Большакова Е.И., Воронцов К.В.,
Ефремова Н.Э., Клышинский Э.С.,
Лукашевич Н.В., Сапин А.С.

Автоматическая обработка текстов на естественном языке и анализ данных

НИУ ВШЭ, 2017 год

УДК 81’32+004.8
ББК 32.813
Б 79

Б 79 Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017. — 269 с.
ISBN 978–5–9909752-1-7

В учебном пособии рассматриваются базовые вопросы компьютерной лингвистики: от теории лингвистического и математического моделирования до вариантов технологических решений. Дается интерпретация основных лингвистических объектов и единиц анализа. Приведены сведения, необходимые для создания отдельных подсистем, отвечающих за анализ текстов на естественном языке. Рассматриваются вопросы анализа тональности и тематического моделирования текстов, извлечения информации из текстов. Предназначено для студентов и аспирантов высших учебных заведений, работающих в области обработки текстов на естественном языке.

УДК 81’32+004.8
ББК 32.813

Содержание

Глава 1

Компьютерная лингвистика: задачи, подходы, ресурсы

Большакова Е.И.

1.1 Введение

Появление сети Интернет и бурный рост доступной текстовой информации значительно ускорило развитие научной области, существующей уже много десятков лет и известной как **автоматическая обработка текстов** (**Natural Language Processing**) и **компьютерная лингвистика** (**Computational Linguistics**). В рамках этой области предложено много перспективных идей по автоматической обработке текстов на естественном языке (ЕЯ), которые были воплощены во многих прикладных системах, в том числе коммерческих. Сфера приложений компьютерной лингвистики постоянно расширяется, появляются все новые задачи, которые успешно решаются, в том числе с привлечением результатов смежных научных областей. О научных достижениях области можно получить представление по интернет-сайту ACL (Association of Computational Linguistics) [1] — международной Ассоциации по Компьютерной Лингвистике, на ко-

тором агрегируются работы многочисленных научных конференций в этой области.

Компьютерная лингвистика (КЛ) — междисциплинарная область, которая возникла на стыке таких наук, как лингвистика, математика, информатика (Computer Science), искусственный интеллект (Artificial Intelligence). В своем развитии она до сих пор вбирает и применяет (при необходимости адаптируя) разработанные в этих науках методы и инструменты.

Истоки КЛ восходят к исследованиям известного американского лингвиста Н. Хомского по формализации структуры естественного языка [6], к первым экспериментам по машинному переводу, выполненным программистами и математиками, а также к разработанным в области искусственного интеллекта первым программам понимания естественного языка (например, [28]).

Поскольку в КЛ объектом обработки выступают тексты естественного языка, ее развитие невозможно без базовых знаний в области общей лингвистики (языкознания) [32]. Лингвистика изучает общие законы естественного языка — его структуру и функционирование, и включает такие области:

- **фонология** — изучает звуки речи и правила их соединения при формировании речи;
- **морфология** — занимается внутренней структурой и внешней формой слов речи, включая части речи и их категории;
- **синтаксис** — изучает структуру предложений, правила сочетаемости и порядка следования слов в предложении, а также общие его свойства как единицы языка.
- **семантика** и **прагматика** — тесно связанные области: семантика занимается смыслом слов, предложений и других единиц речи, а прагматика — особенностями выражения этого смысла в связи с конкретными целями общения;
- **лексикография** описывает лексикон конкретного ЕЯ — его отдельные слова, их грамматические и семантические свойства, а также методы создания словарей.

Наиболее тесно компьютерная лингвистика связана с областью искусственного интеллекта (ИИ) [37], в рамках которой разрабатываются программные модели отдельных интеллектуальных функций. Несмотря на очевидное пересечение исследований в области компьютерной лингвистики и ИИ (поскольку владение языком относится к интеллектуальным функциям), ИИ не поглощает всю КЛ, поскольку она имеет свой теоретический базис и методологию. Общим для указанных наук является компьютерное моделирование как основной способ и итоговая цель исследований, эвристический характер многих применяемых методов.

Несколько упрощенно задача компьютерной лингвистики может быть сформулирована как разработка методов и средств построения **лингвистических процессоров** для различных прикладных задач по автоматической обработке текстов на ЕЯ. Разработка лингвистического процессора для некоторой прикладной задачи предполагает формальное описание лингвистических свойств обрабатываемого текста (хотя бы самое простое), которое может рассматриваться как **модель текста** (или **модель языка**).

1.2 Приложения компьютерной лингвистики

Область приложений КЛ постоянно расширяется, поэтому охарактеризуем здесь наиболее известные прикладные задачи, решаемые ее инструментами.

Машинный перевод (**Machine Translation**) [20] — самое раннее приложение КЛ, вместе с которым возникла и развивалась сама эта область. Первые программы перевода были построены в середине прошлого века и были основаны на простейшей стратегии пословного перевода. Однако довольно быстро было осознано, что машинный перевод требует гораздо более полной лингвистической модели. Такая модель была разработана в отечественной системе ЭТАП [24], а также в нескольких других системах, выполняющих перевод научных текстов.

В настоящее время существует целый спектр компьютерных систем машинного перевода (разного качества), от больших интернациональных исследовательских проектов до коммерческих автоматических переводчи-

ков. Существенный интерес представляют проекты многоязыкового перевода с использованием промежуточного языка, на котором кодируется смысл переводимых фраз. Современное направление — **статистическая трансляция**, опирающаяся на статистику переводных пар слов и словосочетаний. Несмотря на многие десятилетия исследований этой задачи, качество машинного перевода ещё далеко до совершенства. Существенный прорыв в этой области связывают с использованием машинного обучения и нейронных сетей (возникших и исследуемых в рамках ИИ).

Ещё одно довольно старое приложение компьютерной лингвистики — это **информационный поиск** (**Information Retrieval**) [39] и связанные с ним задачи индексирования, реферирования, классификации и рубрицирования документов.

Полнотекстовый поиск документов в больших базах текстовых документов предполагает **индексирование** текстов, требующее их простейшей лингвистической предобработки, и создание специальных индексных структур. Известны несколько моделей информационного поиска, наиболее известной и применяемой является векторная модель, при которой информационный запрос представляется в виде набора слов, а подходящие (релевантные) документы определяются на основе похожести запроса и вектора слов документа. Современные интернет-поисковики реализуют эту модель, выполняя индексирование текстов по употребляемым в них словам и используя для выдачи релевантных документов весьма изощренные процедуры ранжирования. Актуальное направление исследований в области информационного поиска — многоязыковой поиск по документам.

Реферирование текста (**Summarization**) — сокращение его объема и получение краткого изложения его содержания — реферата, что делает более быстрым поиск в коллекциях документов. Реферат может составляться также для нескольких близких по теме документов (например, по кластеру новостных документов). Основным методом автоматического реферирования до сих пор является отбор наиболее значимых предложений реферируемого текста на основе статистики слов и словосочетаний, а также структурных и лингвистических особенностей текстов.

Близкая к реферированию задача — **аннотирование** текста документа, т. е. составление его аннотации. В простейшей форме аннотация представляет собой перечень основных (ключевых) тем текста, для выделения которых используются статистические и лингвистические критерии.

При обработке больших коллекций документов актуальны задачи **классификации** (**Categorization**) и **кластеризации** текстов (**Text Clustering**) [27]. Классификация означает отнесение каждого документа к определенному классу с заранее известными параметрами, а кластеризация — разбиение множества документов на кластеры, т. е. подмножества тематически близких документов. Для решения этих задач применяются методы машинного обучения, в связи с чем эти прикладные задачи часто относят к направлению **Text Mining**, рассматриваемому как часть научной области **Data Mining** (**интеллектуальный анализ данных**) [25]. Задача классификации получает все большее распространение, она решается, например, при распознавании спама, классификации SMS-сообщений и др.

Очень близка к классификации задача **рубрицирования** текста (**Text Classification**) — отнесение текста к одной из заранее известных тематических рубрик (обычно рубрики образуют иерархическое дерево тематик).

Относительно новая задача, связанная с информационным поиском — **формирование ответов на вопросы** (**Question Answering**) [10]. Пример возможного вопроса: «Кто придумал вилку?». Задача решается путем определения типа вопроса, поиском текстов, потенциально содержащих ответ на этот вопрос (при этом обычно применяются поисковые машины), и затем извлечением ответа из выданных текстов.

Актуальная прикладная задача, часто относимая к направлению Text Mining — это **извлечение информации** из текстов (**Information Extraction**) [9], что требуется при решении задач экономической и производственной аналитики. При решении этой задачи осуществляется выделение в тесте ЕЯ определенных объектов — именованных сущностей (имен персоналий, географических названий, названий фирм и пр.), их отношений и связанных с ними событий. Как правило, это реализуется на

основе частичного синтаксического анализа текста, позволяющего выполнять обработку больших массивов текстов, в частности, потоков новостей от информационных агентств. Выделенные данные тем или иным образом структурируются или визуализируются.

К направлению Text Mining относятся и две другие близкие задачи — **выделение мнений** (**Opinion Mining**) и **анализ тональности текстов** (**Sentiment Analysis**) [17], привлекающие внимание все большего числа исследователей в силу своей актуальности. В первой задаче происходит поиск (в блогах, форумах, интернет-магазинах и пр.) мнений пользователей о товарах и других объектах, а также производится анализ этих мнений. Вторая задача близка к классической задаче контент-анализа текстов массовой коммуникации, в ней оценивается общая тональность высказываний и текста в целом.

Ещё одна прикладная задача, которая возникла более 50 лет назад и развитие которой стимулировало появление сети Интернет, — это **поддержка диалога** на ЕЯ. Ранее эта задача чаще всего решалась в рамках какой-либо информационной системы, в частности, для обработки запросов на ЕЯ к специализированной базе данных — в этом случае язык запросов достаточно ограничен (лексически и грамматически), что позволяет использовать упрощенные метода анализа вопросов, а ответы строить по шаблонам. В настоящий момент все более широкое распространение в Интернете получают **чат-боты**, поддерживающие беседу с человеком на некоторую тему и являющиеся наследниками известной системы ELIZA (разработанной в области ИИ в 70 гг.). Очевидный успех этого направления в том, что появились программы (например, программа-собеседник «Евгений Гусман»), которые проходят известный тест Тьюринга.

Совершенно иное прикладное направление, которое развивается хотя и медленно, но устойчиво — это **автоматизация подготовки и редактирования** текстов на ЕЯ. Одними из первых достижений в этом направлении были программы автоматического определения переносов слов и программы орфографической проверки текста (спеллеры, или автокорректоры). Проверка орфографии уже давно реализована в коммерческих системах, выявляются также достаточно частотные синтаксические ошибки

(например, ошибки согласования слов). В то же время в автокорректорах пока не реализовано распознавание более сложных ошибок, в частности, неправильное употребление предлогов и лексические ошибки, возникающие в результате опечаток (*правки* вместо *справки*) или неверного использования схожих слов (например, *весовой* вместо *весомый*). В современных исследованиях КЛ разрабатываются методы автоматизированного выявления и исправления подобных ошибок на основе статистики встречаемости слов и словосочетаний [22].

Ещё одной прикладной задачей является **обучение естественному языку**, в рамках этого направления создаются компьютерные системы, поддерживающие изучение отдельных аспектов (морфологии, лексики, синтаксиса) языка — английского, русского и др. (подобные системы можно найти в Интернете). Разрабатываются также многофукциональные компьютерные словари, не имеющие текстовых аналогов и ориентированные на широкий круг пользователей, например, словарь сочетаемости слов русского языка КроссЛексика [26], дополнительно предоставляющий справки по синонимам, антонимам и другим смысловым связям слов.

Следующее прикладное направление, которое стоит упомянуть — это **автоматическая генерация** текстов на ЕЯ [2]. В принципе, эту задачу можно считать подзадачей уже рассмотренной выше задачи машинного перевода, однако в рамках направления есть ряд специфических задач. Такой задачей является многоязыковая генерация, т. е. автоматическое построение сразу на нескольких языках специальных документов — патентных формул, инструкций по эксплуатации технических изделий или программных систем, исходя из их формальной спецификации.

Активно развивающимся направлением является **распознавание и синтез звучащей речи**. Неизбежно возникающие ошибки распознавания исправляются автоматическими методами на основе словарей и морфологических моделей, также применяется машинное обучение.

1.3 Сложности моделирования естественного языка

Сложность моделирования в КЛ связана с тем, что ЕЯ — большая открытая многоуровневая система знаков, возникшая для обмена информацией в процессе практической деятельности человека, и постоянно изменяющаяся в связи с этой деятельностью [32, 35].

Текст на ЕЯ составлен из отдельных единиц (знаков), и возможно несколько способов разбиения (членения) текста на единицы, относящиеся к разным уровням.

Общепризнано существование следующих уровней [32]:

- уровень предложений (высказываний) — **синтаксический уровень**;
- уровень слов (словоформ — слов в определенной грамматической форме, например, *ручка, дружбой*) — **морфологический уровень**;
- уровень фонем (отдельных звуков, с помощью которых формируются и различаются слова) — **фонологический уровень**.

Фонологический уровень выделяется для устной речи, а для письменных текстов в языках с алфавитным способом записи (в частности, в европейских языках) он соответствует **уровню символов** (фонемы приблизительно соответствуют буквам алфавита).

Уровни, по сути, есть подсистемы общей системы ЕЯ (взаимосвязанные, но в достаточной степени автономные), и в них самих могут быть выделены подсистемы. Так, морфологический уровень включает также **подуровень морфем**. **Морфема** — это минимальная значащая часть слова (корень, приставка, суффикс, окончание, постфикс).

Вопрос о количестве уровней и их перечне в лингвистике до сих пор остается открытым. Как отдельный может быть выделен **лексический уровень** — уровень лексем. **Лексема** — это слово, как совокупность всех его конкретных грамматических форм (к примеру, лексему *лист* образуют формы *лист, листа, листу, листом*). Точнее, лексема — семантический инвариант всех словоформ. В тексте встречаются **словоформы** (лексемы в определенной форме), а в словаре ЕЯ — лексемы, точнее, в словаре записывается каноническая словоформа лексемы, называемая также **лем-**

мой (например, для существительных это форма именительного падежа единственного числа: *лист*).

В рамках синтаксического уровня может быть выделен **подуровень словосочетаний** — синтаксически связанных групп слов (*видел лес, синий шар*), и надуровень **сложного синтаксического целого**, которому примерно соответствует абзац текста. Сложное синтаксическое целое, или **сверхфразовое единство** — это последовательность предложений (высказываний), объединенных смыслом и лексико-грамматическими средствами [35]. К таким средствам относятся в первую очередь лексические повторы и **анафорические ссылки** — ссылки на предшествующие слова текста, реализуемые при помощи местоимений и местоименных слов (*они, этот, там же* и т. д.).

Иерархия уровней проявляется в том, что единицы более высокого уровня разложимы на единицы более низкого (например, словоформы на морфы); более высокий уровень в большой степени обуславливает организацию нижележащего уровня — так, синтаксическая структура предложения в значительной мере определяет, какие должны быть выбраны словоформы.

Можно также говорить ещё об одном уровне — **уровне дискурса** [35], под которым понимается связный текст в его коммуникативной направленности. Под дискурсом понимается последовательность взаимосвязанных друг с другом предложений текста, обладающая определенной смысловой целостностью, за счет чего он выполняет определенную прагматическую задачу. Во многих типах связных текстов проявляется традиционная схематическая (**дискурсивная**) структура, организующая их общее содержание, например, определенную структуру имеют описания сложных технических систем, патентные формулы, научные статьи, деловые письма и др.

