматическом режиме, другим же необходимы обучающие примеры, заданные экспертной группой, поэтому данное требование удовлетворено частично.

6.4. **Возможность внедрения в систему с дефицитом данных о ключевых словах.**

Дефицит информации приводит к деградации качества разработанных моделей. Отмечается при этом, что разработанные решения демонстрируют адекватные результаты на небольших объемах данных, начиная от 1000 наборов ключевых слов. Дополнительным источником качества в случае дефицита данных служит информация о взаимосвязанных объектах. Поскольку возможность внедрения присутствует, требование считается выполненным.

6.5. **Адаптируемость к добавлению новых сущностей и отношений между ними в системе.**
Благодаря возможности дообучения, а также возможности относительно быстро пересчитать все требуемые модели с начала, данное требование считается выполненным.

6.6. **Развертываемость комплекса внутри новой системы не должна занимать много времени работы экспертов. Необходимо лишь наладить поставку данных в нужном формате и сконфигурать модули для наиболее эффективного решения задач конкретной системы.**
На текущей момент высокоуровневые модели требуют небольшое количество экспертной информации из предметной области для правильной настройки моделей. Поэтому требование считается выполненным частично.

6.7. **Устойчивость к пропускам и неточностям в данных.**
Благодаря разработанным семантическим моделям имеется возможность частично восстановить информацию о ключевых словах, которые были упущены при написании набора ключевых слов. Исправление опечаток не является предметом исследования данной работы, однако разработанные модели семантической близости пары ключевых слов, а также модели кластеризации ключевых слов позволяют во многих случаях отнести слово с опечаткой в кластер, которые в числе прочего содержит и верное написание данного слова. В связи с этим, требование считается выполненным.

4.8 Выводы

В настоящей главе представлено описание разработанных автором моделей и алгоритмов, востребованных практикой приложений, а также реализующих их программных комплексов. Основой для них являются подходы, представленные в главах 2, 3. Показано, что разработанный программный комплекс соответствует предъявляемым к нему требованиям.

На настоящее время разработанный комплекс имеет новые точки для роста. Основным направлением развития может стать дальнейшее уменьшение завивисимости работы комплекса от наличия экспертов. Принимаю во внимание, что решение большинства задач, возникающих в области семантического анализа объективно требует экспертной оценки, в рамках работы над настоящей диссертацией были созданы программные модули, позволяющий значительно уменьшить количество ручной работы высококвалифицированных экспертов. К числу таких относятся, например, задачи:

- создания искусственной обучающей выборки для задачи определения семантической близости пары ключевых слов;
- использования методов машинного обучения, позволяющих абстрагироваться от предметной области;
- поиска эффективных механизмов семантической кластеризации ключевых слов, требующих минимального числа параметров для настройки.

В качестве дальнейшей цели исследования может рассматриваться улучшение качества существующих моделей за счет применения технологий отличных от тех, которые рассмотрены в данной работе. Например, в рамках исследования, представленного в настоящей диссертации, в явном виде практически не использовались синтаксические и лексические свойства слов. Кроме этого могут быть использованы представления данных более сложные, чем графовые. Однако, и представленные в настоящей диссертации модели, алгоритмы и их программные реализации позволяют решать многие задачи анализа информации в системах больших данных. Одним из направлений развития может быть увеличение числа задач интеллектуального анализа, востребованных в различных средах национальной экономики.

Заключение

В настоящей диссертации представлено описание методологии реализованных подходов к анализу данных, характеризующих объекты в информационно-аналитической наукометрической системе. Предложены модели, алгоритмы и реализующие их программные средства. В основе разработанных подходов лежат методы теории графов, анализа текстов на естественном языке, машинного обучения и программной инженерии.

Отмечается, что предложенные автором подходы к построению моделей семантического анализа могут применяться не только для наукометрических систем, но также и для других систем, объекты которой описываются набором слов естественного языка. Важным преимуществом разработанных моделей является возможность их эффективного внедрения в системы, не обладающие большим объемом исходных данных. Другое их достоинство заключается в способности использовать произвольные связи между объектами системы для улучшения качества определения уровня семантической близости.

В ходе работ по подготовке настоящей диссертации получены следующие **основные результаты**.

- Разработан ряд моделей вычисления уровня семантической близости между ключевыми словами интеллектуальной системы. Программные реализации моделей позволяют вычислять такую близость, автоматически учитывая специфику системы, в которую модели внедряются. Проведены многочисленные тестовые испытания, подтверждающие высокий уровень полученных результатов. Получены аналитические оценки, характеризующие вычислительную сложность программных реализаций этих моделей.
- Созданы уникальные методы автоматической генерации обучающей выборки для определения семантически похожих ключевых слов. Разработанные методы избавляют от больших человеческих трудозатрат для выставления экспертных оценок. Кроме того, алгоритмы генерации позволяют использовать методы машинного обучения с учителем для решения задачи семантической близости пары слов. Это обстоятельство в значительной мере упрощает подбор параметров модели и улучшает качество определения семантической близости.

– Разработана модель и ее программная реализация для вычисления семантической близости между парой объектов. Для решения задачи были использованы дополнительные связи между сущностями системы. Модель протестирована на данных из ИАС «ИСТИНА». Показано, что модель удовлетворяет предъявляемым к ней требованиям.
– На основе разработанных моделей решены важные, востребованные практикой задачи поиска экспертов в различных областях научных знаний, кластеризации ключевых слов, определения тематической направленности объекта информационно-аналитической системы. Программные реализации решений опробированы на данных из ИАС «ИСТИНА», включены в состав ее программного обеспечения, а также получили высокие оценки качества квалифицированных экспертов. Получен акт о внедрении.

Благодарности. Автор выражает огромную благодарность своему научному руководителю доктору физико-математических наук, профессору Валерию Александровичу Васенину за внимание к работе на всех ее этапах, редакторскую правку диссертационной работы и опубликованных статей, а также за ценные наставления, терпение и понимание.

Автор благодарит кандидата физико-математических наук, доцента С.А.Афонина за активное участие в постановке задач, за плодотворное обсуждение результатов работы, а также за техническую помощь в проведении исследований и экспериментов.

