Федеральное государственное бюджетное учреждение науки
Институт математики и механики им. Н.Н.Красовского
Уральского отделения Российской академии наук

На правах рукописи

Усталов Дмитрий Алексеевич

Модели, методы и алгоритмы построения семантической сети слов для задач обработки естественного языка

Специальность 05.13.17 —
«теоретические основы информатики»

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
кандидат технических наук
Созыкин Андрей Владимирович

Екатеринбург — 2017

Оглавление

Стр.

Стр.

Введение

Актуальность темы. Сегодня наблюдается взрывной рост количества информации, создаваемой людьми и машинами на естественном языке. Аналитическое агентство IDC прогнозирует рост совокупного объема данных, накопленных человечеством, до 163 зеттабайт к 2025 году. Основной частью таких данных являются неструктурированные данные, такие как фотографии, видеозаписи, аудиозаписи, а также тексты на естественном языке.

Язык обладает многозначностью, которая проявляется на разных уровнях: от уровня отдельных звуков в устной речи до уровня значения отдельных слов и предложений в письменном тексте. Несмотря на то, что люди хорошо справляются с разрешением многозначности самостоятельно, проблема машинного понимания естественного языка является сложной и требует специальных автоматических методов. Постоянное увеличение интенсивности потока входящей текстовой информации делает все более важной задачу математического моделирования естественного языка, в частности — русского языка.

Важнейшей проблемой является лексическая многозначность, требующая от машины понимания контекста и предметной области, в которой употребляется каждое многозначное слово. Такие сведения представляются в семантических сетях — специальных высококачественных базах знаний, представляющих машиночитаемые сведения об окружающем мире в виде понятий и связей между ними. Связи между понятиями задают семантическую иерархию, которая позволяет решать различные задачи машинного понимания естественного языка и является критически важным элементом семантических сетей. В настоящее время, наиболее известной семантической сетью в области обработки естественного языка является семантическая сеть WordNet для английского языка, связи в которой формируются между синсетами — множествами синонимов.

Семантические сети применяются при решении большого количества важнейших прикладных задач обработки естественного языка. В системах

разрешения лексической многозначности и системах машинного перевода, семантические сети представляют известные значения слов заданного языка. В вопросно-ответных системах, таких как IBM Watson, семантические сети задают сведения об объектах предметной области и связях между ними. В системах поиска сущностей, таких как Google Knowledge Graph, семантические сети представляют атрибуты, понятные и людям, и машинам. Высококачественные семантические сети широко используются в качестве золотого стандарта для оценки эффективности систем автоматической обработки естественного языка.

Создание высококачественных баз знаний вручную является длительной и ресурсоемкой задачей, поэтому исследователи уделяют большое внимание вопросу автоматического построения семантических ресурсов, таких как семантические сети. Существующие методы автоматического построения семантических сетей используют высококачественные исходные данные, что затрудняет их применение для автоматической обработки текста на языках, представляющих другие языковые группы. Например, славянских и балтийских языков. Основное внимание исследователей уделяется английскому языку, для которого сегодня доступно большое количество высококачественных баз знаний и других языковых ресурсов.

Проблема доступности и качества машиночитаемых семантических ресурсов осложняется наличием ошибок или пропущенными данными в существующих словарях. Методы машинного обучения, особенно — методы обучения без учителя, позволяют обнаруживать скрытые закономерности в неструктурированных данных. Применение таких методов может повысить полноту доступных семантических ресурсов. Таким образом, **актуальной** является задача развития методов автоматического построения семантических сетей за счет структурирования и расширения существующих слабоструктурированных словарей, не содержащих сведений о значениях слов.

Степень разработанности темы. В настоящее время наблюдается большой научный интерес к области автоматического построения семантических ресурсов, в том числе семантических сетей. Классические методы автоматического построения семантических ресурсов основаны на теоретико-графовых методах

и представлены в трудах Джона Совы (John Sowa), Эдуарда Хови (Eduard Howe), Роберто Навильи (Roberto Navigli), Патрика Пантель (Patrick Pantel), Деканга Лина (Dekang Lin), Криса Биманна (Chris Biemann), Ирины Гуревич (Iryna Gurevych), Криштианы Феллбаум (Christiane Fellbaum), Хайнриха Шютце (Hinrich Schütze). Современные методы основаны на дистрибутивных моделях и векторных представлениях слов, описанных в работах Томаса Миколова (Tomas Mikolov), Идо Дагана (Ido Dagan), Ричарда Сошера (Richard Socher), и др. Среди российских исследователей наибольший вклад в данную область внесли научные группы, возглавляемые Н. В. Лукашевич, П. И. Браславским, И. В. Азаровой, Е. В. Падучевой, С. О. Шереметьевой, Ю. А. Загорулько.

На сегодняшний день область научных исследований, связанная с автоматическим построением семантических сетей, продолжает активно развиваться. Одной из важных нерешенных проблем является задача разработки моделей, методов и алгоритмов построения семантической сети на основе слабоструктурированных языковых ресурсов без использования дополнительных высококачественных баз знаний в процессе построения.

Цель и задачи исследования. *Целью* данной работы является разработка моделей, методов и алгоритмов построения семантической сети, связывающей лексические значения слов семантическим отношением на основе материалов слабоструктурированных словарей, а также разработка на их основе комплекса программ автоматического построения такой семантической сети.

Для достижения этой цели необходимо было решить следующие *задачи*:

1. Разработать математическую модель представления лексических значений слов и связей между ними в виде семантической сети слов.
2. Разработать метод и алгоритм построения синсетов на основе разрешения многозначности слов.
3. Разработать метод и алгоритм построения и расширения однозначных семантических связей между многозначными словами.
4. Реализовать разработанные модели, методы и алгоритмы в виде комплекса программ, позволяющего построить семантическую сеть слов на основе слабоструктурированных языковых ресурсов.

5. Провести вычислительные эксперименты, подтверждающие эффективность предложенных методов.

Научная новизна работы заключается в следующем:

– разработана оригинальная модель представления значений слов и семантических связей между ними в виде семантической сети слов;
– предложены новый метод и алгоритм построения синсетов путем формирования и кластеризации вспомогательного графа значений слов;
– предложены новый метод и алгоритм построения и расширения однозначных семантических связей между многозначными словами на основе иерархических контекстов;
– разработан комплекс программ автоматического построения семантической сети слов на основе предложенных моделей, методов и алгоритмов.

Теоретическая ценность работы состоит в том, что в ней дано формальное описание методов, алгоритмов и архитектурных решений, позволяющих производить автоматическое построение семантической сети слов на основе слабоструктурированных языковых ресурсов. **Практическая ценность** работы заключается в том, что на базе разработанных моделей, методов и алгоритмов разработан комплекс программ автоматического построения семантической сети слов, позволяющий повысить полноту сведений о семантических связях. Разработанные методы, алгоритмы и программное обеспечение могут применяться для построения интеллектуальных поисковых систем, систем машинного понимания текста, систем общения, и других информационных систем, основанных на знаниях.

Методология и методы исследования. Методологической основой исследования является теория множеств и теория графов. Для построения синсетов и связывания понятий использовались методы компьютерной лингвистики и машинного обучения. При разработке комплекса программ построения семантической сети слов применялись методы объектно-ориентированного проектирования и язык UML.

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

1. Предложена модель семантической сети слов, связывающей лексические значения слов семантическим отношением.
2. Разработан метод и алгоритм построения синсетов путем формирования и кластеризации вспомогательного графа значений слов.
3. Разработан метод и алгоритм построения и расширения однозначных семантических связей между многозначными словами.
4. Выполнена реализация комплекса программ автоматического построения семантической сети слов.
5. Проведены вычислительные эксперименты, подтверждающие высокую эффективность разработанных моделей, методов и алгоритмов.

Степень достоверности результатов. Все полученные результаты подтверждаются экспериментами, проведенными в соответствии с общепринятыми стандартами.

Апробация результатов исследования. Основные положения диссертационной работы, разработанные модели, методы, алгоритмы и результаты вычислительных экспериментов докладывались автором на следующих международных научных конференциях:

- 55-я международная конференция Ассоциации по компьютерной лингвистике (ACL 2017) (30 июля – 4 августа 2017 г., Канада, г. Ванкувер);
- 23-я международная конференция по компьютерной лингвистике «Диалог 2017» (31 мая – 3 июня 2017 г., Москва);
- 15-я международная конференция европейского отделения Ассоциации по компьютерной лингвистике (EACL 2017) (3–7 апреля 2017 г., Испания, г. Валенсия);
- Открытая международная конференция ИСП РАН (1–2 декабря 2016 г., Москва);
- 17-я всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям (30 октября – 3 ноября 2016 г., Новосибирск);

– 5-я международная конференция по анализу изображений, социальных сетей и текстов (АИСТ’2016) (7–9 апреля 2016 г., Екатеринбург);
– 21-я международная конференция по компьютерной лингвистике «Диалог 2015» (27–30 мая 2015 г., Москва);
– 16-я международная суперкомпьютерная конференция «Научный сервис в сети Интернет: многообразие суперкомпьютерных миров» (22–27 сентября 2014 г., Новороссийск);
– 14-я международная конференция европейского отделения Ассоциации по компьютерной лингвистике (EACL 2014) (26–30 апреля 2014 г., Швеция, г. Гетеборг);
– 3-я международная конференция по анализу изображений, социальных сетей и текстов (АИСТ’2014) (10–12 апреля 2014 г., Екатеринбург).

Публикации соискателя по теме диссертации. Основные результаты диссертации опубликованы в следующих научных работах.

Статьи в журналах из перечня ВАК

1. *Усталов Д., Созыкин А.* Комплекс программ автоматического построения семантической сети слов // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2017. Т. 6, № 2. С. 69–83.
2. *Усталов Д.* Семантические сети и обработка естественного языка // *Открытые системы. СУБД*. 2017. № 2. С. 46–47.
3. *Усталов Д.* Обнаружение понятий в графе синонимов // *Вычислительные технологии*. 2017. Т. 22, Спецвып. 1. С. 99–112.
4. *Ustalov D.* Joining Dictionaries and Word Embeddings for Ontology Induction // *Proceedings of the Institute for System Programming*. 2016. Vol. 28, no 6. P. 197–206.

Статьи в изданиях, индексируемых в Scopus и Web of Science

5. *Ustalov D.* Expanding Hierarchical Contexts for Constructing a Semantic Word Network // Computational Linguistics and Intellectual Technologies: Papers from the Annual conference “Dialogue”. Volume 1 of 2. Computational Linguistics: Practical Applications, May 31 – June 3, 2017, Moscow, Russia. Moscow, Russia: RSUH, 2017. P. 369–381.

6. *Ustalov D., Arefyev N., Biemann C., Panchenko A.* Negative Sampling Improves Hypernymy Extraction Based on Projection Learning // Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017): Volume 2, Short Papers, April 3–7, 2017, Valencia, Spain. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. P. 543–550.
7. *Ustalov D.* Russian Thesauri as Linked Open Data // Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue". Volume 1 of 2. Main conference program, May 27–30, 2015, Moscow, Russia. Moscow, Russia: RGGU, 2015. P. 616–625.

Статьи в других изданиях

8. *Ustalov D., Panchenko A., Biemann C.* Watset: Automatic Induction of Synsets from a Graph of Synonyms // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017) (Volume 1: Long Papers), July 30 – August 4, 2017, Vancouver, BC, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. P. 1579–1590.

Свидетельства о регистрации программ для ЭВМ

9. *Усталов Д.* Свидетельство Роспатента о государственной регистрации программы для ЭВМ «Программа подбора проекционной матрицы для векторных представлений слов» № 2017615703 от 22.05.2017.

В работе 1 научному руководителю Созыкину А. В. принадлежит постановка задачи, Усталову Д. А. — все полученные результаты. В работе 6 постановка задачи принадлежит Биманну К. и Панченко А. И., результаты экспериментов по материалам англоязычных словарей принадлежат Арефьеву Н. В., разработанный метод и результаты экспериментов по материалам русскоязычных словарей принадлежат Усталову Д. А. В работе 8 результаты экспериментов по материалам англоязычных словарей принадлежат Панченко А. И. и Биманну К., все остальные результаты принадлежат Усталову Д. А.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и двух приложений. Полный объем диссертации составляет

129 страниц, включая 24 рисунка и 9 таблиц. Список литературы содержит 105 наименований.

Содержание работы. Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

Первая глава посвящена обзору работ по автоматизированному построению семантических сетей для решения задач автоматической обработки естественного языка. Перечислены трудности, возникающие при построении семантических сетей. В настоящее время не разработаны методы построения семантической сети путем интеграции существующих слабоструктурированных языковых ресурсов; существующие методы предполагают высокое качество исходных данных.

Вторая глава посвящена разработке модели семантической сети слов, методов и алгоритмов ее автоматического построения. Вводится семантическая сеть слов. Описывается оригинальный метод построения синсетов на основе графа синонимов и приводится соответствующий алгоритм. Описывается оригинальный метод построения и расширения семантических связей между значениями слов на основе иерархических контекстов и приводится соответствующий алгоритм.

Третья глава посвящена разработке архитектуры комплекса программ, реализующего предложенные модели, методы и алгоритмы. На основе предложенной архитектуры реализован комплекс программ с использованием языков программирования Python, AWK, Java. Использованы внешние библиотеки scikit-learn, Gensim, TensorFlow и Raptor. Результат работы методов представляется в формализме RDF в виде троек «субъект–предикат–объект» с использованием моделей SKOS и Lemon.

Четвертая глава посвящена проверке адекватности разработанных методов на основе сравнения полученных результатов с результатами, полученными путем использования методов, опубликованных в открытой литературе.

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

В **приложении 1** приводятся основные обозначения, используемые в диссертационной работе.

В **приложении 2** приводятся определения основных терминов, используемых в диссертационной работе.

Глава 1. Семантические сети в задачах обработки естественного языка

В данной главе рассматриваются тенденции в области обработки естественного языка и выполняется обзор научных исследований в области современных методов автоматического построения семантических сетей. Основное внимание уделяется методам автоматического построения семантических сетей и тезаурусов. Анализируются публикации, наиболее близко относящиеся к теме диссертации.

1.1. Обработка естественного языка

Обработка естественного языка — общее направление искусственного интеллекта и математической лингвистики, изучающее проблемы компьютерного анализа и синтеза естественных языков [14]. Методы обработки естественного языка лежат в основе технологий распознавания речи, информационного поиска, средств проверки правописания, систем общения, и др.

Основные трудности в обработке естественного языка вызваны проблемой *многозначности* языка [13], выражающейся на всех стадиях его обработки: от фонетического до семантического уровня с точки зрения лингвистической теории «Смысл ⇔ Текст» [12]. Таким образом, методы обработки естественного языка направлены на разрешение многозначности в различных ее проявлениях. Например, смысл предложения «Я напился из ключа.» зависит от того, в каком значении употреблено многозначное слово «ключ».

Ранние системы обработки естественного языка, возникшие в конце 40-х гг. XX века, были ориентированы на решение задачи машинного перевода и использовали большое количество *правил*, составленных людьми вручную. Успешная демонстрация автоматического перевода шестидесяти предложений из научных

статей по органической химии с русского языка на английский, проведенная компанией IBM в рамках Джорджтаунского эксперимента в 1954 г. [56], привела к существенному росту внимания к обработке естественного языка и увеличению объема финансирования исследований и разработок в этой области. Организаторы эксперимента заявляли о решении проблемы машинного перевода в течение 3–5 лет, но проблема оказалась гораздо сложнее.

В конце 60-х годов XX века развитие компьютерной лингвистики серьезно замедлилось из-за пессимистичного отчета Наблюдательного комитета по автоматической обработке языка (англ. *Automatic Language Processing Advisory Committee*, сокр. *ALPAC*) в 1966 г. В отчете заявлялось о недостаточной результативности исследований прошедших десяти лет, что привело к резкому снижению финансирования научно-исследовательских работ и стало одной из причин наступления т. н. «зимы искусственного интеллекта» [56]. Несмотря на возникший кризис завышенных ожиданий, исследования продолжались. В основе методов обработки естественного языка стали использоваться статистические модели, построенные при помощи методов машинного обучения с использованием больших коллекций документов — корпусов текста [14]. Статистический подход хорошо зарекомендовал себя; на нем основано большинство современных подходов к решению задач автоматической обработки естественного языка. Основатель распознавания речи, Фредерик Йелинек, в шутку заявлял: «Каждый раз, когда лингвист покидал коллектив, качество распознавания речи увеличивалось.»

Широкое распространение доступа в Интернет и взрывной рост популярности Всемирной паутины в 90-е гг. привели к необходимости каталогизации и систематизации информации, представленной на просторах Сети. Это привело к появлению специальных систем обработки естественного языка — поисковых машин, например Google (1998 г.) и «Яндекс» (2000 г.). Поисковые машины осуществляют обработку и индексирование опубликованных в Интернете документов с целью предоставления наиболее *релевантных* некоторому запросу, сформулированному пользователем на естественном языке [11]. Возник рынок контекстной рекламы, состоящей в показе тематических объявлений на

страницах результатов поиска. Это повысило требования к качеству поиска и способности поисковой машины учесть информационную потребность пользователя. Несмотря на то, что качество поиска зависит не только от анализа текстов, но и от моделей поведения пользователя и структуры Всемирной паутины, инвестиции в область обработки естественного языка значительно увеличились.

Сегодня технологии обработки естественного языка прочно вошли в повседневную жизнь и помогают людям лучше понимать друг друга и быстрее ориентироваться по поступающей информации. В этом помогают технологии машинного перевода, анализа эмоциональной окраски текстов, автоматического реферирования документов, распознавания и синтеза речи, и т. д. Несмотря на высокую популярность статистических методов обработки естественного языка, существуют задачи, для решения которых требуются знания об окружающем мире. Среди таких задач важно отметить разрешение лексической многозначности, построение вопросно-ответных систем, автоматическая рубрикация документов, и др. [9] Решение таких задач производится при помощи систем, основанных на знаниях. Такие системы используют специализированные ресурсы — *онтологии*.

1.2. Семантические сети

В литературе слова «семантическая сеть» и «онтология» встречаются в достаточно близких контекстах, связанных с областью инженерии знаний или различными разделами искусственного интеллекта как научной дисциплины [4]. Слово «онтология» имеет два значения:

- философская дисциплина, изучающая наиболее общие характеристики бытия;
- структура, описывающая значения элементов некоторой системы.

В данной диссертационной работе будет использоваться второе значение этого слова. Несмотря на существование большого количества определений, следующее определение будет использоваться в качестве рабочего определения [9]:

***Определение* 1.** *Онтология — это формальная теория, ограничивающая возможные концептуализации.*

Данное определение означает, что онтология задает совокупность *концептуализаций* — структур реальности, рассматриваемых независимо от предметной области и конкретной ситуации [7]. Онтология предоставляет некоторый формализм, позволяющий оперировать понятиями и высказываниями об этих понятиях. Можно выделить пять основных компонентов онтологии:

- *классы* или *понятия* — описания группы индивидуальных сущностей, объединенных на основании наличия общих свойств;
- *атрибуты* — свойства классов и экземпляров, предназначенные для хранения информации;
- *связи* — компоненты, описывающие типы взаимодействия между понятиями;
- *аксиомы* или *правила вывода* — очевидные утверждения, из которых могут быть выведены другие утверждения;
- *экземпляры* — единичные сущности, принадлежащие классам онтологии.

Среди известных работ по построению онтологий верхнего уровня стоит отметить онтологию Cyc [64], включающую как онтологию среднего уровня и онтологии нескольких предметных областей, онтологию SUMO [79], составленную из общих понятий, и др. В зависимости от задачи и предметной области, некоторые элементы могут быть опущены или же, наоборот, детализированы. Исследователи выделяют различные условные виды онтологий по степени формальности [63]:

- *словарь* — список однозначных терминов;
- *глоссарий* — словарь многозначных терминов с указанием их значений;
- *тезаурус* — глоссарий с заданной системой семантических связей;
- *формальная таксономия* — тезаурус со строгим соблюдением транзитивности родо-видовой связи;
- *формальные экземпляры* — формальная таксономия с наличием экземпляров классов;

– и т. д.

Наиболее распространенным видом онтологии в области обработки естественного языка и информационного поиска является тезаурус [9, 40]. Простейшим, но часто используемым видом онтологии, является словарь или *словник*.

***Определение* 2.** *Словник V — это множество всех лексических единиц заданного языка.*

Несмотря на близость контекстов, слова «семантическая сеть» и «онтология» обозначают два никак не связанных понятия. Одно из слов характеризует *способ представления* знаний, а другое — *способ хранения* знаний [7]. В частности, онтология задает предмет описания, в то время как семантическая сеть определяет способ представления знаний в виде ориентированного графа.

На заре инженерии знаний и обработки естественного языка под семантической сетью понимался размеченный ориентированный граф, вершины которого соответствуют некоторым сущностям (понятиям, событиям, характеристикам или значениям), а дуги выражают связи между этими сущностями [93]. Одной из первых работ, в которых фигурирует понятие, близкое к семантической сети, является работа А. М. Коллинса и М. Р. Квиллиана о семантической памяти (англ. *semantic memory*) [31]. Замечено, что люди воспринимают окружающий мир как иерархию понятий, связанных отношениями общего и частного. Например, если человек знает, что канарейка — это птица, то он сможет предположить, что у нее есть крылья. Определения других авторов хорошо согласуются с этим определением [4, 32, 94]. Исследователи выделяют шесть различных типов семантических сетей [94]. Наиболее близкими из которых к данной диссертационной работе являются *сети определений* (англ. *definitional networks*) — семантические сети, выражающие классы и подклассы понятий, связанные родо-видовым отношением (англ. *is-a*). Поскольку наиболее распространенным классом семантических отношений являются бинарные отношения [97], в качестве рабочего определения в данной работе будет использовано следующее определение:

Определение 3. *Семантическая сеть — это ориентированный граф, вершины которого — понятия, а дуги — связи между понятиями.*

Семантические сети не накладывают ограничений на структуру знаний или конкретную предметную область до тех пор, пока эти знания возможно представить в виде ориентированного графа [32,94]. Семантические сети являются одной из форм *представления* знаний. Существуют и другие формы представления знаний, такие как продукционные правила, фреймы и формальные логические модели [4]. Их рассмотрение выходит за рамки данной диссертационной работы. Таким образом, основное внимание в данной работе будет посвящено *семантическим сетям определения* как способу представления знаний и *тезаурусам* как к способу хранения знаний.

В семантических сетях представлены различные семантические отношения между понятиями. Отношения бывают симметричными и асимметричными. Примерами *симметричных* семантических отношений являются синонимия и антономия. Несмотря на важнейшую роль, которую играет отношение синонимии в лингвистике, существуют различные подходы к его определению [105]. Поскольку данное отношение является рефлексивным и симметричным, но не обязательно удовлетворяет свойству транзитивности [54], в данной работе отношение *синонимии* будет считаться отношением толерантности на словнике или на множестве лексических значений слов. Примерами *асимметричных* семантических отношений являются «часть–целое», «причина–следствие», «противоположность», и др. [2, 46].

На сегодняшний день, наиболее известной семантической сетью в области обработки естественного языка является семантическая сеть WordNet [9], построенная на основе формализации человеческого восприятия окружающего мира. На рис. 1.1 представлен пример семантической сети на основе верхнего уровня тезауруса WordNet. В семантической сети WordNet, понятия сформированы на основе отношения синонимии, поэтому называются *синсетами* (от англ. *set of synonyms* — «множество синонимов», сокр. *synset*) [40].

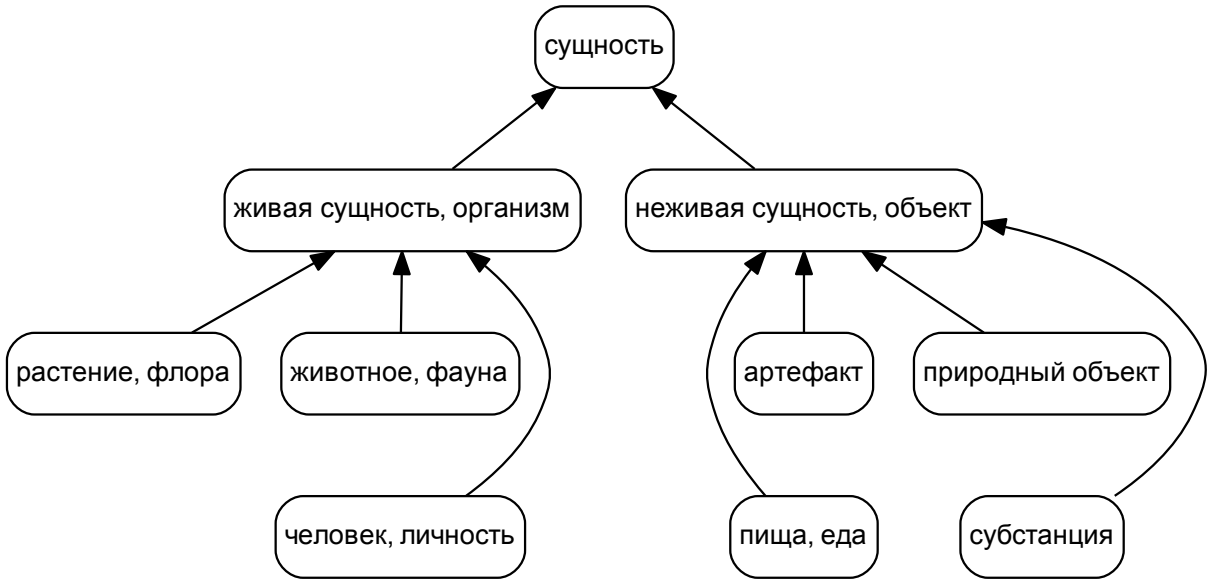


Рис. 1.1 — Верхний уровень тезауруса WordNet: понятия образованы множествами синонимов (синсетами) и связаны друг с другом при помощи асимметричного семантического отношения; исходная сеть представлена на английском языке

Основная задача, в которой широко используются семантические сети — разрешение лексической многозначности (англ. *word sense disambiguation*). Задача разрешения лексической многозначности состоит в определении конкретного значения каждого многозначного слова в заданном тексте. В этом случае семантические сети, такие как как WordNet [40] и BabelNet [76], используются как инвентарь смыслов (англ. *sense inventory*) [17, С. 229].

Наиболее известным широкой публике применением семантической сети является задача построения вопросно-ответных систем (англ. *question answering*). Система IBM Watson, известная тем, что победила профессионального игрока в *Jeopardy!* («Своя игра»), объединяет большое количество различных семантических сетей предметной области для поиска ответов на вопросы [41].

Семантические сети используются в современных методах машинного перевода. Поскольку в задаче машинного перевода необходимо не только корректно определить лексическое значение слова на исходном языке, но и корректно выбрать лексическое значение слова на целевом языке, семантические сети применяются в качестве инвентаря смыслов [78]. Кроме того, поскольку в таких

семантических сетях, как WordNet и BabelNet, к синсетам представлены толкования и примеры употребления, то эти сведения также используются для построения векторных представлений значений слов в машинном переводе [88].

Семантические сети применяются для решения задачи автоматической рубрикации текстовых документов [9]. В этом случае используются знания о синонимах и лексических отношениях, описанных в семантических сетях. На основе этих знаний производится построение классификатора при помощи машинного обучения с учителем. Классификатор используется для назначения класса ранее неизвестным документам.

Концепция Семантической паутины (англ. *Semantic Web*), предложенная в 2001 г., предполагает представление Всемирной паутины в виде большой семантической сети, в которой документы доступны в форме, удобной для чтения как человеком, так и машиной в виде автоматических агентов [24]. В настоящее время данный подход реализован лишь отчасти в виде форматов и публикуемых машиночитаемых метаданных документов. Это привело к возникновению технологии *поиска сущностей* или *семантического поиска*, когда на странице результатов поиска представляются фактографическая информация об организации, событии или личности, которой был посвящен поисковый запрос, см. Google Knowledge Graph [96].

Важнейшим применением высококачественных семантических сетей, таких как WordNet [40] и РуТез [9], является оценка систем автоматической обработки естественного языка и различных языковых ресурсов. В этом случае семантические сети используются в качестве «золотого стандарта» — общепринятого эталона, с которым выполняется сопоставление нового метода или языкового ресурса на основании некоторой заданной количественной меры качества.

Одним из применений семантических сетей, не имеющее непосредственного отношения к задачам обработки текста, является разметка объектов на фотографиях и других изображениях. В проекте ImageNet [33] объекты на изображениях выделены граничной рамкой, каждая из которых ссылается на соответствующий синсет в онтологии WordNet [40].

Как правило, высококачественные семантические сети, применяемые для решения задач обработки естественного языка, создаются коллективами экспертов-лексикографов. Процесс создания семантической сети занимает длительное время, что в свою очередь оправдывается высоким качеством результата. Известными примерами семантических ресурсов, построенных традиционным путем, являются тезаурус Роже [92] и семантическая сеть WordNet [40]. Процесс работы экспертов трудноформализуем, однако отмечаются его общие этапы [7]:

– отбираются разные примеры употребления слова в текстах;
– значения слов разделяются и группируются в понятия;
– строятся семантические отношения между понятиями.

При построении семантической сети необходимо формировать и следовать определенной концепции такого семантического ресурса [30], отвечающую на вопросы об актуальности представленной лексики и наличия неологизмов и устаревших слов, подходах к выявлению синонимии и иных семантических отношений, степени внимания к уровням онтологии, наличия определений понятий и примеров их употребления, формате словарных статей, а также других особенностях целевого языка и используемых допущениях о его строении.

Среди доступных ресурсов для русского языка важно отметить тезаурус РуТез [9], также разработанный коллективом лексикографов-экспертов, но ориентированный на решение задач информационного поиска. Некоторое время назад возникла WordNet-подобная версия данного ресурса под названием RuWordNet [67]. Кроме того, существует тезаурус Yet Another RussNet [30], построенный при помощи краудсорсинга, сочетающий как машиночитаемые данные доступных семантических словарей, так и данные, полученные от волонтеров, участвующих в сборке синсетов и отношений. Анализ современного состояния тезаурусов русского языка показывает [6], что другие ресурсы либо недоступны, либо качество представленных в них данных недостаточно высоко для применения в качестве эталона.

Достаточно важной проблемой является фактическая доступность ресурсов и особенности их лицензирования. С одной стороны, не все существующие ресурсы доступны для скачивания или использования. Например, создатели

тезауруса RussNet [2] на сегодняшний день опубликовали только часть существующих данных. С другой стороны, существуют высококачественные ресурсы, распространяющиеся на условиях открытой лицензии, такие как РуТез [9] и BabelNet [76], но эта лицензия ограничивает их коммерческое применение. Другие ресурсы, такие как Russian WordNet [21], по всей видимости, утрачены безвозвратно. Это существенно затрудняет повторное использование данных для создания производных ресурсов. Семантические сети для русского языка, находящиеся в открытом доступе в машиночитаемом виде:

– тезаурус РуТез [9], построенный коллективом экспертов-лексикографов для решения задач информационного поиска, доступный на условиях лицензии Attribution-NonCommercial-ShareAlike 3.0 Unported и включающий в себя около 31 тыс. синсетов и 110 тыс. лексем (слов и словосочетаний);
– тезаурус UNLDC [34], доступный на условиях лицензии Attribution-ShareAlike 3.0 и около 62 тыс. синсетов и 90 тыс. отношений между ними;
– тезаурус RuWordNet [67], построенный путем автоматизированного преобразования тезауруса РуТез в WordNet-подобную структуру, доступный на условиях лицензии Attribution-NonCommercial-ShareAlike 3.0 Unported и включающий в себя около 50 тыс. синсетов и 110 тыс. лексем;
– тезаурус Yet Another RussNet [30], построенный при помощи краудсорсинга, доступный на условиях лицензии Attribution-ShareAlike 3.0 и включающий в себя около 2 тыс. синсетов и 9 тыс. лексем после процедуры фильтрации, состоящей в удалении всех синсетов, имеющих менее восьми правок от участников проекта.

Основной сложностью при построении семантических сетей является большой объем работы, который необходимо выполнить для получения ресурса. Работа по созданию тезаурусов при помощи традиционного экспертного лексикографического подхода длится годы, прежде чем ресурс может применяться на практике [6]. Применение автоматизированных подходов снижает длительность процесса, но существенно повышает требования к контролю качества.