Особым является вопрос об **уровне семантики**. В принципе, смысл есть всюду, где есть знаковые единицы языка (морфемы, слова, предложения). Подтверждением самостоятельности уровня семантики считается то, что человек обычно запоминает смысл высказывания, а не его конкретную языковую форму. До сих пор не ясна организация этого уровня, предполагается, что существует универсальный набор элементарных семантических

единиц (называемых **семами**), примерно 2 тысячи, при помощи которых можно выразить смысл любого высказывания.

Кроме многоуровневости системы ЕЯ сложность его моделирования связана с постоянно происходящими в нем изменениями (что вполне ощутимо по прошествии одного-двух десятилетий). Изменения касаются не только словарного запаса языка (новые слова и новые смыслы старых), но также синтаксиса, морфологии и фонетики. Как следствие, принципиально невозможно единожды разработать формальную модель конкретного ЕЯ и построить соответствующий лингвистический процессор. Требуется постоянное пополнение знаний о языке на всех его уровнях и коррекция существующих моделей.

Одним из следствий долгого исторического развития ЕЯ является нестандартная сочетаемость (**синтактика**) единиц на каждом уровне языка. В отличие от искусственных формальных языков (языков логики, языков программирования), в которых сочетаемость знаков диктуется их семантикой и может быть зафиксирована синтаксически (грамматически), в естественных языках соединение слов в предложениях лишь частично может быть описана законами грамматики. В любом языке достаточно много грамматически правильных сочетаний реально не употребляется, например, в русском языке употребительным сочетанием является *крепкий чай*, но не *тяжелый чай* (как в английском *strong tea*).

Одной из самых больших сложностей при обработке текстов на ЕЯ является неоднозначность (многозначность) его единиц, проявляющаяся на всех его уровнях, что выражается в явлениях полисемии, омонимии, синонимии.

Полисемия — наличие у одной единицы языка нескольких связанных между собой значений, в частности, полисемия слов, например: *земля* — суша, почва, конкретная планета. **Синонимия** — полное или частичное совпадение значений разных единиц, например: синонимия слов: *негодяй* и *подлец*, синонимия приставок (морфов) *пре-* и *пере-* (*прекрасный*, *пересохший*). **Омонимия** — совпадение по форме двух разных по смыслу единиц (в отличие от полисемии нет смысловой связи между совпавшими по форме единицами). Различают следующие виды омонимии.

- **Лексическая омонимия** означает одинаково звучащие и пишущиеся слова, не имеющие общих элементов смысла, например, *рожа* — лицо и вид болезни.
- **Морфологическая омонимия** — совпадение форм одного и того же слова (лексемы), например, словоформа *карандаш* соответствует именительному и винительному падежам.
- **Лексико-морфологическая омонимия** (наиболее частый вид) возникает при совпадении словоформ двух разных лексем, например, *стих* — два омонима: глагол в единственном числе мужского рода и существительное в единственном числе, именительном падеже.
- **Синтаксическая омонимия** означает неоднозначность синтаксической структуры, что приводит к нескольким интерпретациям: *Студенты из Минска поехали в Москву*, *Flying planes can be dangerous* (известный пример Хомского) и др.

1.4 Общие этапы и модули обработки текстов

Сложность формального описания ЕЯ и его обработки ведет к разбиению этого процесса на отдельные этапы, соответствующие уровням языка. Большинство современных лингвистических процессоров относятся к модульному типу, в котором каждому уровню/этапу анализа или синтеза текста соответствует отдельный модуль процессора. В случае анализа текста отдельные модули ЛП выполняют:

- **графематический анализ** (**сегментация**), т. е. выделение в тексте предложений и словоформ, точнее **токенов** (т. к. в тексте могут быть не только слова) — переход от символов к словам;
- **Морфологический анализ** — переход от словоформ к их леммам (словарным формам лексем) или **основам** (ядерным частям слова, за вычетом словоизменительных морфем);
- **Синтаксический анализ** — выявление синтаксических связей слов и грамматической структуры предложений;

- **Семантический и прагматический анализ**, при котором определяется смысл фраз и соответствующая реакция системы, в рамках которой работает ЛП.

Таким образом, лингвистический процессор можно рассматривать как многоэтапный преобразователь, переводящий в случае анализа текста каждое его предложение во внутреннее представление его смысла и наоборот в случае синтеза.

Возможны разные схемы объединения и взаимодействия модулей рассмотренных этапов, однако отдельные уровни — морфология, синтаксис и семантика обычно обрабатываются разными механизмами. При решении некоторых прикладных задач можно обойтись без представления в процессоре всех этапов/уровней (к примеру, в ранних экспериментальных программах КЛ обрабатываемые тексты относились к очень узким проблемным областям с ограниченным набором слов, так что не требовался морфологический и синтаксический анализ).

Модули морфологического анализа словоформ различаются в основном по следующим параметрам:

- результату работы — лемма или основа с набором морфологических характеристик (род, число, падеж, вид, лицо и т.п.) заданной словоформы;
- методу анализа — с опорой на словарь словоформ языка или на словарь основ, либо же бессловарный метод;
- возможности обработки словоформы лексемы, не включенной в словарь.

При морфологическом синтезе исходными данными являются лексема и конкретные морфологические характеристики запрашиваемой словоформы данной лексемы, возможен и запрос на синтез всех форм заданной лексемы (так называемой **парадигмы** слова). Результат как морфологического анализа, так и синтеза в общем случае неоднозначен.

Для реализации синтаксического этапа в рамках КЛ предложено большое число разных идей и методов, отличающихся способом описания синтаксиса языка, способом использования этой информации при анализе или синтезе предложений, а также способом представления синтаксиче-

ской структуры предложения [5]. Можно выделить три основных подхода: генеративный подход, восходящий к идеям порождающих грамматик Н. Хомского [6]; подход, восходящий к идеям И. Мельчука и представленный в лингвистической модели «Смысл⇔Текст» [40], а также подход, в рамках которого делаются те или иные попытки преодолеть ограничения первых двух подходов, в частности, теория синтаксических групп [30].

В рамках генеративного подхода синтаксический анализ производится, как правило, на основе формальной контекстно-свободной грамматики, описывающей фразовую структуру предложения, или же на основе некоторого расширения контекстно-свободной грамматики. Эти грамматики исходят из последовательного линейного членения предложения на фразы (различные словосочетания) и отражают поэтому одновременно как его синтаксическую, так и линейную структуры. Полученная в результате иерархическая синтаксическая структура предложения ЕЯ описывается **деревом составляющих**, в листьях которого находятся слова предложения, поддеревья соответствуют входящим в предложение синтаксическим конструкциям (фразам), а дуги выражают отношения вложения конструкций. Данный подход был значительно развит в ряде работ, в частности, в [18].

В рамках второго подхода для представления синтаксической структуры предложения используется более наглядный способ — **деревья зависимостей**. В узлах дерева расположены слова предложения (в корне — слово-предикат, обычно глагол-сказуемое), а каждая дуга дерева, связывающая пару узлов, интерпретируется как синтаксическая **подчинительная связь** между ними, причем направление связи соответствует направлению данной дуги. Поскольку при этом синтаксические связи слов и порядок слов в предложении отделены, то на основе деревьев подчинения могут быть описаны разорванные и **непроективные** конструкции [32], достаточно часто возникающие в языках со свободным порядком слов.

Деревья составляющих больше подходят для описания языков с жёстким порядком слов, представление с их помощью разорванных и непроективных конструкций требует расширения используемого грамматического формализма. Зато в рамках этого подхода более естественно описывают-

ся конструкции с неподчинительными отношениями. В то же время общая трудность для обоих подходов — представление однородных членов предложения.

Синтаксические модели во всех описанных подходах пытаются учесть ограничения, накладываемые на соединение языковых единиц в речи, при этом так или иначе используется понятие валентности [35]. **Валентность** — это способность слова или другой единицы языка присоединять другие единицы определенным синтаксическим способом; **актант** — это слово или синтаксическая конструкция, заполняющая эту валентность. Например, русский глагол *передать* имеет три основные валентности, которые можно выразить следующими вопросительными словами: *кто? кому? что?* В рамках генеративного подхода валентности слов (прежде всего, глаголов) описываются преимущественно в виде специальных фреймов (**subcategorization frames**) [4], а в рамках подхода, основанного на деревьях зависимостей — как модели управления [35].

Модули синтаксического анализа в обоих рассмотренных подходах опираются на **грамматики** ЕЯ. Общее число правил грамматики может быть от нескольких десятков до нескольких сотен, в зависимости от используемого словаря: чем больше информации представлено в словаре, тем короче может быть грамматика и наоборот. Так, в модели «Смысл⇔Текст» [40] упор делается на словарь, а не на грамматику; в применяемом словаре хранится информация, относящаяся к разным уровням языка, в частности, о моделях управления слов и нестандартной сочетаемости слов.

Этап семантического анализа текста наименее проработан в рамках КЛ. Для локального семантического анализа, т. е. анализа предложений были предложены так называемые падежные грамматики и **семантические падежи** (валентности) [8], на базе которых семантика предложения описывается через связи главного слова (обычно глагола) с его семантическими актантами, т. е. через семантические падежи. Например, глагол *передать* описывается семантическими падежами *дающего* (агенса), *адресата* и *объекта передачи*. Используя терминологию ИИ, совокупность семантических падежей часто называют **семантическим фреймом**, описы-

вающим соответствующую ситуацию (в используемом примере — ситуация передачи).

Для представления семантики всего текста обычно используются два формализма (оба они детально описаны в рамках ИИ [37]):

- формулы исчисления предикатов, выражающие свойства, состояния, процессы, действия и отношения;
- семантические сети — размеченные графы, в которых вершины соответствуют понятиям, а дуги — отношениям между ними.

Мало исследован в КЛ уровень прагматики и дискурса, к которому анализ текст в целом. В основном разработаны методы анализа локальной связности текста, в первую очередь, разрешение анафорических ссылок [15]. Среди работ, идеи которых все чаще применяются, следует указать теорию риторических структур [14]; в работе [38] предложена модель синтеза дискурсивной структуры описательных текстов.

Используемые в компьютерной лингвистике модели ЕЯ обычно строятся с учетом лингвистических теорий и моделей; выделим особенности именно моделей КЛ [4]:

- формальность и, в конечном счете, алгоритмизируемость;
- функциональность (воспроизведение функций языка как «черного ящика», без построения точной модели синтеза и анализа речи человеком);
- опора на лингвистические ресурсы;
- экспериментальная обоснованность, предполагающая тестирование модели на разных текстах.

1.5 Лингвистические ресурсы: построение и применение

Разработка и применение лингвистических процессоров опирается на использование тех или иных лингвистических ресурсов: лексических (словарных) и текстовых. К лексическим ресурсам относятся словари, тезаурусы, онтологии.

Словари являются наиболее традиционной формой представления лексической информации; они различаются своими единицами (обычно слова или словосочетания), структурой, охватом лексики (словари терминов конкретной проблемной области, словари общей лексики, словари синонимов или паронимов и т.п.). Единица словаря называется **словарной статьей**, в ней представляется информация о лексеме. Лексические омонимы обычно представляются в разных словарных статьях.

К лексическим ресурсам относятся **базы словосочетаний**, в которые отбираются наиболее типичные словосочетания конкретного языка. Такая база словосочетаний русского языка (более миллиона единиц) составляет ядро системы КроссЛексика [26].

Более сложными видами лексических ресурсов являются **тезаурусы** и **онтологии**. Тезаурус — это семантический словарь, т. е. словарь, в котором представлены смысловые связи слов — синонимические, отношения Род-Вид (иногда называемые отношением Выше-Ниже), Часть-Целое, ассоциации. В качестве характерного примера можно привести информационно-поисковый тезаурус РуТез для русского языка, охватывающего общественно-политическую лексику [36].

С понятием тезауруса тесно связано понятие онтологии [11]. Онтология — набор понятий, сущностей определенной области знаний, ориентированный на многократное использование для различных задач. Онтологии могут создаваться на базе существующей в языке лексики — в этом случае они называются **лингвистическими**.

Подобной лингвистической онтологией считается система WordNet [21] — большой лексический ресурс, в котором собраны слова английского языка: существительные, прилагательные, глаголы и наречия и представлены их смысловые связи нескольких типов. Для каждой из указанных частей речи слова сгруппированы в группы синонимов (**синсеты**), между которыми установлены отношения антонимии, гипонимии (отношение род-вид), меронимии (отношение часть-целое), тропонимии. Ресурс содержит примерно 117 тыс. понятий-синсетов (около 155 тысяч лексем), число уровней иерархии для отношения род-вид в среднем равно 6–7, достигая

порою 15. Верхний уровень иерархии формирует общую онтологию — систему основных понятий о мире.

По схеме английского WordNet были построены аналогичные лексические ресурсы для других европейских языков, объединённые под общим названием EuroWordNet.

Текстовые ресурсы, служащие для построения модулей лингвистических процессоров, охватывают коллекции текстов (обычно для конкретных проблемных областей) и **текстовые корпуса**.

Корпус текстов — это представительный массив текстов, собранный по определённому принципу (по жанру, авторской принадлежности и т.п.) и обладающий **лингвистической разметкой** — морфологической, акцентной, синтаксической, дискурсивной или др. [3]. В настоящее время известно несколько сотен различных корпусов (для разных ЕЯ и с различной разметкой), в России наиболее известными являются Национальный корпус русского языка (НКРЯ) [41], OpenCorpora [16] и ГИКРЯ [29], они отличаются целями и методами создания, набором включенных русскоязычных текстов.

Размеченные корпуса создаются обычно экспертами-лингвистами и используются как для лингвистических исследований, так и для настройки (обучения) лингвистических процессоров на основе методов машинного обучения. Поскольку разметка текстов — достаточно трудоёмкая и долгая работа, требующая специалистов, для ускорения создания корпусов прибегают к краудсорсингу, при котором разметка выполняется волонтёрами, а затем модерируется. Другой способ — полуавтоматическая разметка, когда сначала работает уже готовый модуль анализа текста, а его результаты подправляются человеком-экспертом. Еще один путь — поиск естественной разметки текста. К примеру, для машинного обучения в задачах оценки тональности текстовых отзывов могут быть использованы тексты интернет-отзывов с уже проставленными оценками.

Заметим, что поскольку корпуса и коллекции текстов всегда ограничены по представленным в них языковым явлениям, в качестве более полного источника образцов современной речи могут рассматриваться тексты сети

Интернет. В частности, из собранных интернет-текстов составлен русскоязычный корпус ГИКРЯ.

1.6 Подходы к построению модулей и систем КЛ

В настоящее время для создания модулей лингвистических процессоров применяется два главных подхода: **основанный на правилах** (**rule-based**), или **инженерный**, и **основанный на машинном обучении** (**machine learning**).

Исторически первым является подход на правилах, который заключается в описании необходимой лингвистической информации в виде формальных правил. В ранних системах правила были встроены в программный код, сейчас же для записи правил используется либо уже готовый формальный язык, либо подобный язык специально создаётся для разрабатываемого приложения. Правила создаются лингвистами или специалистами по проблемной области обрабатываемых текстов.

В рамках подхода, основанного на машинном обучении, источником лингвистической информации выступают не правила, а отобранные тексты проблемной области. Среди методов, применяемых в рамках подхода, выделяют методы **обучения с учителем** (**supervised**), методы **обучения без учителя** (**unsupervised**), методы **частичного обучения с учителем** (**bootstrapping**).