Список литературы

1. *Frawley*, *W.* Linguistic Semantics / W. Frawley. — L. Erlbaum Associates, 1992.
2. Semantic Similarity from Natural Language and Ontology Analysis / S. Harispe [и др.] // CoRR. — 2017. — T. abs/1704.05295. — arXiv: 1704.05295. — URL: http://arxiv.org/abs/1704.05295.
3. *Levenshtein*, *V.* Binary Codes Capable of Correcting Deletions, Insertions and Reversals / V. Levenshtein // Soviet Physics Doklady. — 1966. — T. 10. — C. 707.
4. *Miller*, *F. P.* Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance / F. P. Miller, A. F. Vandome, J. McBrewster. — Alpha Press, 2009.
5. *Ristad*, *E. S.* Learning string edit distance / E. S. Ristad, P. N. Yianilos, S. Member // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1998. — T. 20. — C. 522—532.
6. *McCallum*, *A.* A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance / A. McCallum, K. Bellare, F. C. N. Pereira // CoRR. — 2012. — T. abs/1207.1406. — arXiv: 1207.1406. — URL: http://arxiv.org/abs/1207.1406.
7. *Jaro*, *M. A.* Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida / M. A. Jaro // Journal of the American Statistical Association. — 1989. — T. 84, № 406. — C. 414—420.
8. *Jaro*, *M. A.* Probabilistic linkage of large public health data file / M. A. Jaro // Statistics in Medicine. T. 14. — Carg□se, France., 1995. — C. 491—498.
9. *Ukkonen*, *E.* Approximate String-matching with Q-grams and Maximal Matches / E. Ukkonen // Theor. Comput. Sci. — Essex, UK, 1992. — Янв. — T. 92, № 1. — C. 191—211. — URL: http://dx.doi.org/10.1016/0304-3975(92)90143-4.
10. *Huang*, *A.* Similarity Measures for Text Document Clustering / A. Huang. — 2008.
11. *Jacobs*, *J.* Finding words that sound alike. The SOUNDEX algorithm. / J. Jacobs // Byte 7. — 1982. — C. 473—474.

12. *Hixon, B.* Phonemic Similarity Metrics to Compare Pronunciation Methods / B. Hixon, E. Schneider, S. L. Epstein // INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011. — 2011. — С. 825—828. — URL: http://www.isca-speech.org/archive/interspeech_2011/i11_0825.html.

13. *Dunning, T.* Accurate Methods for the Statistics of Surprise and Coincidence / T. Dunning // Comput. Linguist. — Cambridge, MA, USA, 1993. — Март. — Т. 19, № 1. — С. 61—74. — URL: http://dl.acm.org/citation.cfm?id=972450.972454.

14. Class-based N-gram Models of Natural Language / P. F. Brown [и др.] // Comput. Linguist. — Cambridge, MA, USA, 1992. — Дек. — Т. 18, № 4. — С. 467—479. — URL: http://dl.acm.org/citation.cfm?id=176313.176316.

15. *Church, K. W.* Word Association Norms, Mutual Information, and Lexicography / K. W. Church, P. Hanks // Comput. Linguist. — Cambridge, MA, USA, 1990. — Март. — Т. 16, № 1. — С. 22—29. — URL: http://dl.acm.org/citation.cfm?id=89086.89095.

16. *Chen, S. F.* An Empirical Study of Smoothing Techniques for Language Modeling / S. F. Chen, J. Goodman // Proceedings of the 34th Annual Meeting on Association for Computational Linguistics. — Santa Cruz, California : Association for Computational Linguistics, 1996. — С. 310—318. — (ACL '96). — URL: https://doi.org/10.3115/981863.981904.

17. *Department, C.-M. U. C. S.* Adaptive Statistical Language Modeling: a Maximum Entropy Approach / C.-M. U. C. S. Department, R. Rosenfeld. — School of Computer Science, Carnegie Mellon University, 1994. — (Adaptive statistical language modeling: a maximum entropy approach ; т. 94—138). — URL: https://books.google.ru/books?id=8AgFngEACAAJ.

18. *Schneider, K.-M.* Weighted Average Pointwise Mutual Information for Feature Selection in Text Categorization / K.-M. Schneider // Knowledge Discovery in Databases: PKDD 2005 / под ред. A. M. Jorge [и др.]. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. — С. 252—263.

19. *Dagan, I.* Similarity-Based Models of Word Cooccurrence Probabilities / I. Dagan, L. Lee, F. C. N. Pereira // Machine Learning. — 1999. — Февр. — Т. 34, № 1. — С. 43—69. — URL: https://doi.org/10.1023/A:1007537716579.

20. *Bouma*, *G.* Normalized (pointwise) mutual information in collocation extraction / G. Bouma // From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009. Normalized. — Tübingen, 2009. — С. 31—40.

21. *Thanopoulos*, *A.* Comparative Evaluation of Collocation Extraction Metrics. / A. Thanopoulos, N. Fakotakis, G. Kokkinakis // Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002). — Las Palmas, Canary Islands - Spain : European Language Resources Association (ELRA), 05.2002. — URL: http://www.lrec-conf.org/proceedings/lrec2002/pdf/128.pdf ; ACL Anthology Identifier: L02-1128.

22. *Bollegala*, *D.* Measuring semantic similarity between words using web search engines / D. Bollegala, Y. Matsuo, M. Ishizuka // WWW '07: Proceedings of the 16th international conference on World Wide Web. — Banff, Alberta, Canada : ACM, 2007. — С. 757—766.

23. *Terra*, *E.* Frequency Estimates for Statistical Word Similarity Measures / E. Terra, C. L. A. Clarke // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. — Edmonton, Canada : Association for Computational Linguistics, 2003. — С. 165—172. — (NAACL '03). — URL: https://doi.org/10.3115/1073445.1073477.

24. *Roark*, *B.* Discriminative N-gram Language Modeling / B. Roark, M. Saraclar, M. Collins // Comput. Speech Lang. — London, UK, UK, 2007. — Апр. — Т. 21, № 2. — С. 373—392. — URL: http://dx.doi.org/10.1016/j.csl.2006.06.006.

25. *Bickel*, *S.* Predicting Sentences Using N-gram Language Models / S. Bickel, P. Haider, T. Scheffer // Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. — Vancouver, British Columbia, Canada : Association for Computational Linguistics, 2005. — С. 193—200. — (HLT '05). — URL: https://doi.org/10.3115/1220575.1220600.

26. *Pauls*, *A.* Faster and Smaller N-gram Language Models / A. Pauls, D. Klein // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. — Portland, Oregon :

Association for Computational Linguistics, 2011. — С. 258—267. — (HLT ’11). — URL: http://dl.acm.org/citation.cfm?id=2002472.2002506.

27. *Chelba, C.* N-gram Language Modeling using Recurrent Neural Network Estimation : тех. отч. / C. Chelba, M. Norouzi, S. Bengio ; Google. — 2017. — URL: https://arxiv.org/abs/1703.10724.