Автоматические методы построения семантической сети на основе приобретения знаний (англ. *ontology learning*, *taxonomy induction*) осуществляют извлечение онтологических понятий и связей из корпуса текстов [25,71,77]. Можно выделить общие этапы таких методов:

– предварительная графематическая и морфологическая обработка текста;
– извлечение терминов-кандидатов в понятия;
– формирование этих понятий на основе терминов;
– выявление и построение связей между понятиями;
– удаление противоречивых связей.

Примерами таких методов являются OntoLearn [75,100], OntoGain [37], TAXI [82], и др. Известны работы по выявлению значений слов путем сочетания автоматических методов и краудсорсинга, такие как TWSI [27]. Среди ранних подходов следует упомянуть шаблоны Херст (англ. *Hearst patterns*) формата «X является видом Y», до сих пор широко используемые в задачах приобретения знаний [53]. Важно отметить, что исследования последних лет показывают перспективность использования векторных представлений слов в задачах расширения онтологий [86] и обнаружения семантических отношений путем как линейного преобразования векторных представлений гипонимов [45], так и классификации синтаксических отношений [95], однако эти результаты пока не интегрированы в существующие методы. Как правило, методы на основе приобретения знаний используют для построения предметных онтологий, где неоднозначность терминов исключена.

Автоматические методы построения семантических сетей на основе интеграции знаний (англ. *knowledge integration*, *knowledge engineering*) предполагают сочетание существующих ресурсов известного качества с целью их повторного использования. Можно выделить общие этапы таких методов:

– объединение множеств понятий исходных ресурсов;
– разрешение многозначности и слияние омонимичных понятий;
– формирование связей между объединенными понятиями.

Исследователи стремятся обеспечить два свойства производных ресурсов: многоязычность и лексическое покрытие. Например, большая многоязычная семантическая сеть BabelNet [76] построена на основе англоязычного тезауруса WordNet и материалов многоязычной Википедии, и благодаря этому доступна на 271 различном языке. Другим примером многоязычной семантической сети является онтология UBY, построенная на материалах WordNet, FrameNet, OmegaWiki и Википедии для английского и немецкого языков [50]. Русский Викисловарь, формируемый сообществом добровольцев на основе материалов открытых словарей, также можно отнести к ресурсам, полученным на основе интеграции знаний [8, 30]. Кроме того, такие методы, как ECO [49], используют процедуру онтологизации [87] для выравнивания семантической сети относительно другого ресурса. Как правило, методы на основе интеграции знаний используют для связывания неструктурированных или слабоструктурированных языковых ресурсов с уже существующими семантическими ресурсами высокого качества.

1.3. Критерии качества семантических сетей

Семантические сети создаются коллективами экспертов-лексикографов или автоматически при помощи методов машинного обучения. Это существенно затрудняет оценку качества таких ресурсов, поскольку полученные семантические ресурсы должны корректно описывать окружающий мир или заданную предметную область [3]. В настоящее время интегральная оценка качества семантических сетей является открытой научной проблемой. Существует три общепринятых подхода к оценке качества семантических сетей:

– экспертная оценка;
– сопоставление с золотым стандартом;
– проведение сравнительной дорожки.

При экспертной оценке приглашенный эксперт или коллектив экспертов формирует оценку качества лексико-семантического ресурса по заранее

разработанной методологии. В этом случае каждый элемент семантической сети оценивается по заданной шкале измерений, позволяющий произвести качественную или количественную оценку [3]. Важно отметить, что в качестве экспертной оценки может использоваться коллективное суждение большого количества людей, полученное при помощи краудсорсинга [60]. При использовании краудсорсинга существует проблема надежности результатов [61], вызванная неопределенным составом участников, чьи суждения используются для оценки качества данных. Несмотря на это, краудсорсинг является популярным методом построения и оценки качества языковых ресурсов [27].

Сопоставление с золотым стандартом — самый популярный подход к оценке качества, используемый в области обработки естественного языка и информационного поиска [11]. Этот подход предполагает сравнение содержимого построенной семантической сети с материалами некоторого заранее подготовленного золотого стандарта — набора данных, обладающего известным высоким качеством. Преимущество данного подхода состоит в возможности оценки близости нового ресурса к уже существующему ресурсу на основании некоторой заданной количественной меры качества. Основная трудность при оценке семантических сетей при помощи такого подхода состоит в невозможности однозначного сопоставления понятий и связей между тестируемой онтологий и золотым стандартом. Поэтому качество понятий и связей оценивается отдельно.

В области обработки естественного языка широко используются информационно-поисковые критерии оценки качества данных [11]: точность Pr, полнота Re и их среднее гармоническое значение — F_1-мера:

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{F}_1 = 2\frac{\mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}, \tag{1.1}$$

где TP — количество верных положительных ответов, FP — количество ложных положительных ответов, FN — количество ложных отрицательных ответов.

Меры качества понятий. Задача оценки качества построения понятий в семантической сети состоит в оценке похожести двух нечетких кластеризаций одних и тех же объектов, образующих понятия, являющихся синсетами [57]. Среди

популярных подходов к оценке качества понятий можно отметить следующие подходы:

– попарная точность, полнота и F_1-мера;
– нечеткий коэффициент B^3;
– нормализованная модифицированная чистота;
– нечеткая нормализованная взаимная информация.

Попарная точность, полнота и F_1-мера (англ. *paired F_1-score*), основанные на сопоставлении пар объектов, образующих понятия [55, 69]. В этом случае оцениваемый набор данных K и золотой стандарт G преобразуются во множества пар слов. Каждый синсет, содержащий n слов, порождает $\frac{n(n-1)}{2}$ пар слов. Это позволяет определить положительные и отрицательные ответы следующим образом:

$$\text{TP} = |K \cup G|, \qquad \text{FP} = |K \setminus G|, \qquad \text{FN} = |G \setminus K|, \tag{1.2}$$

где TP — количество пар слов, одновременно присутствующих в оцениваемом наборе данных и в золотом стандарте, FP — количество пар слов, отсутствующих в золотом стандарте, но присутствующих в оцениваемом наборе данных, FN — количество пар слов, отсутствующих в оцениваемом наборе данных, но присутствующих в золотом стандарте. Проблема использования данной меры качества состоит в том, что она не учитывает внутреннюю структуру синсетов, а только попарную встречаемость слов в этих синсетах. Эта проблема решена в других упомянутых выше мера качества понятий. В свою очередь, эта мера является легко интерпретируемой: она принимает нулевое значение при отсутствии совпадений и она равна единице при совпадении каждой пары элементов между двумя кластеризациями.

Нечеткий коэффициент B^3 (англ. *fuzzy B^3*) [57] является модификацией коэффициента B^3, используемого для оценки качества жесткой кластеризации [20]. Данный подход предполагает вычисление F_1-меры на основе критериев точности (Pr) и полноты (Re), модифицированных для задачи сравнения двух нечетких кластеризаций X и Y:

$$\text{Pr} = \text{avg}_i \, \text{avg}_{j \neq i \in \mu_Y(i)} \, P(i, j, X), \tag{1.3}$$

$$\text{Re} = \text{avg}_i \, \text{avg}_{j \neq i \in \mu_X(i)} \, R(i, j, X), \tag{1.4}$$

где $\mu_X(i)$ — множество кластеров в наборе данных X, содержащих элемент i, $P(i,j,X)$ — поэлементная функция точности для элементов i и j, $R(i,j,X)$ — поэлементная функция полноты для элементов i и j, avg — оператор усреднения. Для определения поэлементных функций точности и полноты используется следующая функция корректности:

$$C(i,j,X) = \sum_{k \in \mu_X(i) \cup \mu_X(j)} 1 - |w_k(i) - w_k(j)|, \tag{1.5}$$

где $\mu_X(i)$ — множество кластеров в наборе данных X, содержащих элемент i, $w_k(i)$ — вес элемента i в кластере k в наборе данных X. Таким образом, поэлементные функции полноты и точности для элементов i и j имеют следующий вид:

$$P(i,j,X) = \frac{\min(C(i,j,X), C(i,j,Y))}{C(i,j,X)}, \tag{1.6}$$

$$R(i,j,X) = \frac{\min(C(i,j,X), C(i,j,Y))}{C(i,j,Y)}. \tag{1.7}$$

Известно, что максимальное значение нечеткого коэффициента B^3 достигается в вырожденном случае, когда все имеющиеся элементы находятся в одном и том же единственном кластере [22]. Это не позволяет использовать данную меру качества для исследования эффективности методов построения понятий. Несмотря на это, применение данной меры качества полезно для получения интегральной оценки однородности понятий.

Нормализованная модифицированная чистота (англ. *normalized modified purity*) [59] является модификацией классического метода вычисления чистоты кластеризации [11], используемого в информационном поиске. Данный подход предполагает вычисление F_1-меры на основе критериев нормализованной модифицированной чистоты и нормализованной обратной частоты. Нормализованная модифицированная чистота для оцениваемого набора данных K и золотого стандарта G вычисляется следующим образом:

$$\text{nmPU} = \frac{1}{N} \sum_{i:|K_i|>1} \max_j \delta_{K_i}(K_i \cap G_j), \tag{1.8}$$

$$\text{niPU} = \frac{1}{N} \sum_j \max_i \delta_{G_j}(K_i \cap G_j), \tag{1.9}$$

где N — количество уникальных элементов в кластерах оцениваемого набора данных K, $|K_i|$ — размер кластера K_i. Функция $\delta_{K_i}(K_i \cap G_j)$ выражает сумму элементов кластера K_i:

$$\delta_{K_i}(K_i \cap G_j) = \sum_{v \in K_i \cap G_j} c_{iv}, \tag{1.10}$$

где c_{iv} — v-й компонент кластера K_i. Проблема использования критерия чистоты кластеров состоит в том, что максимальное значение данной меры качества достигается в случае, когда каждый элемент находится в отдельном кластере [11]. Это не позволяет использовать данную меру качества для исследования эффективности методов построения понятий. Несмотря на это, применение данной меры качества полезно для получения интегральной оценки атомарности понятий.

Нечеткая нормализованная взаимная информация (англ. *fuzzy normalized mutual information*) [57] является модификацией классического метода вычисления нормализованной взаимной информации [11], используемого в информационном поиске. Нечеткая нормализованная взаимная информация также допускает включение одного объекта в один или более кластеров. Известно, что максимальное значение нормализованной взаимной информации достигается в вырожденном случае, когда каждый элемент находится в отдельном кластере [22], который не пересекается с другими кластерами. Это не позволяет использовать данную меру качества для исследования эффективности методов построения понятий по тем же причинам, что и критерий нормализованной модифицированной чистоты.

Меры качества связей. Задача оценки качества построения связей в семантической сети состоит в сравнении похожести двух иерархий, связывающие одни и те же объекты [29]. В настоящее время не существует общепринятого подхода к оценке семантических связей. Среди доступных подходов к оценке качества связей можно отметить следующие подходы:

– попарная точность, полнота, F_1-мера;
– точность, полнота и F_1-мера на основе путей в графе;
– кумулятивный коэффициент Фоулкса–Мэллоуса.

Попарная точность, полнота и F_1-мера предполагает сравнение ребер двух иерархий аналогичным образом, как это делается при вычислении попарной точности, полноты и F_1-меры для оценки качества понятий [29, 68]. В частности, пусть оцениваемая иерархия представлена в виде ориентированного графа $H = (V_H, E_H)$, где V_H — множество вершин, E_H — множество дуг; золотой стандарт представлен в виде графа $G = (V_G, E_G)$, где V_G — множество вершин, E_G — множество дуг. Тогда положительные и отрицательные ответы определяются следующим образом:

$$\text{TP} = |E_H \cup E_G|, \qquad \text{FP} = |E_H \setminus E_G|, \qquad \text{FN} = |E_G \setminus E_H|, \tag{1.11}$$

где TP — количество пар слов, направленный путь между которыми одновременно присутствует в оцениваемом наборе данных и в золотом стандарте, FP — количество пар слов, направленный путь между которыми отсутствует в золотом стандарте, но присутствует в оцениваемом наборе данных, FN — количество пар слов, направленный путь между которыми отсутствует в оцениваемом наборе данных, но присутствует в золотом стандарте. Основной проблемой попарной точности, полноты и F_1-мера при оценке качества построения семантических связей является игнорирование транзитивности таких связей. Если пара слов в оцениваемом наборе данных связана напрямую, а в золотом стандарте эта пара слов связана через промежуточное слово, то данный подход к оценке качества некорректно засчитает такой случай в качестве ложного отрицательного ответа.

Подход на основе точности, полноты и F_1-меры на основе проверки существования путей в графе учитывают транзитивность связей [39, 58, 87]. При использовании такого подхода для каждой пары ребер в оцениваемом ресурсе и золотом стандарте проверяется существование ориентированного пути от одной вершины к другой вместо проверки существования отдельных ребер из оцениваемой иерархии в составе иерархии золотого стандарта:

$$\text{TP} = |(u, v) \in H : \exists u \xrightarrow{G} v|, \tag{1.12}$$

$$\text{FP} = |(u, v) \in H : \nexists u \xrightarrow{G} v|, \tag{1.13}$$

$$\text{FN} = |(u, v) \in G : \nexists u \xrightarrow{H} v|, \tag{1.14}$$

где TP — количество верных положительных ответов, FP — количество ложных положительных ответов, FN — количество ложных отрицательных ответов, $u \xrightarrow{G} v$ означает ориентированный путь от вершины u до вершины v в графе G. Проблема данного подхода состоит в том, что он занижает количество ложных отрицательных срабатываний в случае, когда пары слов в золотом стандарте расположены в оцениваемой иерархии на большом расстоянии.

Кумулятивный коэффициент Фоулкса–Мэллоуса (англ. *cumulative Fowlkes–Mallows index*) [100] является модификацией коэффициента Фоулкса–Мэллоуса, предназначенного для сравнения методов иерархической кластеризации [42]. Сравнение двух иерархий C_1 и C_2 при помощи кумулятивного коэффициента Фоулкса–Мэллоуса производится путем оценки близости этих иерархий на i-м уровне, записываемом как C_1^i и C_2^i, соответственно:

$$B_{1,2}^i = \frac{n_{11}^i}{\sqrt{(n_{11}^i + n_{10}^i) \cdot (n_{11}^i + n_{01}^i)}}, \tag{1.15}$$

где n_{11}^i — количество пар объектов, находящихся в одном кластере в C_1^i и C_2^i, n_{10}^i — количество пар объектов, находящихся в одном кластере в C_1^i, но в разных кластерах в C_2^i, n_{01}^i — количество пар объектов, находящихся в одном кластере в C_2^i, но в разных кластерах в C_1^i. Кумулятивный индекс получается путем суммирования данной меры качества на каждом уровне $0 \leqslant i < k$:

$$B_{1,2} = \frac{\sum_{i=0}^{k-1} \frac{i+1}{k} B_{1,2}^i}{\frac{k+1}{2}}, \tag{1.16}$$

где $k = \max(k_1, k_2)$ — максимальная глубина сравниваемых иерархий C_1 и C_2. Таким образом, как и подход на основе проверки существования пути в графе, подход к оценке качества построения семантической сети на основе кумулятивного коэффициента Фоулкса–Мэллоуса учитывает транзитивность семантических связей. Для вычисления каждого уровня производится поиск в глубину [98], причем каждый последующий уровень включает в себя содержимое всех предыдущих уровней. Проблема данного подхода состоит в высокой ресурсоемкости, что делает возможным его применение только на семантических сетях небольшого размера.

Следует отметить, что при оценке методов, возвращающих несколько ответов, используется мера качества hit@k — доля объектов, для которых метод вернул хотя бы один корректный ответ среди первых k ответов [44]:

$$\text{hit@}k = \frac{\sum_{(x,y)\in R} \mathbb{1}_{f_k(x)}(y)}{|R|}, \tag{1.17}$$

где R — выборка, x — входной параметр метода $f_k(x)$, y — корректный ответ, k — количество ответов, возвращаемых методом, $\mathbb{1}_{f_k(x)}$ — индикаторная функция множества ответов $f_k(x)$.

Популярным, но не всегда доступным вариантом оценки качества семантической сети, является сравнение методов по материалам специально подготовленной «дорожки» (англ. *shared task*). Наиболее популярным подходом к оценке качества семантических сетей является выполнение дорожки по разрешению семантической многозначности [57, 69, 77], когда на единой коллекции документов необходимо правильно указать номер значения каждого выделенного слова. Таким образом, измеряется пригодность использования семантической сети для решения практической задачи. В качестве количественных критериев качества применяются те же критерии, что и при сопоставлении с золотым стандартом. Проведение такой дорожки требует длительной подготовки, разработки методологии оценки и разметку проверочных наборов данных.

Эксперименты, представленные в данной работе, используют материалы трех различных семантических ресурсов для русского языка: RuWordNet [67] и Yet Another RussNet [30]. При использовании методов машинного обучения с учителем в литературе широко применяется подход на основе t-критерия равенства средних [102]. При использовании методов машинного обучения без учителя нет возможности примнения этого критерия, поскольку кластеризация данных выполняется на том же наборе данных, что и оценка качества. В данной работе для оценки статистической значимости результатов оценки методов машинного обучения без учителя будет использоваться подход на основе пословной оценки [91]. Меры качества, такие как точность, полнота и F_1-мера, вычисляются для каждого слова отдельно. Затем формируется выборка, каждый элемент которой соответствует значению выбранной меры качества, вычисленной относительно каждого

слова. Затем используется знаковый ранговый критерий Уилкоксона для сравнения равенства средних значений между этими выборками [104].

1.4. Обзор работ по теме диссертации

В данном разделе выполняется обзор работ, наиболее близко относящихся к теме диссертации. Некоторые описанные методы производят только построение понятий, причем только наиболее распространенного типа понятий — синсетов [40]. Некоторые другие рассмотренные методы осуществляют только построение асимметричных семантических связей. Представленные в обзоре методы требуют предварительного подбора гиперпараметров под целевой язык и конкретный набор данных. Под *гиперпараметром* в данной работе подразумевается параметр метода машинного обучения, для вычисления которого отсутствует аналитическая формула [62, С. 64–65].

Метод вывода значений слова при помощи жесткой кластеризации графа. В работе [36] предложен подход, позволяющий обнаружить значения неизвестного слова u на основе анализа окрестности вершины графа, соответствующей этому слову. Сначала из исходного графа $G = (V, E)$ извлекается *эго-сеть* $G_u = (V_u, E_u)$ — окрестность вершины $u \in V$ в графе G [43]. Затем из графа G_u исключается вершина u и производится кластеризация данного графа при помощи какого-либо метода жесткой кластеризации графа [26]. Каждый полученный кластер в результате кластеризации соответствует отдельному лексическому значению заданного слова $u \in V$.

Данный подход широко используется в теоретико-графовых методах вывода значений слова [71, 77]. Двумя наиболее популярными алгоритмами являются марковский алгоритм кластеризации [35] и алгоритм испорченного телефона [26]. Оба алгоритма принимают на вход неориентированный взвешенный граф и возвращают множество непересекающихся кластеров.

Марковский алгоритм кластеризации (англ. *Markov Clustering*, сокр. *MCL*) — алгоритм жесткой кластеризации взвешенного графа, основанный на моделировании потока в графе [35]. Данный алгоритм выполняет две операции: расширение (англ. *expansion*) и накачивание (англ. *inflation*). Алгоритм основан на двух предположениях:

– вершины, относящиеся к одному кластеру, соединены большим количеством ребер, поэтому вес этих ребер будет выше, чем вес ребер, соединяющих отдельные кластеры;
– случайный обход графа будет, как правило, проходить по вершинам одного кластера и редко выходить за его пределы.

Алгоритм испорченного телефона (англ. *Chinese Whispers*, сокр. *CW*) — простой алгоритм жесткой кластеризации взвешенного графа [26]. Алгоритм состоит из двух этапов:

– на этапе инициализации, каждой вершине графа назначается метка — уникальное случайное число;
– на этапе обновления меток, до тех пор, пока метки вершин изменяются, для каждой вершины графа выбирается наибольшее значение метки среди меток смежных вершин; полученные метки используются в качестве идентификаторов кластеров.

Метод вывода значений слова при помощи жесткой кластеризации графа позволяет обнаружить значения многозначных слов, но не позволяет объединить эти значения в синсеты. Несмотря на этот недостаток, данный метод широко используется для построения инвентарей смыслов, используемых для решения важной обратной задачи — разрешения лексической многозначности [77].

Стоит отметить, что метод вывода значений слова при помощи жесткой кластеризации графа не накладывает ограничений на тип ребер в исходном графе до тех пор, пока эти ребра порождены симметричным семантическим отношением. Известно, что этот метод показывает хорошие результаты и при использовании семантически связанных слов, не являющихся синонимами [83].

Метод нечеткой кластеризации MaxMax. В работе [55] представлен MaxMax — метод нечеткой кластеризации, позволяющей каждой вершине графа являться элементом одного или нескольких кластеров. Слова, оказавшиеся в составе нескольких кластеров, неявным образом получают метки различных значений исходного слова. Несмотря на то, что метод MaxMax является специализированным методом кластеризации для графов совместной встречаемости слов, авторы не накладывают ограничений на его использования для обработки других типов графов. Метод MaxMax включает два основных шага:

- построение вспомогательного ориентированного графа $G' = (V, E')$ на основе исходного неориентированного взвешенного графа $G = (V, E)$;
- извлечение пересекающихся кластеров из вспомогательного графа G'.

Авторы алгоритма MaxMax используют понятие *максимально близкой вершины*: вершина $v \in V$ является максимально близкой вершиной для вершины $u \in V$, если среди всех вершин, смежных с u, вес $\text{weight}(u, v) \in \mathbb{R}$ максимален. Это позволяет использовать два предположения:

- если u является максимально близкой вершиной для v, то из этого не следует, что v является максимально близкой вершиной для u;
- пара вершин $\{u, v\}$ находится в одном кластере тогда и только тогда, когда u является максимально близкой вершиной для v и v является максимально близкой вершиной для u.

Таким образом, на этапе *преобразования графа* производится построение такого ориентированного невзвешенного графа, что множество его вершин совпадает с множеством вершин исходного графа, а множество ребер формируется путем подбора максимально близкой вершины для каждой вершины.

В полученном ориентированном графе выполняется *извлечение кластеров*. Для этого сначала все вершины помечаются как *корневые*. Затем, для каждой вершины $u \in V$, все вершины, достижимые из нее, помечаются как *не корневые*. В результате такой операции получаются кластеры, которые можно извлечь из ориентированного графа при помощи запуска алгоритма поиска в глубину [98] из каждой корневой вершины.

В отличие от метода вывода значений слова при помощи жесткой кластеризации графа, метод MaxMax естественным образом производит и вывод значений слов, и объединение этих слов в синсеты. Присутствие слова в составе нескольких различных кластеров интерпретируется как наличие нескольких значений данного слова, соответствующих количеству содержащих его кластеров. Недостатком этого метода является зависимость от подхода к взвешивания графа, поскольку данный метод спроектирован с учетом неравномерного распределения весов ребер.

Метод перколяции клик. В работе [80] предложен метод перколяции клик (англ. *Clique Percolation Method*, сокр. *CPM*) — метод нечеткой кластеризации невзвешенного ориентированного графа. Этот метод широко используется в области анализа социальных сетей для поиска пересекающихся сообществ (кластеров).

Поскольку множество синонимов, выражающих одно и то же понятие, образует связанное скопление вершин в графе синонимов [47, 58], то данный метод является подходящим для построения синсетов в графе. Метод перколяции клик формирует кластеры путем обнаружения в графе k-клик, соответствующим полносвязным подграфам, содержащим $k \in \mathcal{N}$ вершин. Две k-клики считаются смежными при условии, что у них есть $(k-1)$ общих вершин. Кластер определяется как максимальное объединение k-клик, которые достижимы друг от друга через последовательности из смежных k-клик.

Недостатком метода перколяции клик является труднодостижимое на практике допущение о том, что связанные скопления вершин образуют полные клики в графе синонимов. В методе не предусмотрено средств для восстановления пропущенных ребер в исходном графе.

Метод построения связей на основе лексико-синтаксических шаблонов. В работе [53] предложен простой и широко используемый подход извлечения из неструктурированных текстов связей между словами $R \subset V \times V$ на основе лексико-синтаксических шаблонов, также известный как *шаблоны Херст*

(англ. *Hearst patterns*). Данный подход предполагает извлечение из коллекции документов упорядоченных пар слов $(w, h) \in R$, соответствующих множеству заранее составленных шаблонов следующего вида:

– «w является видом h»;
– «такие w, как h»;
– и т. д.

Первоначально, данный метод применялся для обработки текстов на английском языке и заслужил высокую популярность. К сожалению, этот метод возвращает ненадежные результаты, содержащие ошибки. При этом часто встречающиеся пары слов, извлеченные при помощи лексико-синтаксических шаблонов, выражают устойчивые связи между словами [81]. Поэтому использование такого подхода в общем виде требует исключения редко встречающихся пар слов. Это производится при помощи простого порогового фильтра, который удаляет из набора данных таких пар слов, которые встретились меньше, чем заданное количество раз. Пороговое значение зависит от конкретной коллекции документов и подбирается индивидуально.

Обработка флективных языков, таких как русский язык, при помощи подобных лексико-синтаксических шаблонов осложняется тем, что формы слов изменяются при согласовании предложений. Существует три различных подхода к решению этой проблемы:

– использование более высоких значений порогового фильтра;
– применение методов, преобразующих каждую форму слова в тексте в исходную форму слова — *лемму*;
– разработка специализированных лексико-синтаксических шаблонов.

В работе [5] предложены специализированные лексико-синтаксические шаблоны, позволяющие извлекать родо-видовые связи из словарных определений — толкований Малого академического словаря под ред. А. П. Евгеньевой [15]. Использование специализированных шаблонов позволяет извлечь достоверные связи между парами слов, но разработкатаких шаблонов требует большего времени на адаптацию под выбранную текстовую коллекцию, по сравнению с применением лексико-синтаксических шаблонов общего назначения.

Общей проблемой метода построения связей на основе лексико-синтаксических шаблонов является *разреженность* данных. Например, если пара слов $(\textit{кот}, \textit{животное})$ присутствует в коллекции документов, а пара слов $(\textit{кошка}, \textit{животное})$ нет, то не существует возможности восстановить связь между словами, близкими по значению без использования внешних семантических ресурсов.

Метод построения связей при помощи линейного преобразования векторных представлений слов. В работе [45] предложен подход к построению асимметричных связей между словами на основе подбора матрицы линейного преобразования векторных представлений слов. Пусть задано асимметричное отношение $R \subset V \times V$ и каждому слову $w \in V$ ставится в однозначное соответствие вектор $\vec{w} : w \to \mathbb{R}^d, d \in \mathbb{N}, |V| \gg d$. Примером такого преобразования являются дистрибутивные модели, такие как Skip-gram [74]. Особенность такого векторного представления слов состоит в том, что слова, близкие по смыслу, находятся близко, а слова с разными смыслами находятся друг от друга далеко.

С целью учета неоднородностей векторного пространства $\mathbb{R}^d$, производится предварительная кластеризация этого пространства на $k \in \mathbb{N}$ кластеров при помощи алгоритма k-средних [52] с использованием смещений векторов $(\vec{h} - \vec{w}), \forall (w, h) \in R$. Пусть R_i — подмножество обучающей выборки, соответствующее i-му кластеру. Для i-го кластера независимым образом подбираются значения элементов матрицы Φ_i^*:

$$\Phi_i^* \in \arg\min_{\Phi_i} \left(\frac{1}{|R_i|} \sum_{(\vec{w},\vec{h}) \in R_i} \left\| \Phi_i \vec{w} - \vec{h} \right\|^2 \right), \tag{1.18}$$

где $1 \leqslant i \leqslant k$ — номер кластера, $\vec{w}$ — вектор нижестоящего слова w, $\vec{h}$ — вектор вышестоящего слова h, Φ_i — матрица линейного преобразования для кластера k.

Далее производится построение связей на основе полученного семейства из k матриц линейного пребразования. Это выполняется исходя из допущения о том, что $\Phi^* \vec{w} = \vec{h}, \forall (w, h) \in R$. Таким образом, считается, что упорядоченная пара слов $(w, h), w \in V, h \in V$ порождена асимметричным семантическим

отношением R, если результат умножения матрицы на векторное представление нижестоящего слова находится на некотором евклидовом расстоянии $\delta \in \mathbb{R}^+$ от векторного представления вышестоящего слова: $\|\Phi^* \vec{w} - \vec{h}\| < \delta$.

Основная проблема данного подхода состоит в том, что он явным образом не учитывает многозначность слов, если при использовании метода не предоставляется гарантия того, что все слова в словнике являются однозначными. Кроме того, данный подход также явным образом не учитывает информацию об асимметричности связи между словами в минимизируемую функцию (1.18).

Метод построения связей при помощи краудсорсинга. В работе [8] предложен метод построения семантических связей путем разбора слабоструктурированных материалов Викисловаря [103]. Викисловарь — это большой многоязычный семантический ресурс, доступный в том числе на русском языке. Построением Викисловаря занимается большое количество участников-редакторов, широко применяющих материалы других словарей и иных источников для оперативного реагирования на изменения языка [72]. Русский Викисловарь пополняется на основе материалов существующих электронных словарей русского языка [8], при этом участниками ресурса производится очистка и дооформление данных. Викисловарь и его машиночитаемая версия широко используется для решения задач, требующих сведений о семантических отношениях [99].

Важной проблемой Русского Викисловаря как семантической сети является отсутствие объединения слов в синсеты, поэтому семантические связи в этом ресурсе сформированы только между отдельными словами. Другой важной проблемой всех ресурсов, создаваемых волонтерами в сети Интернет с использованием т. н. «вики-подхода» является достоверность сведений и необходимости обеспечения качества правок, вносимых участниками [61]. Эксперимент по построению семантической сети русского языка Yet Another RussNet путем интеграции материалов русскоязычных тезаурусов с применением совместного редактирования выявил две основные проблемы данного подхода [30]:

– при недостаточном патрулировании правок участники игнорируют результаты друг друга и создают понятия-дубликаты;

– без проведения тщательного инструктажа участники не всегда различают тонкости построения связей между словами.

Метод «Извлечение, кластеризация, онтологизация». В работе [49] предложен метод «Извлечение, кластеризация, онтологизация» (англ. *Extraction, Clustering, Ontologisation*, сокр. *ECO*) для автоматического построения WordNet-подобного тезауруса на основе извлечения семантических отношений из корпуса текстов и использовании такой информации для обогащения существующей семантической сети. Метод ECO предполагает три этапа обработки данных:

– извлечение экземпляров семантических онтошений между лексическими входами;
– кластеризация синонимичных лексических входов, полученных из отношений, для получения синсетов;
– связывание синсетов («онтологизация») путем установления семантических отношений между ними [87].

Исходными данными в методе ECO являются словарные определения. На этапе извлечения отношений при помощи шаблонов Херст [53] извлекаются такие семантические отношения, как синонимия, гипо-/гиперонимия, причина-следствие, атрибут. Затем связи между синонимами используются на этапе кластеризации для построения синсетов, а другие отношения используются для построения связей между синсетами.

На этапе кластеризации сведения о синонимах используются для построения графа синонимов, вершинами которого являются слова, а ребрами — пары слов, связанные отношением синонимии. Результатом данного этапа является нечеткая кластеризация графа, допускающая включение вершин в несколько различных кластеров. Это позволяет учесть многозначность слов. Подход к кластеризации в методе ECO основан на марковском алгоритме кластеризации [35], который является алгоритмом жесткой кластеризации графа, определяющим каждую вершину в единственный кластер.

***Определение* 4.** *Граф синонимов $W = (V, E)$ — это неориентированный взвешенный граф, множество вершин V которого является словником, а множество ребер E порождается отношением синонимии на словнике.*

Несмотря на существование подходов, позволяющих произвести при помощи такого алгоритма вывод отдельных значений слов [26, 36], при построении тезауруса Onto.PT используется собственный подход к кластеризации [48]. В вес каждого ребра графа синонимов вносится случайный шум, после чего производится кластеризация графа при помощи марковского алгоритма кластеризации [35]; полученные кластеры сохраняются. Такая операция производится тридцать раз. Затем для каждой пары слов оценивается вероятность попадания в один и тот же кластер и производится формирование синсетов на основе пар слов, вероятность попадания в один кластер для которых превышает некоторый заданный порог.