Чаще всего применяется обучение с учителем, при котором происходит построение математической и программной модели — **машинного классификатора**, который умеет распознавать различные классы единиц текста (слов, словосочетаний и других конструкций) или самих текстов. Построение классификатора происходит на специально размеченном текстовом корпусе (**обучающей выборке**), в котором распознаваемым единицам (или самим текстам) приписаны метки, кодирующие важные признаки распознаваемых единиц/текстов. Обучение представляет собой, по сути, выявление общих закономерностей, присущих текстам на ЕЯ, на основе данных обучающей выборки.

Оба рассмотренных подхода имеют свои достоинства и недостатки. Создание правил трудоемко и требует достаточно квалифицированного труда, как правило, лингвиста. Очень часто даже лингвист не может предусмотреть заранее все частные случаи, которые надо отразить в правилах. В то же время правила обычно декларативны и легко понимаемы, поэтому их просто поддерживать: модифицировать и расширять, тем самым отлаживая функционирование процессора. Машинное обучение не требует ручного труда по составлению правил и сокращает время разработки систем, однако необходимы знания для выбора подходящих методов обучения. Кроме того, результирующие модели (классификаторы) непрозрачны для понимания, т. к. не имеют явной лингвистической интерпретации. Также машинное обучение предполагает наличие подходящего размеченного корпуса текстов, что не всегда возможно. Создание такого корпуса в любом случае требует значительных объемов ручного труда.

Сравнивая применение этих подходов, можно заметить, что ранее чаще применялся подход на правилах, поскольку было мало размеченных текстовых корпусов. С появлением различных размеченных данных все чаще прибегают к машинному обучению, как быстрому способу получения нужного приложения КЛ.

Современная тенденция — модульные, многокомпонентные системы автоматической обработки текстов (**multi-component, pipelined systems**), причем разные модули могут быть созданы в рамках разных подходов, например, модуль графематического анализа — на основе машинного обучения, а морфологического — на основе правил.

Машинное обучение довольно часто применяется для обработки коллекций текстовых документов, с использованием **признаковой модели текста**, при которой признаки определены для каждого документа по отдельности. Признаками могут выступать различные информационные характеристики текста: как лингвистические, так статистические и структурные: например, частота определенных слов (или их категорий) в документе, частота использования спецзнаков, соотношение частей речи слов, наличие определенных синтаксических конструкций или разделов текста, дата создания и др.

Разновидностями признаковой модели являются модель **BOW** (**bag of words** — мешок слов), в которой текст характеризуется набором своих значимых слов (обычно это все знаменательные слова, точнее, их леммы), а также **векторная модель текста**, в которой указанный набор упорядочен. Векторная модель применяется, например, в информационном поиске, при этом в качестве признаков чаще берутся не слова, а более сложные характеристики, такие как показатель TF-IDF [39] для слов.

Особняком стоит **статистическая языковая модель** (**Language Model**), характеризующая язык в целом, а не отдельный текст [12, 19]. Классическая языковая модель строится по представительному массиву текстов конкретного ЕЯ (например, английского) путем подсчета частот N**-грамм** слов (т. е. стоящих рядом слов). Чаще всего рассматриваются биграммы ($N = 2$) и триграммы ($N = 3$). Модель призвана давать ответ на вопрос, насколько вероятно появление заданного слова, если непосредственно перед ним встречались определенные слова. Вероятности рассчитываются на основе собранной статистики. Такая модель применяется, к примеру, для разрешения лексической неоднозначности. Разновидности модели: N-граммы частей речи слов текста или N-граммы букв текста (возможны и другие модели) применяются для разрешения морфологической омонимии или для выявления опечаток в тексте соответственно.

1.7 Заключение

Компьютерная лингвистика демонстрирует вполне осязаемые результаты в различных приложениях по автоматической обработке и анализу текстов на ЕЯ. В большинстве приложений используются простые и редуцированные модели ЕЯ, которые однако дают приемлемые или даже хорошие результаты; нередко качество результатов достигает экспертного уровня — обычно там, где мнения экспертов могут расходиться. Дальнейший прогресс в области КЛ связан как с более точным учетом лингвистических особенностей текстов на различных этапах его обработки и применением более детальных лингвистических моделей, так и с развитием методов ма-

шинного обучения и поиском более эффективных методов и их комбинаций для каждой прикладной задачи.

Более подробно с различными подходами, методами, системами и инструментами компьютерной лингвистики можно ознакомиться в книгах [26, 12, 23, 27, 31, 34, 42].

1.8 Список литературы

[1] ACL Anthology: A Digital Archive of Research Papers in Computational Linguistics. http://aclweb.org/anthology/

[2] Bateman, J., Zock M. Natural Language Generation. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p.304.

[3] Biber, D., Conrad S., and Reppen D. Corpus Linguistics. Investigating Language Structure and Use. Cambridge University Press, Cambridge, 1998.

[4] Bolshakov, I.A., Gelbukh A. Computational Linguistics. Models, Resources, Applications. Mexico, IPN, 2004.

[5] Carroll J R. Parsing. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 233-248.

[6] Chomsky, N. Syntactic Structures. The Hague: Mouton, 1957.

[7] Feldman R., Sanger J. (ed.). The text mining handbook: advanced approaches in analyzing unstructured data. — Cambridge University Press, 2007.

[8] Fillmore C. J. The Case for Case. In: Bach and Harms (Ed.): Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, 1968.

[9] Grishman R., Information Extraction. In: The Handbook of Computational Linguistics and Natural Language Processing. A. Clark, C. Fox, and S. Lappin (Eds), Wiley-Blackwell, 2010, pp. 515-530.

[10] Harabagiu, S., Moldovan D. Question Answering. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 560-582.

[11] Hirst, G. Ontology and the Lexicon. In.: Handbook on Ontologies in Niformation Systems. Berlin, Springer, 2003.

[12] Jurafsky D., Martin J. Speech and Language Processing. An Introduction to Natural Language Processing, Comp. Linguistics and Speech Recognition. Prentice Hall, 2000.

[13] Manning, Ch. D., H. Schutze. Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[14] Mann, W.C., Thompson S.A. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text, 8 (3), 1988, p. 243-281.

[15] Mitkov R. Discourse Processing. In: The Handbook of Computational Linguistics and Natural Language Processing. A. Clark, C. Fox, and S. Lappin (Eds), Wiley-Blackwell, 2010.

[16] Open Corpora: Открытый корпус. http://opencorpora.org

[17] Pang Bo, Lee L. Opinion Mining and Sentiment Analysis. In: Foundations and Trends® in Information Retrieval. Now Publishers, 2008.

[18] Polard C., Sag I. Head-Driven phrase structure grammer/ Chicago University Press, 1994.

[19] Samuelsson C. Statistical Methods. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 358-375.

[20] Somers, H. Machine Translation: Latest Developments. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 512-528.

[21] Word Net: an Electronic Lexical Database. /Edit. by Christiane Fellbaum. Cambridge, MIT Press, 1998.

[22] Wu J., Yu-Chia Chang Y., Teruko Mitamura T., Chang J. Automatic Collocation Suggestion in Academic Writing. In: Proceedings of the ACL 2010 Conference Short Papers, 2010.

[23] Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И. и др. — М.: МИЭМ, 2011.

[24] Апресян Ю.Д. и др. Лингвистическое обеспечение системы ЭТАП-2. М.: Наука, 1989.

[25] Барсегян А.А. и др. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP — 2-е изд. — СПб.: БХВ-Петербург, 2008.

[26] Большаков, И.А. КроссЛексика — большой электронный словарь сочетаний и смысловых связей русских слов. // Комп. лингвистика и интеллект. технологии: Труды межд. Конф. «Диалог 2009». Вып. 8 (15) М.: РГГУ, 2009, с. 45-50.

[27] Васильев В. Г., Кривенко М. П. Методы автоматизированной обработки текстов. — М.: ИПИ РАН, 2008.

[28] Виноград Т. Программа, понимающая естественный язык — М.: Мир, 1976.

[29] ГИКРЯ: генеральный интернет-корпус русского языка.
`http://www.webcorpora.ru/`

[30] Гладкий А.В. Синтаксические структуры естественного языка в автоматизированных системах общения. — М.: Наука, 1985.

[31] Ингерсолл Г.С., Мортон Т.С., Фэррис Э.Л. Обработка неструктурированных текстов. Поиск, организация и манипулирование / Пер. с англ. — М.: ДМК Пресс, 2015.

[32] Касевич В.Б. Элементы общей лингвистики. — М.: Наука, 1977.

[33] Кобозева И.М. Лингвистическая семантика. — М., 2009.

[34] Леонтьева Н. Н. Автоматическое понимание текстов: Системы, модели, ресурсы: Учебное пособие — М.: Академия, 2006.

[35] Лингвистический энциклопедический словарь /Под ред. В. Н. Ярцевой, М.: Советская энциклопедия, 1990, 685 с.

[36] Лукашевич Н.В. Тезаурусы в задачах информационного поиска. — М.: Изд-во Московского университета, 2011..

[37] Люгер Дж. Искусственный интеллект: стратегии и методы решения сложных проблем. М., 2005.

[38] Маккьюин К. Дискурсивные стратегии для синтеза текста на естественном языке // Новое в зарубежной лингвистике. Вып. XXIV. М.: Прогресс, 1989, с.311-356.

[39] Маннинг К., Рагхаван П., Шютце Ч. Введение в информационный поиск — М.: Вильямс, 2011.

[40] Мельчук И.А. Опыт теории лингвистических моделей «СМЫСЛ ⇔ ТЕКСТ». — М.: Наука, 1974.

[41] Национальный Корпус Русского Языка. `http://ruscorpora.ru`

[42] Прикладная и компьютерная лингвистика / Под ред. Николаева И.С. и др. — М.: ЛЕНАНД, 2016

Глава 2

Морфологический анализ текстов

Клышинский Э.С., Сапин А.С.

2.1 Морфологический анализ

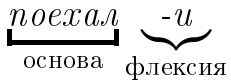
2.1.1 Основные понятия

Одной из основ языка является его словарь. Но что содержит в себе словарь? Так как термин «слово» является слишком многозначным для того, чтобы употреблять его в строгом научном тексте, введём несколько понятий.

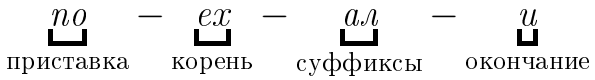
Если из текста извлечь все подстроки, не содержащие разделителей (пробелов, некоторых знаков препинания и др.), мы получим множество **токенов**. Например, токеном будут являться слова *подъезд* или *под* (так как они могут встретиться в тексте сами по себе), но не будет являться подстрока *подъ* (если она, конечно, не написана в тексте именно в таком виде).

Считается, что для каждого токена существует его **начальная** (или **нормальная**) **форма** (также называемая **леммой**). От этой начальной формы создаются все остальные формы слова путем **флексии**, то есть

некоторых изменений этой начальной формы. Например:



Образование новых слов или их форм происходит на уровне комбинирования **морфов** — минимальных значащих единиц языка. Морфы делятся на корневые (корень слова) и служебные: **префикс** (приставка), **суффикс**, **флексия** (окончание), **постфикс**. Носителем основного смысла слова является корень, а служебные, в общем случае, придают дополнительный смысл. Разбиение слова на морфы называется **морфемным разбором**. Например,



Некоторые служебные морфы (например, приставки и суффиксы) отвечает за образование новых слов, другие (например, окончания) — за образование форм слов. Изменение формы слова привязывается к набору **грамматических параметров (тегов)**: часть речи, род, число, падеж, притяжательность и проч.

Под **словоформой** мы будем понимать группу (кортеж), состоящую из токена, связанной с ним начальной формы и множества грамматических параметров[1]. Например, словоформой является множество ⟨кошачьего, кошачий, {прил, муж. род, ед. ч., род. п}⟩, которое содержит в себе строку «кошачьего», связанную с начальной формой «кошачий», и которая характеризуется следующими параметрами: прилагательное в мужском роде, единственном числе и родительном падеже. Под **лексемой** мы будем понимать множество всех словоформ, связанных с данной начальной формой.

Исходя из данных определений, можно сказать, что задачей морфологического анализа (лемматизации) является нахождение в словаре слово-

[1]Обратите внимание, что здесь термин словоформа используется нестандартным образом. Обычно под *словоформой* понимается то, что мы выше определили как токен, то есть подстроку в тексте, находящуюся между знаками препинания. Сам термин *токен* при этом не используется или сохраняет своё значение. В данной работе нам необходимо разделять понятия подстроки и результата разбора. В связи с этим мы вводим новое определение для *словоформы*.

формы по ее токену. Задача морфологического синтеза прямо противоположная: по словоформе необходимо вернуть ее токен.

Или более формально:

- **морфологический анализ** — это получение леммы или основы (псевдоосновы) заданного токена, а при необходимости, морфологических параметров;
- **морфологический синтез** — это генерация нужной словоформы слова или всей его парадигмы по нормальной форме (или основе) и морфологическим характеристикам.

Под **словоупотреблением** мы будем понимать вхождение словоформы в текст. В зависимости от контекста, под словоупотреблением может пониматься либо только строка словоформы, либо словоформа как множество. Например, фраза *«Косой косой косил косой косой за песчаной косой»* содержит в себе 8 словоупотреблений, 7 словоформ, 6 лексем и 4 уникальных токена. Это становится очевидным, если вместо строк записать словоформы.

```
<косой, косой, {прил, муж. род, ед. ч., им. п.}>
<косой, косой, {сущ, муж. род, ед. ч., им. п.}>
<косил, косить, {гл, 3 лицо, ед. ч., прош. вр., муж. род}>
<косой, косой, {прил, жен. род, ед. ч., тв. п.}>
<косой, коса, {сущ, жен. род, ед. ч., тв. п.}>
<за, за, {предл}>
<песчаной, песчаный, {прил, жен. род, ед. ч., тв. п.}>
<косой, косой, {сущ, жен. род, ед. ч., тв. п.}>
```

Здесь мы неявно использовали запись грамматических параметров, подразумевающую раздельное написание **имени** и **значения** параметра. Именем параметра может служить род, число, время, склонение, краткость формы прилагательного и другие признаки слов, принятые в данном языке. Значение параметра — это конкретное значение, которое может принимать данный признак. Так, например, падеж может быть именительным, родительным, местным, аккузативным; род может быть мужским, женским, средним; число — единственным, множественным, двойственным и т. д.

Подобная форма записи называется **списковой** и является удобной в ряде случаев: удобнее читать отдельные параметры, само наличие параметра с таким именем может являться показателем, названия значений параметров могут совпадать, параметры могут следовать в произвольном порядке, количество параметров является переменным. В некоторых ситуациях удобнее оказывается **позиционная** запись, когда первая буква обозначает часть речи, а следом за ней идёт фиксированное количество букв, задающих значения параметров. Количество параметров задаётся частью речи, последовательность параметров фиксируется. Подобная запись удобна в тех случаях, когда список параметров не будет изменяться. Она занимает меньше места, но при этом требует дополнительной информации о расположении параметров. Сравните тот же пример, в котором используется несколько разных форматов записи.

```
<косой, косой, {прил, муж. род, ед. ч., им. п.}>
<косой, косой, {pos=сущ, род=муж, число=ед, падеж=им}>
<косил, косить, ГЗЕПМ>
<косой, косой, AFSI>
<косой, коса, N, {gender=f, number=sg, case=instr}>
<за, за, PREP>
<песчаной, песчаный, Adj,Fem,Sg,Abl>
<косой, косой, {сущ, жен. род, ед. ч., тв. п.}>
```

Помимо конкретных значений, для параметров иногда задают заменители. Например, если необходимо указать любое значение параметра, можно использовать знак звёздочки, а для указания совпадения значений параметров знак минус.

```
<*, *, {pos=прил, род=+, число=+, падеж=+}>                    (a)
<*, *, {pos=сущ, род=+, число=+, падеж=+}>

<*, *, П+++><*, *, С+++>                                       (b)
```

Здесь показан пример записи именной группы, содержащей по одному прилагательному и существительному, согласующемуся по роду, числу и падежу, в списковой (a) и позиционной (b) записях.