28. N-gram-based Machine Translation / J. B. Mariòo [и др.] // Comput. Linguist. — Cambridge, MA, USA, 2006. — Дек. — Т. 32, № 4. — С. 527—549. — URL: http://dx.doi.org/10.1162/coli.2006.32.4.527.

29. Bilingual n-gram statistical machine translation / R. E. Banchs [и др.] // In Proc. of Machine Translation Summit X. — 2005. — С. 275—282.

30. Statistical Machine Translation of Euparl Data by Using Bilingual N-grams / R. E. Banchs [и др.] // Proceedings of the ACL Workshop on Building and Using Parallel Texts. — Ann Arbor, Michigan : Association for Computational Linguistics, 2005. — С. 133—136. — (ParaText ’05). — URL: http://dl.acm.org/citation.cfm?id=1654449.1654478.

31. *Kondrak, G.* N-gram Similarity and Distance / G. Kondrak // Proceedings of the 12th International Conference on String Processing and Information Retrieval. — Buenos Aires, Argentina : Springer-Verlag, 2005. — С. 115—126. — (SPIRE’05). — URL: http://dx.doi.org/10.1007/11575832_13.

32. *Albatineh, A. N.* Correcting Jaccard and other similarity indices for chance agreement in cluster analysis / A. N. Albatineh, M. Niewiadomska-Bugaj // Advances in Data Analysis and Classification. — 2011. — Окт. — Т. 5, № 3. — С. 179—200. — URL: https://doi.org/10.1007/s11634-011-0090-y.

33. *Sørensen, T.* A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons / T. Sørensen. — I kommission hos E. Munksgaard, 1948. — (Biologiske skrifter). — URL: https://books.google.ru/books?id=rpS8GAAACAAJ.

34. *Levandowsky, M.* Distance between sets [5] / M. Levandowsky, D. Winter // Nature. — 1971. — Т. 234, № 5323. — С. 34—35.

35. *Cilibrasi, R. L.* The Google Similarity Distance / R. L. Cilibrasi, P. M. Vitanyi // IEEE Transactions on Knowledge and Data Engineering. — Los Alamitos, CA, USA, 2007. — Т. 19. — С. 370—383.

36. *Shirude, S. B.* Identifying Subject Area/s of User Using n-Gram and Jaccard's Similarity in Profile Agent of Library Recommender System / S. B. Shirude, S. R. Kolhe // Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies. — Udaipur, Rajasthan, India : ACM, 2014. — 23:1—23:6. — (ICTCS '14). — URL: http://doi.acm.org/10.1145/2677855.2677878.

37. *Salton, G.* Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer / G. Salton. — Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1989. — C. 189—225.

38. *Miller, G. A.* WordNet: A Lexical Database for English / G. A. Miller // Commun. ACM. — New York, NY, USA, 1995. — Нояб. — Т. 38, № 11. — С. 39—41. — URL: http://doi.acm.org/10.1145/219717.219748.

39. RussNet: Building a Lexical Database for the Russian Language / I. Azarova [и др.] // In: Proceedings: Workshop on Wordnet Structures and Standardisation and How this affect Wordnet Applications and Evaluation. Las Palmas. — 2002. — С. 60—64.

40. *Braslavski, P.* A Spinning Wheel for YARN: User Interface for a Crowdsourced Thesaurus / P. Braslavski, D. Ustalov, M. Mukhin // Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. — Gothenburg, Sweden : Association for Computational Linguistics, 04.2014. — С. 101—104. — URL: http://www.aclweb.org/anthology/E14-2026.

41. YARN: Spinning-in-progress / P. Braslavski [и др.] // Proceedings of the 8th Global WordNet Conference, GWC 2016. — Global WordNet Association, 2016. — С. 58—65.

42. *Лукашевич, Н.* Тезаурусы в задачах информационного поиска / Н. Лукашевич. — Изд-во Моск. ун-та, 2011. — URL: https://books.google.ru/books?id=J4XlkQEACAAJ.

43. Creating Russian WordNet by Conversion / N. V. Loukachevitch [et al.] // Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. — Rossiiskii Gosudarstvennyi Gumanitarnyi Universitet, 2016. — P. 405—415.

44. *Resnik, P.* Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language / P. Resnik // CoRR. — 2011. — T. abs/1105.5444. — arXiv: 1105.5444. — URL: http://arxiv.org/abs/1105.5444.

45. *Budanitsky, A.* Evaluating WordNet-based Measures of Lexical Semantic Relatedness / A. Budanitsky, G. Hirst // Comput. Linguist. — Cambridge, MA, USA, 2006. — Март. — Т. 32, № 1. — С. 13—47. — URL: http://dx.doi.org/10.1162/coli.2006.32.1.13.

46. *Matar, Y.* KWSim: Concepts Similarity Measure. / Y. Matar, E. Egyed-Zsigmond, S. Lajmi // CORIA. — Université de Renne 1, 08.07.2009. — С. 475—482. — URL: http://dblp.uni-trier.de/db/conf/coria/coria2008.html#MatarEL08.

47. Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies / E. G. M. Petrakis [и др.] // In 4 th Workshop on Multimedia Semantics (WMS'06. — 1998. — С. 44—52.

48. *Gabrilovich, E.* Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis / E. Gabrilovich, S. Markovitch // Proceedings of the 20th International Joint Conference on Artifical Intelligence. — Hyderabad, India : Morgan Kaufmann Publishers Inc., 2007. — С. 1606—1611. — (IJCAI'07). — URL: http://dl.acm.org/citation.cfm?id=1625275.1625535.

49. *Turdakov, D.* Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation / D. Turdakov, P. Velikhov.

50. Distributed Representations of Words and Phrases and Their Compositionality / T. Mikolov [и др.] // Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. — Lake Tahoe, Nevada : Curran Associates Inc., 2013. — С. 3111—3119. — (NIPS'13). — URL: http://dl.acm.org/citation.cfm?id=2999792.2999959.

51. *Pennington, J.* Glove: Global vectors for word representation / J. Pennington, R. Socher, C. D. Manning // In EMNLP. — 2014.

52. StarSpace: Embed All The Things! / L. Wu [и др.]. — 2017. — URL: http://arxiv.org/abs/1709.03856 ; cite arxiv:1709.03856.

53. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // CoRR. — 2018. — Т. abs/1810.04805. — arXiv: 1810.04805. — URL: http://arxiv.org/abs/1810.04805.

54. *Roller*, *S.* Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora / S. Roller, D. Kiela, M. Nickel // ACL (2). — Association for Computational Linguistics, 2018. — С. 358—363. — URL: http://arxiv.org/abs/1806.03191.