Синсеты, полученные на этапе кластеризации, связываются друг с другом при помощи процедуры *онтологизации* [87]. Данная процедура направлена на разрешение многозначности значений слов в семантических отношениях, полученных на этапе извлечения. Для этого оценивается близость значений слов с учетом заданного семантического отношения по материалам существующего золотого стандарта. Выполнение такой процедуры формирует семантическую иерархию синсетов.

Несмотря на то, что метод ECO позволяет построить семантическую сеть в условиях ограниченности доступных языковых ресурсов, данный метод имеет ограничения: на этапе извлечения используются толковые словари высокого качества, доступные на коммерческой основе, отсутствует достоверное подтверждение эффективности используемого метода нечеткой кластеризации графа, а этап онтологизации предполагает применение синсетов из сторонних высококачественных ресурсов, построенных с использованием традиционного экспертного лексикографического подхода [49].

1.5. Выводы по главе 1

В главе 1 выполнен обзор современного состояния исследований и разработок в области построения семантических сетей для задач обработки естественного языка. На сегодняшний день существует большое количество актуальных задач машинного понимания текста на естественном языке, использующих системы, основанные на знаниях. Среди таких задач следует отметить разрешение лексической многозначности, построение вопросно-ответных систем, объектный поиск. Однако для использования таких методов требуется наличие семантических ресурсов, таких как семантические сети.

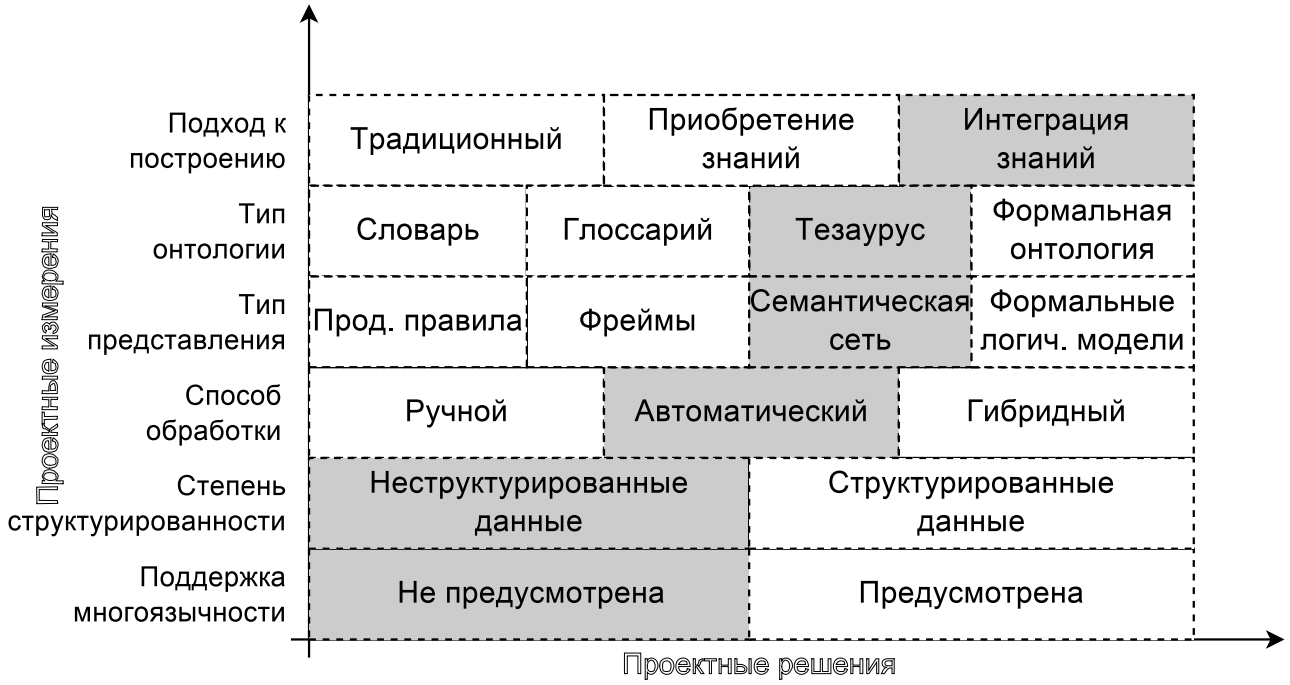


Рис. 1.2 — Проектные решения и измерения

Существующие методы автоматического и автоматизированного построения семантических сетей предполагают либо наличие доступных готовых ресурсов высокого качества для интеграции (ECO [49]), либо формируют только понятия (вывод значений слов [26, 35, 36], MaxMax [55], CPM [80]), либо формируют только семантические отношения (лексико-синтаксические шаблоны [5, 53], подбор матрицы линейного преобразования [45]). Кроме того, современные методы построения связей между понятиями основаны на построении семантической

иерархии и ее выравнивании относительно готовой иерархии [87], в том числе построенной для другого языка [76]. Это вызвано невозможностью формирования такой структуры автоматическим путем без внешних источников из-за необходимости связать знания об объектах окружающего мира.

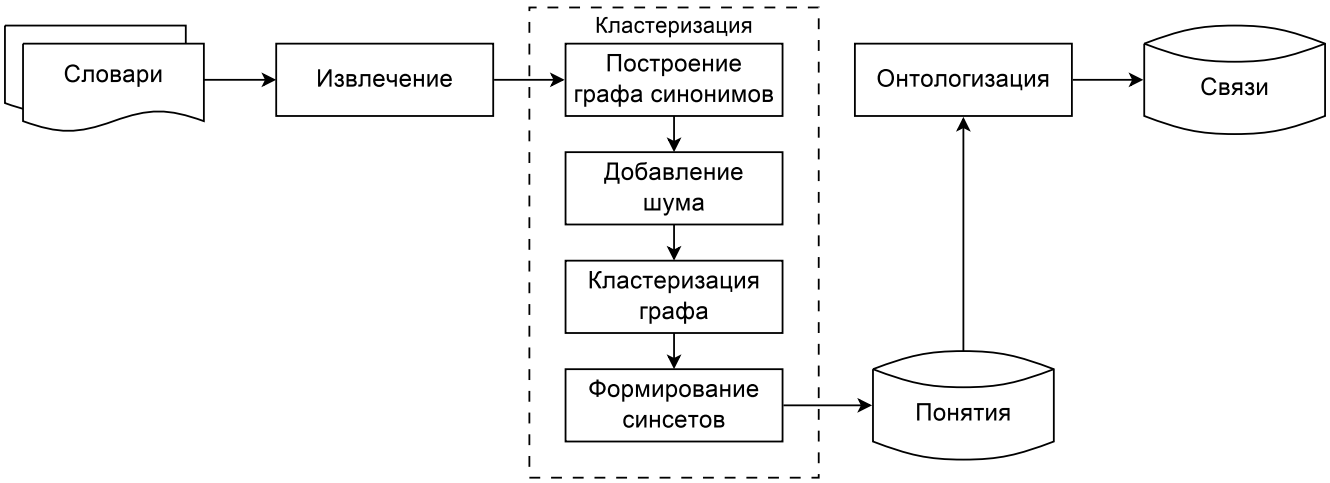


Рис. 1.3 — Общая схема метода «Извлечение, кластеризация, онтологизация»

На основании обзора в разделе 1.4 построена матрица проектных решений, представленная на рис. 1.2. Наиболее близким методом из предложенных является метод «Извлечение, кластеризация, онтологизация» (англ. *ECO*), представленный на рис. 1.3. Ключевое достоинство метода ECO состоит в том, что данный метод осуществляет и построение понятий, и построение отношений между ними на основе слабоструктурированных словарей. В свою очередь, выделены две основные проблемы метода ECO:

– отсутствие достоверного подтверждения эффективности используемого метода нечеткой кластеризации графа;
– зависимость от существующих семантических ресурсов высокого качества при выполнении процедуры онтологизации, формирующей связи между синсетами [87].

Решение обозначенных выше проблем делает актуальной задачу разработки новых моделей, методов и алгоритмов построения семантической сети на основе материалов слабоструктурированных словарей для задач обработки естественного языка.

Глава 2. Методы построения семантической сети слов

Семантические связи в традиционных семантических сетях формируются между отдельными синсетами [9, 40, 76]. Автоматическое построение семантических связей является трудноразрешимой задачей из-за нехватки данных и различий в лексическом покрытии доступных языковых ресурсов. Источниками данных о семантических связях слов являются как словари, составленные людьми, так и словари, построенные путем извлечения пар слов из корпуса текстов [53]. К сожалению, доступность и полнота таких наборов данных ограничена, что делает актуальной задачу расширения лексического покрытия таких ресурсов [6]. Это делает невозможным применение процедуры отнологизации [87] и непосредственное использование метода ECO [49] для построения связей между синсетами, поскольку требуется наличие доступного тезауруса высокого качества для выравнивания связей на его основе. Подобные высококачественные семантические ресурсы доступны не для всех естественных языков и их создание требует большого количества работы экспертов-лексикографов [7].

С целью решения проблемы доступности и полноты данных, предлагается не производить построение семантических связей между целыми синсетами, но формировать такие связи между отдельными лексическими значениями слов. Это согласуется с человеческим восприятием окружающего мира, поскольку такие лексические значения соответствуют различным «концептам слов» (англ. *word concepts*) [89]. Таким образом, представление знаний будет осуществляться в виде специальной модели — *семантической сети слов*. В такой семантической сети слов семантические связи формируются между отдельными лексическими значениями слов. Это позволяет как повторно использовать, так и расширить доступные лексико-семантические ресурсы для автоматического построения семантической сети слов без непосредственного участия человека.

В данной главе описывается модель представления знаний в виде семантической сети слов, а также предлагаются методы и алгоритмы ее построения.

Предложен метод построения семантической сети слов, включающий метод построения синсетов на основе графа синонимов и метод построения связей между значениями слов. Предложенные модели, методы и алгоритмы предназначены для устранения проблем лексической многозначности и неполноты данных в языковых ресурсах.

2.1. Семантическая сеть слов

Поскольку вершинами в семантической сети слов являются не множества слов, обозначающих одно и то же понятие, а отдельные лексические значения слов («концепты слов»), то в целях определения семантической сети слов вносится уточнение в исходное определение семантической сети (Определение 3).

Пусть словник V — множество всех слов. Пусть $\mathcal{V}$ — множество всех лексических значений слов. Каждому слову $u \in V$ ставится в соответствие множество лексических значений $\text{senses}(u) \subseteq \mathcal{V}$. Пусть $\mathcal{R} \subset \mathcal{V} \times \mathcal{V}$ — асимметричное отношение между лексическими значениями слов. Пусть $(w, h) \in \mathcal{R}$ является такой упорядоченной парой, что $w \in \mathcal{V}$ является нижестоящим значением слова по отношению к вышестоящему значению слова $h \in \mathcal{V}$.

***Определение* 5.** *Семантическая сеть слов $\mathcal{N} = (\mathcal{V}, \mathcal{R})$ — это семантическая сеть, понятия которой — лексические значения слов $\mathcal{V}$, а множество дуг $\mathcal{R} \subset \mathcal{V} \times \mathcal{V}$ порождается асимметричным отношением на множестве $\mathcal{V}$.*

Таким образом, под *понятием* будет подразумеваться элемент множества $\mathcal{V}$, то есть «концепт слова» [89]. При обозначении слова и идентификатора его конкретного значения используется нотация, принятая в BabelNet [76], но без явного указания части речи в нижнем индексе. Запись *слово*^{i} указывает значение указанного слова под номером $1 \leqslant i \leqslant |\,\text{senses}(\textit{слово})|$. Например, слово «лук» имеет не меньше двух значений: *лук*^{1} как растение и *лук*^{2} как оружие. В целях обеспечения компактности записи, операция однозначного преобразования слова $u \in V$ в его векторное представление будет записываться в виде оператора $\vec{u}$ [74].

Для построения семантической сети слов предлагается модифицированный метод ECO [49], общая схема которого представлена на рис. 2.1. Входными данными для метода построения семантической сети слов являются:

– *слабоструктурированные словари*, содержащие пары однозначных или многозначных слов, принадлежащие некоторому бинарному отношению на словнике, необходимые для построения графа синонимов и его кластеризации, а также для построения иерархических контекстов и связывания значений слов;
– *неразмеченный корпус текстов* для построения векторных представлений слов, необходимых для взвешивания графа синонимов на этапе кластеризации и расширения иерархических контекстов на этапе связывания.

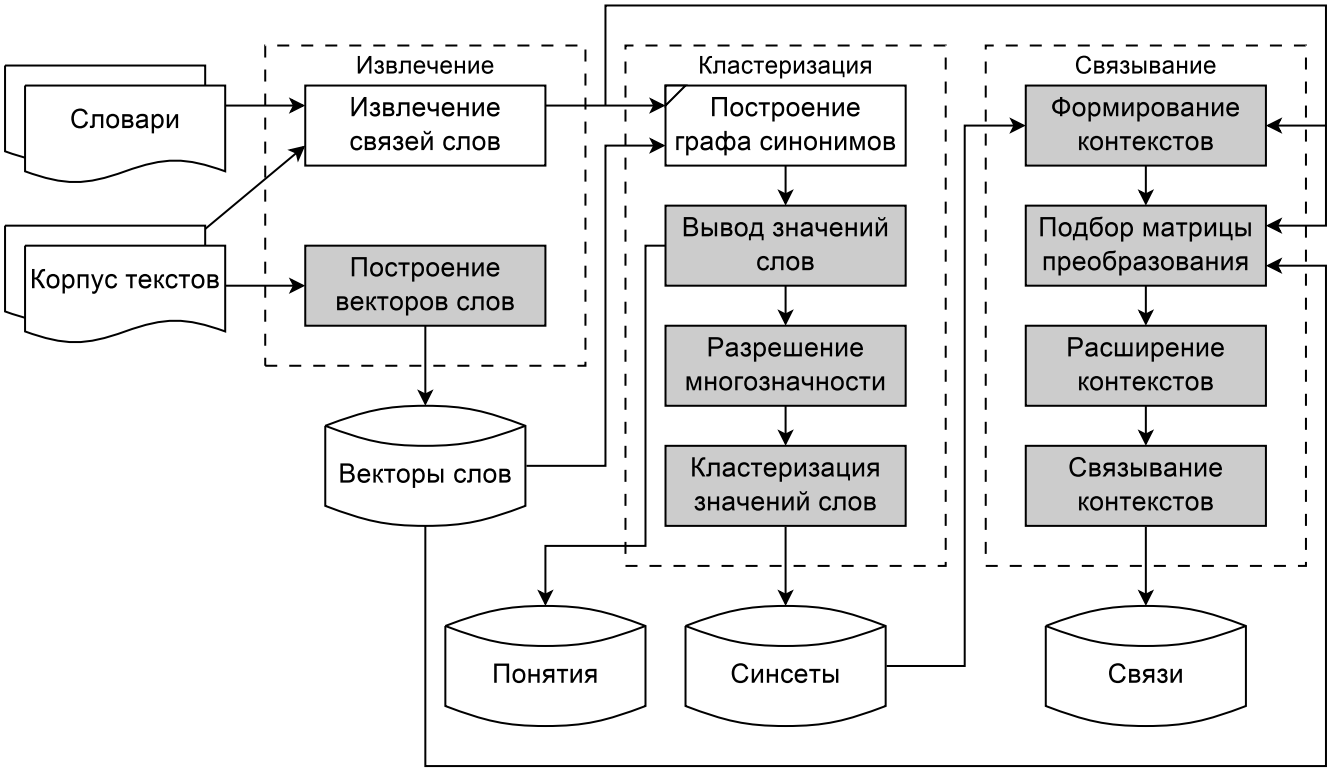


Рис. 2.1 — Общая схема предлагаемого метода построения семантической сети слов: модифицированные блоки метода ECO помечены уголком сверху слева; новые блоки выделены цветом

Метод построения семантической сети слов состоит из следующих этапов:

– На этапе *извлечения* производится извлечение связей между словами как из семантических словарей, так и из корпуса текстов стандартным образом [53]. Формируются векторные представления слов в пространстве низкой размерности при помощи общепринятых подходов [74].
– На этапе *кластеризации* осуществляется построение понятий семантической сети слов путем построения на основе графа синонимов вспомогательного графа значений слов путем вывода значений слов, разрешения многозначности в контекстах; синсеты формируются путем кластеризации графа значений слов.
– На этапе *связывания* выполняется связывание понятий семантической сети слов путем построения, расширения и связывания иерархических контекстов.

Результатом выполнения предлагаемого метода является семантическая сеть слов, семантические связи в которой сформированы между значениями слов. Помимо этого, основые отличия предложенного метода построения семантической сети слов от метода ECO заключаются в следующем:

– Исходными данными для построения семантической сети слов являются слабоструктурированные словари и неразмеченный корпус текстов. Слабоструктурированные словари представляют собой пары однозначных или многозначных слов. Пары слов в таких словарях могут принадлежать как симметричному отношению (синонимия), так и асимметричному («род–вид», «часть–целое», и др.). Корпус текстов предназначен для построения векторов слов при помощи таких методов, как Skip-gram [74].
– На этапе *извлечения* предлагается использовать векторные представления слов в пространстве низкой размерности [74], построенные на основе корпуса текстов, для взвешивания графа синонимов на этапе кластеризации и для расширения иерархических контекстов на этапе связывания.
– На этапе *кластеризации* предлагается использовать новый метод построения синсетов на основе графа синонимов. Данный метод основан на построении вспомогательного графа значений слов путем вывода

значений слов [26, 36, 83] с применением процедуры разрешения многозначности в контекстах [38]. Вершинами графа значений слов являются значения слов, а множество его ребер порождается отношением синонимии на множестве лексических значений слов. Формирование синсетов осуществляется путем кластеризации этого вспомогательного графа при помощи какого-либо метода жесткой кластеризации графа.

– Этап отнологизации предлагается заменить на специализированный этап *связывания* значений слов, использующий новый метод связывания значений слов. На основе существующих слабоструктурированных словарей строятся иерархические контексты, представляющие вышестоящие слова для слов в синсетах. Поскольку доступность таких словарей ограничена, производится расширение иерархических контекстов при помощи модифицированного метода подбора матрицы линейного преобразования [45]. Построение семантической сети слов производится путем подбора значений слов в иерархических контекстах.

2.2. Метод построения синсетов

На этапе кластеризации (рис. 2.1) производится построение синсетов на основе словаря синонимов, входящих в исходные данные метода построения семантической сети слов. Основная трудность построения синсетов заключается в учете многозначности слов: один синсет представляет единое значение для множества образующих его лексических единиц [40].

Пусть словарь синонимов $D \subseteq V \times V$ — это отношение синонимии между словами. В соответствии с Определением 4 на основе элементов данного отношения возможно построить граф синонимов W, связанные скопления вершин в котором обозначают одно и то же понятие [47, 58]. Задача обнаружения таких скоплений вершин решается путем поиска клик [28] или поиска сообществ [101]

в графе синонимов, однако при использовании упомянутых подходов не учитывается явным образом многозначность слов. Поскольку словник V содержит как однозначные, так и многозначные слова, то в данном разделе предлагается новый метод построения синсетов на основе графа синонимов, явным образом учитывающий явление полисемии.

В литературе, как правило, под синсетом подразумевается множество слов, являющихся синонимами [9,40,76]. С целью упрощения нотации, в данной работе под элементами синсета будут пониматься не элементы словника V, а элементы множества лексических значений слов $\mathcal{V}$. Для этого вносится уточнение в определение синсета.

***Определение* 6.** *Синсет $S \in \mathcal{S}$ — это множество $S \subseteq \mathcal{V}$, такое, что все пары элементов S принадлежат отношению синонимии.*

В случае, если словник V содержит только однозначные слова, то искомое множество синсетов $\mathcal{S}$ можно было бы получить путем выполнения какого-либо алгоритма жесткой кластеризации графа синонимов W. Например, алгоритма испорченного телефона [26] или марковского алгоритма кластеризации [35]. В данной диссертационной работе такое допущение не используется, поэтому каждому слову $u \in V$ ставится в соответствие множество значений $\text{senses}(u) \in \mathcal{V}$, $|\,\text{senses}(u)| \geqslant 1$, то предлагается построить вспомогательный граф значений слов $\mathcal{W} = (\mathcal{V}, \mathcal{E})$. Поскольку вершинами такого графа являются не слова, а значения слов, то кластеризация графа W при помощи какого-либо метода жесткой кластеризация графа приведет к получению множества синсетов $\mathcal{S}$.

***Определение* 7.** *Граф значений слов $\mathcal{W} = (\mathcal{V}, \mathcal{E})$ — это неориентированный взвешенный граф, множество вершин которого состоит из лексических значений слов, а множество ребер порождается отношением синонимии на множестве лексических значений слов.*

Таким образом, используется следующая *постановка задачи* построения синсетов: найти все синсеты $\mathcal{S}$ в графе $\mathcal{W}$ такие, что в каждом синсете $S \in \mathcal{S}$

любая пара значений слов $a \in S, b \in S$ находится в отношении синонимии. Общая схема предлагаемого метода построения синсетов представлена на рис. 2.2 и включает в себя четыре шага:

– построение графа синонимов;
– вывод лексических значений каждого слова;
– построение вспомогательного графа значений слов;
– кластеризация графа значений слов.

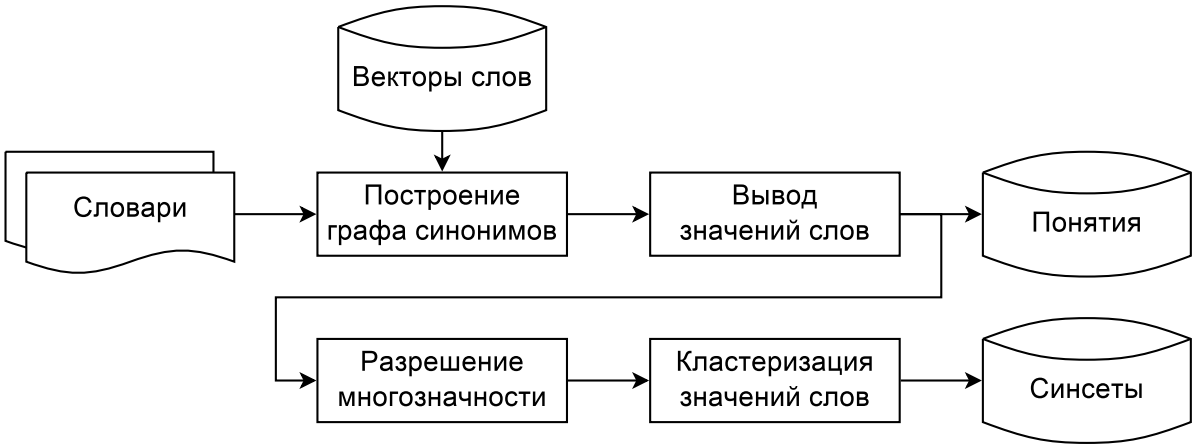


Рис. 2.2 — Общая схема метода построения синсетов

2.2.1. Построение графа синонимов

Построение графа синонимов $W = (V, E)$ осуществляется на основе словаря синонимов $D \subseteq V \times V$ таким образом, что множество ребер E данного графа является множеством, элементами которого являются двухэлементные множества, состоящие из неодинаковых слов в словаре синонимов:

$$E = \{\{u, v\} \in V \times V : (u, v) \in D, u \neq v\}. \tag{2.1}$$

Пусть задана некоторая мера семантической близости слов $\text{sim}_{\text{word}} : (u, v) \to \mathbb{R}, u \in V, v \in V$. Тогда взвешивание ребер E осуществляется путем вычисления меры sim_{word} между словами, соответствующими вершинам, инцидентным каждому ребру $\{u, v\} \in E$. Например, для этого

возможно использовать косинус угла между векторными представлениями слов [74], или любую другую меру.

В результате выполнения данных операций получен взвешенный граф синонимов $W = (V, E)$, на основе которого будет построен вспомогательный граф значений слов.

2.2.2. Вывод лексических значений слов

Вершинами графа синонимов W являются как однозначные, так и многозначные слова. С целью определения значений слов и построения множества значений слов $\mathcal{V}$ предлагается воспользоваться подходом к выводу значений слов на основе кластеризации окрестностей вершин, описанным в работах [26, 36, 83].

Окрестность вершины u в графе синонимов W — это неориентированный граф $W_u = (V_u, E_u)$, множество вершин V_u которого не включает u:

$$V_u = \{v \in V : \{u, v\} \in E\}, \tag{2.2}$$

$$E_u = \{\{v, w\} \in E : v \in V_u, w \in V_u\}. \tag{2.3}$$

С целью определения лексических значений выполняется следующая процедура для каждого слова $u \in V$. Сначала из графа W извлекается окрестность W_u. Затем производится кластеризация графа W_u при помощи какого-либо метода жесткой кластеризации графа. Результатом кластеризации является множество кластеров $C : V_u = \bigcup_{1 \leqslant i \leqslant |C|} C_i$. Кластеры принимаются в качестве значений слова u: $\text{senses}(u) = \{u^i : 1 \leqslant i \leqslant |C|\}$. Каждому значению слова $u^i \in \text{senses}(u)$ ставится в соответствие контекст $\text{ctx}(u^i) = C_i$.

***Определение* 8.** *Контекст* $\text{ctx}(u^i) \subset V$ *— множество синонимов слова* $u \in V$ *в значении под номером* $1 \leqslant i \leqslant |\,\text{senses}(u)|$.

Процедура вывода значений слов позволяет определить лексические значения слов и сформировать контексты, представляющие синонимы слов в соответствующих лексических значениях. Кроме того, в результате выполнения

этой процедуры можно построить множество понятий $\mathcal{V}$ семантической сети слов, являющееся объединением множеств лексических значений слов:

$$\mathcal{V} = \bigcup_{u \in V} \text{senses}(u). \tag{2.4}$$

На рис. 2.3 представлен пример окрестности слова «программа». При кластеризации целевое слово исключается. Это приводит к появлению трех кластеров, соответствующих разным значениям этого слова: $\{$*план*, *проект*, *график*$\}$, $\{$*программное обеспечение*, *приложение*$\}$ и $\{$*манифест*$\}$.

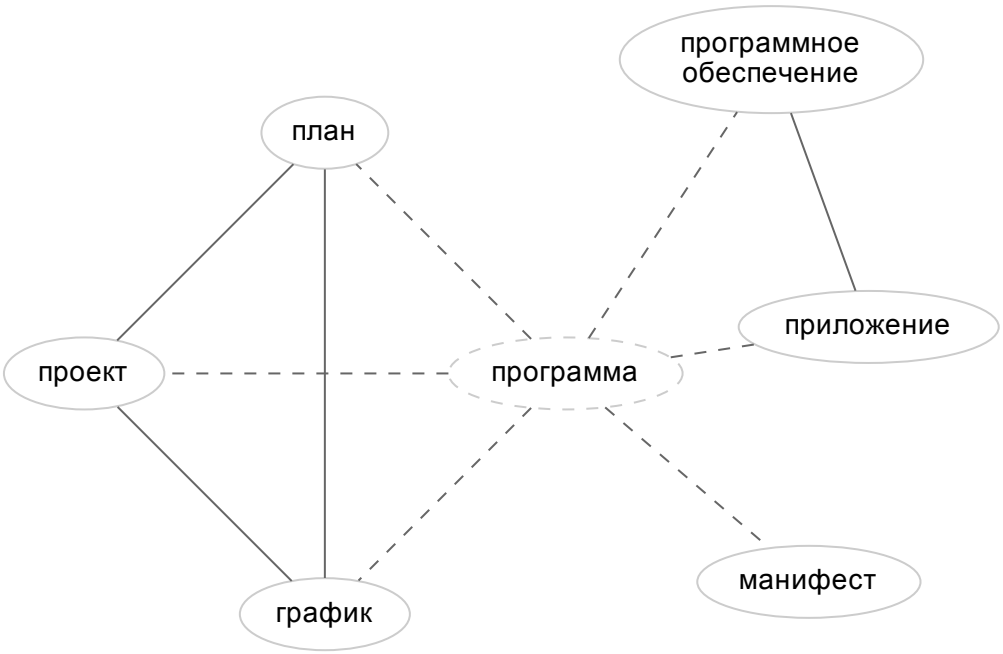


Рис. 2.3 — Пример кластеризации окрестности слова «программа»: многозначное слово «программа» исключено из окрестности

В табл. 1 приведены результаты вывода значений слова «программа» из примера на рис. 2.3: в колонке «Значение» представлены обнаруженные значения слова, в колонке «Контекст» перечислены контексты для каждого из значений.

Таблица 1 — Пример контекстов слова «программа»

Значение	**Контекст**
программа^1	$\{$*план*, *проект*, *график*$\}$
программа^2	$\{$*программное обеспечение*, *приложение*$\}$
программа^3	$\{$*манифест*$\}$

2.2.3. Построение графа значений слов

Для построения вспомогательного графа значений слов $\mathcal{W} = (\mathcal{V}, \mathcal{E})$ необходимо сформировать множество его ребер $\mathcal{E}$, порожденное отношением синонимии на множестве лексических значений слов $\mathcal{V}$. Этого возможно достичь путем определения значений слов в ребрах графа синонимов $W = (V, E)$ с использованием контекстов значений слов.

Пусть задана некоторая мера близости контекстов $\text{sim}_{\text{ctx}} : (\text{ctx}(a), \text{ctx}(b)) \to \mathbb{R}, \forall a \in \mathcal{V}, b \in \mathcal{V}$. Поскольку элементами контекстов являются слова без указания значений, производится разрешение многозначности контекста каждого значения слова $s \in \mathcal{V}$. Каждому элементу $u \in \text{ctx}(s)$ ставится в соответствие значение $\hat{u} \in \mathcal{V}$ с наиболее близким контекстом:

$$\hat{u} \in \arg\max_{u' \in \text{senses}(u)} \text{sim}(\text{ctx}(s), \text{ctx}(u')), \tag{2.5}$$

где $\text{senses}(u)$ — множество всех значений слова $u \in V$. Затем, каждому элементу $s \in \mathcal{V}$ ставится в соответствие контекст с разрешенной многозначностью $\widehat{\text{ctx}}(s) \subset \mathcal{V}$:

$$\widehat{\text{ctx}}(s) = \{\hat{u} : u \in \text{ctx}(s)\}. \tag{2.6}$$

На основе полученных контекстов со снятой многозначностью формируется множество ребер $\mathcal{E}$ графа значений слов $\mathcal{W} = (\mathcal{V}, \mathcal{E})$:

$$\mathcal{E} = \{\{\hat{u}, \hat{v}\} \in \mathcal{V} \times \mathcal{V} : \hat{v} \in \widehat{\text{ctx}}(\hat{u})\}. \tag{2.7}$$

При построении множества ребер $\mathcal{E}$ графа значений слов каждому ребру назначается вес, равный весу ребра графа синонимов, инцидентного вершинам, соответствующим словам, значения которых определялись в данной процедуре.

На рис. 2.4 представлен пример графа значений слов, полученного путем вывода значений слов (рис. 2.3) и разрешения неоднозначности в контекстах (табл. 1). В данном примере слово «программа» участвует в образовании трех синсетов, связанных с различными значениями этого слова: $\{$*программа*^{1}, *проект*^{1}, *план*^{1}, *график*$^2\}$,

$\{$*программа*^{2}, *программное обеспечение*^{1}, *приложение*$^2\}$ и $\{$*программа*^{3}, *манифест*$^1\}$.