Помимо заменителей вводятся ещё два значения: пустое и произвольное (или нулевое). Пустое значение показывает отсутствие значения у параметра, например, род у глагола в настоящем времени. Произвольное значение показывает, что удобнее считать, что данный параметр есть у слова, но он не принимает никакого конкретного значения. Например, значение рода у существительных, употребляемых только в множественном числе, лучше считать нулевым. Пустое значение особенно удобно использовать в позиционной системе записи, когда необходимо показать отсутствие параметра в данной форме слова.

Заметим, что внутри одной лексемы (то есть набора словоформ) значения некоторых грамматических параметров не изменяются. Например, у существительных фиксируется род, у глаголов — притяжательность. С другой стороны, прилагательное и причастие в общем случае не обладают ни одним постоянным параметром. На основании этого признака разделим параметры на **словообразовательные** и **формообразовательные** (хотя в целом в лингвистике нет устоявшейся терминологии на этот счёт; см., например, [14, с. 123]). Словообразовательные параметры присущи всем словоформам, входящим в одну лексему. Как следствие, в некоторых случаях мы можем хранить их один раз для всей лексемы. Формообразовательные параметры присущи только данной словоформе и должны храниться вместе с ней.

2.1.2 Дополнительные функции систем морфологического анализа

Прежде чем провести морфологический анализ, необходимо выделить из текста отдельные слова. В связи с этим вместе с системой морфологического анализа иногда поставляется подсистема графематического анализа. Входной поток символов разбивается на токены нескольких классов: буквенные последовательности, числа, цифробуквенные комплексы, пунктуация, разделители, иероглифы. При этом каждый класс токенов имеет собственный набор тегов, в частности, для слов это может быть язык (кириллица или латиница) и регистр.

Анализ несловарных слов является важной функцией систем морфологического анализа. Как показывает практика, составить абсолютно полный словарь практически невозможно, ведь естественный язык является постоянно развивающейся системой. При увеличении размера анализируемых текстов до нескольких десятков миллиардов слов словарь так и не выходит на насыщение. Это связано с появлением текстов и лексики из новых предметных областей; всё новыми ошибками, появляющимися в текстах; новыми именами собственными, которые авторы придумывают для беллетристики, или которые приходят с новостями из заграницы. На одном из круглых столов конференции Диалог[2] коллеги из Яндекса говорили, что пользователи делают около 480 опечаток в день в слове *одноклассники*. В месяц количество уникальных опечаток превышает 1500. При этом от 80% до 90% всех опечаток отстоят от оригинала на одну ошибку.

Ещё одним ярким примером активного словообразования является немецкий язык с его составными существительными. В немецком языке принято некоторые понятия, которые должны описываться несколькими словами, давать одним словом. Это новое слово получается за счёт «склеивания» входящих в него слов по определенным правилам. Так, в [15, с. 17] приводится в качестве примера слово *Donaudampfschifffahrtskapitän* — капитан рейса, выполняемого пароходом по Дунаю: *Donau* — Дунай, *Dampfschiff* — пароход (в свою очередь тоже составное слово), *Fahrt* — рейс, *Kapitän* — капитан. Там же говорится о том, что река Дунай вполне может быть заменена на Неву или другую реку, в результате чего будет получено новое слово. Подобное образование слов особенно принято в формальных областях человеческой деятельности: названия мероприятий, министерств, регламентов и др.

Аналогичная ситуация наблюдается и в русском языке, хотя и в значительно меньшей степени. Несколько неудачно, но вполне художественно смотрятся такие фразы, как, например, *краснопогонное общество* или *интернетоговорящая публика*. Аналогично образуются слова через дефис: *серо-буро-малиновый, кубик-кубик* и др. Помимо этого, не следует скиды-

[2] `http://dialog-21.ru/`

вать со счетов слова новояза, принятые в некоторых сообществах: *даунлоадить*, *бэксайд*, *гуфи*, *запитонить*, *напитонил и в продакшн* и пр.

В связи с наличием подобных явлений в языке в систему морфологического анализа необходимо вводить модуль анализа несловарных слов. Обычно он реализуется с помощью набора эвристик, таких как отсечение префиксов, аналогия по окончанию и правила для слов с дефисом.

Ещё одной важной функцией является снятие морфологической омонимии. В разных системах реализуются два различных подхода для решения этой задачи: контекстное и бесконтекстное снятие. Бесконтекстное снятие выполняется на основе подсчёта статистики по размеченному корпусу, а контекстное снятие — с помощью классификатора, настроенного при помощи одного из методов машинного обучения.

2.1.3 Представление текста в виде вектора

В примере выше мы уже считали количество словоупотреблений, словоформ, лексем и токенов. Аналогично, мы можем рассчитать, например, сколько раз встретилась каждая из лексем. Для этого надо предварительно провести лемматизацию. В итоге для нашего примера мы получим вектор, показанный в таблице 2.1.

Таблица 2.1. Вектор лексем для текста *«Косой косой косил косой косой за песчаной косой»*

токен	за	коса	косить	косой (прил)	косой (сущ)	песчаный
частота	1	2	1	2	1	1

Аналогичным образом можно построить вектор частот для любого текста, состоящего из произвольного количества предложений. Вектор может быть построен как для лемм, так и для словоформ или токенов, в зависимости от того, какая перед нами стоит задача.

Векторное представление позволяет перейти от текста к его описанию в некотором пространстве. Представим себе, что каждая лемма задаёт собственное направление в некотором многомерном пространстве (размерность которого будет равна количеству лемм в тексте). В таком случае, текст можно представить как точку или вектор в этом многомерном про-

странстве. Более того, если у нас имеется несколько текстов, то мы можем объединить все леммы (словоформы, токены) этих текстов и получить пространство большей размерности. В этом новом пространстве можно будет представить в виде точки или вектора каждый из имеющихся текстов.

Переход к многомерному пространству позволяет измерять расстояния между текстами, то есть степень, в которой они похожи или не похожи друг на друга. Логично предположить, что если в двух текстах употребляются примерно одни и те же слова, то эти тексты посвящены одной (или сходной) теме. И наоборот, если в тексте отсутствуют одинаковые слова, есть большая вероятность, что они относятся к разным предметным областям (если только над текстом не поработал опытный копирайтер, поставивший себе задачу не оставить в переписываемом тексте ни одного слова).

Заметим, что привычная Евклидова метрика работает в таком пространстве очень и очень плохо. Но в данной главе мы не будем останавливаться на вопросе определения меры сходства между текстами. Рассмотрим пока особенности хранения таких векторов.

Средняя новостная заметка содержит порядка сотни словоупотреблений (или нескольких сотен в случае аналитической статьи), среди которых встречается в несколько раз меньше лемм. Однако если мы будем анализировать все новостные заметки, полученные из одного источника, размер текста будет уже около сотни миллионов словоупотреблений и порядка 300 000 лемм[3]. Но при этом мы продолжаем работать с отдельными заметками или их группами. Получается, что для работы с одной заметкой в сотню слов нам необходим вектор размерностью в несколько сотен тысяч параметров. При количестве заметок, приближающемся к миллиону, разместить всю информацию в оперативной памяти становится проблематично. А если нам необходимо работать с сотнями новостных лент?

Для того чтобы избежать подобной траты памяти, мы можем сократить объем хранимой информации. Для слов, которые не встретились в данной заметке, соответствующее значение в векторе будет равно нулю. Если слова будут храниться в алфавитном порядке, то мы можем хранить только те слова, частота которых не равна нулю. В этом случае можно

[3]Конкретные цифры очень сильно зависят от источников. Например, объем текстов у РИА Новости измеряется уже миллиардами словоупотреблений, а словарь значительно богаче.

предположить, что все слова, которые расположены между двумя словами в векторе, встретились ноль раз. То есть, фактически мы храним только ненулевые значения, а размерность вектора может быть приведена к общему количеству слов во всех текстах.

2.1.4 Представление слов в словаре

Вернёмся теперь к той информации, которая хранится в словаре. Списки грамматических параметров разных словарей существенно отличаются между собой. Так, например, **морфологическая языковая модель**, заложенная в словарь Национального корпуса русского языка, включает в себя по два дательных, родительных, винительных и предложных падежа, а также звательный падеж. Морфологическая языковая модель ABBYY предполагает разметку притяжательных местоимений как прилагательных. Включение в языковую модель параметра одушевлённости может приводить как к положительным, так и к отрицательным эффектам: можно повысить точность морфологической разметки, но при этом возрастёт сложность предсказания новых слов.

Таким образом, результаты морфологической разметки будут отличаться (как визуально, так и содержательно) в зависимости от наполнения словаря, используемой языковой модели и формата записи. Как следствие, получаемые результаты также могут отличаться.

В русском языке слова, принадлежащие одной лексеме, обычно отличаются только окончаниями. За счёт этого в рамках одной лексемы можно выделить **псевдооснову** — начальную часть слова, которая не изменяется во всех словоформах. Часть слова, не принадлежащая псевдооснове, называется **псевдоокончанием**. Множество псевдоокончаний с привязанными к ним наборами параметров называется **морфологической парадигмой**. Пример парадигмы для слова *завод* показан в таблице 2.2.

Таблица 2.2. Словоизменительная парадигма слова *завод*

Единственное число						Множественное число					
Им.	Род.	Дат.	Вин.	Тв.	Пр.	Им.	Род.	Дат.	Вин.	Тв.	Пр.
∅	а	у	∅	ом	е	ы	ов	ам	ы	ами	ах

Разные слова могут обладать одинаковыми парадигмами. Например, слова *вектор*, *завод* и *стол* обладают одной и той же парадигмой. Будут ли обладать одной и той же парадигмой слова *вектор* и *лектор* зависит от используемой морфологической модели: при наличии параметра одушевлённости слова будут относиться к разным парадигмам, при отсутствии — к одной. Очевидно, что, например, глаголы и существительные будут обладать разными парадигмами, тогда как для прилагательных и существительных, производных от прилагательных, разница будет заключаться лишь в части речи.

Псевдоокончание не всегда будет совпадать с окончанием, так как в некоторых словах при образовании словоформы может происходить изменение корня, изменяться, появляться или выпадать суффиксы, происходить другие изменения. Например, для глагола *гнать* часть форм будет образовываться с добавлением гласной *-о*:

$$\overbrace{г}\overbrace{нать} \; — \; \overbrace{г}\overbrace{онит}$$

Псевдооснова в данном примере будет *г-*, псевдоокончания *-нать* и *-онит*, соответственно.

Для глагола *идти* псевдооснова оказывается пустой, так как для форм *идти* и, например, *шёл* общих подстрок не находится.

$$\overbrace{}\overbrace{идти} \; — \; \overbrace{}\overbrace{шёл}$$

Понятием, тесно связанным с морфологической парадигмой, является **словоизменительный класс** — подкласс слов одной части речи с одной морфологической парадигмой. Таким образом, в словаре для каждой лексемы можно хранить только ее часть речи и словоизменительный класс.

В словаре бывает необходимо хранить и составные слова, в которых меняется каждое входящее в их состав слово, например, *Римский-Корсаков*. В этом случае псевдооснова будет идти до первой изменившейся буквы, то есть фактически будет состоять только из псевдоосновы первого слова.

$$\overbrace{Римск}\overbrace{ий\text{-}Корсаков} \; — \; \overbrace{Римск}\overbrace{ого\text{-}Корсакова}$$

Для некоторых языков проблема хранения изменений стоит гораздо острее. Например, в арабском языке слово образуется от трёхбуквенной основы добавлением новых букв между имеющихся букв, а также аффиксов в начале и конце слова. С другой стороны, добавление некоторых частей меняет смысл слов, то есть их можно считать новыми словами, которые надо хранить отдельно. В немецком языке проблему представляют отделяемые приставки у глаголов: в некоторых случаях определённые приставки отсоединяются от глагола и переходят в конец предложения. Как и в русском языке, приставка существенно меняет смысл слова. Получается, что для восстановления начальной формы необходимо знать не только сам токен, но и ещё один, находящийся в определённом месте предложения.

Мы рассмотрели только один вариант образования новых форм слов — флексию. Но, например, в тюркских языках новые формы образуются путём конкатенации основы с аффиксами, причём сами аффиксы присоединяются в строго определённом порядке. Подобные языки называются агглютинативными. За счёт чётких правил присоединения, к основе может добавляться довольно много аффиксов. Помимо этого, к слову может быть присоединено одно подчинённое слово. Так, например, в одно слово можно сказать *kitap_lar_ım_da_ki_ler_i*: *те (вин. падеж), что лежат на моих книгах*. В результате количество форм у одного слова может доходить до нескольких тысяч, при том, что их анализ человеком не затрудняется. Хранить подобные объёмы становится затруднительно.

Наконец, сущим бедствием могут стать полисинтетические языки, в которых несколько членов предложения могут склеиваться в одно слово. *Mamihlapinatapai* — слово из яганского языка (племя Яган, Огненная Земля), указано в книге рекордов Гиннесса в качестве «наиболее сжатого слова» и считается одним из самых трудных для перевода слов. Оно означает «*Взгляд между двумя людьми, в котором выражается желание каждого в том, что другой станет инициатором того, чего хотят оба, но ни один не хочет быть первым*». Слово состоит из рефлексивного / пассивного префикса *ма-* (*MAM*-перед гласным), корень *ihlapi*, что значит быть в недоумении, как то, что делать дальше, то *stative* суффикса *-н*, достижение Суффикс *-ate*, и двойной

суффикс *-apai*, который в составе с рефлексивным *mat-* есть взаимные чувства. Пример восьмипорядковой деривации в эскимосском языке: *igdlo_ssua_tsia_lior_fi_gssa_liar_qu_gamiuk* (*дом-большой-довольно-изготавливать-место-быть-идти-велеть-когда-он-его*), «*Велев ему пойти туда, где строился довольно большой дом*».

Таким образом, методы хранения словаря могут зависеть от того, со словарём какого языка мы работаем. Ниже мы рассмотрим несколько разных вариантов хранения.

2.1.5 Морфологическая омонимия

В примере со словом «косой», мы увидели, что одному и тому же токену может соответствовать несколько разных словоформ. Подобное явление называется **лексической неоднозначностью**. Лексическая неоднозначность включает в себя несколько явлений. **Омонимия** — это явление, при котором два слова сходны по написанию и звучанию, но различные по смыслу. Различают **лексическую** (совпадение форм разных лексем), **грамматическую** (совпадение токенов одной лексемы), **синтаксическую** (различные корректные трактовки одной и той же последовательности слов) и другие виды омонимии. В нашем случае *песчаная* *коса*, *косая* *коса* и *косой* *косой* в соответствующих формах будут именно омонимами. Лексическая неоднозначность, помимо омонимии, включает в себя также различные варианты анализа слова в рамках одной лексемы: токен *косой* может быть проанализирован одновременно как именительный падеж мужского рода и творительный падеж женского рода в рамках одной лексемы ⟨*косой, прилагательное*⟩.