55. *Kalman*, *D.* A singularly valuable decomposition: The SVD of a matrix / D. Kalman // College Math Journal. — 1996. — Т. 27. — С. 2—23.

56. *Kenter*, *T.* Short Text Similarity with Word Embeddings / T. Kenter, M. de Rijke // Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. — Melbourne, Australia : ACM, 2015. — С. 1411—1420. — (CIKM '15). — URL: http://doi.acm.org/10.1145/2806416.2806475.

57. Information Retrieval in Folksonomies: Search and Ranking / A. Hotho [и др.] // Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications. — Budva, Montenegro : Springer-Verlag, 2006. — С. 411—426. — (ESWC'06). — URL: http://dx.doi.org/10.1007/11762256_31.

58. *Srinivas*, *G.* A Weighted Tag Similarity Measure Based on a Collaborative Weight Model / G. Srinivas, N. Tandon, V. Varma // Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents. — Toronto, ON, Canada : ACM, 2010. — С. 79—86. — (SMUC '10). — URL: http://doi.acm.org/10.1145/1871985.1871999.

59. The PageRank citation ranking: Bringing order to the Web / L. Page [и др.] // Proceedings of the 7th International World Wide Web Conference. — Brisbane, Australia, 1998. — С. 161—172. — URL: citeseer.nj.nec.com/page98pagerank.html.

60. *Jeh*, *G.* SimRank: a measure of structural-context similarity / G. Jeh, J. Widom // KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. — New York, NY, USA : ACM Press, 2002. — С. 538—543. — URL: http://dx.doi.org/10.1145/775047.775126.

61. *Jeh*, *G.* Scaling Personalized Web Search / G. Jeh, J. Widom // Proceedings of the 12th International Conference on World Wide Web. — Budapest, Hungary : ACM, 2003. — С. 271—279. — (WWW '03). — URL: http://doi.acm.org/10.1145/775152.775191.

62. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia / J. Lehmann [и др.] // Semantic Web Journal. — 2015. — Т. 6, № 2. — С. 167—195. — URL: http://jens-lehmann.org/files/2015/swj_dbpedia.pdf.

63. Freebase: a collaboratively created graph database for structuring human knowledge / K. Bollacker [и др.] // In SIGMOD Conference. — 2008. — С. 1247—1250.

64. *Mahdisoltani*, *F.* YAGO3: A Knowledge Base from Multilingual Wikipedias / F. Mahdisoltani, J. Biega, F. M. Suchanek. — 2015.

65. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion / X. L. Dong [и др.] // The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. — 2014. — С. 601—610. — URL: http://www.cs.cmu.edu/~nlao/publication/2014.kdd.pdf ; Evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz.

66. A Review of Relational Machine Learning for Knowledge Graphs / M. Nickel [и др.] // Proceedings of the IEEE. — 2016. — Янв. — Т. 104, № 1. — С. 11—33.

67. *Adamic*, *L. A.* Friends and Neighbors on the Web / L. A. Adamic, E. Adar // SOCIAL NETWORKS. — 2001. — Т. 25. — С. 211—230.

68. *Barabasi*, *A.-L.* Emergence of Scaling in Random Networks / A.-L. Barabasi, R. Albert // Science. — 1999. — Т. 286, № 5439. — С. 509—512. — eprint: http://www.sciencemag.org/cgi/reprint/286/5439/509.pdf. — URL: http://www.sciencemag.org/cgi/content/abstract/286/5439/509.

69. *Katz*, *L.* A new status index derived from sociometric analysis / L. Katz // Psychometrika. — 1953. — Март. — Т. 18, № 1. — С. 39—43. — URL: http://ideas.repec.org/a/spr/psycho/v18y1953i1p39-43.html.

70. *Leicht*, *E. A.* Vertex similarity in networks. / E. A. Leicht, P. Holme, M. Newman // Physical review. E, Statistical, nonlinear, and soft matter physics. — 2006. — Т. 73 2 Pt 2. — С. 026120.

71. *Nickel*, *M.* Tensor Factorization for Multi-relational Learning / M. Nickel, V. Tresp // Machine Learning and Knowledge Discovery in Databases / под ред. H. Blockeel [и др.]. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2013. — С. 617—621.

72. *Kolda*, *T. G.* Tensor Decompositions and Applications / T. G. Kolda, B. W. Bader // SIAM Review. — 2009. — Т. 51, № 3. — С. 455—500. — eprint: https://doi.org/10.1137/07070111X. — URL: https://doi.org/10.1137/07070111X.

73. *Lao*, *N.* Random Walk Inference and Learning in a Large Scale Knowledge Base / N. Lao, T. Mitchell, W. W. Cohen // Proceedings of the Conference on Empirical Methods in Natural Language Processing. — Edinburgh, United Kingdom : Association for Computational Linguistics, 2011. — С. 529—539. — (EMNLP '11). — URL: http://dl.acm.org/citation.cfm?id=2145432.2145494.

74. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks / Y. Sun [и др.] // In VLDB' 11. — 2011.

75. *U*, *L. H.* PathSimExt: Revisiting PathSim in Heterogeneous Information Networks / L. H. U, K. Yao, H. F. Mak // WAIM. Т. 8485. — Springer, 2014. — С. 38—42. — (Lecture Notes in Computer Science).

76. *Pham*, *P.* W-PathSim: Novel Approach of Weighted Similarity Measure in Content-Based Heterogeneous Information Networks by Applying LDA Topic Modeling / P. Pham, P. Do, C. D. C. Ta // ACIIDS (1). Т. 10751. — Springer, 2018. — С. 539—549. — (Lecture Notes in Computer Science).

77. *Blei*, *D. M.* Latent Dirichlet Allocation / D. M. Blei, A. Y. Ng, M. I. Jordan // J. Mach. Learn. Res. — 2003. — Март. — Т. 3. — С. 993—1022. — URL: http://dl.acm.org/citation.cfm?id=944919.944937.

78. *Nickel*, *M.* Tensor factorization for relational learning : дис. … канд. / Nickel M. — Ludwig Maximilians University Munich, 2013. — С. 1—145.

79. *Ristoski*, *P.* RDF2Vec: RDF Graph Embeddings for Data Mining / P. Ristoski, H. Paulheim // The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I. — 2016. — С. 498—514. — URL: https://doi.org/10.1007/978-3-319-46523-4%5C_30.

80. Expeditious Generation of Knowledge Graph Embeddings. — 2018.

81. Convolutional 2D Knowledge Graph Embeddings / T. Dettmers [и др.] // AAAI. — AAAI Press, 2018. — С. 1811—1818.

82. Reasoning With Neural Tensor Networks for Knowledge Base Completion / R. Socher [и др.] // Advances in Neural Information Processing Systems 26 / под ред. C. J. C. Burges [и др.]. — Curran Associates, Inc., 2013. — С. 926—934. — URL: http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf.