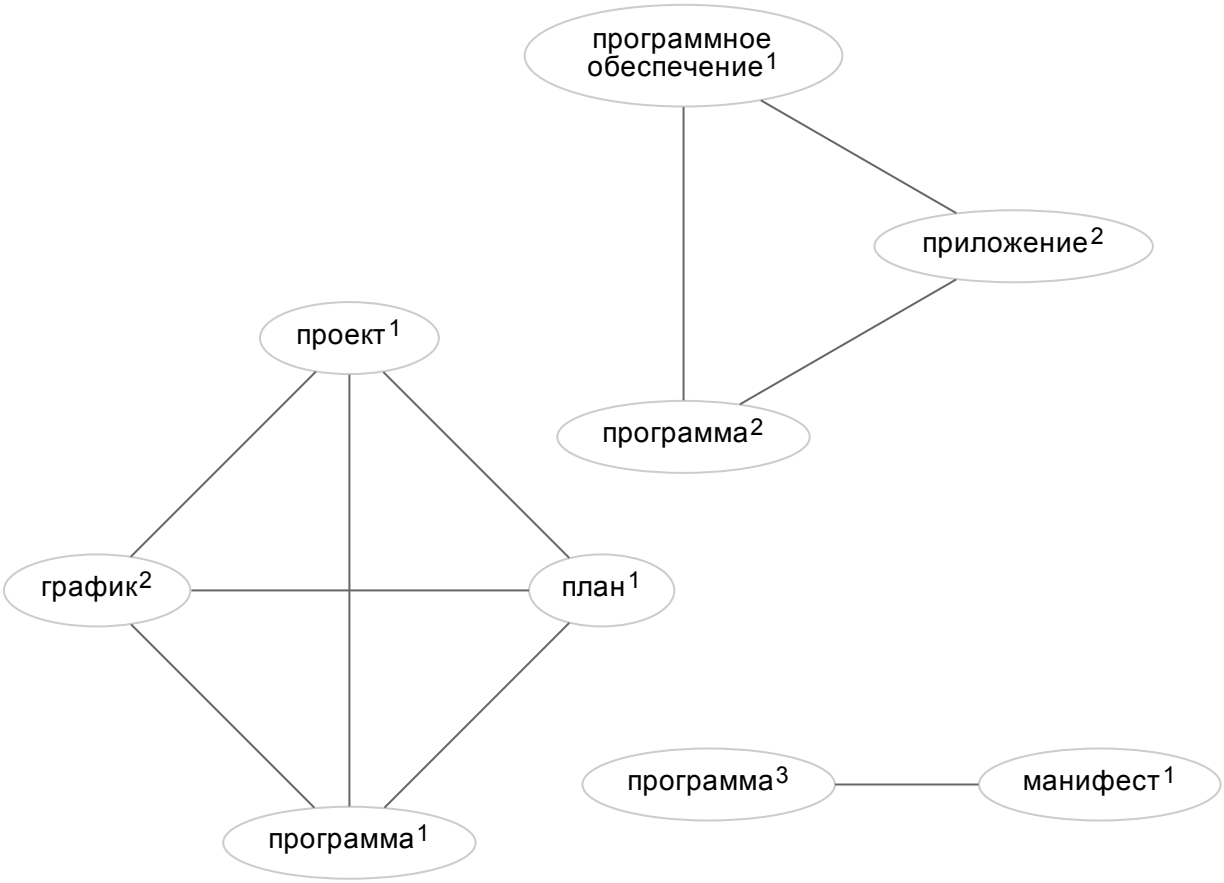


Рис. 2.4 — Пример графа значений слов

2.2.4. Кластеризация графа значений слов

Поскольку ребра графа значений слов $\mathcal{W} = (\mathcal{V}, \mathcal{E})$ порождаются отношением синонимии на множестве лексических значений слов, то справедливо допустить, что связанные скопления вершин в таком графе также обозначают одно и то же понятие [47, 58]. В качестве заключительного шага производится кластеризация графа значений слов при помощи какого-либо алгоритма жесткой кластеризации графа. Полученное в результате кластеризации данного вспомогательного графа множество кластеров является искомым множеством синсетов $\mathcal{S}$.

2.2.5. Алгоритм построения синсетов Watset

На основе метода построения синсетов предложен алгоритм Watset. Название алгоритма образовано от английских слов *what* — «что» и *set* — «множество», и обозначает особенность его работы. Входными данными для алгоритма является словарь синонимов $D \subseteq V \times V$. Результатом работы алгоритма является множество лексических значений слов $\mathcal{V}$ и множество синсетов $\mathcal{S}$, т. е. множество кластеров отдельных лексических значений слов. Алгоритм имеет четыре гиперпараметра:

- $\text{Cluster}_{\text{Local}}$ — алгоритм жесткой кластеризации графа, используемый для кластеризации окрестностей вершин в графе синонимов при выводе лексических значений слов;
- $\text{Cluster}_{\text{Global}}$ — алгоритм жесткой кластеризации графа, используемый для поиска синсетов в графе значений слов;
- $\text{sim}_{\text{word}} : (u, v) \to \mathbb{R}$ — мера близости слов $u \in V$ и $v \in V$;
- $\text{sim}_{\text{ctx}} : (\text{ctx}(a), \text{ctx}(b)) \to \mathbb{R}$ — мера близости контекстов значений слов $a \in \mathcal{V}$ и $b \in \mathcal{V}$.

Алгоритм Watset состоит из головной процедуры и трех вспомогательных процедур построения графа синонимов, вывода значений заданного слова и разрешения многозначности контекстов слова. Применение алгоритмов жесткой кластеризации графа обусловлено тем, что метод построения синсетов спроектирован с допущением о том, что в результате кластеризации одна вершина может входить в состав не более, чем одного кластера. Предложенный алгоритм не накладывает ограничений на используемые алгоритмы жесткой кластеризации графа $\text{Cluster}_{\text{Local}}$ и $\text{Cluster}_{\text{Global}}$. В области обработки естественного языка большой популярностью пользуется марковский алгоритм кластеризации [35] и алгоритм испорченного телефона [26], но возможно использование и любого другого алгоритма.

Головная процедура. В общем виде, головная процедура выглядит следующим образом:

Шаг 1. **Построить граф синонимов** W;

Шаг 2. Для всех слов $u \in V$ выполнить цикл

Шаг 2.1. **Произвести вывод значений слова** u;

Шаг 3. Конец цикла;

Шаг 4. Построить множество значений всех слов: $\mathcal{V} \leftarrow \bigcup_{u \in V} \text{senses}(u)$;

Шаг 5. Для всех значений слов $s \in \mathcal{V}$ выполнить цикл.

Шаг 5.1. **Разрешить многозначность контекста** s;

Шаг 6. Конец цикла;

Шаг 7. Построить множество ребер: $\mathcal{E} \leftarrow \{\{\hat{u}, \hat{v}\} \in \mathcal{V} \times \mathcal{V} : \hat{v} \in \widehat{\text{ctx}}(\hat{u})\}$;

Шаг 8. Выполнить кластеризацию графа $\mathcal{W} = (\mathcal{V}, \mathcal{E})$:
$\mathcal{S} \leftarrow \text{Cluster}_{\text{Global}}(\mathcal{W})$;

Шаг 9. Стоп.

Процедура построения графа синонимов. Данная процедура предназначена для построения графа синонимов. Входными данными для процедуры является множество словарей синонимов D. Результатом выполнения процедуры является граф синонимов $W = (V, E)$, взвешенный при помощи меры семантической близости слов sim_{word}. Процедура выглядит следующим образом:

Шаг 1.1. $V \leftarrow \bigcup_{(u,v) \in D} \{u, v\}$;

Шаг 1.2. $E \leftarrow \{\{u, v\} \in V \times V : (u, v) \in D, u \neq v\}$;

Шаг 1.3. Для всех ребер $\{u, v\} \in E$ выполнить цикл

Шаг 1.3.1. $\text{weight}(u, v) \leftarrow \text{sim}_{\text{word}}(u, v)$;

Шаг 1.4. Конец цикла;

Шаг 1.5. Конец процедуры.

Процедура вывода значений слова. Данная процедура предназначена для определения значений слова $u \in V$. Входными данными для процедуры является граф синонимов $W = (V, E)$ и заданное слово u. Результатом выполнения

процедуры является множество $\text{senses}(u)$, содержащее все обнаруженные значения слова u, причем для каждого обнаруженного значения составлен контекст, представляющий синонимы слова в данном значении (см. пример в табл. 1). Процедура выглядит следующим образом:

Шаг 2.1.1. $\text{senses}(u) \leftarrow \varnothing$;

Шаг 2.1.2. Извлечь вершины окрестности вершины u:
$V_u \leftarrow \{v \in V : \{u, v\} \in E\}$;

Шаг 2.1.3. Извлечь ребра окрестности вершины u:
$E_u \leftarrow \{\{v, w\} \in E : v \in V_u, w \in V_u\}$;

Шаг 2.1.4. Выполнить кластеризацию графа $W_u = (V_u, E_u)$:
$C \leftarrow \text{Cluster}_{\text{Local}}(W_u)$;

Шаг 2.1.5. $i \leftarrow 1$;

Шаг 2.1.6. $\text{ctx}(u^i) \leftarrow C_i$;

Шаг 2.1.7. $\text{senses}(u) \leftarrow \text{senses}(u) \cup \{u^i\}$;

Шаг 2.1.8. Если $i < |C|$, то $i \leftarrow i + 1$ и перейти на шаг 2.1.6;

Шаг 2.1.9. Конец процедуры.

Процедура разрешения многозначности контекста. Данная процедура предназначена для построения контекста с разрешенной многозначностью для элемента $s \in \mathcal{V}$. Входными данными для процедуры является заданное значение слова s и контексты значений всех слов, входящих в контекст $\text{ctx}(s)$. Результатом выполнения процедуры является контекст с разрешенной многозначностью $\widehat{\text{ctx}}(s)$. Процедура выглядит следующим образом:

Шаг 6.1.1. $\widehat{\text{ctx}}(s) \leftarrow \varnothing$;

Шаг 6.1.2. Для каждого слова в контексте $u \in \text{ctx}(s)$ выполнить цикл

Шаг 6.1.2.1. $\hat{u} \leftarrow \arg\max_{u' \in \text{senses}(u)} \text{sim}_{\text{ctx}}(\text{ctx}(s), \text{ctx}(u'))$;

Шаг 6.1.2.2. $\widehat{\text{ctx}}(s) \leftarrow \widehat{\text{ctx}}(s) \cup \{\hat{u}\}$;

Шаг 6.1.3. Конец цикла;

Шаг 6.1.4. Конец процедуры.

Теорема. *Пусть* $\deg_{\max}$ *— максимальная степень вершины графа* $W = (V, E)$. *Тогда вычислительная сложность процедуры разрешения многозначности*

контекста всех значений слов составляет $O(|V|\deg^4_{\max})$ *при использовании косинусной меры близости контекстов значений слов.*

Доказательство. Пусть $m_{\text{senses}} = \max_{u \in V} |\operatorname{senses}(u)|$ — наибольшее количество значений слова, $m_{\text{ctx}} = \max_{s \in \mathcal{V}} |\operatorname{ctx}(s)|$ — размер наибольшего контекста. Поскольку процедура разрешения многозначности контекста выполняется для каждого слова в каждом контексте каждого элемента $s \in \mathcal{V}$, то перечисление всех слов, входящих в контексты каждого значения слова в словнике требует $O(|V| \times m_{\text{senses}} \times m_{\text{ctx}})$ шагов. В каждом контексте слову $u \in \operatorname{ctx}(s)$ ставится в соответствие значение $\hat{u} \in \mathcal{V}$, для чего требуется перечисление всех возможных значений такого слова с вычислением косинусной меры близости между контекстом $\operatorname{ctx}(s)$ и контекстом каждого значения $u' \in \operatorname{senses}(u)$. Эта операция требует $O(m_{\text{senses}} \times m_{\text{ctx}})$ шагов. Таким образом, процедура разрешения многозначности контекста выполняется за $O(|V| \times m^2_{\text{senses}} \times m^2_{\text{ctx}})$ шагов. Поскольку наибольшее количество значений слова достигается в случае, когда количество компонент связности окрестности $W_u = (V_u, E_u)$ вершины $u \in V$ в графе W равно количеству вершин в такой окрестности, причем количество таких вершин по определению (2.3) не превышает $\deg_{\max}$, то справедливо считать $m_{\text{senses}} \leqslant \deg_{\max}$. Поскольку контекст с наибольшим размером образуется в случае, когда граф W_u является полносвязным, причем такой контекст по Определению 8 включает не более $\deg_{\max}$ слов, то справедливо считать $m_{\text{ctx}} \leqslant \deg_{\max}$. Следовательно, вычислительная сложность процедуры разрешения многозначности контекста при использовании косинусной меры близости контекстов составляет $O(|V|\deg^4_{\max})$. □

2.3. Метод построения связей

На этапе связывания (рис. 2.1) производится построение однозначных асимметричных связей между словами, входящих в исходные данные метода

построения семантической сети слов. Основная трудность построения связей заключается в учете многозначности слов: в слабоструктурированных словарях не представлена информация о значениях слов, входящих в пары, принадлежащие асимметричному отношению [53].

Пусть $R \subset V \times V$ — асимметричное отношение, порожденное на словнике. Пусть $(w, h) \in R$ является такой упорядоченной парой, что $w \in V$ является нижестоящим словом по отношению к вышестоящему слову $h \in V$. Данные для построения отношения R представлены в материалах слабоструктурированных словарей и не указывают значения связанных слов. Поскольку словник V содержит как однозначные, так и многозначные слова, невозможно построить множества дуг $\mathcal{R}$ семантической сети слов на основе только элементов отношения R. В данном разделе предлагается новый метод построения и расширения асимметричных семантических связей на основе иерархических контекстов, представляющих наиболее типичные вышестоящие слова синсетов. В свою очередь, выстраивание семантической иерархии между синсетами является трудной задачей, решаемой путем выравнивания относительно другой высококачественной семантической сети [49, 87]. Поэтому целесообразно осуществлять построение связей между отдельными лексическими значениями слов, что позволит построить множество дуг $\mathcal{R}$ семантической сети слов $\mathcal{N} = (\mathcal{V}, \mathcal{R})$.

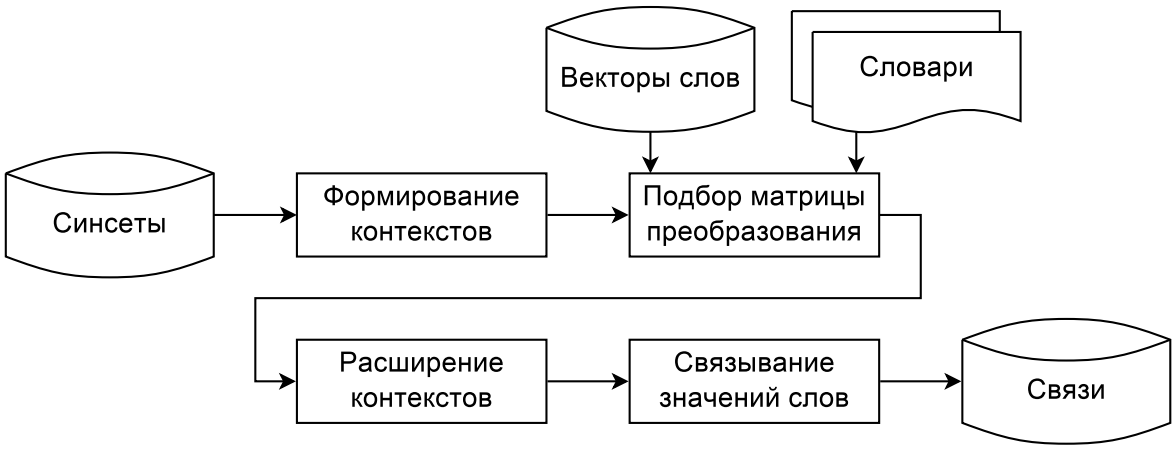


Рис. 2.5 — Схема метода построения связей

Таким образом, используется следующая *постановка задачи* построения связей: для каждого синсета $S \in \mathcal{S}$ найти множество вышестоящих значений слов

$\widehat{\text{hctx}}(S) \subset \mathcal{V}$, такое, что каждый элемент $\hat{h} \in \widehat{\text{hctx}}(S)$ является вышестоящим значением по отношению к каждому элементу $s \in S$. Общая схема предлагаемого метода построения связей представлена на рис. 2.5 и включает четыре шага:

– формирование иерархических контекстов синсетов;
– подбор семейства матриц линейного преобразования;
– расширение иерархических контекстов;
– связывание значений слов при помощи иерархических контекстов.

2.3.1. Построение иерархических контекстов

Метод построения связей основан на допущении о том, что множества вышестоящих слов для каждого элемента некоторого синсета совпадают, по крайней мере, частично. Такое допущение справедливо, поскольку в распространенных семантических сетях элементами иерархии являются множества синонимов [9, 40, 76]. Для этого вводится понятие иерархического контекста синсета.

***Определение* 9.** *Иерархический контекст* $\text{hctx}(S) \subset V$ *синсета* $S \in \mathcal{S}$ *— это объединение множеств вышестоящих слов для каждого слова синсета* S.

Пусть $\text{words}(S) \subseteq V$ — множество слов, значения которых включены в синсет S. Тогда каждому синсету $S \in \mathcal{S}$ на основе вышеупомянутого допущения $\left|\bigcap_{w \in \text{words}(S)} \{h \in V : (w, h) \in R\}\right| > 0, \forall S \in \mathcal{S}$ ставится в соответствие иерархический контекст:

$$\text{hctx}(S) = \{h \in V : (w, h) \in R, w \in \text{words}(S), h \notin \text{words}(S)\}. \qquad (2.8)$$

Поскольку значимость слов в иерархических контекстах различается, предлагается для взвешивания элементов контекстов использовать меру tf–idf, широко применяющуюся в информационном поиске [11]. Таким образом, в иерархическом контексте $\text{hctx}(S)$ синсета $S \in \mathcal{S}$ вес каждого слова $h \in \text{hctx}(S)$ вычисляется по формуле

$$\text{tf–idf}(h, S, \mathcal{S}) = \text{tf}(h, S) \times \text{idf}(h, \mathcal{S}), \qquad (2.9)$$

где $\mathrm{tf}(h, S)$ — частота слова h в синсете S, $\mathrm{idf}(h, \mathcal{S})$ — обратная частота слова h во множестве синсетов $\mathcal{S}$. При этом значение частоты слова h в синсете S определяется как отношение количества появлений этого слова в иерархическом контексте среди суммы количества появлений других слов в нем:

$$\mathrm{tf}(h, S) = \frac{|h' \in \mathrm{hctx}(S) : h = h'|}{|\,\mathrm{hctx}(S)|}. \tag{2.10}$$

В свою очередь, значение обратной частоты документа выражается как отношение количества всех синсетов $|\mathcal{S}|$ к количеству синсетов, иерархические контексты которых включают h:

$$\mathrm{idf}(h, \mathcal{S}) = \log \frac{|\mathcal{S}|}{|S' \in \mathcal{S} : h \in \mathrm{hctx}(S')|}. \tag{2.11}$$

В табл. 2 приведены примеры иерархических контекстов для двух синсетов со словом «программа». Видно, что синсеты содержат информацию о конкретных значениях слов. Иерархические контексты, в свою очередь, не содержат такой информации и содержат многозначные слова: слово «знак» может употребляться в значении «дорожный знак», а может употребляться в значении «денежный знак», и т. д.

Таблица 2 — Пример иерархических контекстов синсетов со словом «программа»

Синсет	**Иерархический контекст**
{*программа*^{1}, *план*^{1}, …}	{*документ*, *перечень*, …}
{*программа*^{2}, *приложение*^{2}, …}	{*запись*, *информация*, …}
{*программа*^{3}, *манифест*^{1}, …}	{*документ*, *заявление*, …}

2.3.2. Расширение иерархических контекстов

Расширение иерархических контекстов предназначено для добавления в иерархические контексты слов, подходящих по смыслу контекста, в целом, но

отсутствующих в асимметричном отношении R. Пусть $h \in \text{hctx}(S)$ — некоторое вышестоящее слово иерархического контекста синсета $S \in \mathcal{S}$. Пусть $\vec{h}$ — векторное представление данного слова в пространстве низкой размерности [74]. Пусть $\text{NN}_n(\vec{h}) \in V$ — операция поиска $n \in \mathbb{Z}^+$ слов, векторные представления которых соответствуют векторам-ближайшим соседям векторного представления $\vec{h}$ слова h. Поскольку в таких моделях, как Word2Vec, ближайшими соседями слов являются синонимы, когипонимы, партонимы, и морфологические варианты лексической единицы [74], то необходимо производить проверку осмысленности связи слова-кандидата со словами синсета. Поэтому для каждого синсета $S \in \mathcal{S}$ расширение иерархического контекста $\text{hctx}(S)$ осуществляется в два шага: формирование кандидатов и проверка кандидатов.

Сначала формируется множество кандидатов $M_S \subset V$ путем объединения множеств ближайших соседей каждого элемента иерархического контекста $\text{hctx}(S)$ без учета слов, уже являющихся элементами контекста:

$$M_S = \bigcup_{h \in \text{hctx}(S)} \text{NN}_n(\vec{h}) \setminus \text{hctx}(S), \tag{2.12}$$

Пусть Φ^* — такая матрица, что $\Phi^* \vec{w} = \vec{h}, \forall (w, h) \in R$. Проверка основана на допущении о том, что если слово $h \in M_S$ действительно является вышестоящим по отношению к любому слову $w \in \text{words}(S)$, то вектор, полученный путем умножения матрицы Φ^* на векторное представление нижестоящего слова $\vec{w}$, находится от вектора вышестоящего слова $\vec{h}$ на евклидовом расстоянии, не превышающем некоторый заданный порог $\delta \in \mathbb{R}$ [45]. Таким образом, кандидат в вышестоящее слово $h \in M_S$ добавляется в иерархический контекст $\text{hctx}(S)$ тогда и только тогда, когда выполняется условие

$$\exists w \in \text{words}(S) : \|\Phi^* \vec{w} - \vec{h}\| < \delta. \tag{2.13}$$

Рассмотрим пример на рис. 2.6. Слово «организация» является известным гиперонимом слова «банк». В качестве ближайших соседей слова «банк» выделены слова «супермаркет», «корпорация», «учреждение», и др. Суть проверки состоит в вычислении евклидова расстояния между вектором, полученным путем

умножения матрицы Φ^* на векторное представление слова «банк» между векторным представлением каждого слова-кандидата. В данном случае, векторы только двух слов находятся не дальше, чем δ от гиперонима: «корпорация» и «учреждение». Эти слова будут добавлены в иерархический контекст, при этом слово «супермаркет» добавлено не будет.

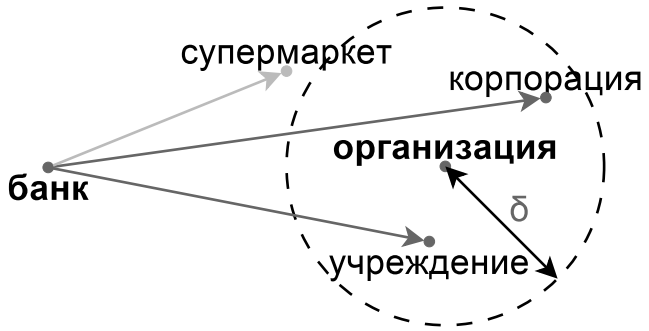


Рис. 2.6 — Выбор слов в δ-радиусе слова «организация», являющихся вышестоящими по отношению к слову «банк»: слово «супермаркет» не прошло проверку

2.3.3. Подбор матрицы линейного преобразования

С целью повышения аккуратности преобразования векторных представлений слов в векторные представления вышестоящих слов, предлагается модифицировать метод подбора матрицы линейного преобразования [45]. Известно, что векторы семантических близких слов расположены достаточно близко друг к другу, а векторы как вышестоящих, так и семантически не связанных слов достаточно далеко друг от друга [74]. Несмотря на то, что знание семантической близости между словами не позволяет надежно предсказать тип семантической связи между словами, описанное наблюдение позволяет внести в данную модель дополнительную информацию о взаимосвязях слов. Из свойства асимметричности отношения R следует, что если в каждой паре слов $(w, h) \in R$ слово h является вышестоящим по отношению к слову w, то слово w не может быть вышестоящим по отношению к слову h. В целях внесения такой информации, в модель (1.18)

вводится член стабилизации H, влияние которого определяется значением коэффициента λ:

$$\Phi_i^* \in \arg\min_{\Phi_i} \left(\frac{1}{|R|} \sum_{(\vec{w},\vec{h})\in R} \left\| \Phi_i \vec{w} - \vec{h} \right\|^2 + \lambda H \right). \tag{2.14}$$

Член стабилизации H увеличивает значение минимизируемой функции исходя из допущения, что повторное применение данного линейного преобразования к вектору $\Phi_i^* \vec{w}$ не должно порождать вектор $\Phi^2 \vec{w}$, близкий к исходному вектору $\vec{w}$:

$$H = \sum_{(\vec{w},\vec{h})\in R} ((\Phi^2 \vec{w})^T \vec{w})^2. \tag{2.15}$$

Предложенный стабилизатор H не зависит от внешних языковых ресурсов: для подбора матрицы преобразование требуется только обучающая выборка, представленная в виде отношения $R \subset V \times V$. Пример данного наблюдения представлен на рис. 2.7: расстояние между близкими по смыслу словами «кот» и «кошка» достаточно мало по сравнению с расстоянием от этих слов до гиперонимов «млекопитающее» и «животное».

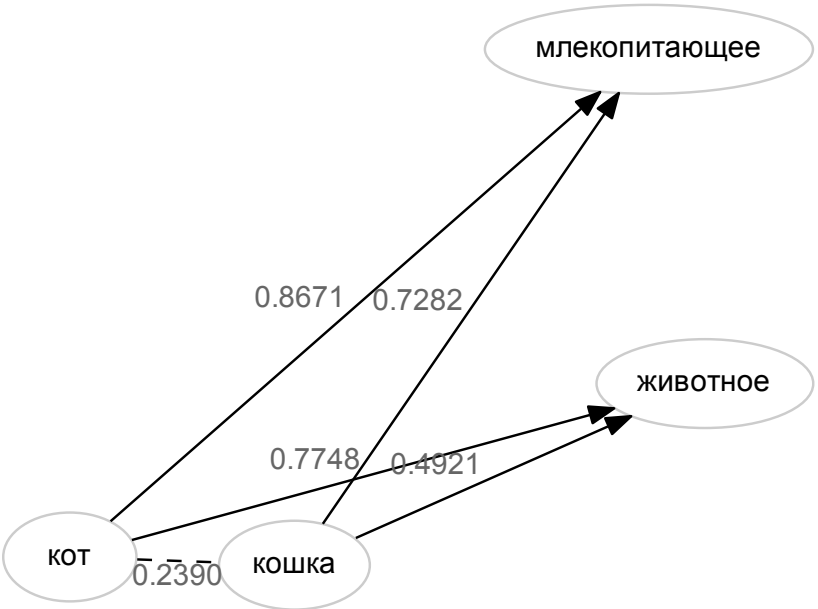


Рис. 2.7 — Семантическая близость близких и вышестоящих слов: в качестве расстояния использовано косинусное расстояние между соответствующими векторами

2.3.4. Связывание иерархических контекстов

Для построения семантической сети слов $\mathcal{N} = (\mathcal{V}, \mathcal{R})$ необходимо сформировать множество дуг $\mathcal{R}$, порожденное асимметричным отношением на множестве лексических значений слов $\mathcal{V}$. Это возможно сделать путем подбора значений слов в иерархических контекстах синсетов.

Пусть задана некоторая мера близости иерархического контекста и слов синсета $\text{sim}_{\text{hctx}} : (\text{hctx}(A), \text{words}(B)) \to \mathbb{R}, \forall A \in \mathcal{S}, B \in \mathcal{S}$. Поскольку элементами иерархических контекстов являются слова без указания значений, производится разрешение многозначности иерархического контекста каждого синсета $S \in \mathcal{S}$. Каждому элементу $h \in \text{hctx}(S)$ ставится в соответствие значение $\hat{h} \in \mathcal{V}$, включенное в наиболее близкий синсет:

$$\hat{h} \in \arg\max_{h' \in \text{senses}(h): S' \in \mathcal{S}, h' \in S', S \neq S'} \text{sim}_{\text{hctx}}(\text{hctx}(S), \text{words}(S')), \tag{2.16}$$

где $\text{words}(S)$ — множество слов, значения которых включены в синсет S. Затем, каждому синсету $S \in \mathcal{S}$ ставится в соответствие иерархический контекст с разрешенной многозначностью $\widehat{\text{hctx}}(S) \in \mathcal{V}$:

$$\widehat{\text{hctx}}(S) = \{\hat{h} : h \in \text{hctx}(S)\}. \tag{2.17}$$

На основе синсетов и контекстов со снятой многозначностью формируется множество дуг $\mathcal{R}$ семантической сети слов $\mathcal{N} = (\mathcal{V}, \mathcal{R})$, вершинами которой являются лексические значения слов, а множество дуг порождается асимметричным отношением на множестве лексических значений слов:

$$\mathcal{R} = \bigcup_{S \in \mathcal{S}} S \times \widehat{\text{hctx}}(S). \tag{2.18}$$

Пример семантической сети слов для многозначного слова «программа» приведен на рис. 2.8. Слова с различными значениями, но с совпадающими лексемами, не имеют общих связей.

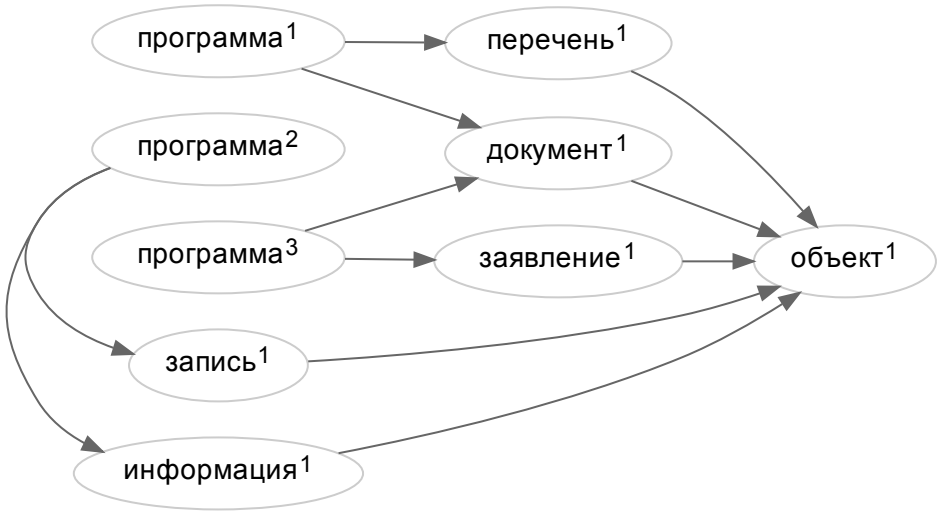


Рис. 2.8 — Пример фрагмента семантической сети слов: три различных значения слова «программа» не имеют общих связей, т. е. *программа*^{1} означает план деятельности, *программа*^{2} означает программное обеспечение, *программа*^{3} означает изложение целей и задач организации

2.3.5. Алгоритм построения связей Watlink

На основе метода построения связей предложен алгоритм Watlink. Название алгоритма образовано от английских слов *what* — «что» и *link* — «связь», и обозначает особенность его работы. Входными данными для алгоритма является множество синсетов $\mathcal{S}$, асимметричное отношение $R \subset V \times V$, и однозначное векторное представление $\vec{u}$ каждого слова $u \in V$ в пространстве $\mathbb{R}^d$, где $|V| \gg d$. Множество синсетов $\mathcal{S}$ может быть получено в результате работы алгоритма Watset (раздел 2.2.5). Результатом работы алгоритма является семантическая сеть слов $\mathcal{N}$. Алгоритм имеет пять гиперпараметров:

- $n \in \mathbb{Z}^+$ — количество ближайших соседей, возвращаемых при расширении контекстов;
- $k \in \mathbb{N}$ — количество подпространств при подборе матрицы линейного преобразования;
- $\lambda \in \mathbb{R}$ — влияние стабилизации на функцию потерь при подборе матрицы линейного преобразования;

– $\delta \in \mathbb{R}^+$ — максимальное расстояние до ближайшего соседа, включаемого при расширении иерархического контекста;
– $\text{sim}_{\text{hctx}} : (\text{hctx}(S), \text{words}(S')) \to \mathbb{R}$ — мера близости иерархического контекста $\text{hctx}(S) \subseteq V$ и слов синсета $S' \in \mathcal{S}$: $\text{words}(S') \subseteq V$.

Алгоритм Watlink состоит из головной процедуры и трех вспомогательных процедур подбора матриц линейного преобразования, построения иерархического контекста синсета, разрешения многозначности иерархического контекста.