Снятие (или **разрешение**) **омонимии** — этап анализа текста, на котором проводится выбор единственного варианта морфологического анализа для каждого токена.

Если словарь не содержит данный токен, будем называть этот токен несловарным. Если с данным токеном связано более одной словоформы, будем называть такой токен неоднозначным. Как уже было сказано выше, словоформа содержит в себе лемму, часть речи и набор грамматических параметров. Следовательно, неоднозначность может проявить себя в одной

из этих частей или их комбинации. Помимо несловарных и однозначных слов введём для токенов ещё четыре класса омонимии.

- **Неоднозначные по параметрам** — в анализе присутствуют словоформы с различными множествами грамматических параметров, но совпадающей леммой и частью речи. Например, прилагательное *косой* может выражать как именительный падеж мужского рода, так и творительный падеж женского рода.
- **Неоднозначные по части речи** — в анализе присутствуют словоформы, совпадающие по лемме, но отличающиеся по части речи. Так как части речи не совпадают, то наборы параметров у словоформ также будут отличаться. В связи с этим сравнение параметров проводиться не может. Например, прилагательное и существительное *раненый*.
- **Неоднозначные по лемме** — в анализе присутствуют словоформы, отличающиеся по лемме, но имеющие одинаковую часть речи. Здесь параметры могут как совпадать, так и отличаться. Примером совпадающих параметров является токен *смели* — третье лицо единственного числа от лемм *сметь* и *смолоть* (ударение у токенов будет различаться, но мы не увидим этого в тексте). Примером различающихся параметров будет являться токен *вина*, соответствующий именительному падежу единственного числа леммы *вина* и именительному падежу множественного числа леммы *вино*.
- **Неоднозначные по части речи и лемме** — в анализе присутствуют словоформы, отличающиеся как по лемме, так и по части речи. Сравнение параметров здесь также проводиться не может. Примером здесь может служить токен *стекло*, соответствующий именительному падежу единственного числа существительного *стекло* и третьему лицу среднего рода прошедшего времени глагола *стечь*.

2.1.6 Подходы к морфологическому анализу

В зависимости от постановки задачи, морфологический анализатор может возвращать разную информацию. Если нам необходимо рассчитать вектор частотности употребления слов в тексте, это может быть только лемма или лемма и часть речи. Для языков с невысокой флективностью

применяется **стемминг** — определение основы слова путём отбрасывания окончаний из известного набора (возможно, псевдоосновы за счёт отбрасывания псевдоокончаний). Отличие стемминга от лемматизации можно продемонстрировать на примере слова *выходцы*: процесс лемматизации определит лемму *выходец*, в то время как стемминг вернёт псевдооснову *выход*.

По функциональным возможностям морфологические процессоры делятся на несколько видов:

- выполняющие только лемматизацию или стемминг,
- определяющие часть речи,
- осуществляющие полный морфологический анализ, т. е. лемматизацию и определение всех морфологических характеристик словоформы.
- осуществляющие морфемный анализ, выделяющие морфы, входящие в состав слова.

Введение в морфоанализатор функции морфемного разбора расширяет его применимость. Например, с помощью данной функции может быть реализован поиск семантически близких слов разных частей речи (однокоренных), что может быть полезно в ряде задач компьютерной лингвистики, включая задачу распознавания конструкций по шаблонам.

Первые морфологические анализаторы русского языка были простыми и практически не использовали словарной информации. С ростом вычислительных мощностей и объёмов оперативной памяти, а также появлением новых алгоритмов и структур данных стало возможно эффективное использование больших словарей, что значительно улучшило качество реализуемой модели. Далее мы проведём обзор различных подходов к морфологическому анализу.

Бессловарная морфология

Бессловарные морфологии являются одним из первых подходов к решению задачи морфоанализа русского языка. Однако бессловарными они являются лишь условно: в них отсутствуют большие словари лексических единиц, но фактически используются небольшие словари с информацией о флексии. Такой словарь можно представить в виде таблицы — см. таблицу 2.3. В первом столбце содержатся флексии словоформ, во втором —

флексия нормальной формы, а в третьем — морфологические характеристики, соответствующие исходной словоформе с данной флексией. Конечно, из данной таблицы можно удалить второй столбец, договорившись, что первая запись содержит в себе псевдоокончание начальной формы.

Таблица 2.3. Таблица флексий

Флексия	**Флексия нач. формы**	**Морф. характеристики**
-онок	-онок	СУЩ., неод., м. р., ед. ч., им. п.
-онока	-онок	СУЩ., неод., м. р., ед. ч., род. п.
-оноку	-онок	СУЩ., неод., м. р., ед. ч., дат. п.
-оноком	-онок	СУЩ., неод., м. р., ед. ч., твор. п.
...	...	...

Для бессловарной морфологии анализ сводится к поиску наиболее длинного окончания анализируемой словоформы в этом словаре и выборке соответствующих морфологических характеристик, например:

$$позв\ \underbrace{\text{-}онок}_{\text{окончание}} \rightarrow СУЩ.,\ неод.,\ м.\ р.,\ ед.\ ч.,\ им.\ п.$$

Существенным плюсом здесь является то, что определение морфологических характеристик и нормальной формы возможно практически для любой словоформы, для которой нашлась подходящая флексия. Однако такой подход зачастую оказывается неточным из-за наличия большого количества исключений в языке.

Словарная морфология на основе словаря основ

Одним из классических подходов к морфологическому анализу русского языка является построение словарной морфологии на основе словаря основ [9]. Основой модели выступает словарь основ, который содержит все (псевдо)основы лексем языка. Он связан со вспомогательными словарями, в которых содержится список флексий всех словоизменительных классов, для каждой из которых указан набор значений морфологических характеристик, которые она может выражать. Также, как правило, хранится дополнительная информация об особенностях словоизменения, например,

о чередовании букв в основах или беглых гласных. Зачастую присутствует дополнительный словарь исключительных случаев.

Разбор словоформ происходит по следующей схеме.

1. Последовательно отсекаются возможные окончания длиной от 0 до n букв, таким образом слово разбивается на основу и флексию.
2. Для полученного окончания находится его словоизменительный класс.
3. Проверяется наличие полученной основы в словаре основ и находится номер её словоизменительного класса.
4. В случае совпадения словоизменительных классов выбираются соответствующие морфологические характеристики и строится лемма, которые и являются результатом анализа.

Словарная морфология на основе словаря словоформ

Для высокофлективного языка наиболее частым подходом к решению задачи морфологического анализа является словарная морфология на основе словаря словоформ [9]. База морфологического процессора, построенного на такой морфологии, заключается в создании словаря всех форм языка, который может быть представлен в виде таблицы — см. пример в таблице 2.4.

Таблица 2.4. Простейшая таблица словоформ

Словоформа	**Нач. форма**	**Морфологические характеристики**
АБАТ	АБАТ	СУЩ., од., м. р., им. п.
АБАТА	АБАТ	СУЩ., од., м. р., род. п.
АБАТУ	АБАТ	СУЩ., од., м. р., дат. п.
АБАТОМ	АБАТ	СУЩ., од., м. р., твор. п.
...	...	...

Определение нормальной формы и морфологических характеристик сводится к поиску словоформы в таблице. Синтез словоформы, похожим образом, сводится сначала к поиску нормальной формы, а затем формы, соответствующей запрошенным морфологическим характеристикам.

Значительной проблемой всех словарных морфологий, особенно на основе словаря словоформ, является структура хранения словаря, который

может занимать несколько гигабайт. Однако, при использовании эффективной структуры данных, даже более обширный словарь словоформ становится не только хорошей теоретической моделью, но и удобным практическим средством. Достоинствами являются возможность выполнения как анализа, так и синтеза. К основным недостаткам словарных морфологий, относятся: проблема анализа слов, которых нет в словаре, необходимость качественного и объёмного словаря, проблема морфологической омонимии.

2.1.7 Морфемный разбор

Морфемный разбор является плохо изученной задачей в области компьютерной лингвистики, при этом работы для русскоязычных текстов практически отсутствуют. Однако, как отмечалось выше, введение функции морфемного разбора может значительно расширить приложения морфологического процессора, особенно для такого высокофлективного языка как русский.

Задача морфемного разбора слов заключается в разбиении слова на упорядоченный набор морфов. Например:

$$impossible \rightarrow \underbrace{im}_{\text{приставка}} - \underbrace{poss}_{\text{корень}} - \underbrace{ible}_{\text{суффикс}}$$

В настоящий момент наиболее известными подходами к автоматизированному морфемному разбору являются: метод Харриса [31], метод Дежона [29], метод Бернхард [20], а также метод, реализованный в системе Morfessor [41]. Каждый из них относится к методам, не требующим размеченных данных, и целиком опирается на информацию из неразмеченных корпусов.

Метод Харриса [31] базируется на простой идее подсчета количества различных букв в словах корпуса или словаря, идущих после различных начальных частей слова и перед конечными частями слова. На рисунке 2.1 приведен пример такого подсчета: в верхней строке находятся количества различных букв в словах словаря, идущих после начальной части данной длины, в нижней находятся количества перед конечной частью. Разбиение

слова на морфемы происходит с помощью нахождения пиков (локальных максимумов) в каждом из рядов. В том месте, где обнаружен пик, и находится граница морфем слова.

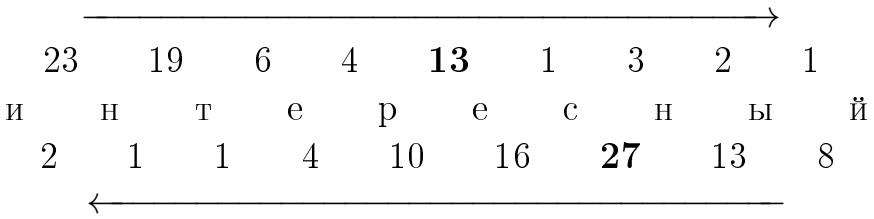


Рис. 2.1. Количество различных букв с аффиксами слова

Данный метод требует достаточно большого корпуса слов и является неточным для высокофлективных языков, таких как русский.

Метод Дежона [29] является расширением метода Харриса и состоит из трёх этапов: поиск изначальных морфем, поиск дополнительных морфем, сегментация слов. На первом этапе алгоритм использует метод Харриса, однако происходит не поиск пиков в количестве различных последующих букв, а нахождение значений, больше определенного порога (половина длины алфавита). На втором этапе происходит дополнение списка морфем: с помощью набора, обнаруженного на первом этапе, от слова отделяют известную морфему и если оставшаяся часть встречается с другим аффиксом более n раз, то этот аффикс также заносится в список морфем. Таблица 2.5 иллюстрирует этап 2 на примере слов с корнем *light*.

После первых двух этапов список морфем построен и происходит сегментация слова с помощью жадного алгоритма: в наборе морфем ищется суффикс или префикс максимальной длины, который совпадает с окончанием или началом слова соответственно.

Данный метод существенно зависит от исходного корпуса слов. Основным недостатком метода для высокофлективных языков является проблема чрезмерного разделения полученного аффикса на набор морфем меньшей длины (*oversegmentation*). Как отмечают авторы, они избегали такого разделения из-за наличия однобуквенных морфем, иначе слово превращалось бы в набор однобуквенных морфем.

Метод Бернхард [20] также базируется на методе Харриса и имеет несколько дополнительных шагов для уточнения результата. На первом

Таблица 2.5. Пример второго этапа алгоритма Дежона

Известные морфемы	Слова	Новые морфемы
	light	
-s	lights	
-ed	lighted	
-ing	lighting	
-ly	lightly	
-er	lighter	
	lightness	-ness
	lightest	-est
	lighten	-en

этапе применяется метод Харриса для получения списка всех суффиксов и префиксов из входных слов. На втором этапе происходит выделение основ слов с помощью простого перебора всех возможных аффиксов слова, полученных различными комбинациями из аффиксов, выявленных на первом шаге. Набор основ в таком случае получается очень большим, поэтому используются некоторые эвристики, например, предположение о том, что длина основы не может быть менее трех символов. На третьем этапе происходит сегментация слова на морфы с помощью сравнения слов с одинаковыми основами. Сравнение заключается в нахождении границ между общими и различающимися частями слова. После третьего шага для одного слова может быть обнаружено несколько различных разборов. Для выбора наилучшего используется жадный алгоритм *поиск «лучший-первый»*. В случае выбора между несколькими аффиксами выбирается наиболее частотный. Также применяются простые эвристики, например, префикс не может идти после суффикса.

Метод был протестирован на нескольких языках, в результате были получены оценки F-меры в пределах от 24 до 60 процентов.

Метод системы Morfessor [41] также использует статистику, собранную по неразмеченному корпусу, однако не базируется на методе Харриса и основывается на алгоритме машинного обучения. Алгоритм пытается найти баланс компактного описания словаря морфов и в то же время, компактного разбиения слов корпуса с помощью этих морфов. Первая версия Morfessor основывалась на рекурсивном алгоритме MDL [28] и была кон-

текстно независимой. Основными недостатками данной версии были проблемы недоразбиения (*undersegmentation*) частотных морфов и чрезмерного разбиения редких морфов. Для устранения этих недостатков авторы использовали скрытые марковские модели [19], с помощью которых происходит учёт контекста ближайших слов.

Morfessor был протестирован [23] на английском, финском и турецком языках. Наилучший результат был достигнут на корпусе турецких слов, F-мера составила около 70%.

2.2 Обзор модулей морфологического анализа

В данном разделе будут рассмотрены наиболее популярные морфологические процессоры русского языка, которые доступны для открытого использования. Данные проекты были участниками соревнований по морфологическому анализу русского языка [10] и нередко используются в практических приложениях [12, 1]. Морфопроцессоры рассматривались как с точки зрения функциональных возможностей: стемминг, лематизация, полный морфологический анализ, синтез словоформ, снятие омонимии, так и технологических характеристик, таких как размер словаря, открытость исходных кодов, используемая модель морфологии и др.

2.2.1 Система Диалинг–АОТ

Проект АОТ [1] включает в себя практически все этапы автоматического анализа текстов на ЕЯ, в том числе и морфологический. Проект разрабатывался при поддержке группы лингвистов из РГГУ, основным разработчиком выступал А. Сокирко. Морфологический процессор был выложен в открытый доступ в 2004 году. Лексической основой служит словарь Зализняка [6].

Морфологический процессор АОТ [17] предоставляет все функции полного морфологического анализа, включая нахождение леммы и морфологических характеристик словоформы, а также синтез словоформ.

АОТ базируется на словарной морфологии, в качестве словаря используется русский грамматический словарь А.А.Зализняка, который включает более 161 тыс. лемм. Логическая структура компьютерного словаря представляется в виде нескольких таблиц: лемм, флексий, приставок, морфологических характеристик. Таблица лемм содержит список псевдооснов слов со ссылками на таблицы флексий и приставок. В таблице флексий каждому из окончаний слов соответствует ссылка на соответствующие морфологические характеристики. Морфологический разбор слова по словарю состоит в поиске максимально совпадающей псевдоосновы в таблице лемм, поиск соответствующих приставки и окончания, а затем нахождение по таблице окончаний соответствующих морфологических характеристик. Синтез осуществляется похожим образом: после нахождения псевдоосновы по морфологическим характеристикам определяется вся парадигма и возвращается пользователю.