83. Complex Embeddings for Simple Link Prediction / T. Trouillon [и др.] // Proceedings of The 33rd International Conference on Machine Learning. Т. 48 / под ред. M. F. Balcan, K. Q. Weinberger. — New York, New York, USA : PMLR, 20–22 Jun.2016. — С. 2071—2080. — (Proceedings of Machine Learning Research). — URL: http://proceedings.mlr.press/v48/trouillon16.html.

84. Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. / J. Weston [и др.] // CoRR. — 2013. — Т. abs/1307.7973. — URL: http://dblp.uni-trier.de/db/journals/corr/corr1307.html#WestonBYU13.

85. Learning Entity and Relation Embeddings for Knowledge Graph Completion / Y. Lin [и др.] // Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. — 2015. — С. 2181—2187. — URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571.

86. TEXTAPPLIANCE – новое решение для интеллектуального поиска и анализа больших массивов текстов / Г. С. Осипов [и др.] // Материалы второго международного профессионального форума «Книга. Культура. Образование. Инновации» («Крым-2016»). — Судак, Россия, 2016. — С. 270—271.

87. Технологии семантического поиска заимствований в научных текстах / Г. С. Осипов [и др.] // Материалы второго международного профессионального форума «Книга. Культура. Образование. Инновации» («Крым-2016»). — Судак, Россия, 2016. — С. 311—313.

88. *Осипов, Г. С.* Семантический анализ научных текстов и их больших массивов / Г. С. Осипов, И. В. Смирнов // Системы высокой доступности. — 2016. — Т. 12, № 1. — С. 41—44.

89. *Ganter, B.* Formal Concept Analysis: Mathematical Foundations / B. Ganter, R. Wille. — Berlin/Heidelberg : Springer, 1999.

90. *Kuznetsov, S.* Comparing performance of algorithms for generating concept lattices / S. Kuznetsov, S. Obiedkov // Journal of Experimental and Theoretical Artificial Intelligence. — 2002. — Т. 14. — С. 189—216. — URL: citeseer.ist.psu.edu/666686.html.

91. RSS-based e-learning recommendations exploiting fuzzy FCA for Knowledge Modeling / C. De Maio [и др.] // Applied Soft Computing. — 2012. — Т. 12, № 1. — С. 113—124. — URL: https://www.sciencedirect.com/science/article/pii/S1568494611003826.

92. Using Formal Concept Analysis for Discovering Knowledge Patterns / M. Rouane-Hacene [и др.] // CLA'10: 7th International Conference on Concept Lattices and Their Applications. — 2010. — Окт. — Т. 672.

93. *Cimiano, P.* Automatic Acquisition of Taxonomies from Text: FCA meets NLP / P. Cimiano, S. Staab, J. Tane // Proceedings of the ECML / PKDD Workshop on Adaptive Text Extraction and Mining. — Cavtat-Dubrovnik, Croatia, 2003. — С. 10—17. — URL: http://www.dcs.shef.ac.uk/~fabio/ATEM03/cimiano-ecml03-atem.pdf.

94. *Kuznetsov, S. O.* Machine Learning and Formal Concept Analysis / S. O. Kuznetsov // Concept Lattices / под ред. P. Eklund. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2004. — С. 287—312.

95. *Kuznetsov, S.* Machine Learning on the Basis of Formal Concept Analysis / S. Kuznetsov // Automation and Remote Control. — 2001. — Окт. — Т. 62. — С. 1543—1564.

96. *Mephu Nguifo, E.* IGLUE: A Lattice-based Constructive Induction System. / E. Mephu Nguifo, P. Njiwoua // Intelligent Data Analysis. — 2001. — Февр. — Т. 5.

97. *Rudolph, S.* Using FCA for Encoding of Closure Operators into Neural Networks / S. Rudolph // Conceptual Structures: Knowledge Architectures for Smart Applications, Proc. ICCS 2007. Т. 4604. — Berlin Heidelberg : Springer-Verlag, 07.2007. — С. 321—332. — (LNAI).

98. *Belohlavek, R.* What is a fuzzy concept lattice? / R. Belohlavek, V. Vychodil // Proc. CLA 2005, 3rd Int. Conference on Concept Lattices and Their Applications. — 2005. — Янв. — Т. 162. — С. 34—45.

99. Relational concept analysis: mining concept lattices from multi-relational data. / M. R. Hacene [и др.] // Ann. Math. Artif. Intell. — 2013. — Т. 67, № 1. — С. 81—108. — URL: http://dblp.uni-trier.de/db/journals/amai/amai67.html#HaceneHNV13.

100. *Kuznetsov, S. O.* Learning of Simple Conceptual Graphs from Positive and Negative Examples / S. O. Kuznetsov // Principles of Data Mining and Knowledge Discovery / под ред. J. M. Żytkow, J. Rauch. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. — С. 384—391.

101. *Liquiere, M.* Structural machine learning with Galois lattice and Graphs / M. Liquiere, J. Sallantin // Proc. of the 1998 Int. Conf. on Machine Learning (ICML'98. — Morgan Kaufmann, 1998. — С. 305—313.

102. *Ferré, S.* A Proposal for Extending Formal Concept Analysis to Knowledge Graphs / S. Ferré // Formal Concept Analysis / под ред. J. Baixeries, C. Sacarea, M. Ojeda-Aciego. — Cham : Springer International Publishing, 2015. — С. 271—286.

103. *Dau, F.* Concept Similarity and Related Categories in Information Retrieval using Formal Concept Analysis / F. Dau, J. Ducrou, P. Eklund // International Journal of General Systems. — 2012. — Нояб. — Т. 41.

104. *Zhao, Y.* Rough concept lattice based ontology similarity measure / Y. Zhao, W. Halang //. — 01.2006. — С. 15.

105. Graph-based Word Clustering Using a Web Search Engine / Y. Matsuo [и др.] // Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. — Sydney, Australia : Association for Computational Linguistics, 2006. — С. 542—550. — (EMNLP '06). — URL: http://dl.acm.org/citation.cfm?id=1610075.1610150.

106. *Newman, M.* Fast algorithm for detecting community structure in networks / M. Newman // Physical Review E. — 2003. — Сент. — Т. 69. — URL: http://arxiv.org/abs/cond-mat/0309508.