Головная процедура. В общем виде, головная процедура выглядит следующим образом:

Шаг 1. **Подобрать матрицы линейного преобразования**;
Шаг 2. Для всех синсетов $S \in \mathcal{S}$ выполнить цикл
Шаг 2.1. **Построить иерархический контекст синсета** S;
Шаг 3. Конец цикла;
Шаг 4. Для всех синсетов $S \in \mathcal{S}$ выполнить цикл
Шаг 4.1. $\text{tf–idf}(h, S, \mathcal{S}) \leftarrow \text{tf}(h, S) \times \text{idf}(h, \mathcal{S})$;
Шаг 5. Конец цикла;
Шаг 6. Для всех синсетов $S \in \mathcal{S}$ выполнить цикл
Шаг 6.1. **Разрешить многозначность иерархического контекста** $\text{hctx}(S)$;
Шаг 7. Конец цикла;
Шаг 8. Построить связи между значениями слов: $\mathcal{R} \leftarrow \bigcup_{S \in \mathcal{S}} S \times \widehat{\text{hctx}}(S)$;
Шаг 9. Построить семантическую сеть слов $\mathcal{N} \leftarrow (\mathcal{V}, \mathcal{R})$;
Шаг 10. Стоп.

Процедура подбора матриц линейного преобразования. Данная процедура предназначена для подбора k матриц линейного преобразования на основе метода [45] с стабилизацией (2.14). Входными данными для процедуры является $k \in \mathbb{N}$ — количество кластеров, R — асимметричное отношение, $\lambda \in \mathbb{R}$ — важность члена стабилизации H. Результатом выполнения процедуры являются k матриц $\Phi_i^* : 1 \leqslant i \leqslant k$. Используется метод k-средних для разбиения исходного

линейного пространства на k подпространств [52] для учета его неоднородностей с использованием смещения $(\vec{h} - \vec{w}), \forall (w, h) \in R$ [45]. Процедура выглядит следующим образом:

Шаг 1.1. Для каждой пары слов $(w, h) \in R$ выполнить цикл

Шаг 1.1.1. offsets$(w, h) \leftarrow (\vec{h} - \vec{w})$;

Шаг 1.2. Конец цикла;

Шаг 1.3. $C \leftarrow$ k-means(offsets, k);

Шаг 1.4. $i \leftarrow 1$;

Шаг 1.5. $\Phi_i^* \leftarrow \arg\min_{\Phi_i} \frac{1}{|R_i|} \sum_{(\vec{w},\vec{h}) \in R_i} \left(\left\| \Phi_i \vec{w} - \vec{h} \right\|^2 + \lambda((\Phi_i^2 \vec{w})^T \vec{w})^2 \right)$;

Шаг 1.6. Если $i < k$, то $i \leftarrow i + 1$ и перейти на шаг 1.4;

Шаг 1.7. Конец процедуры.

Процедура построения иерархического контекста. Данная процедура осуществляет построение иерархического контекста синсета. Входными данными для процедуры является синсет $S \in \mathcal{S}$ и k матриц линейного преобразования, если запрошено расширение контекста. Результатом выполнения процедуры является иерархический контекст hctx(S). Процедура выглядит следующим образом:

Шаг 2.1.1. hctx$(S) \leftarrow \{h \in V : (w, h) \in R, w \in \text{words}(S), h \notin \text{words}(S)\}$;

Шаг 2.1.2. Сформировать множество слов-кандидатов в иерархический контекст: $M_S \leftarrow \bigcup_{h \in \text{hctx}(S)} \text{NN}_n(\vec{h}) \setminus \text{hctx}(S)$;

Шаг 2.1.2. Для каждой пары слов $(w, h) \in \text{words}(S) \times M_S$ выполнить цикл

Шаг 2.1.2.1. Выбрать одну из k матриц линейного преобразования Φ^* для пары слов (w, h) на основе смещения $(\vec{h} - \vec{w})$;

Шаг 2.1.2.2. Если $\|\vec{w}\Phi^* - \vec{h}\| < \delta$, то hctx$(S) \leftarrow$ hctx$(S) \cup \{h\}$;

Шаг 2.1.3. Конец цикла;

Шаг 2.1.4. Конец процедуры.

Процедура разрешения многозначности иерархического контекста. Данная процедура предназначена для разрешения многозначности иерархического контекста. Входными данными для процедуры является синсет $S \in \mathcal{S}$ и его иерархический контекст hctx(S). Результатом выполнения процедуры является

иерархический контекст синсета S с разрешенной многозначностью $\widehat{\text{hctx}}(S)$. Процедура выглядит следующим образом:

Шаг 6.1.1. $\widehat{\text{hctx}}(S) \leftarrow \varnothing$;

Шаг 6.1.2. Для каждого слова $h \in \text{hctx}(S)$ выполнить цикл

Шаг 6.1.2.1. $\hat{h} \leftarrow \arg\max_{h' \in \text{senses}(h): S' \in \mathcal{S}, h' \in S', S \neq S'} \text{sim}_{\text{hctx}}(\text{hctx}(S), \text{words}(S'))$;

Шаг 6.1.2.2. $\widehat{\text{hctx}}(S) \leftarrow \widehat{\text{hctx}}(S) \cup \{\hat{h}\}$;

Шаг 6.1.3. Конец цикла;

Шаг 6.1.4. Конец процедуры.

2.4. Выводы по главе 2

В главе 2 описана модель представления знаний в виде семантической сети слов. Предложен метод построения семантической сети слов, метод построения синсетов в графе синонимов, метод построения связей между значениями слов. Предложенные модели, методы и алгоритмы предназначены для устранения проблем лексической многозначности и неполноты данных в семантических словарях.

Метод построения синсетов, предложенный в данной главе, предназначен для автоматического определения многозначных слов и их группировки в синсеты. Это позволяет сформировать вершины семантической сети слов — понятия. Предложенный метод отличается от аналогичных методов тем, что производит построение вспомогательного графа значений слов путем вывода значений слов и разрешения их многозначности. Это позволяет использовать доступные методы жесткой кластеризации графа. На основе метода построения синсетов предложен алгоритм Watset, осуществляющий построение синсетов по материалам слабоструктурированных словарей, представляющих отношение синонимии.

Метод построения связей, предложенный в данной главе, предназначен для автоматического определения наиболее подходящих вышестоящих слов по отношению к словам в синсетах. Это позволяет сформировать дуги семантической

сети слов. Предложенный метод отличается от аналогичных методов тем, что производит расширение доступных лексико-семантических ресурсов и осуществляет разрешение многозначности слов при помощи иерархических контекстов. На основе метода построения связей предложен алгоритм Watlink, осуществляющий построение и расширение семантических связей между значениями слов по материалам слабоструктурированных словарей, представляющих асимметричное отношение.

Глава 3. Комплекс программ построения семантической сети слов

На основе описанных в главе 2 моделей, методов и алгоритмов разработан комплекс программ автоматического построения семантической сети слов SWN (сокр. англ. *semantic word network* — семантическая сеть слов). Комплекс программ представляет собой совокупность приложений командной строки, позволяющий сформировать граф синонимов, определить лексические значения слов, построить граф значений слов, провести его кластеризацию и получить синсеты, построить и расширить иерархические контексты, и записать результат работы в виде семантической сети слов.

В данной главе описывается архитектура и особенности реализации моделей, методов и алгоритмов построения семантической сети слов в виде комплекса программ SWN. Исходные тексты разработанных программ доступны в сети Интернет по адресу https://github.com/dustalov/watset, https://github.com/dustalov/projlearn и https://github.com/dustalov/watlink.

3.1. Архитектура комплекса программ

Архитектура разработанного комплекса программ автоматического построения семантической сети слов SWN представлена в виде UML-диаграммы пакетов на рис. 3.1. В целях обеспечения тестируемости программ и возможности запуска различных сочетаний гиперпараметров предложенных алгоритмов, комплекс программ имеет модульную структуру и включает четыре основных модуля:

– модуль построения синсетов (`Watset`) реализует алгоритм Watset, описанный в разделе 2.2;
– модуль связывания (`Watlink`) реализует алгоритм Watlink, описанный в разделе 2.3;

– модуль подбора матрицы линейного преобразования (`Hyperstar`) реализует стабилизированный метод, описанный в разделе 2.3.3;
– модуль экспорта данных (`SWNRDF`) реализует преобразование семантической сети слов в стандартный формат представления семантических сетей RDF [24].

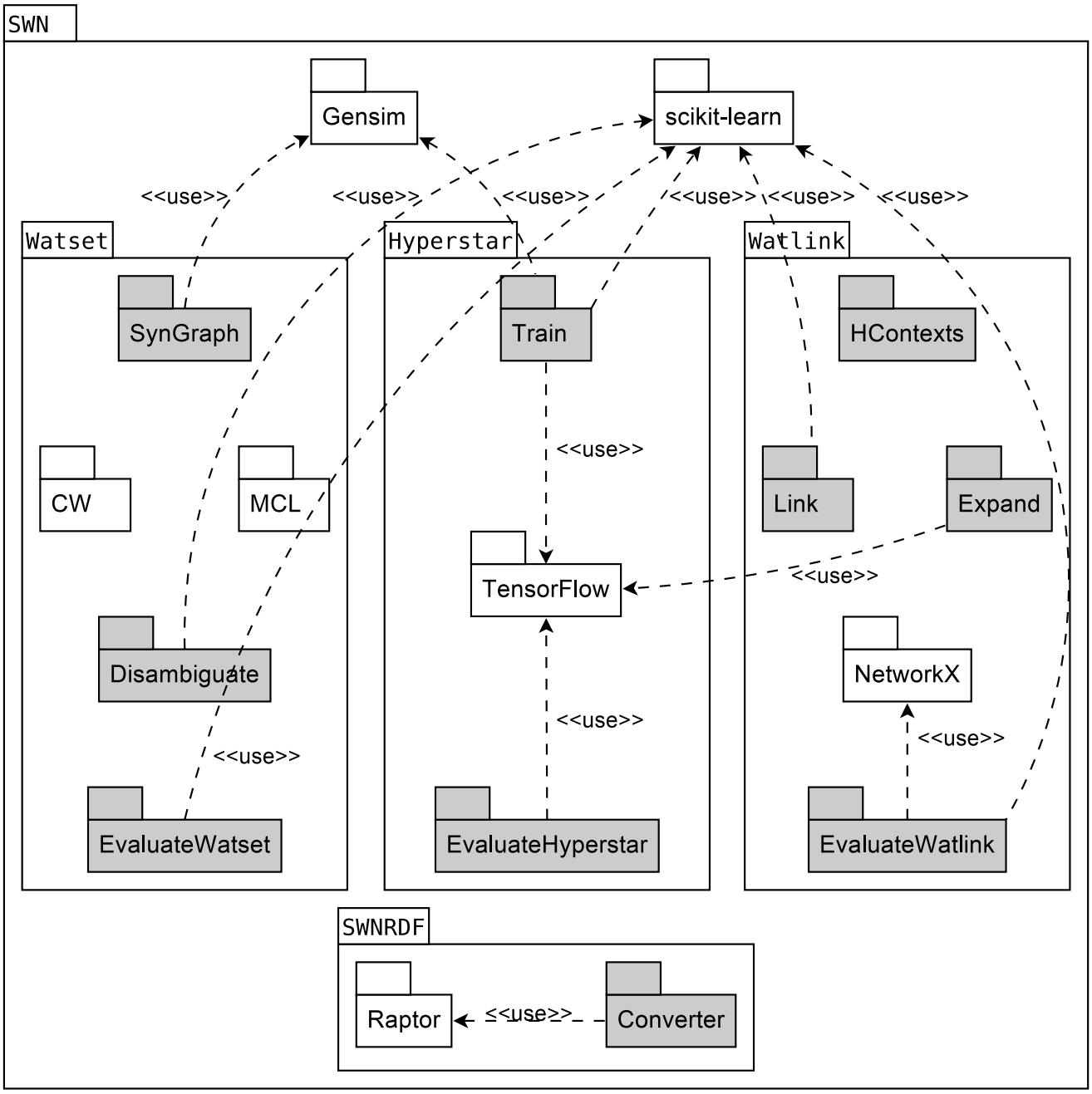


Рис. 3.1 — UML-диаграмма пакетов; цветом выделены программы, разработанные в рамках данной диссертационной работы

В состав комплекса входят следующие программы:

– программа построения графа синонимов *SynGraph*;
– программа разрешения многозначности в контекстах *Disambiguate*;
– программа оценки качества построения синсетов *EvaluateWatset*;
– программа подбора матрицы линейного преобразования *Train*;
– программа оценки качества подбора матрицы линейного преобразования *EvaluateHyperstar*;
– программа построения иерархических контекстов *HContexts*;
– программа расширения иерархических контекстов *Expand*;
– программа связывания значений слов *Link*;
– программа оценки качества связания значений слов *EvaluateWatlink*;
– программа преобразования семантической сети слов в формат Семантической паутины *Converter*.

3.1.1. Модуль построения синсетов

Модуль `Watset` реализует метод построения синсетов на основе графа синонимов, описанный в разделе 2.2. На рис. 3.2 представлена UML-диаграмма активности построения синсетов, состоящая из трех шагов:

– подготовка данных (программа *SynGraph*);
– построение синсетов (программы *CW*, *MCL* и *Disambiguate*);
– тестирование (программа *EvaluateWatset*).

Сначала производится загрузка материалы слабоструктурированных словарей и извлечение из них множества пар синонимов. При необходимости, вычисляется значение семантической близости между парами синонимов на основе косинусной меры близости между векторами слов. При отсутствии векторного представления некоторого слова, используется среднее значение близости, вычисленное по всем парам слов. Эти сведения используются при построении графа

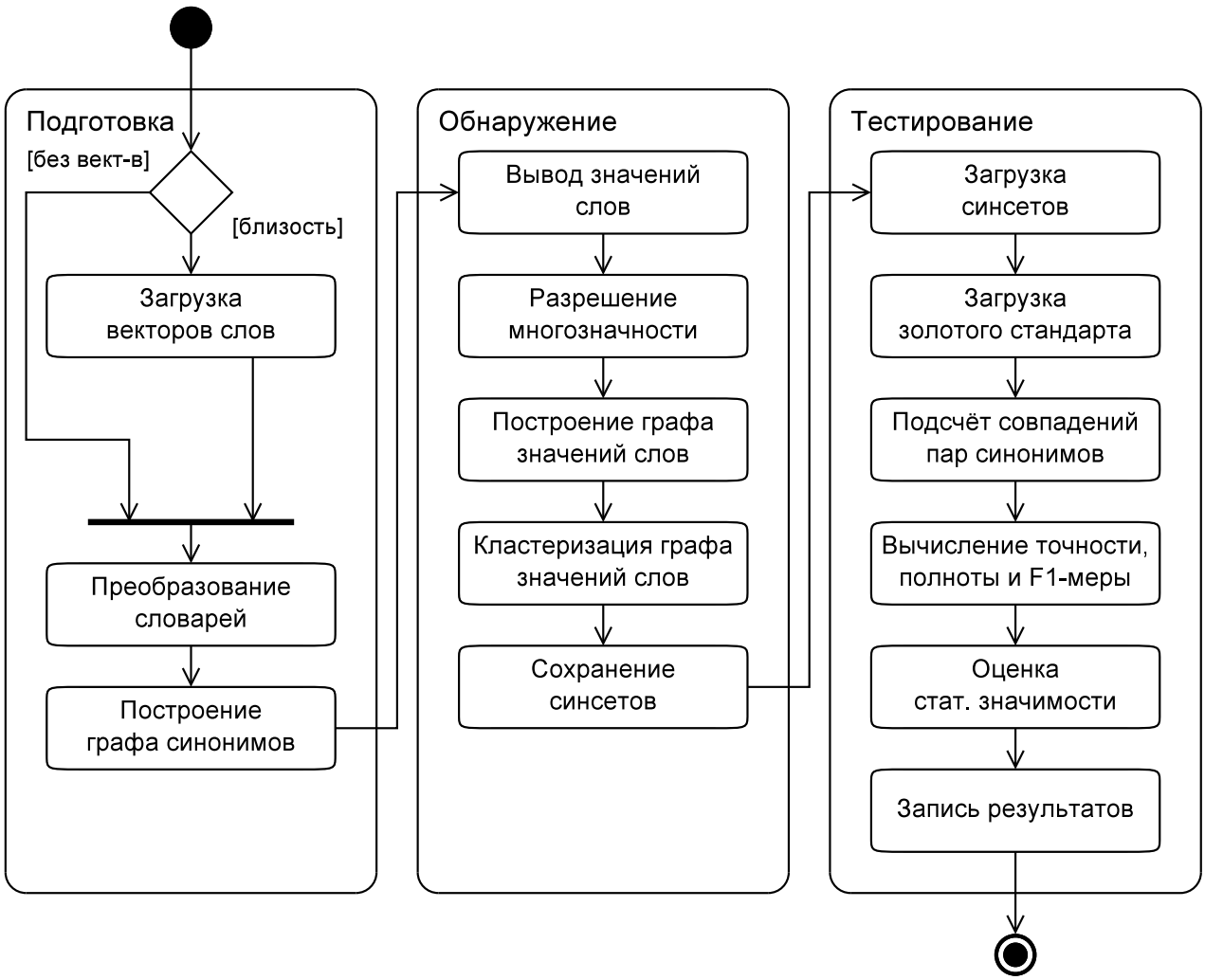


Рис. 3.2 — UML-диаграмма активности построения синсетов

синонимов. При отсутствии сведений о семантической близости слов предусмотрено два альтернативных варианта: использование единичных весов для каждого ребра графа синонимов или подсчет количества появлений пары синонимов в исходных словарях.

На этапе вывода значений слов допускается использование двух различных алгоритмов жесткой кластеризации графа: Chinese Whispers [26] или MCL [35]. На этапе разрешения многозначности производится разрешение многозначности в контекстах, причем в целях повышения производительности используется традиционный прием параллелизма по данным: каждое слово обрабатывается независимо в отдельном процессе. Определение номера значения слова в контексте производится путем максимизации косинусной меры близости (2.5). Если лексическое значение слова в контексте определить не удалось, то данное слово исключается из контекста. В результате разрешения многозначности

формируется граф значений слов, кластеризация которого для получения синсетов производится методом Chinese Whispers или MCL. Синсеты получают уникальные номера и записываются в текстовый файл. Это необходимо как для использования данных в других задачах, так и для оценки качества.

При оценке качества загружаются построенные синсеты и синсеты золотого стандарта. Затем, каждый синсет из n значений слов преобразуется во множество из $\frac{n(n-1)}{2}$ пар слов и производится подсчет совпадений пар синонимов в полученном ресурсе и золотом стандарте. Вычисляются значения попарных информационно-поисковых критериев точности, полноты и F_1-меры [69] и оценивается статистическая значимость значения каждого критерия (см. раздел 1.3). После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

3.1.2. Модуль подбора матрицы линейного преобразования

Модуль `Hyperstar` осуществляет подбор матрицы линейного преобразования векторных представлений нижестоящих слов в векторные представления вышестоящих слов на основе модифицированного подхода, первоначально предложенного в [45]. На рис. 3.3 представлена UML-диаграмма активности подбора матрицы линейного преобразования, состоящая из трех условных шагов:

– подготовка данных (программа *Train*, режим подготовки);
– обучение модели (программа *Train*, режим подбора параметров);
– тестирование (программа *EvaluateHyperstar*).

Исходными данными для подбора матрицы являются векторы слов и упорядоченные пары слов, порожденные асимметричным отношением, полученные из словарей. В процессе используются только те пары слов, для которых имеются векторы. Это вызвано тем, что векторы слов строятся на основании большого корпуса текстов с различными подходами к предварительной обработке, например,

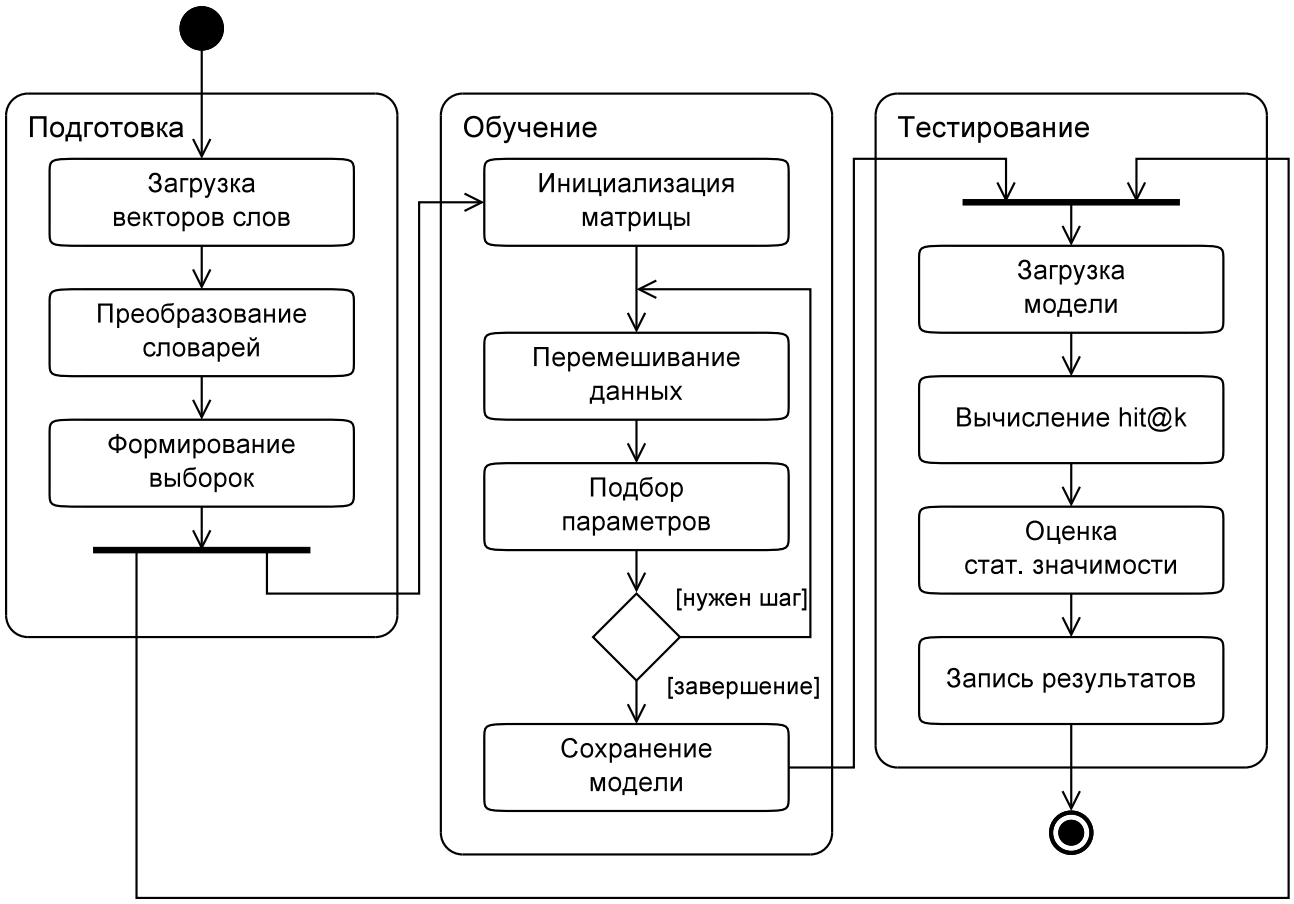


Рис. 3.3 — UML-диаграмма активности подбора матрицы линейного преобразования

фильтрации низкочастотных слов [18]. Полученные пары векторов слов разбиваются на три различные выборки в соотношении: 60 % данных составляют обучающую выборку для подбора параметров, 20 % данных составляют проверочную выборку для подбора гиперпараметров, и оставшиеся 20 % составляют тестовую выборку для оценки качества модели.

В начале процесса обучения все элементы матрицы генерируются как независимые между собой случайные величины, имеющие стандартное нормальное распределение с параметрами $\mu = 0$ и $\sigma = 0{,}1$; допущения о свойствах матрицы не используются [45]. На каждом шаге обучения производится перемешивание данных и выполняется подбор значений элементов матрицы с целью минимизации функции потерь (2.14). Процесс обучения завершается по достижении указанного при запуске количества шагов; двоичное представление полученной матрицы записывается в файл.

Оценка качества предполагает загрузку полученных матриц и вычисление значения критерия hit@k по проверочной выборке для подбора параметров или по тестовой выборке для оценки качества работы метода. Оценивается статистическая значимость значения данного критерия (см. раздел 1.3). После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

3.1.3. Модуль построения связей

Модуль `Watlink` реализует метод построения связей, описанный в разделе 2.3. Также данный модуль осуществляет построение семантической сети слов на основе ранее полученных синсетов и доступных упорядоченных пар слов, порожденных асимметричным отношением. На рис. 3.4 представлена UML-диаграмма активности построения связей, состоящая из четырех условных шагов:

– подготовка данных (программа *HContexts*);
– расширение (программа *Expand*; опциональный шаг)
– связывание (программа *Link*);
– тестирование (программа *EvaluateWatlink*).

Исходными данными для построения связей являются синсеты и материалы слабоструктурированных словарей, содержащих перечисленные в текстовом виде упорядоченные пары слов, порожденные асимметричным отношением. На основе этих сведений формируются иерархические контексты каждого синсета. При необходимости загружаются векторы слов, матрица линейного преобразования, и осуществляется расширение иерархических контекстов с использованием ранее полученной матрицы линейного преобразования.

Разрешение многозначности в иерархических контекстах реализовано с использованием трех различных подходов к взвешиванию каждого вышестоящего слова в иерархическом контексте: tf, idf и tf–idf [11], причем под «термином» понимается слово, а под «документом» понимается иерархический контекст

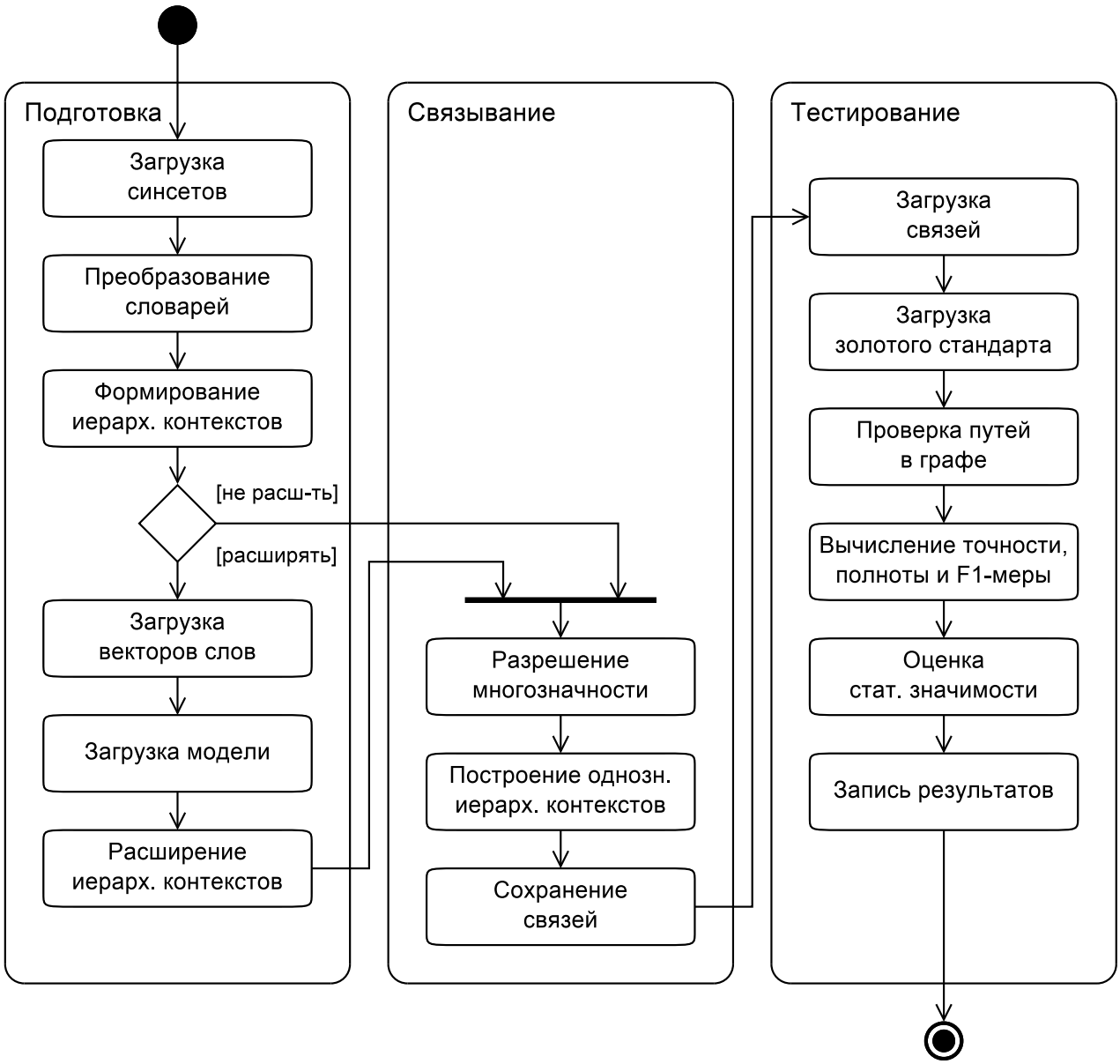


Рис. 3.4 — UML-диаграмма активности построения связей

синсета. В целях повышения производительности, используется традиционный прием параллелизма по данным: каждый синсет обрабатывается независимо в отдельном процессе. В иерархическом контексте $\mathrm{hctx}(S)$ синсета S номер значения каждого слова $h \in \mathrm{hctx}(S)$ определяется путем максимизации косинусной меры близости (2.16). Если лексическое значение слова в иерархическом контексте определить не удалось, то данное слово исключается из иерархического контекста. При построении иерархических контекстов со снятой неоднозначностью используется только несколько элементов иерархического контекста, получившие максимальный вес по итогам выполнения этапа разрешения многозначности. Семантическая сеть слов сохраняется в текстовый файл.

При оценке качества загружаются построенные связи и связи между словами золотого стандарта в виде ориентированных графов. Затем, для каждой пары слов проверяется существование пути от нижестоящего слова к вышестоящему в графе значений золотого стандарта. Вычисляются значения информационно-поисковых критериев точности, полноты и F_1-меры [11] и оценивается статистическая значимость значения каждого критерия (см. раздел 1.3). После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

3.2. Реализация комплекса программ

При реализации комплекса программ SWN использованы языки программирования Python, AWK и Bash. Благодаря доступности высококачественных библиотек для решения задач анализа данных и встроенной поддержке многобайтовых кодировок, в качестве основного языка программирования выбран Python версии 3. Вспомогательные программы обработки и преобразования данных написаны на языке программирования AWK (диалект GNU AWK). Связывание программ на разных языках программирования осуществляется при помощи сценариев командного интерпретатора Bash. Целевой операционной системой является Linux с поддержкой 64-битной адресации памяти.

В целях повышения скорости разработки, используются следующие внешние библиотеки и зависимости:

- библиотека алгоритмов машинного обучения, подготовки и обработки данных scikit-learn [85] для языка программирования Python;
- реализация алгоритма кластеризации Chinese Whispers [26] (CW) на языке программирования Java;
- реализация марковского алгоритма кластеризации [35] (MCL) на языке программирования Си;

– библиотека тематического моделирования и работы с векторами слов Gensim [90] для языка программирования Python;
– библиотека методов оптимизации TensorFlow [16] для языка программирования Python;
– библиотека работы с графами NetworkX [51] для языка программирования Python;
– реализация средств обработки RDF-троек Raptor [23] на языке программирования Си.

На рис. 3.5 представлена UML-диаграмма вариантов использования комплекса программ построения семантической сети слов SWN.