Для физического (бинарного) представления словаря используется структура конечного автомата [27]. Стоит отметить, что хранение морфологических характеристик сильно увеличивает число состояний автомата и, как следствие, время поиска в нем, поэтому в АОТ характеристики находятся в отдельной таблице, а сам автомат хранит ссылки на них. Итоговый размер словаря составляет около 9 МБ, что является небольшим значением для современных компьютеров.

Если словоформа не была найдена в словаре, то в этом случае в АОТ работает морфологическое предсказание. Первым шагом предсказания является попытка найти существующую словоформу языка, которая имела бы максимально общее окончание со входным словом. Если при этом длина левой (неузнанной) части слова не превышает определенного размера (5 символов), а длина общего окончания со словарной словоформой не меньше 4 символов, тогда слово предсказывается по найденной правой части, т. е. берутся морфологические характеристики найденной словоформы. Если же такой подход не сработал, то ищется наиболее длинное совпадающее окончание. Пример такого анализа показан на рисунке 2.2.

$$куз\underbrace{явые}[?] \longrightarrow кор\underbrace{явые}[\text{П,мн,им}]$$

Рис. 2.2. Распознавание неизвестного слова

В настоящий момент морфопроцессор проекта АОТ является полностью открытым и распространяется под лицензией LGPL. Однако проект не поддерживается и не имеет удобных средств для пополнения словаря.

2.2.2 Система TreeTagger

TreeTagger [37] позиционируется как система для определения частей речи слов с возможностью настройки на любой естественный язык при наличии словаря и размеченного корпуса. Она была разработана в 1996 году в университете Штутгарта Хельмутом Шмидтом, и на данный момент доступна для множества языков, включая русский.

Процессор TreeTagger позволяет определять части речи слов и другие морфологические характеристики, а также их нормальную форму. Основной упор в данном процессоре сделан на разрешение морфологической омонимии и предсказание характеристик неизвестных слов.

TreeTagger базируется на словарной морфологии и использует словарь английского языка из проекта Penn TreeBank, содержащий более 2 млн. словоформ. Объем русского словаря неизвестен, однако по объему бинарного файла можно судить о сопоставимости размера словаря с английской версией. В процессе анализа используются 2 вида словарей: словоформ и суффиксов (имеются в виду флексии). Структуры данных, используемые для словарей, похожи на те, что используются в проекте АОТ, и также являются вариациями минимальных конечных автоматов. Автомат суффиксов (флексий) строится из всех флексий слов длиной до пяти символов. При этом каждому из суффиксов приписывается соответствующая флективная часть речи на основе взвешенной энтропии Шенона. При этом узлы, имеющие значение энтропии меньше определенного порога, удаляются из автомата.

Для снятия частеречной омонимии в TreeTagger используются решающие деревья [35] для частей речи, обученные на размеченном корпусе.

В узлах такого дерева находятся предикаты с ответом «да» или «нет» для двух предшествующих слов. При этом в листьях хранятся значения вероятностей для возможных ответов. Построение дерева происходит рекурсивно, с помощью модифицированного алгоритма ID3. На каждом шаге для двух предыдущих слов проверяются предикаты на равенство всем возможным частям речи, при этом для определения предиката, наилучшим образом разбивающего пространство признаков, используется правило максимизации энтропии Шенона. Для определения части речи входного слова достаточно, используя информацию о предыдущих словах, пройти по дереву от корня до листьев и выбрать наиболее вероятное значение.

В настоящий момент TreeTagger распространяется в виде бинарного файла, код самого процессора является закрытым. Проект поддерживается, для него создаются новые словари под различные языки. Синтез словоформ в TreeTagger отсутствует.

2.2.3 Система Pymorphy2

Pymorphy2 [33] — морфологический процессор с открытым исходным кодом, предоставляет все функции полного морфологического анализа и синтеза словоформ.

Процессор базируется на словарной морфологии и использует словарные данные проекта OpenCorpora [13, 21]. Словарь содержит около 250 тыс. лемм, а также является полностью открытым и регулярно пополняемым. Словарь, как и в проекте АОТ, логически представляет собой структуру из трёх таблиц, однако словарные данные хранятся в едином автомате. Для бинарного представления используется автомат с оптимизацией по памяти [25], что позволяет иметь в нем не более чем 2^{32} различных связей, однако для задачи морфологического анализа данное ограничение не является существенным. Итоговый размер словаря составляет около 7 МБ.

В процессе морфологического синтеза, по исходной словоформе и тегам выполняется поиск нормальной формы слова, а затем перебор всех возможных пар $\langle$*окончание, теги*$\rangle$ в найденной лексеме, пока не будет найдена пара с заданными морфологическими тегами. После этого от нормальной

формы отсекается её окончание, а найденное окончание приписывается к полученной псевдооснове.

Для анализа неизвестных слов в Pymorphy2 используются несколько методов, которые применяются последовательно. Изначально от слова отсекается префикс из набора известных префиксов и если остаток слова был найден в словаре, то отсеченный префикс приписывается к результатам разбора. Если этот метод не сработал, то аналогичные действия выполняются для префикса слова длиной от 1 до 5, даже если такой префикс является неизвестным. Затем, в случае неудачи, словоформа разбирается по окончанию. Для этого используется дополнительный автомат всех окончаний, встречающихся в словаре с имеющимися разборами. В процессе построения из автомата удаляются редкие окончания и разборы. Метод анализа по окончанию аналогичен тому, что используется в процессоре АОТ.

Разрешение мофромонимии построено на основе корпусной статистики. Если слово имеет несколько вариантов разбора, то среди всех выбирается наиболее вероятный. Вероятности определяются по следующей формуле:

$$P(w|t) = \frac{Fr(w,t)+1}{Fr(w)+|R(w)|}.$$

В приведенной формуле, $Fr(w)$ — количество раз, которое словоформа w встретилась в корпусе, а $Fr(w,t)$ — количество раз, которое эта словоформа встретилось с тегом t. $|R(w)|$ — число разборов, полученных от анализатора для словоформы w.

В настоящее время Pymorphy2 поддерживается, при этом происходит постоянное пополнение корпуса OpenCorpora, что улучшает характеристики точности и полноты морфологического разбора.

2.2.4 Система Snowball

Данный стеммер разработан Мартином Портером и опубликован в 1980 году [32]. Snowball использует систему суффиксов и окончаний для предсказания части речи и грамматических параметров. Так как одно и то же окончание может принадлежать разным частям речи или различ-

ным парадигмам, его оказывается недостаточно для точного предсказания. Применение суффиксов позволяет повысить точность.

Система реализовывается на языке программирования в виде большого количества условных операторов, анализирующих самый длинный постфикс и его контекст. По окончании анализа слову приписывается часть речи и набор параметров, а найденное окончание (или псевдоокончание) отрезается. В итоге, помимо параметров, система возвращает стем.

Система реализована на многих языках программирования и распространяется в исходных кодах, что позволяет легко встраивать ее в новые системы. Она не требует никакого словаря, однако расширение и уточнение правил выделения окончаний может оказаться нетривиальной задачей. Точность работы данного стеммера находится на уровне около 80%. Заметим, что использование методов машинного обучения, применённых к размеченному корпусу, позволяет получить гораздо лучшие результаты.

2.2.5 Система MyStem

MyStem — морфологический анализатор, разработанный компанией Яндекс [11]. Первая версия [39] была создана в 90-х годах, однако не имела большой популярности и не находилась в открытом доступе. Стоит отметить, что первая версия предполагала использование словаря небольшого размера, опираясь в основном на методы бессловарной морфологии, в то время как текущие реализации базируются на классическом подходе словарной морфологии.

В настоящий момент MyStem версии 3.0 предоставляет все функции полного морфологического анализа, однако не имеет функции синтеза. Данная версия является наиболее стабильной и доступной для скачивания в бинарном виде.

Морфоанализатор MyStem базируется на словаре НКРЯ [12], который содержит более 200 тыс. лемм. Исходные коды MyStem являются закрытыми, поэтому характеристики использованной структуры данных не известны, однако размер полученного бинарного словаря более 20 МБ.

MyStem производит разрешение морфологической омонимии и делает разбор несловарных словоформ. Для решения этой задачи используются

различные методы машинного обучения. В зависимости от входных данных MyStem снимает омонимию двумя способами: с учетом контекста и без учета контекста [11].

Снятие омонимии без учета контекста происходит благодаря обучению наивного баесовского классификатора на размеченном корпусе со снятой омонимией. Частоты встречаемости факторизируются и отдельно настраиваются для окончаний морфологических парадигм, основ парадигм и самих парадигм. Вероятность принадлежности неизвестного слова $word$, имеющего основу $stem$ и окончание $flex$, к парадигме $para$ рассчитывается по формуле [11]:

$$P(para|word) = \frac{P(word|para) \cdot P(para)}{P(word)} =$$

$$= \frac{P(stem|para) \cdot P(flex|para) \cdot P(para)}{P(word)}.$$

При этом предполагается, что $stem$ и $flex$ являются независимыми случайными величинами.

Контекстное снятие омонимии является подключаемым и использует технологию MatrixNet. Основной идеей является ранжирование разборов на основе ближайших к разбираемому слов (контекстов).

В настоящее время MyStem поддерживается и используется в ряде проектов, таких как НКРЯ. Также он доступен в виде динамической библиотеки для некоммерческих приложений и позволяет подключать собственные словари через опции командной строки или интерфейса библиотеки. В этом случае стандартный словарь полностью заменяется пользовательским.

2.2.6 Сравнение систем морфологического анализа

В таблице 2.6 приведено сравнение характеристик рассмотренных морфологических процессоров.

Все морфопроцессоры предоставляют наиболее важную для русского языка функцию лемматизации словоформ, при этом со снятием омонимии. Данная функция реализуется и для несловарных слов. Функция стеммин-

Таблица 2.6. Характеристики морфологических процессоров

Система	АОТ	**MyStem**	**TreeTagger**	**Pymorphy2**
Открытые исходные коды	да	нет	нет	да
Скорость, слов в секунду	60-90 тыс.	100-120 тыс.	20-25 тыс.	80-100 тыс.
Подключение словарей	нет	да	да	нет
Объем словаря, тыс. слов	160	>250	210	250

га является менее популярной в реализациях, т. к. менее востребована на практике, однако все процессоры, кроме TreeTagger, предоставляют возможность получения словоизменительной парадигмы заданной словоформы, а с её помощью достаточно просто получить псевдооснову слова. Морфологический синтез также реализован лишь в двух из рассмотренных процессоров, хотя во многих задачах компьютерной лингвистики данная функция является важной.

Два из представленных процессоров являются закрытыми и распространяются исключительно в виде бинарных файлов. Словарь MyStem является закрытым, словарь TreeTagger доступен в виде бинарного файла. Скорость обрабатываемых слов у всех процессоров является достаточно высокой. Как правило, существенное замедление обработки наблюдается на более поздних этапах анализа ЕЯ, поэтому скорость морфопроцессора редко становится узким местом. Возможность подключения словаря является особенно важной для задач ограниченных предметных областей. Данную функцию предоставляет MyStem.

Существенной проблемой, связанной с морфологическими процессорами, является использование собственной системы морфологических тегов в каждом из них. Из-за несоответствия морфологических тегов сложно сравнивать работу процессоров, оценивать их точность и полноту на размеченных корпусах. Решением данной проблемы мог бы быть универсальный конвертер из одной системы тегов в другую, который отсутствует во всех рассмотренных анализаторах.

2.3 Методы хранения словарей

2.3.1 Форматы входных и выходных данных

На данный момент существует несколько форматов морфологической разметки. Один из них, формат TEI (Text Encoding Initiative), основывается на применении формата XML [40]. Данный формат регламентирует разметку коллекций разной природы и направленности, от разделения предложения на слова до расшифровки фотографий рукописей с учётом всех особенностей повреждений бумаги, на которой написана рукопись [8].

Для русского языка формат был доработан разработчиками Национального корпуса русского языка (НКРЯ)[12]. Пример разметки показан на рис. 2.3. На рисунке использованы теги для текста, параграфа, предложения, словоупотребления, варианта анализа слова. В случае разметки без снятия омонимии, тег <ana> может повторяться несколько раз внутри одного словоупотребления.

```
<?xml version="1.0" encoding="windows-1251" ?>
<text> <p> <s>
<w> Примерная<ana lex="примерная" pos="A" gr="m,sg,nom"></w>
<w> разметка<ana lex="разметка" pos="S" gr="f,inan,sg,nom"></w>
<w> текста<ana lex="пример" pos="S" gr="m,inan,sg,gen"></w>
</s> </p> </text>
```

Рис. 2.3. Пример разметки текста в формате НКРЯ

Формат CoNLL (Computational Natural Language Learning — конференция и серия соревнований в ее рамках) использует формат TSV (tab separated values), в котором каждое слово представляет собой одну строку, части которой разделены символами табуляции [22]. В состав строки входит идентификатор слова, токен, лемма, часть речи и набор тегов. В формате CoNLL-U набор тегов представляется одновременно в позиционной и списковой формах в модифицированном формате MULTEXT [42]. Пример записи текста показан на рис. 2.4.

Предшественником формата CoNLL можно считать формат выдачи анализатора CLAWS. Данный анализатор выдавал результат в «вертикаль-

1	Då	då	ADV	AB	_
2	var	vara	VERB	VB.PRET.ACT	Tense=Past|Voice=Act
3	han	han	PRON	PN.UTR.SIN.DEF.NOM	Case=Nom|Definite=Def| ...
4	elva	elva	NUM	RG.NOM	Case=Nom|NumType=Card
5	år	år	NOUN	NN.NEU.PLU.IND.NOM	Case=Nom|Definite=Ind| ...
6	.	.	PUNCT	DL.MAD	_

Рис. 2.4. Пример разметки текста в формате CoNLL-U

ном» и «горизонтальном» форматах. Примеры для них приведены на рис. 2.5. Заметим, что горизонтальная разметка очень часто используется в статьях для написания примеров.

0000003	010	The	AT
0000003	020	quick	[JJ/99] RR@/1 NN1%/0
0000003	030	brown	[JJ/93] NN1@/7 VV0%/0
0000003	040	fox	[NN1/100] VV0@/0
0000003	050	jumps	[VVZ/97] NN2@/3
0000003	060	over	[II/59] RP/41 NN1%/0 JJ%/0
0000003	070	the	AT
0000003	080	lazy	JJ
0000003	090	dog	[NN1/100] VV0%/0
0000003	091	.	.

The_AT quick_JJ brown_JJ fox_NN1 jumps_VVZ over_II the_AT lazy_JJ dog_NN1 ._.

Рис. 2.5. Пример «вертикальной» и «горизонтальной» разметки текста в системе CLAWS [34]

Не следует забывать, что часть систем выдаёт результаты анализа в виде структур. Так, например, система Pymorphy возвращает результаты в виде списков и объектов языка Python, которые могут быть выведены в любой из перечисленных выше нотаций.