107. Fast unfolding of communities in large networks / V. D. Blondel [и др.] // Journal of Statistical Mechanics: Theory and Experiment. — 2008. — Окт. — Т. P10008. — С. 1—12. — URL: https://hal.archives-ouvertes.fr/hal-01146070.

108. *Dongen, S.* A Cluster Algorithm for Graphs : тех. отч. / S. Dongen. — Amsterdam, The Netherlands, The Netherlands, 2000.

109. *Pirim*, *H.* A Minimum Spanning Tree Based Clustering Algorithm for High Throughput Biological Data : дис. … канд. / Pirim Harun. — Mississippi State, MS, USA, 2011. — AAI3450335.

110. *Stanchev*, *L.* Fine-Tuning an Algorithm for Semantic Document Clustering Using a Similarity Graph / L. Stanchev // International Journal of Semantic Computing. — 2016. — Т. 10, № 04. — С. 527—555.

111. *Bai*, *Q.* Text Clustering Algorithm Based on Semantic Graph Structure / Q. Bai, C. Jin // 2016 9th International Symposium on Computational Intelligence and Design (ISCID). Т. 2. — 12.2016. — С. 312—316.

112. Semantic Word Clusters Using Signed Normalized Graph Cuts / J. Sedoc [и др.] // CoRR. — 2016. — Т. abs/1601.05403. — arXiv: 1601.05403. — URL: http://arxiv.org/abs/1601.05403.

113. Semantic Clustering and Convolutional Neural Network for Short Text Categorization / P. Wang [и др.] // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). — Beijing, China : Association for Computational Linguistics, 2015. — С. 352—357. — URL: http://www.aclweb.org/anthology/P15-2058.

114. *Chen*, *T.* XGBoost: A Scalable Tree Boosting System / T. Chen, C. Guestrin // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — San Francisco, California, USA : ACM, 2016. — С. 785—794. — (KDD '16). — URL: http://doi.acm.org/10.1145/2939672.2939785.

115. *Tang*, *J.* AMiner: Mining Deep Knowledge from Big Scholar Data / J. Tang // Proceedings of the 25th International Conference Companion on World Wide Web. — Montréal, Québec, Canada : International World Wide Web Conferences Steering Committee, 2016. — С. 373. — (WWW '16 Companion). — URL: https://doi.org/10.1145/2872518.2890513.

116. *Grover*, *A.* Node2vec: Scalable Feature Learning for Networks / A. Grover, J. Leskovec // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — San Francisco, California, USA : Association for Computing Machinery, 2016. — С. 855—864. — (KDD '16). — URL: https://doi.org/10.1145/2939672.2939754.

117. Maximizing Modularity is hard / U. Brandes [и др.] // ArXiv Physics e-prints. — 2006. — Авг. — eprint: physics/0608255.

118. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise / M. Ester [и др.] // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. — Portland, Oregon : AAAI Press, 1996. — С. 226—231. — (KDD'96). — URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

119. *Hartigan, J. A.* A k-means clustering algorithm / J. A. Hartigan, M. A. Wong // JSTOR: Applied Statistics. — 1979. — Т. 28, № 1. — С. 100—108.

120. *Borgatti, S. P.* Centrality and network flow / S. P. Borgatti // Social Networks. — 2005. — Янв. — Т. 27, № 1. — С. 55—71.

121. Translating Embeddings for Modeling Multi-relational Data / A. Bordes [и др.] // Advances in Neural Information Processing Systems 26 / под ред. C. J. C. Burges [и др.]. — Curran Associates, Inc., 2013. — С. 2787—2795.

122. Convolutional 2D Knowledge Graph Embeddings / T. Dettmers [и др.]. — 2018.

123. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network / D. Q. Nguyen [и др.] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). — New Orleans, Louisiana : Association for Computational Linguistics, 06.2018. — С. 327—333.

Работы автора по теме диссертации

Научные статьи, опубликованные в журналах RSCI

124. *Лунев, К. В.* К вычислению смысловой близости предложений / К. В. Лунев // Программная инженерия. — Москва, 2014. — № 8. — С. 30—39.

125. *Лунев*, *К. В.* Выявление тематических направлений в коллекции наборов ключевых слов / К. В. Лунев, С. А. Афонин // Программная инженерия. — Москва, 2015. — № 2. — С. 29—39.

126. *Васенин*, *В. А.* Использование наукометрических информационно-аналитических систем для автоматизации проведения конкурсных процедур на примере информационно-аналитической системы ’ИСТИНА’ / В. А. Васенин, А. А. Зензинов, К. В. Лунев // Программная инженерия. — Москва, 2016. — Т. 7, № 10. — С. 472—480.

127. *Лунев*, *К. В.* Графовые методы определения семантической близости пары ключевых слов и их применения к задаче кластеризации ключевых слов / К. В. Лунев // Программная инженерия. — Москва, 2018. — Т. 9, № 6. — С. 262—271.

128. *Лунев*, *К. В.* Алгоритм автоматизированной генерации обучающей выборки для решения задачи выявления семантической близости между парой ключевых слов методами машинного обучения / К. В. Лунев // Программная инженерия. — Москва, 2021. — Т. 12, № 6. — С. 283—293.

Другие публикации

129. Механизмы системы «ИСТИНА» для интеллектуального анализа состояния и стимулирования хода выполнения проектов в сфере науки и высшего образования / В. А. Васенин [и др.] // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23-28 сентября 2019 г., г. Новороссийск) / под ред. В. В. Воеводин. — ИПМ им. М.В.Келдыша Москва, 2019. — С. 210—221.

130. Methods for Intelligent Data Analysis Based on Keywords and Implicit Relations: The Case of “ISTINA” Data Analysis System / V. Valery [et al.] // Actual Problems of Systems and Software Engineering — APSSE 2019. — United States : United States, 2019. — P. 151—155. — (IEEE Conference Proceedings).

Приложение А

Требования к качеству программной системы анализа ключевых слов

Настоящее приложение содержит характеристики и показатели, определяющие требования, которые предъявляются к качеству разрабатываемого программного комплекса в соответствии со стандартом ГОСТ Р ИСО/МЭК 9126-93.

А.1 Функциональные требования

Система должна поддерживать следующие функциональные требования.

1. Наличие для каждого из используемых в модуле интеллектуального анализа на основе ключевых слов программного средства строго описанных алгоритмов, на которых они реализованы и моделей, в рамках которых эти алгоритмы построены:
    а) модель семантической близости пары ключевых слов;
    б) модель семантической близости наборов ключевых слов;
    в) модель семантической близости пары сущностей системы.