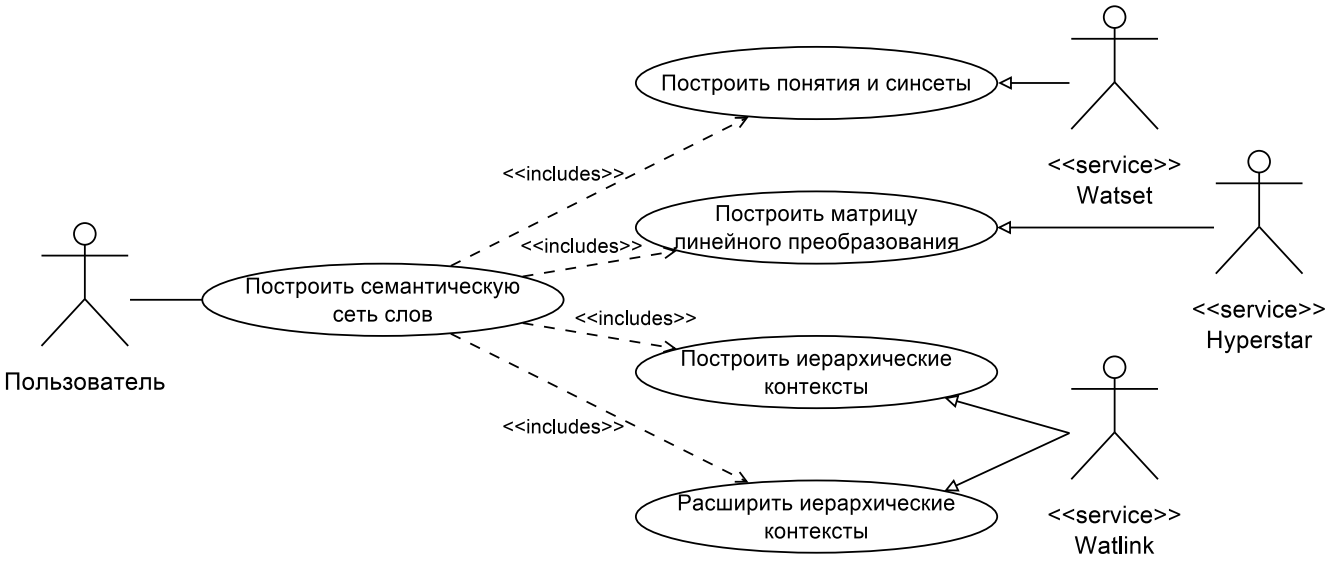


Рис. 3.5 — UML-диаграмма вариантов использования комплекса программ построения семантической сети слов

При реализации операций максимизации или минимизации (2.5, 2.14, 2.16), использован, соответственно, аргумент максимизации или минимизации, при прочих равных, с минимальным идентификатором. В качестве меры близости во всех случаях использована косинусная мера близости. В качестве векторного представления контекстов, иерархических контекстов, а также слов в синсетах, использована общепринятая модель «мешка слов» [11].

Запись синсетов и семантической сети слов производится в текстовых файлах в кодировке UTF-8. Поля разделяются запятыми. В целях представления

лексических значений слов в текстовых файлах используется расширение нотации, используемой в BabelNet [76]. Запись *слово*^i_t означает i-е значение слова, принадлежащее части речи t. Например, запись *лук*^2_n означает, что слово «лук» является именем существительным (англ. *noun*) и использовано во втором значении («лук как стрелковое оружие»).

С одной стороны, такое представление в текстовых файлах требует заданных разделителей для трех полей: лексемы, части речи, и номера значения. С другой стороны, такое представление не позволяет указать словарные пометы и частоту встречаемости данного значения, хотя такие сведения, особенно частотные, очень важны для решения практических задач [10].

Для решения этой проблемы записи значений слов в текстовые файлы использована контекстно-свободная грамматика на рис 3.6, составленная в виде расширенной формы Бэкуса — Наура при помощи утилиты ANTLR [84]. Данная грамматика позволяет выражать слова без значений (`кот`), слова со значениями с указанием и без указания части речи (`кот^NOUN#1` и `кот#1`), а также словарные пометы (`котенок#1_уменьш-ласк`) и частоту (`котенок:10.5`). Многословные выражения и множественные пометы разделяются знаком подчеркивания (_). Знаки пробела, табуляции и перевода строки не допускаются.

Например, запись `нечистая^ADJF_сила^NOUN#1_перен:2.28` означает лексическую единицу из двух слов (имени прилагательного «нечистая» и имени существительного «сила»), употребленную в первом значении с пометой переносного смысла. Частотность данной лексической единицы составляет 2,28. Разбор данного примера представлен на рис. 3.7. Цветом выделены терминалы, соответствующие словам, частям речи, номеру значения, словарной помете, частотности, а также разделительным символам.

```
sememe     : lexeme (HASH sense)? (COLON frequency)? EOF ;

lexeme     : span (UNDERSCORE span)* ;

span       : lemma (HAT pos)? ;

lemma      : STRING ;

pos        : STRING ;

sense      : id labels? ;

id         : INTEGER ;

labels     : (UNDERSCORE label)+ ;

label      : STRING ;

frequency  : INTEGER | DECIMAL ;

INTEGER    : [0—9]+ ;

DECIMAL    : [0—9]* DOT [0—9]+ ;

STRING     : CHAR+ ;

CHAR       : ~[^#:_\t\n\r ] ;

HAT        : '^' ;

HASH       : '#' ;

COLON      : ':' ;

UNDERSCORE : '_' ;

DOT        : '.' ;
```

Рис. 3.6 — Грамматика ANTLR

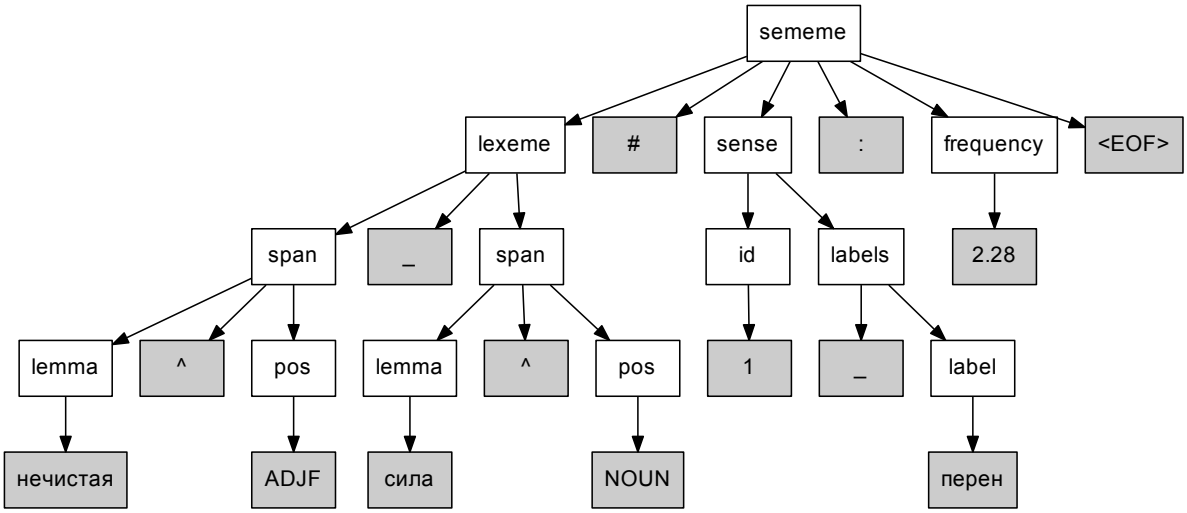


Рис. 3.7 — Пример разбора записи значения слова

3.3. Представление знаний

С целью обеспечения интероперабельности и интеграции комплекса программ с внешними информационными системами, семантическая сеть слов записывается при помощи формализма RDF (англ. *Resource Description Framework*). Нотация RDF предполагает представление знаний в виде троек «субъект–предикат–объект» [24]. Все сущности, полученные в результате работы разработанных методов, преобразованы в RDF-тройки с использованием моделей SKOS [19] и Lemon [70]. На рис. 3.8 изображена диаграмма классов полученной семантической сети слов с точки зрения формализма VOWL [66]. В качестве формата хранения используются текстовые форматы Turtle и N-Triples; для идентификации троек используется префикс `urn:swn`.

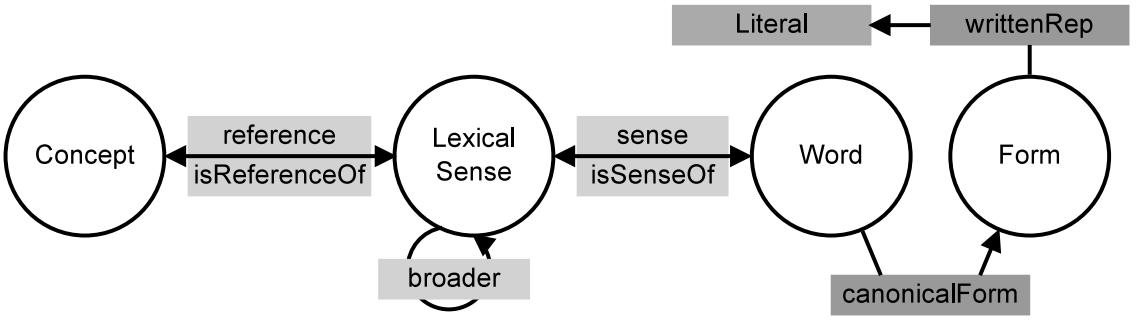


Рис. 3.8 — VOWL-диаграмма семантической сети слов

Таблица 3 — Семантическая сеть слов в виде связанных данных

Сущность	**Назначение**	**Класс**
Слово	Лексическая единица	`lemon:Word`
	Каноничная форма (лемма)	`lemon:Form`
	Связь единицы и леммы	`lemon:canonicalForm`
	Заданное значение слова	`lemon:sense`
	Письменное представление слова	`lemon:writtenRep`
Значение слова	Лексическое значение слова	`lemon:LexicalSense`
	Заданная лексическая единица	`lemon:isSenseOf`
	Заданное понятие	`lemon:reference`
Гипероним	Вышестоящее значение слова	`lemon:broader`
Гипоним	Нижестоящее значение слова	`lemon:narrower`
Понятие	Множество синонимов	`skos:Concept`
	Заданное лексическое значение	`lemon:isReferenceOf`

Преобразование данных осуществляется путем записи каждого элемента данных в виде экземпляров классов, соответствующих моделям SKOS и Lemon (табл. 3). Сначала каждое слово записывается в качестве экземпляра класса `lemon:Word`. Поскольку данный класс справедливо предполагает, что слово может иметь несколько различных форм, то каноничная форма слова (лемма) записывается отдельно в качестве экземпляра класса `lemon:Form`. Между экземпляром слова и его леммой строится свойство каноничной формы `lemon:canonicalForm`. Строковое представление слова записано в виде литерала, связанного с экземпляром леммы свойством `lemon:writtenRep`. Лексические значения слов записываются в виде экземпляров класса `lemon:LexicalSense`. Поскольку в семантической сети слов семантические связи формируется между отдельными значениями слов, то нижестоящие значения слов связываются с вышестоящими при помощи свойства `lemon:broader` и обратного ему свойства `lemon:narrower`. Так как значения слов группируются в синсеты, то для сохранения данной информации каждое

лексическое значение привязывается к экземпляру класса `skos:Concept`, соответствующего синсету, при помощи свойства `lemon:reference` и обратного ему свойства `lemon:isReferenceOf`.

3.4. Выводы по главе 3

В главе 3 представлен комплекс программ SWN, реализующий методы, модели и алгоритмы, предложенные в главе 2. Комплекс программ осуществляет построение семантической сети слов и ее запись в форматах Семантической паутины на основе информационной модели (рис. 3.8). Описана архитектура комплекса программ, включающего в себя программы построения синсетов, построения и расширения семантических связей.

Программы построения семантической сети слов написаны с использованием параллелизма по данным, что позволяет использовать вычислительные узлы с большим количеством доступных ядер центрального процессора для ускорения вычислений. Кроме того, при реализации методов использованы высокоэффективные внешние библиотеки, такие как scikit-learn [85] и TensorFlow [16]. В настоящее время все программы функционируют в режиме командной строки. Связывание программ, написанных на разных языках программирования, осуществляется путем перенаправления потоков стандартного ввода и вывода в сценариях командного процессора Bash и утилиты make.

Входные данные представлены в виде текстовых файлов, поля в которых разделены знаком табуляции. Все промежуточные результаты, кроме матрицы линейного преобразования, также представлены в текстовом виде. Итоговый результат записывается в стандартном формате представления семантических сетей N-Triples [23]. При реализации комплекса программ SWN использованы языки программирования Python, AWK и Bash. Исходные тексты разработанных программ доступны в сети Интернет по адресу https://github.com/dustalov/watset, https://github.com/dustalov/projlearn и https://github.com/dustalov/watlink.

Глава 4. Оценка эффективности разработанных методов

В данной главе производится экспериментальная оценка эффективности разработанных методов. Эксперименты основаны на сопоставлении результатов выполнения предложенных в данной работе методов с материалами *золотого стандарта* — заранее выбранного набора данных известного качества. При использовании подхода к оценке на основе золотого стандарта применяются количественные меры качества, выражающие похожесть исследуемого набора данных на золотой стандарт (более подробные сведения о методологии оценки качества семантических сетей изложены в разделе 1.3).

Понятия и связи являются объектами различной природы и не существует общепринятого и универсального подхода к интегральной оценке (см. раздел 1.3). Экспериментальная оценка понятий и связей в данной главе будет производиться отдельно:

- при оценке качества *понятий* важно, чтобы слова, входящие в синсеты в оцениваемом наборе данных, также являлись синонимами и в золотом стандарте;
- при оценке качества *связей* важно, чтобы связь между словами, представленная в оцениваемом наборе данных, также была представлена и в золотом стандарте.

Наборы данных и используемые в экспериментах меры качества приведены в табл. 4. При оценке качества по материалам золотого стандарта, верным срабатыванием является случай, при котором пара связанных слов из оцениваемого набора данных присутствует и в золотом стандарте, или случай, при котором пара связанных слов, отсутствующая в оцениваемом наборе данных, также отсутствует и в золотом стандарте. Предполагается, что если в золотом стандарте не представлена некоторая имеющееся в оцениваемом наборе данных пара слов, то такая пара слов является некорректно определенной. Такой подход приводит

к выставлению более высоких оценок методам, которые одновременно обладают большим количеством верных срабатываний и низким количеством ложных срабатываний.

Таблица 4 — Методы и меры качества в экспериментах

Метод	**Золотой стандарт**	**Мера качества**
Построение синсетов	RuWordNet, Yet Another RussNet	Попарная точность, полнота и F_1-мера
Построение связей	RuWordNet	Точность, полнота и F_1-мера на основе проверки существования путей в графе
Расширение связей	Русский Викисловарь	Доля примеров с хотя бы одним корректным ответом в десяти первых ответах (hit@10)

Метод построения синсетов является методом нечеткой кластеризации графа синонимов; оценка его качества будет производиться путем сравнения результатов с другими методами кластеризации графа. Для этого используется три меры качества, используемые при оценке методов кластеризации [69]: попарная точность, полнота и F_1-мера. В этом случае и оцениваемый набор данных, и золотой стандарт преобразуется во множество пар синонимов. Оценка производится путем сопоставления пар синонимов оцениваемого набора данных с парами синонимов золотого стандарта.

Метод построения связей оценивается при помощи подхода на основе проверки существования пути между словами в графе золотого стандарта, описанный в разделе 1.3. Такой подход позволяет вычислить точность, полноту и F_1-меру. В этом случае золотой стандарт преобразуется в семантическую сеть слов, с которой происходит сопоставление и проверка наличия пути между вершинами в оцениваемом наборе данных.

Метод подбора матрицы линейного преобразования, используемый для расширения иерархических контекстов при построении связей, оценивается при

помощи общепринятой методологии оценки методов машинного обучения с учителем. Для этого формируется обучающая, проверочная и тестовая выборка. Поскольку метод подбора матрицы линейного преобразования для каждого входного слова возвращает несколько возможных ответов, то для оценки качества используется мера hit@k [44]. Данная мера качества засчитывает засчитывает верный ответ тогда и только тогда, когда хотя бы один из первых k ответов метода совпадает с ответом из тестовой выборки, соответствующей входному слову.

Все вычислительные эксперименты проводились в среде облачных вычислений Azure, доступ к которой предоставлен корпорацией «Майкрософт» в рамках программы Azure for Research [73]. Использовалась виртуальная машина класса NC24 под управлением гипервизора Hyper-V, основные параметры которой приведены в табл. 5.

Таблица 5 — Параметры вычислительной среды

Параметр	**Значение**
Тип центрального процессора	Intel Xeon E5-2690 v3
Количество доступных ядер	24 ядра
Объем оперативной памяти	224 ГБ
Тип графического процессора	NVIDIA Tesla K80
Объем видеопамяти	12 ГБ
Операционная система	CentOS 7.3.1611 (64 бит, Linux)

Сначала производится оценка метода построения синсетов Watset. На основе лучшей конфигурации этого метода производится экспериментальная оценка метода построения связей Watlink без расширения иерархических контекстов, а также стабилизированного метода подбора матрицы линейного преобразования. Затем, на основе лучшей конфигурации метода подбора матрицы линейного преобразования производится экспериментальная оценка метода построения связей с расширением иерархических контекстов.

При проведении всех экспериментов использованы 500-мерные векторные представления слов [18], построенные по материалам электронной библиотеки lib.rus.ec на основе модели Skip-gram [74].

4.1. Оценка метода построения синсетов

Для экспериментальной оценки метода Watset, описанного в разделе 2.2, производится сравнение с аналогичными методами по материалам двух различных золотых стандартов: RuWordNet и Yet Another RussNet. В качестве меры качества используется попарная точность, полнота и F_1-мера. Для этого в золотом стандарте и оцениваемом наборе данных каждый синсет, содержащий $n \in \mathbb{N}$ слов преобразуется в $\frac{n(n-1)}{2}$ пар синонимов для оценки [69]. С целью исключения некорректных и непроверенных данных, в набор данных на основе тезауруса Yet Another RussNet включены только пары синонимов, состоящие в синсетах, которые редактировались не менее восьми раз. Таким образом, использовано $278\,381$ пар синонимов из тезауруса RuWordNet [67] и $48\,291$ пар синонимов из тезауруса Yet Another RussNet [30]. Вычисление мер качества производится на основании наличия или отсутствия определенных пар синонимов в золотом стандарте. Лучшими в данном эксперименте считаются методы, получившие высокие значения полноты и F_1-меры.

4.1.1. Описание эксперимента

В рамках данного эксперимента проводилось сравнение шести различных алгоритмов кластеризации графа:

- Алгоритм Watset, описанный в разделе 2.2, являющийся алгоритмом нечеткой кластеризации графа. Использована реализация данного алгоритма на языке программирования Python, описанная в разделе 3.1.1. В экспериментах использованы следующие значения гиперпараметров:
  - $\text{Cluster}_{\text{Local}} \in \{\text{CW}_{\text{top}}, \text{CW}_{\text{nolog}}, \text{CW}_{\text{log}}, \text{MCL}\}$;
  - $\text{Cluster}_{\text{Global}} \in \{\text{CW}_{\text{top}}, \text{CW}_{\text{nolog}}, \text{CW}_{\text{log}}, \text{MCL}\}$;
  - $\text{sim}_{\text{word}} \in \{\text{ones}, \text{count}, \text{sim}\}$;

  - $\mathrm{sim}_{\mathrm{ctx}} = \cos$.
- Алгоритм испорченного телефона (англ. *Chinese Whispers*, сокр. *CW*), являющийся алгоритмом жесткой кластеризации графа [26]. Использована оригинальная реализация алгоритма на языке программирования Java, предоставленная авторами алгоритма и включающая три различные его вариации:
  - $\mathrm{CW}_{\mathrm{top}}$: оригинальный вариант алгоритма испорченного телефона;
  - $\mathrm{CW}_{\mathrm{nolog}}$: вариант алгоритма испорченного телефона с использованием степени вершины для назначения кластеров;
  - $\mathrm{CW}_{\mathrm{log}}$: то же, что и предыдущий вариант, но с использованием натурального логарифма степени вершины.
- Марковский алгоритм кластеризации (англ. *Markov Clustering*, сокр. *MCL*), являющийся алгоритмом жесткой кластеризации графа [35]. Использована оригинальная реализация алгоритма на языке программирования Си, предоставленная автором алгоритма.
- Алгоритм MaxMax, являющийся алгоритмом нечеткой кластеризации графа [55]. Поскольку реализация данного алгоритма отсутствует в публичном доступе, использована собственная реализация алгоритма на языке программирования Java. Реализация основана на полуформализованном описании, представленном авторами оригинальной статьи в виде псевдокода.
- Алгоритм кластеризации ECO, являющийся алгоритмом нечеткой кластеризации графа [49]. Поскольку реализация данного алгоритма отсутствует в публичном доступе, использована собственная реализация алгоритма на языке программирования Python. Реализация основана на кратком текстовом описании, представленном авторами оригинальной статьи. Из-за нехватки подробностей в описании данного метода, вероятность попадания слов u и v в один кластер сравнивается с заданным

пороговым значением и оценивается как

$$p_{u,v} = \frac{\#(u,v)}{\#(u) + \#(v) - \#(u,v)},$$

где $\#(u,v)$ — количество появлений слов u и v в одном кластере, $\#(u)$ и $\#(v)$ — общее количество появлений слов u и v, соответственно.

– Метод перколяции клик (англ. *Clique Percolation Method*, сокр. *CPM*), являющийся алгоритмом нечеткой кластеризации графа [80]. Использована реализация данного алгоритма на языке программирования Python из библиотеки NetworkX [51]. Алгоритм предназначен для невзвешенных графов, поэтому при выполнении экспериментов веса ребер игнорировались. Использованы следующие значения гиперпараметра, указывающего размер минимальной клики: $k \in \{2, 3, 4\}$.

Запись Watset[MCL, CW_{top}] означает, что для вывода значений слов ($\text{Cluster}_{\text{Local}}$) использован марковский алгоритм кластеризации, а для кластеризации графа значений слов ($\text{Cluster}_{\text{Global}}$) использован оригинальный вариант алгоритма испорченного телефона.

В эксперименте использованы доступные русскоязычные словари синонимов в качестве исходных данных:

– Словарь русских синонимов и сходных по смыслу выражений Н. А. Абрамова [1];
– Русский Викисловарь, извлечение данных из которого выполнено при помощи утилиты Wikokit [8];
– Универсальный словарь концептов [34].

На основе объединения трех исходных словарей построен объединенный граф синонимов — неориентированный граф, состоящий из $83\,092$ вершин, соединенных $211\,986$ ребрами. С целью изучения влияния весов ребер графа синонимов $W = (V, E)$ на результат кластеризации, в эксперименте рассмотрено три различных меры близости слов sim_{word}:

– ones: каждому ребру $\{u, v\} \in E$ назначается одинаковый вес:

$$\text{weight}(u, v) = 1;$$

– count: каждому ребру $\{u, v\} \in E$ назначается вес, равный количеству появлений соответствующей пары слов в словарях синонимов $\mathbb{D}$:

$$\text{weight}(u, v) = \sum_{D \in \mathbb{D}} \mathbb{1}_D((u, v)),$$

где $\mathbb{1}_D$ — индикаторная функция множества D;

– sim: каждому ребру $\{u, v\} \in E$ назначается вес, равный значению косинуса угла между векторными представлениями слов:

$$\text{weight}(u, v) = \cos(\vec{u}, \vec{v}).$$

Поскольку словник используемых словарей отличается от словника золотого стандарта, то при вычислении информационно-поисковых оценок использовались только те пары синонимов, оба слова которых входят в пересечение словника золотого стандарта и объединенного словника наборов данных, полученных в результате выполнения методов кластеризации.

4.1.2. Результаты эксперимента

Результаты сравнения методов на материалах двух золотых стандартов приведены в табл. 6. В этой таблице колонка «# синсетов» означает количество синсетов, выделенных методом; колонка «# пар» означает общее количество пар синонимов, образованных синсетами. Полужирным шрифтом в таблице выделены наибольшие значения соответствующих критериев, причем лучший результат дополнительно выделен подчеркиванием. В таблице приведены результаты только лучших конфигураций всех сочетаний методов. Среди трех вариантов метода испорченного телефона (CW) лучший результат на обоих наборах данных показал CW_{nolog}. Среди использованных значений параметра k метода перколяции клик (CPM) лучший результат на обоих наборах данных показало значение $k = 3$.

На рис. 4.1 представлено сравнение результатов работы оцениваемых методов в зависимости от используемого подхода к взвешиванию графа синонимов.

Видно, что оценки, полученные при использовании подхода sim выше, чем при использовании подходов ones и count, на всех наборах данных. Таким образом, анализ результатов будет осуществляться только на результатах взвешивания графа синонимов при помощи семантической близости слов, обозначаемое как sim.

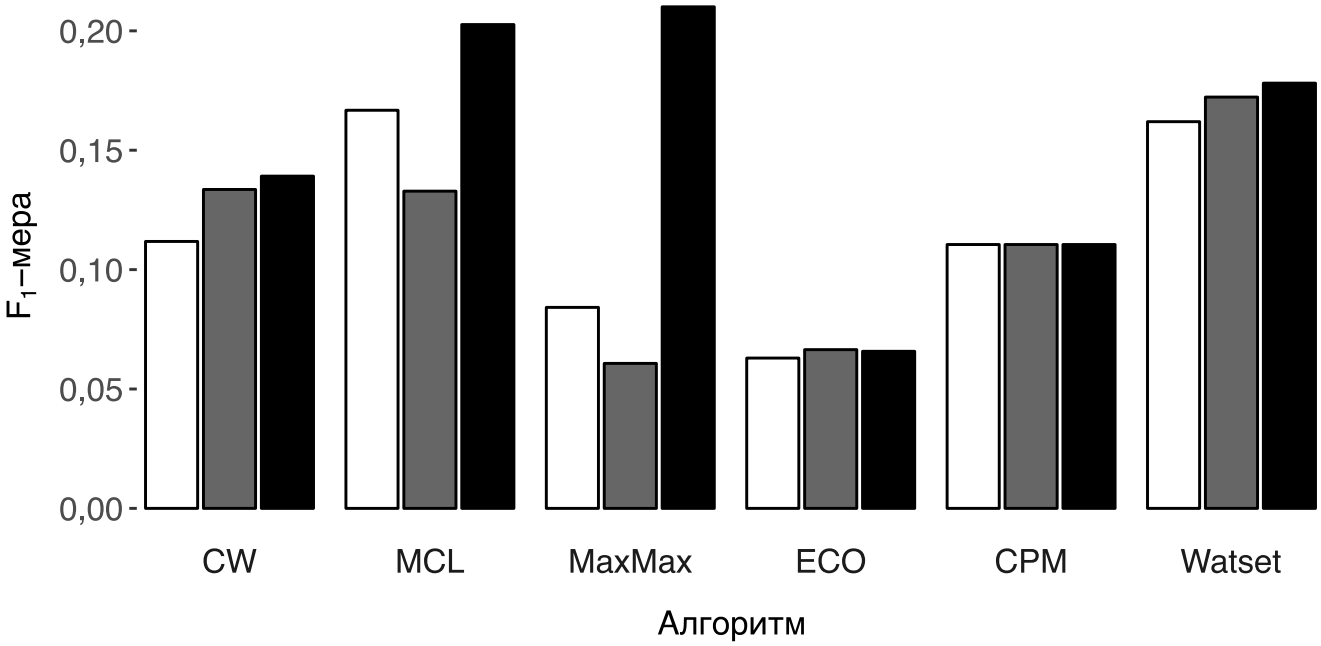
Важно отметить, что при определенных условиях методы MaxMax и CPM генерировали синсеты, содержащие 150 и более слов. Анализ результатов показал нерелевантность таких синсетов, что привело к их исключению из процесса попарной оценки. Другие методы не демонстрировали подобного поведения.

Таблица 6 — Сравнение методов построения синсетов по материалам RuWordNet и Yet Another RussNet

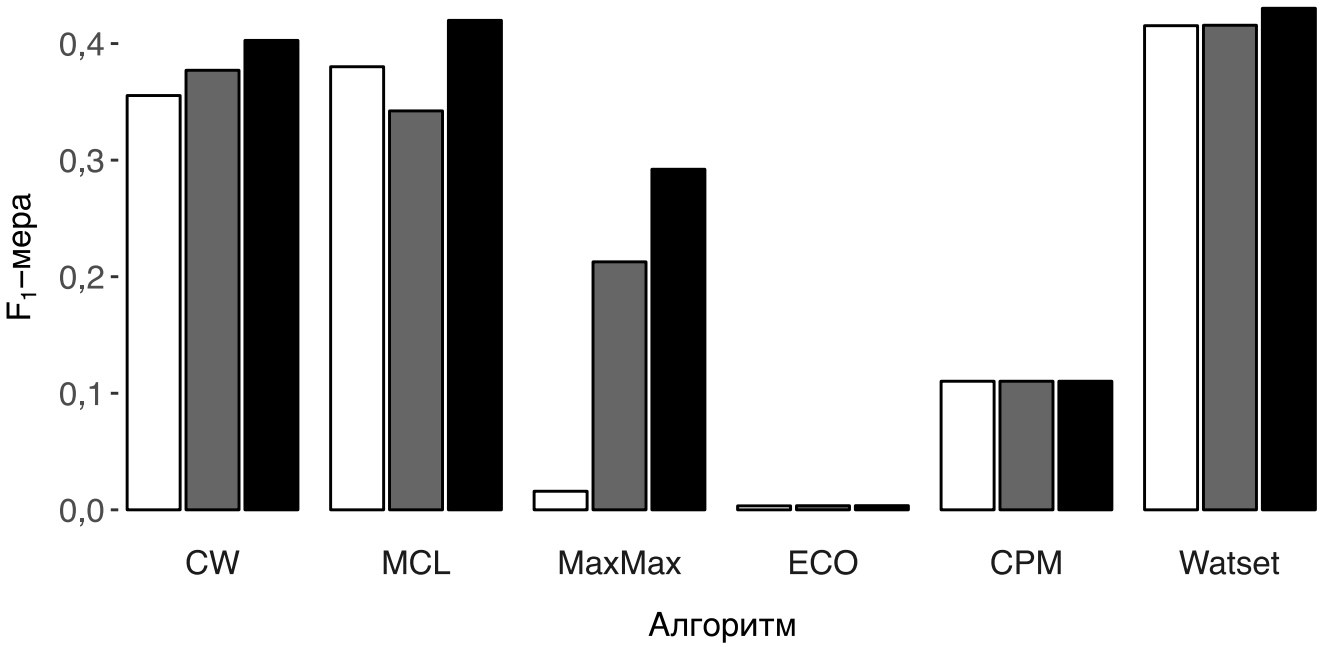
Метод	**# синсетов**	**# пар**	**RuWordNet**			**Yet Another RussNet**		
			Точность	**Полнота**	**F_1-мера**	**Точность**	**Полнота**	**F_1-мера**
Watset[CW_{nolog}, MCL]	55 369	332 727	0,120	**0,349**	**0,178**	0,402	**0,463**	**0,430**
Watset[MCL, MCL]	36 217	403 068	0,111	0,341	0,168	0,405	0,455	**0,428**
Watset[CW_{top}, CW_{log}]	55 319	341 043	0,116	**0,351**	0,174	0,386	**0,474**	**0,425**
MCL	21 973	353 848	0,155	0,291	**0,203**	0,550	0,340	0,420
Watset[MCL, CW_{top}]	34 702	473 135	0,097	**0,361**	0,153	0,351	**0,496**	0,411
CW_{nolog}	19 124	672 076	0,087	0,342	0,139	0,364	0,451	0,403
MaxMax	27 011	461 748	**0,176**	0,261	**0,210**	**0,582**	0,195	0,292
$CPM_{k=3}$	4 000	45 231	**0,234**	0,072	0,111	**0,626**	0,060	0,110
ECO	67 645	18 362	**0,724**	0,034	0,066	**0,904**	0,002	0,004

4.1.3. Анализ результатов

Алгоритм испорченного телефона (*Chinese Whispers*) корректно сгенерировал большое количество синсетов, образованных моносемичными словами и именами собственными, например: {*туфля*[1], *полуботинок*[1], …}. В свою очередь,



а) Сравнение алгоритмов по материалам RuWordNet



б) Сравнение алгоритмов по материалам Yet Another RussNet

Обозначения: ☐ ones, ▒ count, ■ sim.

Рис. 4.1 — Влияние подхода к взвешиванию графа синонимов на результаты работы алгоритма построения синсетов по критерию F_1-меры

при проявлении полисемии возникают заметные проблемы, состоящие в объединении никак не связанных друг с другом слов, например: {*лук*[1], *порей*[1], *налучник*[1], *налучь*[1], …}.