Для хранения словарей используется три основных формата: XML, TSV и бинарное представление файла. Их формат отличается от выдачи анализатора или морфологической разметки корпуса. Так, например, в словаре OpenCorpora каждый грамматический параметр представлен в виде отдельного тега, а сами теги задаются в виде списка (см. рис. 2.6a). Система Morphalou использует как открывающие, так и закрывающие теги XML (см. рис. 2.6b). На рисунках хорошо видно, что оба словаря отдельно хранят грамматические параметры, присущие слову в целом (например,

род для существительного или саму часть речи), и параметры форм слов (число, падеж, время и др.).

```
<lemma id="213937">
  <l t="отказав">
    <g v="GRND"/>
    <g v="perf"/>
    <g v="intr"/>
  </l>
  <f t="отказав">
    <g v="past"/>
  </f>
  <f t="отказавши">
    <g v="past"/>
    <g v="V-sh"/>
  </f>
</lemma>
```

(a)

```
<lexicalEntry id="aal\'enien_1">
<formSet>
<lemmatizedForm>
<orthography>aal\'enien</orthography>
<grammaticalCategory>commonNoun</grammaticalCategory>
<grammaticalGender>masculine</grammaticalGender>
</lemmatizedForm>
<inflectedForm>
<orthography>aal\'enien</orthography>
<grammaticalNumber>singular</grammaticalNumber>
</inflectedForm>
<inflectedForm>
<orthography>aal\'eniens</orthography>
<grammaticalNumber>plural</grammaticalNumber>
</inflectedForm>
</formSet>
<originatingEntry target="TLF">AAL\'ENIEN, IENNE, adj.,
et subst. masc.</originatingEntry>
</lexicalEntry>
```

(b)

Рис. 2.6. Фрагмент словаря OpenCorpora (a) и Morphalou (b) в xml-формате

В формате tsv хранят свои словари такие системы, как Freeling (рис. 2.7a) и Polimorphologic (рис. 2.7b). Здесь уже значительно сложнее хранить общие грамматические параметры отдельно. В связи с тем, что словарь Polimorphologic отсортирован по токенам, а не по леммам, отдельные формы одной и той же лексемы могут находиться в разных частях файла словаря. В связи с этим каждая форма слова должна хранить всю необходимую информацию.

Наконец, такие словари, как PyMorphy, AOT и TreeTagger поставляются со словарём в бинарном файле. Такой файл гораздо проще и быстрее загружать в оперативную память, однако он не подходит для редактирования словаря целиком. Для этих целей используются функции API или конвертор из текстового файла в одном из описанных форматов.

Заметим, что практически каждый словарь использует свою собственную языковую морфологическую модель. В связи с этим одновременное использование нескольких словарей затруднено и требует определённой кон-

```
abarbeite abarbeiten VVSP3S
abarbeiten abarbeiten VVIP1P
abarbeiten abarbeiten VVIP3P
abarbeiten abarbeiten VVN000
abarbeiten abarbeiten VVSP1P
abarbeiten abarbeiten VVSP3P
abarbeitend abarbeiten VVP000
abarbeitest abarbeiten VVIP2S
abarbeitest abarbeiten VVSP2S
abarbeitet abarbeiten VVIP2P
```

(a)

```
Aalborg Aalborg subst:sg:acc:m3
  +subst:sg:nom:m3
Aalborgach Aalborg subst:pl:loc:m3
Aalborgami Aalborg subst:pl:inst:m3
Aalborgi Aalborg subst:pl:acc:m3
  +subst:pl:nom:m3+subst:pl:voc:m3
Aalborgiem Aalborg subst:sg:inst:m3
Aalborgom Aalborg subst:pl:dat:m3
Aalborgowi Aalborg subst:sg:dat:m3
Aalborg\'ow Aalborg subst:pl:gen:m3
Aalborgu Aalborg subst:sg:gen:m3
  +subst:sg:loc:m3+subst:sg:voc:m3
```

(b)

Рис. 2.7. Фрагмент словаря Freeling (a) и Polimorphologic (b) в формате tsv

вертации. Так, если вторые дательный и родительный падежи приводятся к единственным достаточно просто, то преобразование причастия и деепричастия к форме соответствующего глагола может потребовать серьёзной работы.

Заметим, что в большинстве случаев разметка отображает примерно одну и ту же информацию. Если изменения в морфологической языковой модели могут быть критичными и не будут позволять восстановить какую-то информацию, то разметка скорее представляет удобный (или не очень) вариант хранения примерно одного и того же. Так, скажем, сложно (но возможно) восстанавливать из файла в формате tsv информацию о лексемах, но с точки зрения добавления новых строк в словарь такой формат может оказаться удобнее.

Вообще, морфология служит лишь вспомогательным этапом для последующего анализа текста (или, например, частью этапа поверхностно-синтаксического-анализа [1]). Поэтому очень часто текст размечается один раз (или берётся уже размеченным) и в дальнейшем многократно используется в следующих этапах. Если в коллекции находятся тексты с разметкой в разных форматах, они либо конвертируются в единый формат, либо пишется несколько функций для их загрузки. Ещё одним подходом является использование высокопроизводительных систем для морфологического

анализа[4]. Тогда скорость морфологического анализа становится сопоставимой со скоростью разбора разметки входных файлов и оказывается проще работать с уже неразмеченными файлами. Конечно, подобный подход не применим в случае, когда работа ведется с корпусами, в которых омонимия снята вручную.

2.3.2 Внутреннее представление морфологического словаря

В словарных морфологиях хранение словаря является значительной проблемой. Для словаря словоформ табличное представление, например, в реляционной базе данных, является крайне неэффективным, как по потребляемой памяти, так и по скорости обработки входного слова. Более целесообразным в данном случае выступает формат хранения в виде пар $\langle$*ключ, значение*$\rangle$, где ключом выступает словоформа, а значением соответствующий набор морфологических характеристик. При этом стоит учитывать особенность самих хранимых данных: с точки зрения низкоуровневого представления это последовательности символов фиксированной длины, как в ключах, так и в значениях.

Хранение словаря в виде направленного ациклического графа

Эффективной структурой данных для данной задачи является направленный ациклический граф слов (DAWG) [24], который также называют детерминированным ациклическим конечным автоматом (DAFSA). Словоформы (ключи) с одинаковыми префиксами хранятся вместе, что позволяет существенно сэкономить потребляемую память. Морфологические теги (значения), хранимые в автомате, находятся сразу после словоформ, как правило, за символом-разделителем, который не может встретиться ни в одной из входных форм слова — см. рис. 2.8. На данном рисунке автомат содержит строки «дом» и «дома», которым соответствуют абстрактные пары тегов $\langle v_1, v_2\rangle$ и $\langle v_3, v_4\rangle$.

[4]Скорость работы таких систем колеблется от нескольких сотен тысяч дл нескольких миллионов токенов для морфологической разметки и несколько сотен тысяч токенов для снятия омонимии.

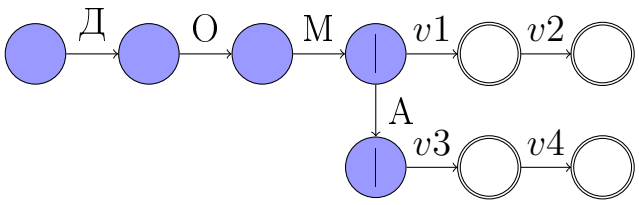


Рис. 2.8. DAWG-автомат, содержащий слова «дом» и «дома»

Поиск словоформы в таком автомате происходит за линейное от длины входной словоформы время: достаточно пройти все состояния автомата, которые соответствуют символам входной словоформы, далее пройти разделительный символ и получить все характеристики словоформы, двигаясь по автомату до достижения конечного состояния.

Сами морфологические характеристики (теги) занимают значительный объем памяти при хранении в виде строк. Решение данной проблемы вытекает из особенности естественных языков — словоформы, относящиеся к одним и тем же парадигмам, имеют одинаковые наборы морфологических характеристик. Всего в словаре OpenCorpora насчитывается около 5 тысяч парадигм, в каждой из которых имеется, в среднем, около 7 различных форм. Таким образом, для однозначного определения морфологических характеристик словоформы достаточно знать номер парадигмы и номер формы в этой парадигме. Поэтому, в CrossMorphy, морфологическая информация хранится отдельно, в виде массива пар номеров парадигм и номеров форм в этой парадигме с конкретными характеристиками, а также таблицы парадигм.

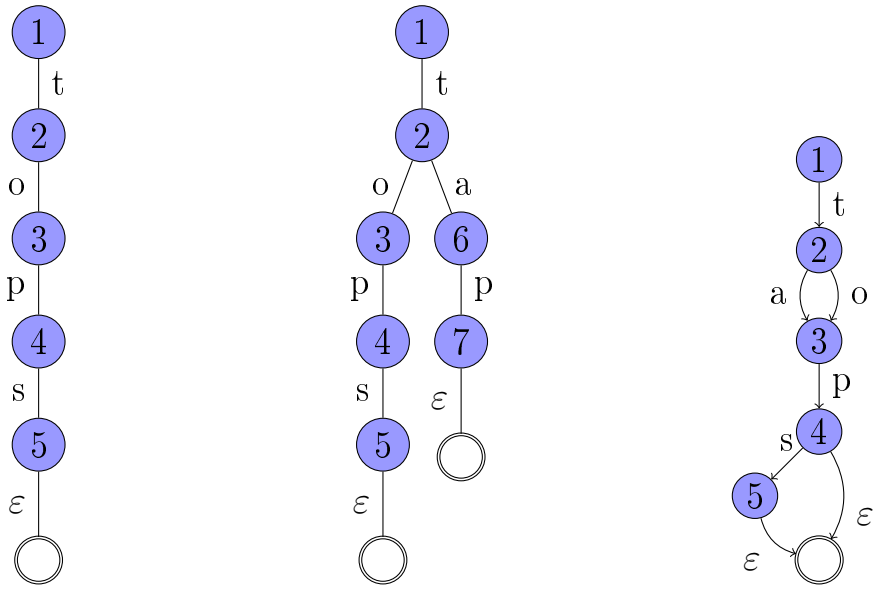
Как уже отмечалось выше, в языке имеется относительно небольшое (на один-два порядка меньшее, чем число лексем, и на три порядка меньше, чем словоформ) количество парадигм. Таким образом, размер автомата можно существенно сократить, если хранить парадигму только один раз. Для этого существует два пути:

- оптимизация автомата с превращением дерева в граф;
- хранение раздельно деревьев псевдооснов и псевдоокончаний.

В случае оптимизации автомата, находится не только самый большой общий префикс, уже имеющийся в графе и начинающийся с начальной вершины, но и самый большой общий постфикс, начинающийся от одной

из листовых вершин. Далее, вместо того чтобы строить дерево, как это показано на рис. 2.8, мы проходим по графу от начальной вершины до конца префикса, создаём путь, соответствующий второй части псевдоосновы, после чего строим дугу, соединяющую последнюю созданную вершину с первой вершиной с найденным постфиксом.

Другим вариантом является собственно оптимизация автомата путём поиска одинаковых путей, ведущих к листовым вершинам, и их объединение. Результат оказывается примерно одинаковым в обоих случаях (см. рис. 2.9).



(a) Исходный минимальный автомат

(b) Добавление нового ключа

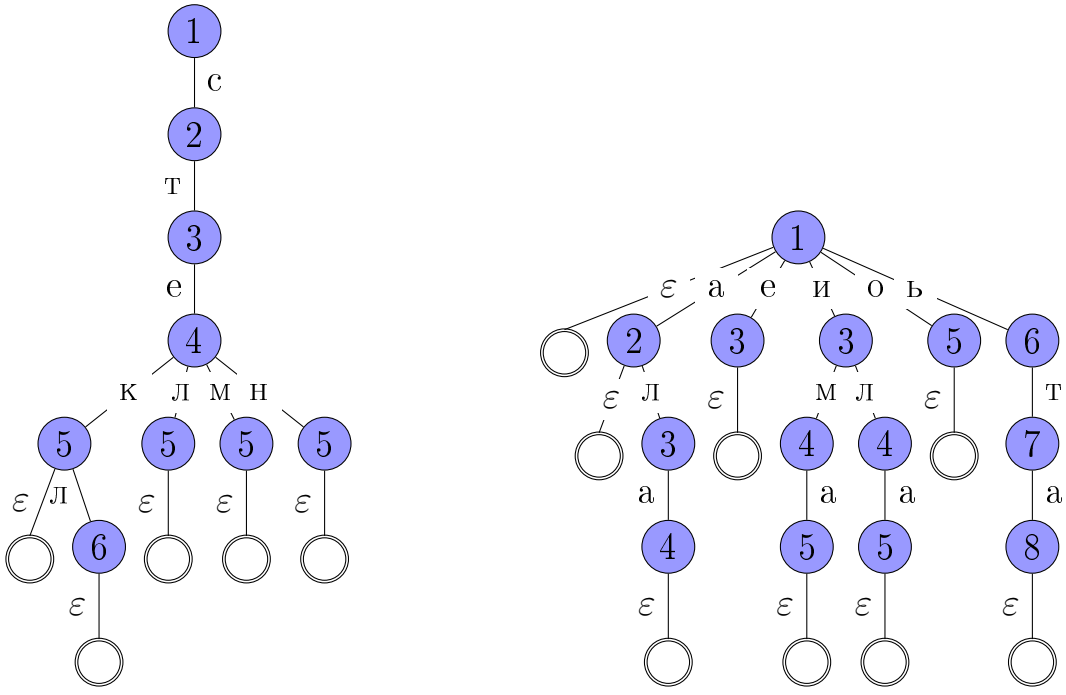
(c) Минимизированный результирующий автомат

Рис. 2.9. Добавление значения в автомат и последующая минимизация

Морфологический анализ с деревьями псевдооснов и псевдоокончаний

Как уже было сказано выше, альтернативой оптимизации дерева является раздельное хранение деревьев псевдооснов и псевдоокончаний. Для простоты будем считать, что слова добавляются по лексемам, содержащим словоформы, в состав которых входит токен, часть речи и набор грамматических параметров. В качестве леммы в этом случае можно использовать токен первой словоформы.

При таком представлении лексемы несложно выделить псевдооснову и парадигму лексемы. Сперва проверяем, есть ли такая парадигма в дереве псевдоокончаний. Если парадигма присутствует, получаем ее идентификатор. При отсутствии парадигмы, добавляем входящие в неё строки, начиная с конца каждой строки. В терминальные вершины добавляем информацию о грамматических параметрах и идентификаторе парадигмы. Добавление строк ведётся так же, как это было показано выше для DAWG-автомата (рис. 2.8). Далее добавляем псевдооснову в соответствующее дерево. К терминальному листу привязываем идентификатор парадигмы. На рис. 2.10 показано дерево псевдооснов (a) и псевдоокончаний (b) для слов *стек*, *стекать*, *стекло*, *стелить*, *стем*, *стена*. Здесь мы предполагаем, что буква *ё* выражается при помощи буквы *е*, парадигмы для слов прописаны не полностью и листовым вершинам приписаны множества, содержащие номера парадигм и списки параметров.



(a) Дерево псевдооснов (b) Дерево псевдоокончаний

Рис. 2.10. Хранение словаря в виде деревьев псевдооснов и псевдоокончаний

Морфологический анализ будет проходить следующим образом. Двигаемся побуквенно по токену, перемещаясь при этом по дереву псевдооснов от корня к листьям. Если переход из текущей вершины по очередной бук-

ве строки отсутствует, то разбор заканчивается. Если достигнута вершина, обозначающая конец псевдоосновы, то информация о парадигме изменения слова переносится в результат с пометкой о длине найденной псевдоосновы.

После анализа псевдоосновы проводится анализ псевдоокончаний. Движемся по токену побуквенно от конца к началу от корня дерева к его листьям. Найдя пометку о конце псевдоокончания, проверяем, заканчивалась ли в той же позиции псевдооснова. Если заканчивалась, проверяем, есть ли совпадающие идентификаторы парадигм у псевдоокончания и псевдоосновы. Если есть, получаем строку начальной формы и отправляем вместе с ней в результат найденные множества грамматических параметров.