   Самым вычислительно сложным является построение модели близости пары ключевых слов. Каждая следующая модель обучается последовательно, поскольку в значительной степени опирается на предыдущую. Весь этап обучения является алгоритмически сложной задачей и включает в себя несколько этапов:
    - предподготовка данных;
    - построение необходимых графов из данных;
    - сбор обучающей выборки для моделей;
    - подсчет многочисленных графовых характеристик по графам для объектов обучающей выборки;
    - непосредственно обучение модели.

   Время полной подготовки данных для входной коллекции слов размером в сотни тысяч наборов ключевых слов должна занимать не более суток.

2. Эффективное обновление имеющихся и добавление новых данных в систему:
    а) добавление/модификация наборов ключевых слов;
    б) добавление/модификация дополнительных графов, связывающих сущности системы различными отношениями.

   При изменении данных системы возникает необходимость переобучения моделей близости для поддержания консистентного состояния между данными и моделями. Сложность данного пункта в том, что наивная переподготовка моделей после изменения данных может занимать продолжительное время. По этой причине возникает необходимость разработки сложных алгоритмов инкрементальной подготовки моделей, что сократит время их дообучения.
3. Эффективная процедура кластеризации ключевых слов системы и поиск необходимого кластера. Данный процесс происходит после обучения модели близости пары ключевых слов. Для коллекции, состоящий из сотен тысяч наборов ключевых слов, процесс кластеризации не должен занимать больше суток времени.
4. Эффективный поиск похожих объектов с помощью обученных моделей:
    а) поиск наиболее похожих ключевых слов к заданному;
    б) поиск сущностей, релевантных заданному набору ключевых слов;
    в) поиск набора ключевых слов, подходящих для заданной сущности.
5. Реализация подмодулей, решающих практически значимые задачи информационного поиска в рамках аналитической системы.
    а) Подмодуль поиска эксперта. Реализация функционала поиска сущностей информационной системы, релевантных поисковому запросу из ключевых слов.
    б) Подмодуль предложенных ключевых слов. Реализация функционала предложения пользователю новых ключевых слов по словам, введенным на данный момент или по имеющейся связанной информации.
6. Сбор пользовательской информации в ходе взаимодействия с комплексом.

а) Подмодуль поиска эксперта. Логирование релевантных и нерелевантных по мнению пользователя результатов.

б) Подмодуль предложенных ключевых слов. Логирование выбранных и невыбранных пользователем ключевых слов из числа предложенных.

Поиск должен выполняться как только пользователь ввел запрос и подтвердил его. Вычисление и показ результатов должны укладываться в несколько секунд. Сложность данного требования в том, что для каждого запроса необходимо подсчитать огромное число графовых характеристик и применить соответствующую предобученную графовую модель близости. В следствии этого данный пункт представляет собой сложную техническую задачу по оптимизации вычислений.

7. Система должна функционировать под управлением ОС с открытым исходным кодом.

А.2 Надежность

Следующие свойства должны быть удовлетворены.

1. Качество обученных моделей должно валидироваться на отложенных выборках после каждого изменения моделей.

а) Для каждой модели и соответствующих ей наборов тестов определяется необходимый уровень качества по выбранным метрикам и уровень производительности и величину ресурсозатратности.

б) Для каждой модели выбирается отложенное множество объектов, на которых модель применяется. При обновлении данных, переобученные модели применяются к тому же множеству объектов и автоматически проверяется, что изменения в предсказаниях оказываются ниже определенного порога. Если это условие не выполняется, то эксперту по системе необходимо детально разбираться в причинах сильных отклонений в предсказаниях. Таким образом в системе реализуется регрессионное тестирование.

2. Стабильная работа в условиях одновременного использования сотрудниками крупной организации.
3. Устойчивость к программным ошибкам и ошибкам интерфейса.

А.3 Практичность

В отношении разрабатываемого комплекса должно выполняться следующее.

1. Комплекс должен иметь простой интуитивный интерфейс для пользователя.
2. Комплекс должен быть легко читаемым и понимаемым для разработчиков.
3. Комплекс должен включать средства обратной связи пользователя с разработчиками.

А.4 Эффективность

Программный комплекс должен быть эффективен в следующих показателях.

1. Удовлетворительные показатели качества работы моделей на сильно ограниченных по объему данных.
2. Этап предподготовки комплекса:
    а) В течение одних суток:
        1) пересчет аналитических моделей определения близости ключевых слов, включая подготовку всех необходимых данных;
        2) пересбор тезауруса ключевых слов.
    б) В течение нескольких часов:
        1) обогащение наборов ключевых слов новой информацией;

2) пересчет аналитических моделей определения близости объектов информационной системы;
3) быстрое добавления новых отношений между сущностями системы.

3. Этап использования моделей:
   а) быстрое построение выдачи по пользовательскому запросу;
   б) быстрое получение кластера ключевых слов содержащее данное;
   в) быстрая реализация поисковых подсказок при вводе запроса.

A.5 Сопровождаемость

Выдвигаются следующие требования к разрабатываемому комплексу по сопровождаемости.

1. Весь комплекс архитектурно должен разбиваться на ряд отдельных модулей. Логика и параметры этих модулей системы должны быть инкапсулированы друг от друга.
2. Иметь возможность быстрого и эффективного способа расширения функционала комплекса.
3. Быть документированной.

A.6 Мобильность

Следующие свойства должны выполняться для разрабатываемого комплекса.

1. Возможность внедрения в различные информационно-аналитические системы произвольной направленности с допустимым уровнем качества моделей. Модели должны иметь возможность обучаться на данных новой системы.

2. Возможность обучения специфических моделей семантической близости, автоматически подстраиваемых к предметной области системы, в которой разворачивается комплекс.
3. Возможность обучения моделей семантической близости без имеющихся обучающих примеров.
4. Возможность внедрения в систему с дефицитом данных о ключевых словах.
5. Адаптируемость к добавлению новых сущностей и отношений между ними в системе.
6. Развертываемость комплекса внутри новой системы не должна занимать много времени работы экспертов. Необходимо лишь наладить поставку данных в нужном формате и сконфигурать модули для наиболее эффективного решения задач конкретной системы.
7. Устойчивость к пропускам и неточностям в данных.

Описанные выше требования задают специфику разрабатываемому программному комплексу. Главные особенности заключаются в следующем:

– комплекс может быть внедрен в систему, не обладающую достаточными объемами данных;
– комплекс поддерживает добавление произвольных отношений различной природы между сущностями.

Приложение Б

Самые абстрактные по смыслу слова для каждой меры центральности

Для каждого алгоритма выписаны 50 самых абстрактных ключевых слов. Жирным шрифтом выделены слова, которые, по мнению авторов, не должны попадать в список самых абстрактных в рамках исследуемого корпуса слов, т.е. ошибочно определённые слова.