Несмотря на то, что алгоритм MaxMax хорошо проявил себя с точки зрения критерия точности на материалах двух золотых стандартов, обнаружены две трудности в его практическом применении, существенно затрудняющие его практическое использование:

– данный алгоритм очень чувствителен к однородности весов ребер на этапе преобразования графа, что выражается в появлении крупных кластеров, связывающих семантически никак не связанные слова, например {*прайс*[1], *бином Ньютона*[1], *программный пакет*[1], …};
– алгоритм не имеет механизма контроля гранулярности синсетов и при определенных условиях генерирует синсеты, состоящие из большого количества семантически связанных слов, не являющимися синонимами, например {*Афродита*[1], *Мефистофель*[1], *Самаэль*[1], …}.

Результаты выполнения метода ECO не согласуются с отчетами об его успешном применении [49]. Синсеты, состоящие из двух или более слов, образованы только моносемичными словами. При этом для многозначных слов вероятность попасть в синсеты с другими словами не превысила пороговое значение, поэтому были образованы некорректные однословные синсеты: {*колонна*[1]}, {*шатун*[1]}, и др. С одной стороны, структура графа синонимов русского языка, использованного в данной работе, может отличаться от структуры графа, использованной в исследовании словарей португальского языка. С другой стороны, вероятность попадания слов в кластер может оцениваться иным образом, но в описании метода ECO не хватает существенных подробностей.

Метод перколяции клик показал неудовлетворительные результаты. Вероятно, потому, что структура k-клик отличается от фактической структуры графа синонимов. Это приводит к появлению таких синсетов, как {*МП*[1], *медицинский пункт*[1], *Московская Патриархия*[1], …} и {*заливное*[1], *студень*[1], *студенческий билет*[1], …}. Как и в случае алгоритма испорченного телефона и метода ECO, моносемичные слова оказались сгруппированы корректно.

Предложенный в данной работе метод Watset при оценке на тезаурусе RuWordNet по критерию F_1-меры уступил только алгоритмам MaxMax и MCL, хотя на этом ресурсе все алгоритмы показали достаточно низкие результаты. При оценке на тезаурусе Yet Another RussNet метод Watset получил максимальные значения как F_1-меры, так и полноты. Построенные синсеты корректно отражают явление полисемии, например {*пустота*^{1}, *бессодержательность*^{1}, *бессмысленность*^{1}, …} и {*вакуум*^{1}, *пустота*^{2}, *ничто*^{1}, …}. На этапе кластеризации значений слов также замечается ранее выявленная в методе MaxMax тенденция к связыванию семантически близких слов, не являющихся синонимами. Это приводит к снижению точности, что особенно заметно при использовании алгоритма MCL для вывода значений слов, генерирующего более крупные по размеру кластеры. Эту проблему можно решить путем предварительной обработки графа синонимов [47].

Важно отметить, что тезаурусы RuWordNet и Yet Another RussNet, использованные в качестве двух золотых стандартов при сравнении методов, имеют различную природу и созданы для решения разных задач. Тезаурус RuWordNet является результатом автоматизированного преобразования в WordNet-подобную структуру тезауруса РуТез для информационного поиска [9]. Тезаурус Yet Another RussNet, в свою очередь, создан в результате эксперимента по комбинированию традиционного лексикографического подхода с привлечением большого количества редакторов-волонтеров при помощи краудсорсинга [30]. Это объясняет разницу в результатах, полученных алгоритмом MaxMax и методом Watset, обусловленную принятыми в этих методах допущениями о структуре исходного графа синонимов. Разница по критерию полноты между результатами работы алгоритма Watset и других алгоритмов по материалам тезауруса RuWordNet статистически значима на основании знакового рангового критерия Уилкоксона [104] на уровне значимости $0{,}01$. Разница по критериям полноты и F_1-меры между результатами работы алгоритма Watset и других алгоритмов по материалам тезауруса Yet Another RussNet статистически значима на основании знакового рангового критерия Уилкоксона на уровне значимости $0{,}01$.

Таким образом, наивысшие оценки по попарным критериям полноты и F_1-меры в результате сравнения алгоритмов получила конфигурация алгоритма Watset, использующая алгоритм испорченного телефона при выводе значений слов (CW_{nolog}) и марковский алгоритм кластеризации при кластеризации графа значений слов (MCL). В табл. 6 эта конфигурация записана как Watset[CW_{nolog}, MCL].

4.2. Оценка метода построения связей

Для экспериментальной оценки метода Watlink, описанного в разделе 2.3, выполняется сравнение значений меры качества на пяти различных наборах данных с парами слов, порожденными асимметричным отношением, до и после использования предложенного метода по материалам золотого стандарта. В качестве меры качества используется точность, полнота и F_1-мера, вычисленные с использованием подхода на основе проверки существования пути в графе (раздел 1.3). Оцениваемый набор данных и золотой стандарт преобразуются в семантическую сеть слов. Оценка качества каждой связи в оцениваемом наборе данных производится путем проверки существования пути от нижестоящего к вышестоящему слову в графе золотого стандарта. Связь считается корректно установленной, если существует путь от нижестоящего значения слова к вышестоящему значению слова в золотом стандарте. Из-за доступности и распространенности, в эксперименте используются только родо-видовые связи. Таким образом, использовано 1 729 090 родо-видовых пар слов из тезауруса RuWordNet [67]. Вычисление мер качества производится на основании наличия или отсутствия пути в золотом стандарте от нижестоящего слова к вышестоящему. Связь между парой слов в оцениваемом ресурсе считается корректно установленной, если в золотом стандарте существует путь от одного значения слова к другому (см. раздел 1.3). Лучшими в данном эксперименте считаются методы, получившие высокие значения полноты и F_1-меры.

4.2.1. Описание эксперимента

В эксперименте используется конфигурация метода Watset, получившая высшие оценки в эксперименте, описанном в разделе 4.1: Watset[CW_{nolog}, MCL]. Данная конфигурация метода использует алгоритм испорченного телефона при выводе значений слов и марковский алгоритм кластеризации при кластеризации графа значений слов. Синсеты, полученные в результате предыдущего эксперимента, используются для построения иерархических контекстов. В эксперименте используются следующие значения гиперпараметров метода Watlink без применения расширения иерархических контекстов:

– количество ближайших соседей при расширении: $n = 0$;
– мера близости иерархических контекстов: $\text{sim}_{\text{hctx}} = \cos$.

Поскольку в данном эксперименте не производится расширение иерархических контекстов, то значения гиперпараметров k, λ и δ не влияют на результаты.

С целью изучения влияния весов слов на результат построения иерархических контекстов, в эксперименте рассматривается три различных подхода к взвешиванию слов в иерархических контекстах:

– tf–idf: используются значение tf–idf в соответствии с формулой (2.9);
– tf: при вычислении tf–idf значение idf принимается за единицу, т. е.

$$\text{tf–idf}(h, S, \mathcal{S}) = \text{tf}(h, S) \times 1;$$

– idf: при вычислении tf–idf значение tf принимается за единицу, т. е.

$$\text{tf–idf}(h, S, \mathcal{S}) = 1 \times \text{idf}(h, \mathcal{S}).$$

Используется пять различных наборов данных с родо-видовыми связями между словами:

– пары слов, извлеченные из электронной библиотеки lib.rus.ec при помощи лексико-синтаксических шаблонов общего назначения [81], всего $1\,597\,651$ пар слов (далее — «Шаблоны»);

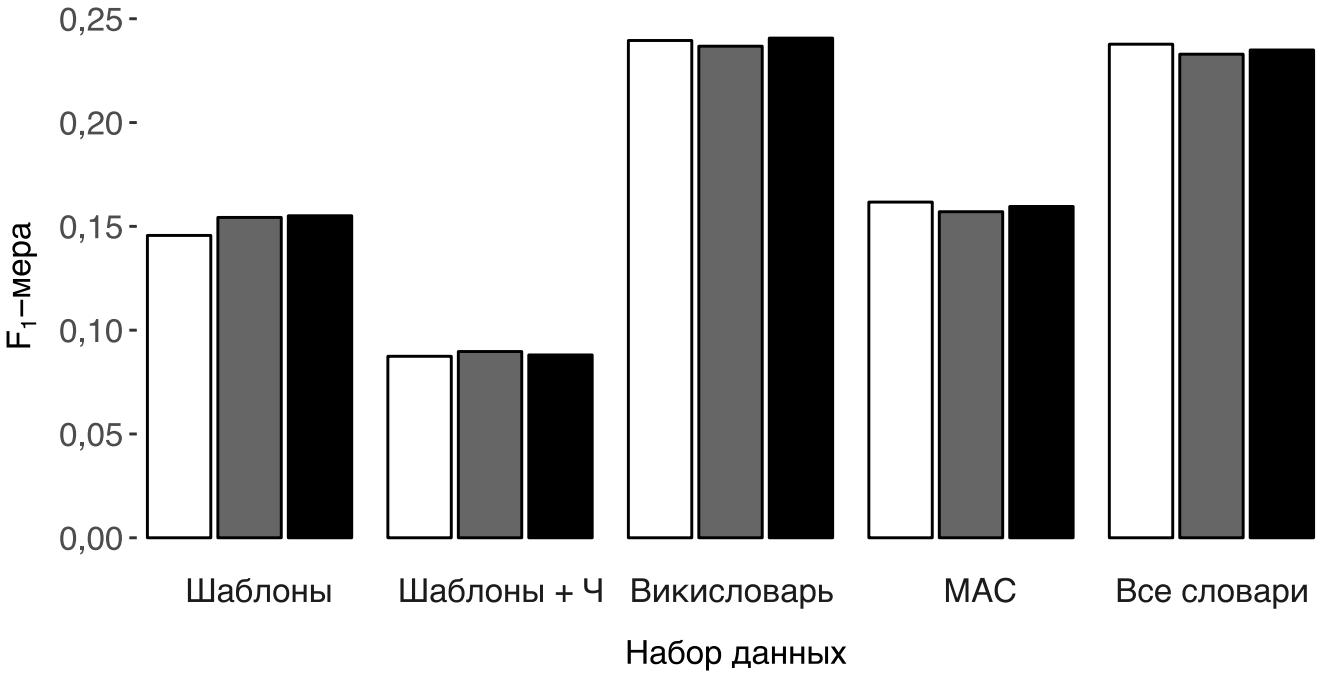
– пары слов из набора данных «Шаблоны», не менее тридцати раз появившиеся в коллекции документов, всего 10 458 пар слов (далее — «Шаблоны + Ч»);
– пары слов из материалов Русского Викисловаря, извлечение данных из которого выполнено при помощи утилиты Wikokit [8], всего 108 895 пар слов (далее — «Викисловарь»);
– пары слов, извлеченные из толкований Малого академического словаря [15] при помощи специализированных лексико-синтаксических шаблонов [5], всего 36 800 пар слов (далее — «МАС»);
– объединение трех ресурсов с удалением пар-дубликатов: «Шаблоны + Ч», «Викисловарь» и «МАС» в единый набор данных из 149 195 пар слов (далее — «Все словари»).

Поскольку словник используемых словарей отличается от словника золотого стандарта, то при вычислении информационно-поисковых оценок использовались только те пары слов, оба слова которых входят в пересечение словника золотого стандарта и объединенного словника наборов данных, полученных в результате выполнения методов построения связей.

4.2.2. Результаты эксперимента

На рис. 4.2 приведены результаты сравнения подходов к взвешиванию слов в иерархических контекстах. Подходы tf и idf по-отдельности показывают нестабильные результаты на различных наборах данных. Таким образом, анализ результатов будет осуществляться только на результатах взвешивания иерархических контекстов при помощи общепринятого подхода tf–idf.

Результаты экспериментальной оценки метода построения связей представлены в табл. 7 и на рис. 4.3. В качестве меры качества использованы информационно-поисковые критерии качества, вычисленные по материалам золотого стандарта — тезауруса RuWordNet. При использовании метода Watlink



Обозначения: □ tf, ■ idf, ■ tf–idf.

Рис. 4.2 — Влияние подхода к взвешиванию иерархических контекстов на результаты работы алгоритма построения связей

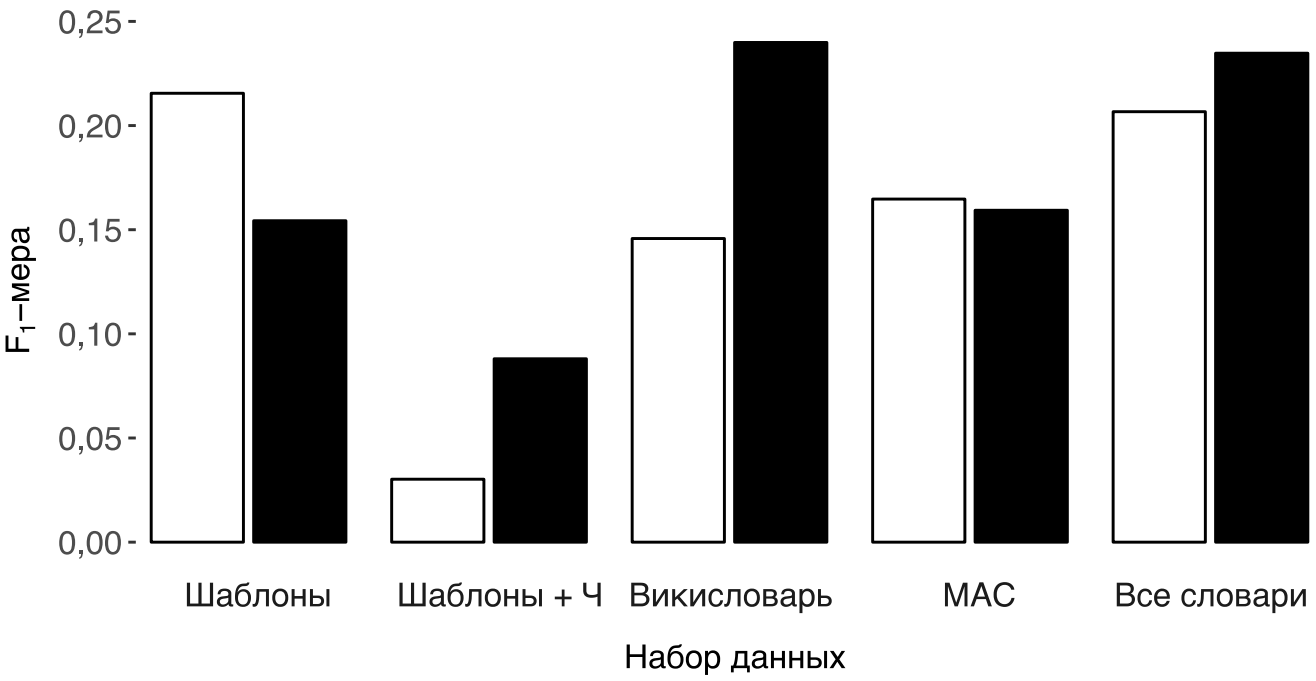
применяется обозначение «+ CCC» (семантическая сеть слов); исходные наборы данных приведены без этого обозначения. Три лучших результата по каждому критерию выделены полужирным начертанием; лучший результат дополнительно обозначен подчеркиванием.

4.2.3. Анализ результатов

Метод Watlink, использующий синсеты, полученные при помощи метода Watset[CW_{nolog}, MCL], демонстрирует лучшие значения по критерию полноты и F_1-меры, статистически значимые на основании критерия Уилкоксона для связных выборок на уровне значимости $0{,}01$ на всех наборах данных, кроме наборов данных «Шаблоны» и «MAC». На наборах данных «Шаблоны» и «MAC» значение F_1-меры значительно уменьшается за счет существенного снижения точности.

Таблица 7 — Сравнение методов построения связей без расширения по материалам RuWordNet

Метод	**# связей**	**Точность**	**Полнота**	**F_1-мера**
Шаблоны	1 597 651	0,1611	**0,3255**	**0,2155**
Шаблоны + ССС	236 922	0,1126	0,2451	0,1543
Шаблоны + Ч	10 458	**0,3773**	0,0157	0,0302
Шаблоны + Ч + ССС	46 758	0,1140	0,0717	0,0880
Викисловарь	108 985	**0,3877**	0,0898	0,1458
Викисловарь + ССС	177 787	0,1836	**0,3460**	**0,2399**
МАС	36 800	**0,1823**	0,1502	0,1647
МАС + ССС	98 085	0,1383	0,1879	0,1593
Все словари	149 195	0,1719	0,2590	0,2067
Все словари + ССС	216 285	0,1685	**0,3865**	**0,2347**



Обозначения: □ исходный набор данных, ■ семантическая сеть слов.

Рис. 4.3 — Сравнение методов построения связей без расширения по материалам RuWordNet по критерию F_1-меры

При запуске на исходном наборе данных «Шаблоны», метод Watlink значительно усилил зашумленные связи, присутствующие в этом наборе данных. Например, в наборе данных «Шаблоны» присутствуют такие связи, как (*ящик*, *сервис*) и (*ячмень*, *земля*). При нехватке других вышестоящих слов для синсетов, некорректно выделенные слова включаются в иерархические контексты, что приводит к распространению некорректных вышестоящих слов на все слова в синсете. Это подтверждается тем, что запуск на наборе данных с частотным фильтром «Шаблоны + Ч» показал значительное увеличение полноты и F_1-меры за счет снижения точности.

Набор данных «МАС», в свою очередь, обладает достаточно ограниченным словником по сравнению с другими наборами данных в эксперименте. Представленные в этом словаре связи основаны на разборе толкований при помощи специализированных лексико-синтаксических шаблонов, причем полнота этих связей достаточно высока. Запуск метода Watlink на этом наборе данных привел к появлению большого количества избыточных связей, которые повышали полноту, но заметно снижали точность. Это приводило к тому, что оценка по критерию F_1-меры снизилась по сравнению с исходным набором данных. Такой результат обусловлен различиями в инвентарях значений слов между множеством значений слов, представленных в Малом академическом словаре и множеством значений слов, записанных в синсетах. Различия приводят к тому, что метод Watlink порождает ложные положительные ответы, в которых вышестоящее слово не соответствует нижестоящему. Например, из-за синонимов корректная пара слов (*табло*, *щит*) привела к порождению некорректной пары слов (*лицо*, *щит*).

Наибольшее значение точности показал набор данных на основе Викисловаря. Это подтверждает высокий потенциал применения краудсорсинга для построения лексических ресурсов [99]. Кроме того, высокое значение точности показал метод на основе лексико-синтаксических шаблонов, запущенный на материалах большой коллекции документов («Шаблоны + Ч»). При частотной фильтрации из набора данных исключались все родо-видовые пары, которые были обнаружены в коллекции документов менее тридцати раз. Общая проблема

всех методов, основанных на лексико-синтаксических шаблонах или краудсорсинге, состоит в низкой полноте. Несмотря на большое количество корректных ответов, их общее количество достаточно невелико. На это указывает низкое значение полноты, полученное на исходных наборах данных «Шаблоны + Ч» и «Викисловарь».

Таким образом, использование метода Watlink позволяет значительно повысить полноту построения связей. Это достигается за счет того, что общие вышестоящие слова по отношению к близким по значению словам распространяются также и на их синонимы, что существенно увеличивает количество верных положительных срабатываний, обеспечивая существенное повышение полноты за счет неизбежного снижения точности.

Важно отметить, что в данном эксперименте не производилась оценка результатов работы метода с применением расширения иерахических контекстов. Это требует наличия матрицы линейного преобразования, значения элементов которой необходимо подобрать на основе оптимальных значений гиперпараметров.

4.3. Оценка метода подбора матрицы линейного преобразования

Для экспериментальной оценки метода расширения связей производится сравнение базового метода подбора матрицы линейного преобразования [45] с его стабилизированным вариантом, представленном в разделе 2.3.3. В качестве меры качества используется мера hit@k [44]. В качестве золотого стандарта для оценки метода подбора матрицы линейного преобразования использованы материалы русского Викисловаря [8]. Таким образом, для каждого нижестоящего слова метод генерирует k ответов, соответствующих вышестоящим словам. Если множество ответов содержит корректный ответ, то данное нижестоящее слово считается корректно обработанным. Лучшим в данном эксперименте считается метод, получивший высокие значения hit@k.

4.3.1. Описание эксперимента

В соответствии с общепринятой методологией машинного обучения с учителем, эксперименты проводятся на трех выборках: обучающей, проверочной и тестовой. С целью предотвращения эффекта лексического переобучения (англ. *lexical overfitting*), смещающего меру качества вверх [65], разбиение исходного набора данных на основе Русского Викисловаря на три выборки произведено таким образом, что ни в одной из получившихся выборок нет нижестоящего слова с общим вышестоящим словом. С целью расширения обучающей выборки, без нарушения данного принципа, добавлены пары слов из набора данных «Шаблоны + Ч», описанные в разделе 4.2. Таким образом, используются выборки следующих размеров:

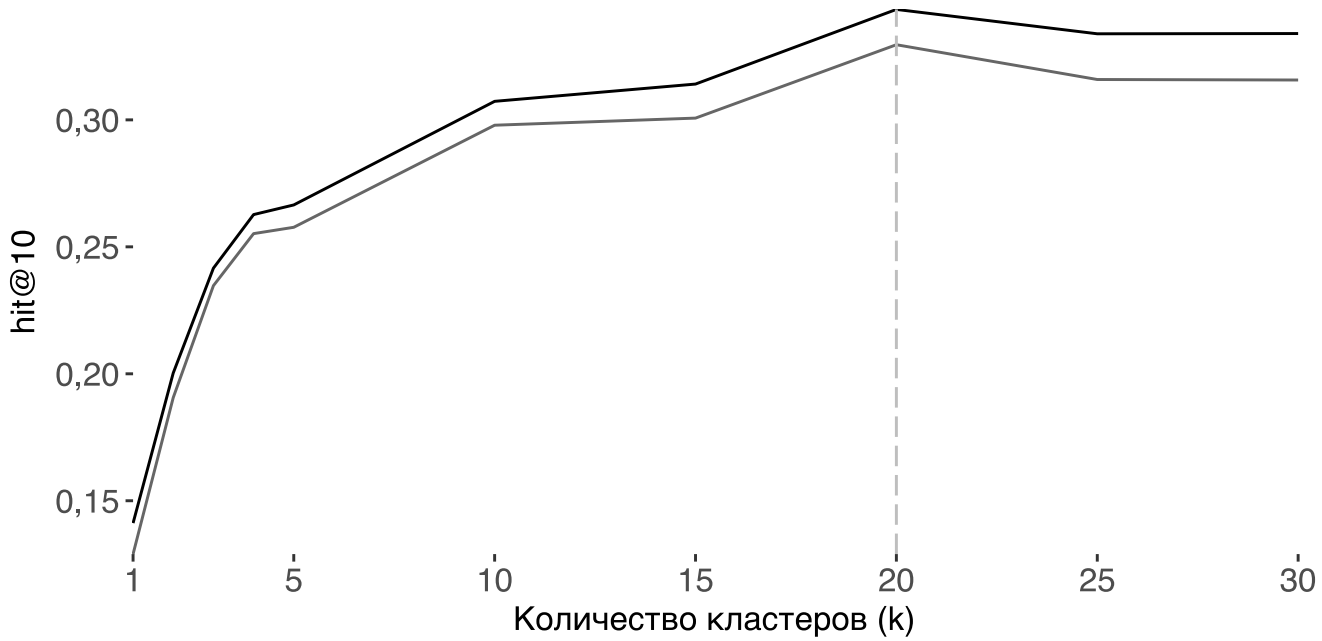
– обучающая выборка: $25\,067$ пар слов;
– проверочная выборка: $8\,192$ пар слов;
– тестовая выборка: $8\,310$ пар слов.

На проверочной выборке осуществляется подбор гиперпараметров, которые используются для сравнения результатов работы базового и стабилизированного метода на тестовой выборке:

– количество кластеров: $1 \leqslant k \leqslant 30$;
– влияние стабилизатора: $\lambda \in \{10^l : l \in \{-1, 0, 1, 2\}\}$.

4.3.2. Результаты эксперимента

Подбор гиперпараметров по проверочной выборке позволил определить их оптимальные значения в указанных интервалах. На рис. 4.4 представлена зависимость величины hit@10 от количества кластеров k. Как в случае единственного кластера $k = 1$, так и в случае обнаруженного оптимального значения $k = 20$, найдено оптимальное значение $\lambda = 1$.



Обозначения: базовый метод, стабилизированный метод.

Рис. 4.4 — Выбор оптимального количества кластеров по проверочной выборке на основании критерия hit@10

Результаты сравнения методов подбора матрицы преобразования по тестовой выборке с использованием меры качества hit@1, hit@5 и hit@10 представлены в табл. 8; лучшие значения выделены полужирным начертанием. Значения критериев hit@1 и hit@5 приведены в иллюстративных целях. Выбор лучшего метода производится на основании значения hit@10.

Таблица 8 — Сравнение методов подбора матрицы линейного преобразования по тестовой выборке

Метод	k	hit@1	hit@5	hit@10
Базовый	1	0,0473	0,1095	0,1297
Стабилизированный	1	**0,0522**	**0,1199**	**0,1403**
Базовый	20	0,2090	0,3031	0,3232
Стабилизированный	20	**0,2119**	**0,3120**	**0,3343**

4.3.3. Анализ результатов

На основании результатов сравнения стабилизированного метода подбора матрицы линейного преобразования с его базовой версией видно, что стабилизированный метод показывает лучшие результаты как в случае единственного кластера ($k = 1$), так и в случае оптимального количества кластеров ($k = 20$). Производилось три запуска с различным зерном генератора случайных чисел. В обоих случаях разница статистически значима на основании t-критерия равенства средних для независимых выборок на уровне значимости $0{,}05$ [102]. Это позволяет сделать вывод о том, что разница в результатах не зависит от удачной случайной совокупности начальных значений элементов матрицы линейного преобразования.

Следовательно, при расширении иерархических контекстов целесообразно использовать матрицу линейного преобразования, элементы которой подобраны на основе использованных данных при помощи стабилизированного метода с гиперпараметрами $k = 20$ и $\lambda = 1$.

4.4. Оценка метода построения связей с расширением

Для экспериментальной оценки метода Watlink, использующего расширение иерархических контекстов, применяется тот же подход, что и в эксперименте по построению связей без их расширения (раздел 4.2). Лучшими в данном эксперименте считаются методы, получившие высокие значения полноты и F_1-меры.

4.4.1. Описание эксперимента

В данном эксперименте используются иерархические контексты, полученные с использованием подхода к взвешиванию tf–idf на основе синсетов, полученных при помощи метода Watset[$\mathrm{CW}_{\mathrm{nolog}}$, MCL] (раздел 4.1). При расширении используется семейство матриц линейного преобразования, полученные в рамках эксперимента в разделе 4.3. Кроме того, исследуются следующие значения гиперпараметров метода Watlink:

– количество ближайших соседей при расширении: $n = 10$;
– количество кластеров при подборе матрицы линейного преобразования: $k = 20$ (оптимальное значение, полученное в разделе 4.3);
– влияние стабилизации на функцию потерь при подборе матрицы линейного преобразования: $\lambda = 1$ (оптимальное значение, полученное в разделе 4.3);
– максимальное расстояние до ближайшего соседа: $\delta = \{\frac{r}{10} : r \in \mathbb{N}, r \leqslant 10\}$;
– мера близости иерархических контекстов: $\mathrm{sim}_{\mathrm{hctx}} = \cos$.

Поскольку набор данных «Шаблоны» содержит большое количество некорректно определенных семантических связей, как показано в разделе 4.2, то этот набор данных был исключен из эксперимента. Вместо него использован только производный набор данных с частотной фильтрацией «Шаблоны + Ч».

4.4.2. Результаты эксперимента

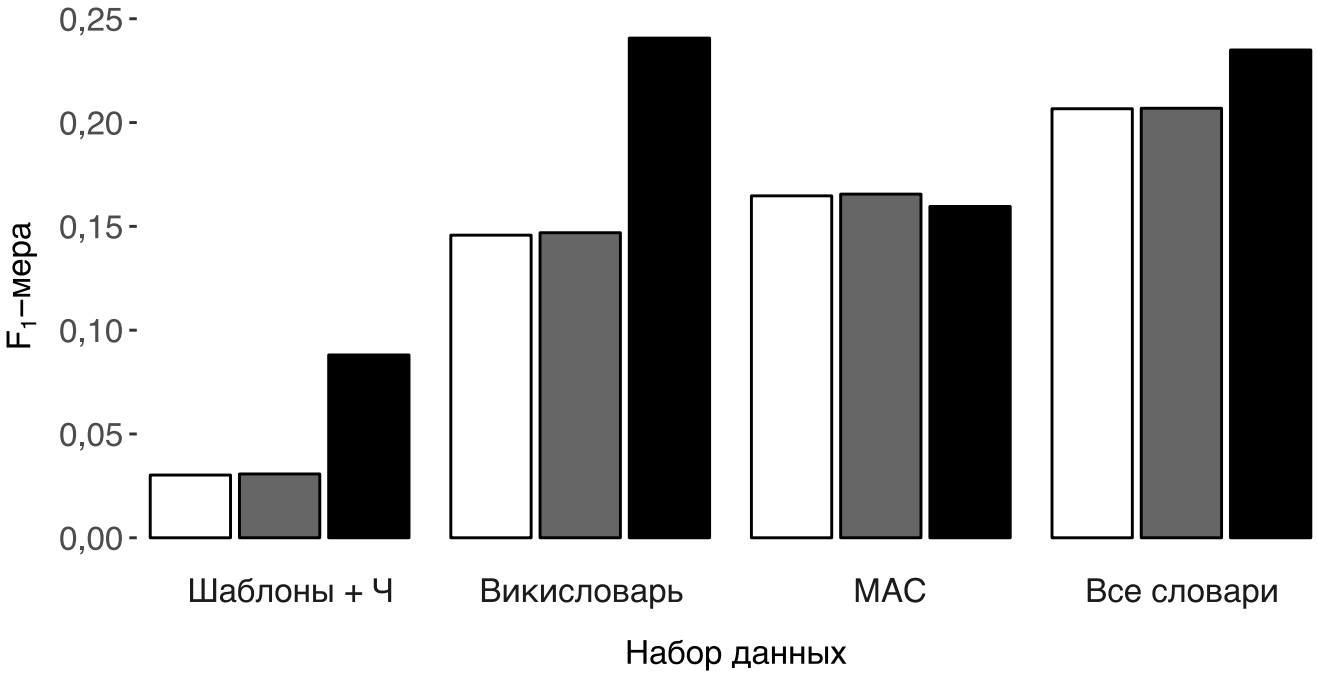
Подбор значений гиперпараметра δ осуществлялся путем независимого запуска алгоритма Watlink в различных конфигурациях на материалах одного и того

же золотого стандарта RuWordNet, представляющего 1 729 090 родо-видовых связей. Метод Watlink продемонстрировал лучшие результаты при значении $\delta = 0{,}6$. Результаты эксперимента далее будут приведены на основании этого значения.

Результаты экспериментальной оценки метода построения связей с расширением иерархических контекстов представлены в табл. 9 и на рис. 4.5. В качестве меры качества использованы информационно-поисковые критерии качества, вычисленные по материалам золотого стандарта RuWordNet, согласно подходу, использованному в разделе 4.2. При использовании расширения семантических связей без построения иерархических контекстов применяется обозначение «+ РЛП» (расширение линейным преобразованием). В этом эксперименте метод Watlink запускался на наборах данных, к которым заранее применялась операция расширения. При использовании метода Watlink применяется обозначение «+ РЛП + ССС» (семантическая сеть слов). Три лучших результата по каждому критерию выделены полужирным начертанием; лучший результат дополнительно обозначен подчеркиванием.

Таблица 9 — Сравнение методов построения связей с расширением по материалам RuWordNet

Метод	**# связей**	**Точность**	**Полнота**	**F_1-мера**
Шаблоны + Ч + РЛП	10 715	**0,3760**	0,0160	0,0307
Шаблоны + Ч + РЛП + ССС	47 387	0,1129	0,0722	0,0881
Викисловарь + РЛП	110 329	**0,3874**	0,0907	0,1469
Викисловарь + РЛП + ССС	179 623	0,1844	**0,3464**	**0,2407**
МАС + РЛП	37 702	**0,1825**	0,1515	0,1655
МАС + РЛП + ССС	99 678	0,1385	0,1883	0,1596
Все словари + РЛП	151 150	0,1720	**0,2594**	**0,2069**
Все словари + РЛП + ССС	218 290	0,1687	**0,3867**	**0,2350**



Обозначения: □ исходный набор данных, ▒ исходный набор данных с расширением, ■ семантическая сеть слов с расширением.

Рис. 4.5 — Сравнение методов построения связей с расширением по материалам RuWordNet по критерию F_1-меры

4.4.3. Анализ результатов

По результатам эксперимента, лучшее значение F_1-меры продемонстрировал набор данных на основе Викисловаря, обработанный методом Watlink с использованием синсетов Watset[CW_{nolog}, MCL]. При сравнении с результатами эксперимента «Викисловарь + ССС», не использующего расширение (табл. 7), обнаружено статистически значимое увеличение точности и F_1-меры на основании критерия Уилкоксона для связных выборок на уровне значимости $0{,}01$. При этом наблюдается отсутствие статистической значимости при сравнении полноты.

Во всех экспериментах по расширению линейным преобразованием без построения семантической сети слов («+ РЛП» без «+ ССС») обнаружено значимое увеличение полноты при одновременном значимом снижении точности на основании критерия Уилкоксона для связных выборок на уровне значимости $0{,}01$. Тем не менее, общая оценка по критерию F_1-меры увеличивается на всех наборах

данных, использующих семантическую есть слов, по сравнению с исходными наборами данных. Использование расширения линейным преобразованием позволяет получить из тех же самых данных от 257 до 1955 семантических связей, на наборах данных «Шаблоны + Ч» и «Все словари», соответственно.

Сочетание расширения линейным преобразованием с построением семантической сети слов показало статистически значимое увеличение полноты на всех наборах данных, кроме объединения словарей с построения семантической сети слов без расширения. Такой результат объясняется достаточно большим количеством родо-видовых связей высокого качества, присутствующих в исходных лексико-семантических ресурсах.

Результаты экспериментов позволяют сделать вывод о высокой эффективности предложенного метода построения и расширения связей.

4.5. Выводы по главе 4

В главе 4 представлены результаты вычислительных экспериментов, выполненных на основе подхода сопоставления с золотым стандартом.

Предложенный в данной работе метод Watset продемонстрировал лучшие значения полноты и F_1-меры по результатами эксперимента на основе сопоставления с материалами золотых стандартов RuWordNet [67] и Yet Another RussNet [30]. По результатам эксперимента, лучшей конфигурацией метода Watset оказалась конфигурация, использующая алгоритм испорченного телефона при выводе значений слов и марковский алгоритм кластеризации для кластеризации графа значений слов.

Предложенный в данной работе метод Watlink продемонстрировал лучшие значения полноты и F_1-меры по результатам эксперимента на основе сопоставления с материалами золотым стандартом RuWordNet [67]. Предложенный в данной работе стабилизированный метод подбора матрицы линейного преобразования

показал лучшие значения меры качества hit@10 на тестовой выборке, построенной на основе материалов Викисловаря [8]. По результатам эксперимента, лучшей конфигурацией метода Watlink оказалась конфигурация, использующая расширение иерархических контекстов на основе стабилизированного метода подбора матрицы линейного преобразования.

Результаты экспериментов показывают и подтверждают, что предложенные в данной работе методы, модели и алгоритмы позволяют эффективно строить семантическую сеть слов.

Заключение

В диссертационной работе были рассмотрены вопросы разработки и исследования эффективных методов автоматического построения семантической сети. Исследованы современные подходы к автоматическому построению семантических ресурсов. Предложена модель семантической сети слов, связывающая лексические значения слов при помощи семантических связей с разрешенной многозначностью. На ее основе разработаны методы и алгоритмы автоматического построения понятий и автоматического построения и расширения семантичеких связей. Корректность предложенных методов подтверждается результатами экспериментов. Разработанные модели, методы и алгоритмы реализованы в виде комплекса программ, который функционирует на многоядерных и многопроцессорных вычислительных системах для выполнения ресурсоемких операций.

Основные результаты, полученные в ходе выполнения диссертационного исследования являются новыми и не покрываются ранее опубликованными научными работами других авторов, обзор которых был дан в разделе 1.4. Следует отметить основные отличия.

Существующие методы постреления синсетов на основе нечеткой кластеризации графа, такие как MaxMax [55], CPM [80] и ECO [49], не осуществляют процедуру вывода значений слов в явном виде и ориентированы на кластеризацию графов совместной встречаемости слов. Методы вывода значений слов [26, 36, 83], в свою очередь, не производят разрешения многозначности полученных значений слов и не используют эти значения слов для построения понятий. Существующий метод разрешения многозначности в контекстах [38] не предполагает построения графа значений слов. Описанный в разделе 2.2 метод обнаружения понятий отличается тем, что использует существующий метод вывода значений слов, затем строит граф значений слов с использованием значений слов с разрешенной многозначностью, после чего производит жесткую кластеризацию

полученного графа значений слов при помощи хорошо известных методов жесткой кластеризации графа [26, 35].

Существующие методы построения связей, такие как онтологизация [87], ECO [49] и BabelNet [76] предполагают построение связей между синсетами на основе заранее подготовленной семантической иерархии высокого качества. В обоих случаях используется тезаурус английского языка WordNet [40]. Описанный в разделе 2.3 метод построения связей не требует такого ресурса для решения задачи. Методы извлечения связей, в первую очередь, шаблоны Херст [53,81] и их вариации для толковых словарей [5] не указывают конкретные значения слов, что приводит к возникновению лексической многозначности. Таким же ограничением обладает Викисловарь и другие общедоступные ресурсы [8], построенные при помощи краудсорсинга. Предложенный в данной работе метод построения связей предназначен позволяет указать конкретные значения связанных слов. Кроме того, подход к расширению иерархических контекстов в данном методе позволяет добавить дополнительные связи, подходящие по смыслу.

Методы подбора матрицы линейного преобразования для поиска вышестоящих слов на основе векторных представлений нижестоящих слов стали разрабатываться относительно недавно. Базовый метод подбора матрицы линейного преобразования [45] не учитывает явным образом асимметричность семантических связей. Описанный в разделе 2.3.3 метод стабилизации функции потерь, используемой для подбора элементов матрицы линейного преобразования, позволяет включить в модель дополнительную информацию о связях слов в виде отрицательных примеров. Другие существующие методы извлечения связей на основе векторных представлений слов [95] требуют трудозатратную операцию полного синтаксического разбора текста.

Можно выделить следующие направления дальнейших исследований:

– использование векторных представлений отдельных лексических значений слов для построения синсетов и связей;
– применение краудсорсинга для пополнения и расширения исходных данных;

– обобщение предложенных методов на другие классы семантических связей;
– формирование связей между отдельными понятиями на основе семантической сети слов;
– разработка интегральной меры качества семантических сетей.

Автор выражает благодарность Н. В. Арефьеву, К. Биманну, П. И. Браславскому, М. Л. Гольдштейну, Д. Г. Ермакову, Д. И. Игнатову, Ю. А. Киселеву, А. А. Крижановскому, Т. М. Ландо, М. Ю. Мухину, А. И. Панченко, Н. В. Пискуновой, В. И. Роговичу, В. А. Соколову, Г. Б. Смирнову, М. А. Черноскутову. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-37-00354 мол_а и при финансовой поддержке РГНФ в рамках научных проектов № 13-04-12020 «Новый открытый электронный тезаурус русского языка» и № 16-04-12019 «Интеграция тезаурусов RussNet и YARN». Поддержка данного проекта осуществлена в рамках благотворительной деятельности, на средства, предоставленные Фондом Михаила Прохорова. Работа выполнена при финансовой поддержке стипендии Президента Российской Федерации молодым ученым и аспирантам № СП-773.2015.5. Автор благодарит компанию Microsoft Research за предоставленные вычислительные ресурсы в облачной среде Microsoft Azure в рамках программы Azure for Research. Автор также выражает благодарность Германской службе академических обменов (Deutscher Akademischer Austauschdienst, DAAD) за поддержку данного исследования.

Литература

1. *Абрамов Н.* Словарь русских синонимов и сходных по смыслу выражений. 7-е изд., стереотип. М.: Русские словари, 1999. 528 с.

2. *Азарова И. В., Митрофанова О. А., Синопальникова А. А.* Компьютерный тезаурус русского языка типа WordNet // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог-2003» (11–16 июня 2003 г., Протвино). М.: 2003. С. 43–50.

3. *Болотникова Е. С., Гаврилова Т. А., Горовой В. А.* Об одном методе оценки онтологий // *Известия Российской академии наук. Теория и системы управления*. 2011. № 3. С. 98–110.

4. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. СПб: Питер, 2000. 384 с.

5. *Киселев Ю. А., Поршнев С. В., Мухин М. Ю.* Метод извлечения родовидовых отношений между существительными из определений толковых словарей // *Программная инженерия*. 2015. № 10. С. 38–48.

6. *Киселев Ю. А., Поршнев С. В., Мухин М. Ю.* Современное состояние электронных тезаурусов русского языка: качество, полнота и доступность // *Программная инженерия*. 2015. № 6. С. 34–40.

7. *Константинова Н. С., Митрофанова О. А.* Онтологии как системы хранения знаний [Электронный ресурс] // *Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы»*. 2008. 54 с. URL: http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf (дата обращения: 20.05.2017).

8. *Крижановский А. А., Смирнов А. В.* Подход к автоматизированному построению общецелевой лексической онтологии на основе данных Викисловаря //

Известия Российской академии наук. Теория и системы управления. 2013. № 2. С. 53–63.

9. *Лукашевич Н. В.* Тезаурусы в задачах информационного поиска. М.: Изд-во Московского университета, 2011. 512 с.

10. *Ляшевская О. Н., Шаров С. А.* Частотный словарь современного русского языка (на материалах Национального корпуса русского языка). М.: Азбуковник, 2009. 1112 с.

11. *Маннинг К. Д., Рагхаван П., Шютце Х.* Введение в информационный поиск. : Пер. с англ. / Под ред. П. И. Браславского, Д. А. Клюшина, И. В. Сегаловича. М.: ООО «И.Д. Вильямс», 2011. 528 с.

12. *Мельчук И. А.* Опыт теории лингвистических моделей «Смысл ⇔ Текст». 2-е изд. М.: Яз. рус. культуры, 1999. 368 с.

13. *Падучева Е. В.* Динамические модели в семантике лексики. М.: Языки славянской культуры, 2004. 609 с.

14. Прикладная и компьютерная лингвистика / Под ред. И. С. Николаева, О. В. Митрениной, Т. М. Ландо. М.: URSS, 2016. 320 с.

15. Словарь русского языка: В 4-х т. / РАН, Ин-т лингвистич. исследований; Под ред. А. П. Евгеньевой. 4-е изд., стер. М.: Рус. яз.; Полиграфресурсы, 1999.

16. *Abadi M. et al.* TensorFlow: A System for Large-Scale Machine Learning // 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), November 2–4, 2016, Savannah, GA, USA. Berkeley, CA, USA: USENIX Association, 2016. P. 265–283.

17. *Allan K.* Concise Encyclopedia of Semantics. Oxford, UK: Elsevier Science, 2009. 1104 pp.

18. *Arefyev N. V., Panchenko A. I., Lukanin A. V. et al.* Evaluating Three Corpus-based Semantic Similarity Systems for Russian // Computational Linguistics and

Intellectual Technologies: Papers from the Annual conference “Dialogue”. Volume 2 of 2. Papers from special sessions, May 27–30, 2015, Moscow, Russia. Moscow, Russia: RGGU, 2015. P. 106–119.

19. *van Assem M., Malaisé V., Miles A., Schreiber G.* A Method to Convert Thesauri to SKOS // 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11–14, 2006 Proceedings. Berlin, Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2006. P. 95–109.

20. *Bagga A., Baldwin B.* Algorithms for Scoring Coreference Chains // Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC), May 26, 1998, Granada, Spain. 1998. P. 563–566.

21. *Balkova V., Sukhonogov A., Yablonsky S.* Russian WordNet // Proceedings of the Second International WordNet Conference—GWC2004, January 20–23, 2004, Brno, Czech Republic. Brno, Czech Republic: Masaryk University Brno, Czech Republic, 2004. P. 31–38.

22. *Bartunov S., Kondrashkin D., Osokin A., Vetrov D. P.* Breaking Sticks and Ambiguities with Adaptive Skip-gram // *Journal of Machine Learning Research*. 2016. Vol. 51. P. 130–138.

23. *Beckett D.* The Design and Implementation of the Redland RDF Application Framework // *Computer Networks*. 2002. Vol. 39, no. 5. P. 577–588.

24. *Berners-Lee T., Hendler J., Lassila O.* The Semantic Web // *Scientific American*. 2001. Vol. 284, no. 5. P. 28–37.

25. *Biemann C.* Ontology Learning from Text: A Survey of Methods // *GLDV-Journal for Computational Linguistics and Language Technology*. 2005. Vol. 20, no. 2. P. 75–93.

26. *Biemann C.* Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems // Proceedings of the

First Workshop on Graph Based Methods for Natural Language Processing (TextGraphs-1), June 9, 2006, New York, NY, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. P. 73–80.

27. *Biemann C.* Creating a system for lexical substitutions from scratch using crowdsourcing // *Language Resources and Evaluation*. 2013. Vol. 47, no. 1. P. 97–122.

28. *Bomze I. M., Budinich M., Pardalos P. M., Pelillo M.* The maximum clique problem // Handbook of Combinatorial Optimization. Springer, 1999. P. 1–74.

29. *Bordea G., Lefever E., Buitelaar P.* SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2) // Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), June 16–17, 2016, San Diego, CA, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016. P. 1081–1091.

30. *Braslavski P., Ustalov D., Mukhin M., Kiselev Y.* YARN: Spinning-in-Progress // Proceedings of the 8th Global WordNet Conference (GWC2016), January 27–30, 2016, Bucharest, Romania. Global WordNet Association, 2016. P. 58–65.

31. *Collins A. M., Quillian M. R.* Retrieval time from semantic memory // *Journal of Verbal Learning and Verbal Behavior*. 1969. Vol. 8, no. 2. P. 240–247.

32. *Deliyanni A., Kowalski R. A.* Logic and Semantic Networks // *Communications of the ACM*. 1979. Vol. 22, no. 3. P. 184–192.

33. *Deng J., Dong W., Socher R. et al.* ImageNet: A Large-Scale Hierarchical Image Database // IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), June 20–25, 2009, Miami, FL, USA. IEEE, 2009. P. 248–255.

34. *Dikonov V. G.* Development of lexical basis for the Universal Dictionary of UNL Concepts // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”, May 29 – June 2, 2013, Bekasovo. Moscow, Russia: RGGU, 2013. P. 212–221.

35. *van Dongen S.* Graph Clustering Via a Discrete Uncoupling Process // *SIAM Journal on Matrix Analysis and Applications*. 2008. Vol. 30, no. 1. P. 121–141.

36. *Dorow B., Widdows D.* Discovering Corpus-Specific Word Senses // 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003), April 12–17, 2003, Budapest, Hungary. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003. P. 79–82.

37. *Drymonas E., Zervanou K., Petrakis E. G. M.* Unsupervised Ontology Acquisition from Plain Texts: The OntoGain System // Proceedings of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB 2010), June 23–25, 2010, Cardiff, Wales, UK. Berlin, Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2010. P. 277–287.

38. *Faralli S., Panchenko A., Biemann C., Ponzetto S. P.* Linked Disambiguated Distributional Semantic Networks // The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II. Cham, Germany: Springer International Publishing, 2016. P. 56–64.

39. *Farhadi A., Hejrati M., Sadeghi M. A. et al.* Every Picture Tells a Story: Generating Sentences from Images // 11th European Conference on Computer Vision (ECCV 2010), Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV. Berlin, Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2010. P. 15–29.

40. *Fellbaum C.* WordNet: An Electronic Database. MIT Press, 1998. 449 pp.

41. *Ferrucci D., Brown E., Chu-Carroll J. et al.* Building Watson: An Overview of the DeepQA Project // *AI Magazine*. 2010. Vol. 31, no. 3. P. 59–79.

42. *Fowlkes E. B., Mallows C. L.* A Method for Comparing Two Hierarchical Clusterings // *Journal of the American Statistical Association*. 1983. Vol. 78, no. 383. P. 553–569.

43. *Freeman L. C.* Centered graphs and the structure of ego networks // *Mathematical Social Sciences*. 1982. Vol. 3, no. 3. P. 291–304.

44. *Frome A., Corrado G. S., Shlens J. et al.* DeViSE: A Deep Visual-Semantic Embedding Model // Advances in Neural Information Processing Systems 26 (NIPS 2013), December 5–10, 2013, Harrah and Harveys, NV, USA. Curran Associates, Inc., 2013. P. 2121–2129.

45. *Fu R., Guo J., Qin B. et al.* Learning Semantic Hierarchies via Word Embeddings // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014) (Volume 1: Long Papers), June 22–27, 2014, Baltimore, MD, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. P. 1199–1209.

46. *Gábor K., Zargayouna H., Tellier I. et al.* Exploring Vector Spaces for Semantic Relations // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), September 9–11, 2017, Copenhagen, Denmark. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. P. 1815–1824.

47. *Gfeller D., Chappelier J.-C., De Los Rios P.* Synonym Dictionary Improvement through Markov Clustering and Clustering Stability // Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA'05), May 17–20, 2005, Brest, France. 2005. P. 106–113.

48. *Gonçalo Oliveira H., Gomes P.* Onto.PT: Automatic Construction of a Lexical Ontology for Portuguese // Proceedings of the 2010 Conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers' Symposium, August 16–20, 2010, Lisbon, Portugal. Amsterdam, The Netherlands: IOS Press, 2010. P. 199–211.

49. *Gonçalo Oliveira H., Gomes P.* ECO and Onto.PT: a flexible approach for creating a Portuguese wordnet automatically // *Language Resources and Evaluation*. 2014. Vol. 48, no. 2. P. 373–393.

50. *Gurevych I., Eckle-Kohler J., Hartmann S. et al.* UBY — A Large-Scale Unified Lexical-Semantic Resource Based on LMF // Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), April 23–27, 2012, Avignon, France. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. P. 580–590.

51. *Hagberg A. A., Schult D. A., Swart P. J.* Exploring Network Structure, Dynamics, and Function using NetworkX // Proceedings of the 7th Python in Science Conference (SciPy2008), August 19–24, 2008, Pasadena, CA, USA. 2008. P. 11–15.

52. *Hartigan J. A., Wong M. A.* Algorithm AS 136: A K-Means Clustering Algorithm // *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. 1979. Vol. 28, no. 1. P. 100–108.

53. *Hearst M. A.* Automatic Acquisition of Hyponyms from Large Text Corpora // Proceedings of the 14th Conference on Computational Linguistics (COLING '92) - Volume 2, August 23–28, 1992, Nantes, France. COLING '92. International Committee on Computational Linguistics, 1992. P. 539–545.

54. *Herrmann D. J.* An old problem for the new psychosemantics: Synonymity // *Psychological Bulletin*. 1978. Vol. 85, no. 3. P. 490–512.

55. *Hope D., Keller B.* MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction // Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24–30, 2013, Proceedings, Part I. Berlin, Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2013. P. 368–381.

56. *Hutchins J.* ALPAC: The (In)Famous Report // *Readings in machine translation*. 2003. Vol. 14. P. 131–135.

57. *Jurgens D., Klapaftis I.* SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses // Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), June 14–15, 2013, Atlanta, GA,

USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2013. P. 290–299.

58. *Kamps J., Marx M., Mokken R. J., de Rijke M.* Using WordNet to Measure Semantic Orientations of Adjectives // Fourth International Conference on Language Resources and Evaluation (LREC 2004), May 26–28, 2004, Lisbon, Portugal. European Language Resources Association (ELRA), 2004. P. 1115–1118.

59. *Kawahara D., Peterson D. W., Palmer M.* A Step-wise Usage-based Method for Inducing Polysemy-aware Verb Classes // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014) (Volume 1: Long Papers), June 22–27, 2014, Baltimore, MD, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. P. 1030–1040.

60. *Kittur A., Chi E. H., Suh B.* Crowdsourcing User Studies with Mechanical Turk // Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’08), April 5–10, 2008, Florence, Italy. New York, NY, USA: ACM, 2008. P. 453–456.

61. *Kittur A., Kraut R. E.* Harnessing the Wisdom of Crowds in Wikipedia: Quality Through Coordination // Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW ’08), November 8–12, 2008, San Diego, CA, USA. New York, NY, USA: ACM, 2008. P. 37–46.

62. *Kuhn M., Johnson K.* Applied Predictive Modeling. 2013th edition. New York, NY, USA: Springer-Verlag New York, 2013. 600 pp.

63. *Lassila O., McGuinness D.* The Role of Frame-Based Representation on the Semantic Web // *Linköping Electronic Articles in Computer and Information Science*. 2001. Vol. 6, no. 005.

64. *Lenat D. B., Guha R. V.* Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. 1st edition. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990. 391 pp.

65. *Levy O., Remus S., Biemann C., Dagan I.* Do Supervised Distributional Methods Really Learn Lexical Inference Relations? // Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015), May 31 – June 5, 2015, Denver, CO, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2015. P. 970–976.

66. *Lohmann S., Negru S., Haag F., Ertl T.* Visualizing Ontologies with VOWL // *Semantic Web*. 2016. Vol. 7, no. 4. P. 399–419.

67. *Loukachevitch N. V., Lashevich G., Gerasimova A. A. et al.* Creating Russian WordNet by Conversion // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”, June 1–4, 2016, Moscow, Russia. Moscow, Russia: RSUH, 2016. P. 405–415.

68. *Luu Anh T., Kim J.-j., Ng S. K.* Taxonomy Construction Using Syntactic Contextual Evidence // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), October 25–29, Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. P. 810–819.

69. *Manandhar S., Klapaftis I., Dligach D., Pradhan S.* SemEval-2010 Task 14: Word Sense Induction & Disambiguation // Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010), July 15–16, 2010, Uppsala, Sweden. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. P. 63–68.

70. *McCrae J., Spohr D., Cimiano P.* Linking Lexical Resources and Ontologies on the Semantic Web with Lemon // The Semantic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 – June 2, 2011, Proceedings, Part I. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2011. P. 245–259.

71. *Medelyan O., Witten I. H., Divoli A., Broekstra J.* Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures // *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2013. Vol. 3, no. 4. P. 257–279.

72. *Meyer C. M., Gurevych I.* Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography // Electronic Lexicography, Ed. by S. Granger, M. Paquot. Oxford: Oxford University Press, 2012. P. 259–291.

73. Microsoft Azure for Research - Microsoft Research [Электронный ресурс]. URL: https://www.microsoft.com/en-us/research/academic-program/microsoft-azure-for-research/ (дата обращения: 22.05.2017).

74. *Mikolov T., Sutskever I., Chen K. et al.* Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems 26 (NIPS 2013), December 5–10, 2013, Harrah and Harveys, NV, USA. Curran Associates, Inc., 2013. P. 3111–3119.

75. *Navigli R., Velardi P.* Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites // *Computational Linguistics*. 2004. Vol. 30, no. 2. P. 151–179.

76. *Navigli R., Ponzetto S. P.* BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network // *Artificial Intelligence*. 2012. Vol. 193. P. 217–250.

77. *Navigli R.* A Quick Tour of Word Sense Disambiguation, Induction and Related Approaches // SOFSEM 2012: Theory and Practice of Computer Science: 38th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 21–27, 2012. Proceedings. Berlin, Heidelberg, Germany: Springer-Verlag, 2012. P. 115–129.

78. *Neale S., Gomes L., Agirre E. et al.* Word Sense-Aware Machine Translation: Including Senses as Contextual Features for Improved Translation Models //

Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), May 23–28, 2016, Portorož, Slovenia. Paris, France: European Language Resources Association (ELRA), 2016.

79. *Niles I., Pease A.* Towards a Standard Upper Ontology // Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS '01) - Volume 2001, October 17–19, 2001, Ogunquit, ME, USA. New York, NY, USA: ACM, 2001. P. 2–9.

80. *Palla G., Derenyi I., Farkas I., Vicsek T.* Uncovering the overlapping community structure of complex networks in nature and society // *Nature*. 2005. Vol. 435. P. 814–818.

81. *Panchenko A., Morozova O., Naets H.* A Semantic Similarity Measure Based on Lexico-Syntactic Patterns // Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), September 19–21, 2012, Vienna, Austria. Vienna, Austria: ÖGAI, 2012. P. 174–178.

82. *Panchenko A., Faralli S., Ruppert E. et al.* TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling // Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), June 16–17, 2016, San Diego, CA, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016. P. 1320–1327.

83. *Panchenko A., Simon J., Riedl M., Biemann C.* Noun Sense Induction and Disambiguation using Graph-Based Distributional Semantics // Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016), September 19–21, 2016, Bochum, Germany. Bochum, Germany: Bochumer Linguistische Arbeitsberichte, 2016. P. 192–202.

84. *Parr T.* The Definitive ANTLR 4 Reference. The Pragmatic Programmers, LLC, 2013. 328 pp.

85. *Pedregosa F., Varoquaux G., Gramfort A. et al.* Scikit-learn: Machine Learning in Python // *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.

86. *Pembeci İ.* Using Word Embeddings for Ontology Enrichment // *International Journal of Intelligent Systems and Applications in Engineering*. 2016. Vol. 4, no. 6. P. 49–56.

87. *Pennacchiotti M., Pantel P.* Ontologizing Semantic Relations // Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006), July 17–21, 2006, Sydney, Australia. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. P. 793–800.

88. *Pu X., Pappas N., Popescu-Belis A.* Sense-Aware Statistical Machine Translation using Adaptive Context-Dependent Clustering // Proceedings of the Second Conference on Machine Translation (WMT 17), September 7–8, 2017, Copenhagen, Denmark. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. P. 1–10.

89. *Quillian M. R.* Word concepts: A theory and simulation of some basic semantic capabilities // *Behavioral Science*. 1967. Vol. 12, no. 5. P. 410–430.

90. *Řehuřek R., Sojka P.* Software Framework for Topic Modelling with Large Corpora // New Challenges for NLP Frameworks Programme: A workshop at LREC 2010, May 22, 2010, Valetta, Malta. European Language Resources Association (ELRA), 2010. P. 51–55.

91. *Riedl M., Biemann C.* Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods // Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016), June 12–17, 2016, San Diego, CA, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016. P. 617–622.

92. *Roget P. M.* Roget's Thesaurus of English Words and Phrases / Ed. by S. M. Lloyd. Harlow, Essex: Longman, 1982. 1247 pp.

93. *Roussopoulos N., Mylopoulos J.* Using Semantic Networks for Data Base Management // Proceedings of the 1st International Conference on Very Large Data Bases (VLDB ’75), September 22–24, 1975, Framingham, MA, USA. New York, NY, USA: ACM, 1975. P. 144–172.

94. *Shapiro S. C.* Encyclopedia of Artificial Intelligence. 2nd edition. New York, NY, USA: John Wiley & Sons, Inc., 1992. 1724 pp.

95. *Shwartz V., Goldberg Y., Dagan I.* Improving Hypernymy Detection with an Integrated Path-based and Distributional Method // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), August 7–12, 2016, Berlin, Germany. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016. P. 2389–2398.

96. *Singhal A.* Introducing the Knowledge Graph: things, not strings [Электронный ресурс]. 2012. URL: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html (дата обращения: 20.05.2017).

97. *Storey V. C.* Understanding Semantic Relationships // *The VLDB Journal*. 1993. Vol. 2, no. 4. P. 455–488.

98. *Tarjan R.* Depth-First Search and Linear Graph Algorithms // *SIAM Journal on Computing*. 1972. Vol. 1, no. 2. P. 146–160.

99. The People’s Web Meets NLP / Ed. by I. Gurevych, J. Kim. Springer Berlin Heidelberg, 2013. 378 pp.

100. *Velardi P., Faralli S., Navigli R.* OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction // *Computational Linguistics*. 2013. Vol. 39, no. 3. P. 665–707.

101. *Wang M., Wang C., Yu J. X., Zhang J.* Community Detection in Social Networks: An In-depth Benchmarking Study with a Procedure-oriented Framework // *Proceedings of the VLDB Endowment*. 2015. Vol. 8, no. 10. P. 998–1009.

102. *Welch B. L.* The generalization of ‘Student’s’ problem when several different population variances are involved // *Biometrika*. 1947. Vol. 34, no. 1-2. P. 28–35.

103. Wiktionary [Электронный ресурс]. URL: https://www.wiktionary.org/ (дата обращения: 20.05.2017).

104. *Wilcoxon F.* Individual Comparisons by Ranking Methods // *Biometrics Bulletin*. 1945. Vol. 1, no. 6. P. 80–83.

105. *Zeng X.-M.* Semantic Relationships between Contextual Synonyms // *US-China Education Review*. 2007. Vol. 4, no. 9. P. 33–37.

Приложение 1

Список сокращений и условных обозначений

V	словник
$\mathcal{V}$	множество лексических значений слов
$u^i \in \mathcal{V}$	i-е лексическое значение слова $u \in V$
$\text{senses}(u) \subseteq \mathcal{V}$	множество значений слова $u \in V$
$\mathcal{W}$	граф значений слов
$\mathcal{S}$	множество синсетов
$S \in \mathcal{S}$	синсет $S \subseteq \mathcal{V}$
$s \in S$	лексическое значение некоторого слова в синсете S
$\text{ctx}(s) \subset V$	множество синонимов слова в заданном значении $s \in \mathcal{V}$
$\widehat{\text{ctx}}(s) \subset \mathcal{V}$	контекст с разрешенной многозначностью
$\text{words}(S) \subseteq V$	множество слов, значения которых включены в синсет S
$R \subset V \times V$	асимметричное отношение, порожденное на словнике
$(w, h) \in R$	упорядоченная пара, состоящая из нижестоящего слова $w \in V$ и вышестоящего слова $h \in V$
$\mathcal{R} \subset \mathcal{V} \times \mathcal{V}$	множество дуг семантической сети слов, порождамое асимметричным отношением на множестве лексических значений слов
$(w, h) \in \mathcal{R}$	упорядоченная пара, состоящая из нижестоящего значения слова $w \in \mathcal{V}$ и вышестоящего значения слова $h \in \mathcal{V}$
$\text{NN}_n(\vec{w}) \subset V$	множество, состоящее из n слов, векторные представления которых являются ближайшими соседями векторного представления слова $\vec{w}$ в некотором векторном пространстве
$\text{hctx}(S) \subset V$	объединение множеств вышестоящих слов для каждого слова синсета $S \in \mathcal{S}$
$\widehat{\text{hctx}}(S) \subset \mathcal{V} \times \mathcal{V}$	иерархический контекст с разрешенной многозначностью
$\mathcal{N} = (\mathcal{V}, \mathcal{R})$	семантическая сеть слов

Приложение 2

Словарь терминов

Гипероним (англ. *hypernym*) — более общее понятие по отношению к гипониму.

Гиперпараметр (англ. *hyperparameter*) — параметр метода машинного обучения, для вычисления которого отсутствует аналитическая формула.

Гипоним (англ. *hyponym*) — более частное понятие по отношению к гипероним у.

Жесткая кластеризация (англ. *hard clustering*) — вид кластеризации, при которой каждый объект включен только в один кластер.

Индикаторная функция — функция $\mathbb{1}_A(x) = \begin{cases} 1, & x \in A, \\ 0, & x \notin A. \end{cases}$

Нечеткая кластеризация (англ. *fuzzy clustering*) — вид кластеризации, при которой один объект включен в один или несколько кластеров.

Окрестность G_u (англ. *neighborhood*) — подграф, индуцированный $G = (V, E)$ на множестве всех вершин, смежных с $u \in V$, не включающий u.

Понятие (англ. *concept*) — отдельное лексическое значение слова; элемент множества вершин семантической сети слов.

Семантическая сеть (англ. *semantic network*) — ориентированный граф, вершины которого — понятия, а дуги — связи между понятиями.

Семантическая сеть слов — семантическая сеть, понятия которой — лексические значения слов, а множество дуг порождается асимметричным отношением на множестве лексических значений слов.

Слабоструктурированный словарь — бинарное отношение, порожденное на словнике, не содержащее явного указания значений многозначных слов.

Синсет (англ. *synset*) — множество значений слов, такое, что все пары элементов такого множества принадлежат отношению синонимии.

Словник (англ. *vocabulary*) — множество всех слов какого-либо языка.