– **Betweenness Centrality**: моделирование, модель, структура, оптимизация, математическая модель, математическое моделирование, управление, **мониторинг**, образование, прогнозирование, эксперимент, **прочность**, методы, методика, **самоорганизация**, история, **адаптация**, **здоровье**, **синтез**, анализ, **эффективность**, свойства, диагностика, **инновации**, **оценка**, технология, **устойчивость**, безопасность, личность, **надежность**, компьютерное моделирование, **взаимодействие**, динамика, качество, термодинамика, **плазма**, **наночастицы**, развитие, исследование, культура, **лазер**, теория, интеграция, модернизация, **деформация**, **метод конечных элементов**, **конкурентоспособность**, численное моделирование, **студенты**, алгоритм.
– **Closeness Centrality**: модель, моделирование, структура, **оптимизация**, управление, прогнозирование, методика, эксперимент, анализ, математическая модель, методы, математическое моделирование, **мониторинг**, **эффективность**, **надежность**, качество, технологии, **прочность**, расчет, **оценка**, планирование, **инновационная культура**, исследование, инновации, синтез, **устойчивость**, **взаимодействие**, образование, проектирование, безопасность, обучение, динамика, свойства, деформация, информационная система, **самоорганизация**, **инновационная деятельность**, вероятность, **профессионализм**, эксплуатация, **здоровье**, интеграция, инновационное развитие, кинетика, **температура**, **вуз**, **адаптация**, **работоспособность**, история, алгоритм.
– **Degree Centrality**: моделирование, математическая модель, математическое моделирование, **оптимизация**, модель, образование, управление, структура, мониторинг, **личность**, **прочность**, инновации, свойства, прогнозирование, **эффективность**, **синтез**, методика, культура, **метод**

конечных элементов, безопасность, **оценка**, компьютерное моделирование, **наночастицы**, **развитие**, **адаптация**, эксперимент, **студенты**, **здоровье**, качество, история, **анизотропия**, **надежность**, технология, **компетентностный подход**, **инновационная деятельность**, численное моделирование, диагностика, модернизация, разрушение, **конкурентоспособность**, творчество, интеграция, **высшая школа**, **компетенции**, **самоорганизация**, **устойчивость**, динамика, **вуз**, **остаточные напряжения**, кинетика.

– **EigenVector Centrality**: образование, управление, модель, инновации, моделирование, **эффективность**, **инновационная деятельность**, наука, **личность**, методика, **оптимизация**, модернизация, технологии, прогнозирование, мониторинг, компетенции, государство, конкурентоспособность, структура, развитие, интеграция, математическая модель, качество, **оценка**, анализ, история, высшая школа, культура, взаимодействие, студенты, **надежность**, инновационное развитие, методы, **власть**, **бизнес**, **вуз**, стратегия, **компетенция**, эксперимент, **инновационная культура**, обучение, планирование, **бакалавриат**, общество, **компетентностный подход**, **здоровье**, **инновационный потенциал**, математическое моделирование, концепция, проект.
– **PageRank Centrality**: моделирование, математическая модель, математическое моделирование, **оптимизация**, модель, образование, **мониторинг**, структура, управление, **метод конечных элементов**, прогнозирование, **прочность**, **наночастицы**, лама, компьютерное моделирование, **личность**, **эффективность**, инновации, развитие, диагностика, численное моделирование, методика, безопасность, **компетентностный подход**, культура, **синтез**, **адаптация**, свойства, **здоровье**, **оценка**, **устойчивость**, технология, **надежность**, **разрушение**, **наноструктуры**, **студенты**, интеграция, история, **роман**, динамика, **анизотропия**, **профессиональное образование**, кинетика, алгоритм, **плазма**, **вуз**, **конкурентоспособность**, качество, **качество образования**, **остаточные напряжения**, **дистанционное обучение**.

Приложение В

Найденные в коллекции документов тематические теги

Жирным шрифтом выделены те теги, которые определены верно.

эпр, медь, алтай, **аудит**, музей, поиск, **право**, доходы, охрана, смазка, стресс, тьютор, услуги, **физика**, катализ, матрица, порошок, контекст, покрытия, преграда, адсорбция, **биометрия**, коррекция, облучение, **семантика**, **кинематика**, **статистика**, предприятие, детали машин, станки с чпу, тестирование, фитопланктон, гидродинамика, дальний восток, самореализация, **конструирование**, диоксид циркония, жидкие кристаллы, пограничный слой, **факторный анализ**, **массовая культура**, преподаватель вуза, имитационная модель, управление знаниями, **нелинейные колебания**, **регрессионный анализ**, **электронное обучение**, ресурсное обеспечение, электроэнцефалограмма, **оптимальное управление**, **физическое моделирование**, образовательная программа, образовательные технологии, поддержка принятия решений, высокоскоростное соударение, **педагогическая деятельность**, международное сотрудничество, научно-образовательный центр, профессиональные компетенции, система менеджмента качества, экспериментальные исследования, **нелинейные динамические системы**, **финансово-хозяйственная деятельность**, федеральный государственный образовательный стандарт, nanoparticles.

Некоторые теги не определяют название дисциплины или направления, но по ним также можно понять тематику документа. Поэтому считается разумным отнести к правильно определенным тематическим тегам следующие:

охрана, покрытия, коррекция, облучение, детали машин, дальний восток, самореализация, управление знаниями, ресурсное обеспечение, образовательная программа, образовательные технологии, профессиональные компетенции, система менеджмента качества.

Далее представлены результаты работы программной реализации алгоритма на данных из Веб.

trade, **testing**, **principal component analysis**, mechanical properties, microstructure, heterogeneity, identification, globalization, **semantic web**, turkey, australia, sensors, information, oxidative stress, wireless sensor networks, tracking, **privacy**, **sustainable development**, **architecture**, feature extraction, obesity,

apoptosis, conservation, **pattern recognition**, **risk assessment**, **kinetics**, poverty, india, depression, **cryptography**, climate, diagnosis, virtual reality, parameter estimation, gene expression, collaboration, **policy**, chaos, detection, finite element method, breast cancer, copper, **optimal control**, algorithms, mems, memory, decomposition, concrete, xml, usa, corrosion, taxonomy, **dynamic programming**, planning, volatility, aggregation, **spectroscopy**, russia, **dynamics**, density, mobility, dna, **cfd**, **sensitivity analysis**.

Аналогично случаю с чистыми данными, можно дополнить список следующими словами:

mechanical properties, microstructure, wireless sensor networks, virtual reality.