Если множество результатов оказалось пустым, возвращаем информацию о неуспешном разборе. В противном случае возвращаем собранное множество результатов.

Морфологический анализ с использованием обычных структур данных

Наконец, для морфологического анализа можно использовать структуры данных, давно уже применяющиеся в программировании. Например, описанные выше примеры хранения словаря описываются с использованием префиксных деревьев [7].

Менее эффективным по скорости и памяти представляется хранение слов в бинарном дереве. В нем вершина имеет не более двух потомков, причём левый потомок хранит данные, которые меньше текущей вершины, а правый — которые больше ее. Таким образом, мы либо находим нужную нам строку и привязанную к ней информацию, либо спускаемся до листовой вершины и сообщаем, что нужной строки в словаре нет.

Более эффективно в данном случае использовать тернарные деревья. Если взять все первые буквы, которые встречаются во всех словах словаря, мы получим почти полный алфавит. Для того чтобы понять, есть ли у нас в словаре слова, начинающиеся с первой буквы пришедшего токена, можно взять полученное множество букв и построить из него бинарное дерево. От каждой буквы, за которой в слове могут следовать другие буквы,

проведём ещё одну связь к следующему дереву, которое хранит все буквы, стоящие за найденной. Таким образом, каждая вершина может иметь до трёх потомков — два из них указывают на буквы, находящиеся в слове на той же позиции, третий указывает на поддерево, хранящее информацию о следующих буквах.

Самым эффективным по скорости на данный момент является метод хранения словаря в виде ассоциативной хеш-таблицы, однако объем памяти, необходимый для такой таблицы, будет весьма значительным. В таком словаре мы напрямую связываем токен с множеством результатов разбора.

2.4 Анализ несловарных слов

Для обработки словоформ, отсутствующих в словаре, существует ряд традиционных решений, которые имеют практические реализации в современных морфологических процессорах. К ним относятся отсечение известных и неизвестных префиксов, аналогия по окончанию и наборы правил для слов, записанных через дефис.

Отсечение известных префиксов

В данном случае делается предположение, что если два слова отличаются только префиксом, то и результаты их анализа будут совпадать. Такое предположение носит общий характер и вытекает из словообразовательной системы русского языка.

В морфопроцессорах AOT и pymorphy2 реализован метод отсечения как известных, так и неизвестных префиксов. Однако отсечение неизвестных префиксов зачастую приводит к генерации заведомо неправильных вариантов, например, для словоформы *вейпер* генерируется разбор с нормальной формой *вейперть* (по аналогии со словом *перть*), который возникает при отсечении неизвестного префикса *вей*.

Аналогия по окончанию

Анализ слова по окончанию происходит в предположении, что если два слова имеют одинаковые окончания, то они относятся к одной и той же парадигме и имеют одинаковые морфологические характеристики. Это предположение объясняется тем, что словообразование в русском языке происходит, в основном, с помощью суффиксов и окончаний, набор которых является достаточно ограниченным.

Для анализа с помощью аналогии по окончанию используется дополнительный словарь окончаний, который строится из исходного словаря словоформ. При построении для каждого окончания определённой длины (1-5 букв) выбираются все разборы, которые встретились в словаре. Однако здесь возникает сразу несколько проблем. Например, окончаниям, особенно коротким, соответствует очень много вариантов разбора (скажем, окончание *-а* может быть почти у любой части речи). Пример анализа слова с помощью такой функции показан на рисунке 2.11.

$$\textit{гл}\underbrace{\textit{окая}} \rightarrow \begin{matrix}\text{окая } (\textit{глубокая, широкая, ...}) \\ \text{ADJF|femn|nomn|sing}\end{matrix} \rightarrow \begin{matrix}\text{глокая} \rightsquigarrow \text{глокий} \\ \text{ADJF|femn|nomn|sing}\end{matrix}$$

Рис. 2.11. Аналогия по окончанию *окая*

Слово *глокая* отсутствует в словаре и не имеет известных префиксов, поэтому для его разбора выполняется аналогия по окончанию. Наибольшим окончанием, которое находится в словаре окончаний, является окончание *-окая*. Оно соответствует парадигме, словоформы которой являются прилагательными женского рода, единственного числа. Внутри этой парадигмы выбираются варианты разбора, соответствующие окончанию, подбирается окончание начальной формы, конкатенируется с полученной псевдоосновой, после чего алгоритм повторяется для следующего возможного варианта разбора.

Для предсказания нам может серьёзно помочь дерево псевдоокончаний. Оно уже хранит все возможные окончания для русского языка. Для незнакомого слова можно выделить все (или почти все) возможные окончания из имеющихся. Но количество полученных вариантов предсказания оказывается слишком большим (в некоторых случаях — до нескольких со-

тен). Особенно много вариантов будет для пустого окончания. В связи с этим следует проанализировать не только само окончание, но и одну-три буквы, которые стоят непосредственно перед ним. Например, пустое окончание может встретиться в разных частях речи. Но наречия обычно заканчиваются на *-о, -е*, некоторые согласные. Если использовать вместо одной буквы несколько, точность предсказания повышается. Эта информация может помочь нам отфильтровать невозможные варианты.

Извлечь информацию о возможных сочетаниях букв, стоящих перед окончанием, нам поможет дерево псевдооснов. В нем необходимо взять листовые вершины и один-два их родителя. Более того, можно посчитать частоты встречаемости таких комбинаций для каждой из парадигм. Такая статистика поможет принять решение о том, какой из вариантов предсказанной парадигмы является наиболее правдоподобным.

Правила для слов с дефисом

Для задачи разбора слов с дефисом не существует традиционных алгоритмов решения. Различные морфологические процессоры опираются на набор эвристических правил, с помощью которых и происходит разбор. Слова с несколькими дефисами разбираются методами, описанными ранее (по префиксу или аналогии по окончанию). Сформулированный набор правил применяется упорядоченно от частных к более общим.

1. Если в слове одна из частей (первая или вторая) является числом или известным префиксоидом (*Маяк-401, вице-директор* и т. д.), то происходит разбор только оставшейся части, которая даёт набор морфохарактеристик и лемму, а результирующая лемма получается приписыванием отсечённой части.
2. Если обе части слова через дефис являются одинаковыми (*гули-гули, тук-тук*), то происходит разбор только одной части.
3. Если для первой или второй части слова с дефисом анализатор не смог найти по словарю ни одного варианта разбора, то по отдельности обрабатываются следующие случаи:

(a) если эта часть — слово в латинице или буквенно-цифровой комплекс (*α-конверсия, ER-метод, 3D-система*), то разбор делается для оставшейся части;

(b) если среди вариантов разбора первой части слова встречается вариант, в котором нормальная форма совпадает с исходной словоформой (*киловатт-часов, веб-дизайн, лексико-семантический*), то разбор делается для второй части слова;

(c) в противном случае (*человек-гора, изба-читальня*) обе части слова с дефисом анализируются независимо и ко всем вариантам разбора второй части приписывается результат разбора первой части, часть речи которого совпадает с частью речи второй части.

2.5 Разрешение морфологической омонимии

Для решения этой проблемы существует три основных подхода:

- основанный на правилах;
- основанный на статистике;
- основанный на машинном обучении.

Метод, основанный на правилах, применяется, например, в работе [5]. Здесь были написаны отдельные модули снятия омонимии, разрешающие ее только в определённых случаях в зависимости от контекста и самих омонимичных слов, их части речи или набора параметров. Суть метода сводится к тому, что в некоторых ситуациях анализ контекста помогает понять синтаксическую структуру части предложения, а с ее помощью и формы слов. Например, в конструкции вида *ни ..., ни ...* оба слова обычно принадлежат одной и той же части речи и находятся в одной и той же форме. Если одно из слов окажется неомонимичным, определить форму второго будет несложно.

Однако данный метод требует ручного составления правил, то есть долгой и кропотливой работы. Для каждого из правил требуется написать самостоятельный программный модуль. Пополнение системы правил становится всё труднее с каждым новым правилом. В связи с этим подобные методы не получили широкого распространения.

Гораздо чаще в современных морфологических процессорах применяются статистические методы и методы, основанные на машинном обучении. Это связано, в первую очередь, с наличием открытых, размеченных корпусов, объёма которых достаточно для построения довольно точных моделей.

Подсчёт статистики различных вариантов разбора по корпусу является простейшим способом снятия морфологической омонимии. При этом по размеченному корпусу со снятой омонимией[5] происходит вычисление апостериорных вероятностей каждого из разборов. В современных системах анализа текстов на естественном языке применяются несколько способов подсчёта таких вероятностей, однако все они оказывают незначительное влияние на точность снятия омонимии. Гораздо большее влияние оказывает сам корпус: его представительность, объём, точность разметки.

Для разрешения омонимии может быть реализован метод простого подсчёта вероятности $P(t|w)$ для каждого из набора тегов и слов корпуса (**униграммный метод**):

$$P(t|w) = \frac{Fr(w,t)}{Fr(w)}$$

где w — слово, t — набор тегов, $Fr(w)$ — сколько раз слово встретилось в корпусе, $Fr(w,t)$ — сколько раз слово встретилось в корпусе с набором тегов t. Фактически в данном методе рассчитывается апостериорная вероятность встретить данную словоформу среди всех вариантов употребления в тексте заданного токена.

Например, если в нашем размеченном корпусе токен *стекло* встретился 100 раз, при этом в форме глагола он встретился 5 раз, в форме именительного падежа существительного — 60 раз и в форме винительного падежа — 35 раз, то вероятности соответствующих форм будут равны 0.05, 0.6 и 0.35. Теперь встречая в новом, неразмеченном, тексте токен *стекло* мы будем принимать решение, что он должен являться именительным падежом существительного (наиболее вероятное решение). Итоговая точность определения леммы будет примерно 0.95 (один раз на двадцать употреблений

[5]Такой корпус имеет ровно один вариант морфологического разбора для каждого словоупотребления.

все-таки попадает глагол), точность определения набора грамматических параметров составит 0.6 (ещё 35 раз из 100 мы пропустим винительный падеж).

Одним из общедоступных источников информации о частотах встречаемости слов является размеченный корпус НКРЯ со снятой омонимией, объёмом около 1 млн. словоупотреблений. Однако к подобным данным надо всё равно относиться с осторожностью. Так, в [15] приводится информация, что точность «золотого стандарта», подготовленного для соревнований морфологических парсеров, проводимого в 2010 году [10], колеблется от 85 до 95% в зависимости от вида разметки.

Более точные результаты даёт учёт контекста слова. Например, если мы встретили в тексте именную группу, состоящую из нескольких прилагательных и существительного, то все слова в ней должны быть согласованы между собой. Более того, если мы встретили предложную группу, то прилагательные и существительное в ней не могут находиться в именительном падеже. Вообще, падеж слов в предложной группе будет определяться предлогом, стоящим в начале. То есть, если мы будем смотреть не на одно слово, а на его соседей, то точность разрешения омонимии должна повыситься.

В этом случае используют **триграммную модель** — анализ слова и его контекста из ещё двух слов. Как показывает практика, триграммная модель показывает значительно лучшие результаты, чем уни- или биграммные. Четырехграммная модель занимает значительно больше места, но серьёзного прироста в точности не даёт [18].

Триграммная модель может использоваться в разных вариантах. Можно выбрать словоформу с максимальной вероятностью встречаемости при условии предыдущих двух слов:

$$w_i = \arg\max P(w_i|w_{i-1}, w_{i-2}).$$

Вообще, под триграммой могут пониматься разные конструкции. Во-первых, это может быть три словоформы, идущие подряд. Однако вероятность встретить именно эти три словоформы в произвольном тексте может оказаться очень низкой. Но если отбросить леммы, вероятность встретить

подобную триграмму становится значительно выше. Но и в этом случае может получиться, что в тексте встретится триграмма, не встречавшаяся ранее (например, в связи с тем, что тексты разного стиля имеют разную частоту встречаемости для частей речи или вообще синтаксических конструкций). При этом составляющие ее биграммы в корпусе уже встречались. В такой ситуации можно использовать комбинации биграмм:

$$w_i = \arg\max P(w_i|w_{i-1}) \cdot P(w_i|w_{i-2}).$$

В общем случае можно использовать **сглаживание**: с разными весовыми коэффициентами берётся информация о триграмме, биграмме и униграмме, на случай, если какая-то часть информации отсутствует в анализируемом тексте:

$$w_i = \arg\max(\lambda_1 P(w_i|w_{i-1}, w_{i-2}) + \lambda_2 P(w_i|w_{i-2}) + \lambda_3 P(w_i)).$$

Во всех рассмотренных вариантах мы пытались предсказать текущее слово по предыдущим. Однако на практике может оказаться, что текущее слово связано с несколькими последующими. Также возможен вариант, когда слово будет связано как с левым, так и с правым контекстом. В такой ситуации можно использовать статистическую информацию не по одной, а по трём триграммам, в которых текущее слово будет занимать разные позиции:

$$w_i = \arg\max P(w_i|w_{i-1}, w_{i-2}) \cdot P(w_i|w_{i-1}, w_{i+1}) \cdot P(w_i|w_{i+1}, w_{i+2}).$$

Наконец, можно использовать комбинации приведённых выше методов.

При использовании триграмм получается следующая ситуация. Третье слово в предложении определяется первыми двумя. Четвёртое слово зависит от второго и третьего (которое также определяется и первым). Пятое слово будет зависеть от всех предыдущих слов и так далее. Таким образом, необходимо выделять не просто слово по его контексту, а найти максимум вероятности для всего предложения. Из-за этого, скорость рабо-

ты метода снятия омонимии растёт по экспоненте от длины предложения, а само снятие омонимии сводится к задаче поиска оптимального решения.

С другой стороны, в текстах на русском языке встречается достаточно много неомонимичных слов. И если в тексте встретится два неомонимичных слова подряд, то предложение можно разделить на две независимые части, оптимизация которых проводится независимо. Такой подход помогает существенно сократить время работы алгоритма. Вообще, статистические методы контекстного снятия омонимии показывают хорошие результаты в этой задаче [16].

Помимо статистических методов, для снятия омонимии сейчас используется целый спектр методов классификации. Представим, что каждая словоформа, являющаяся результатом морфологического анализа выбранного токена, принадлежит одному из двух классов: корректное предсказание и некорректное предсказание. При такой постановке задачи можно провести бинарную классификацию словоформ. В качестве параметров классификации могут браться грамматические параметры данного или соседних слов в некотором окне, их леммы, признаки наличия знаков препинания и проч. Выбор метода классификации во многом зависит от вкусов разработчика, для решения которой используются такие методы машинного обучения, как скрытые марковские модели [19], условные случайные поля [30], рекуррентные нейронные сети [26] и др. (подробнее см. [38]). Для обучения метода классификации также используются размеченные корпуса, например, НКРЯ с вручную снятой омонимией.

Для контекстного снятия омонимии может использоваться, например, метод CRF (условные случайные поля). Этот метод хорошо зарекомендовал себя в задачах компьютерной лингвистики, таких как определение части речи, определение именованных сущностей и др [36]. CRF является дискриминативной вероятностной моделью. Одним из главных достоинств этой модели является то, что она не требует моделировать вероятностные зависимости между так называемыми наблюдаемыми переменными.

Снятие омонимии по всем морфологическим характеристикам (тегам) является сложной для обучения CRF-классификатора из-за большого числа тегов, что требует усложнения модели. Применяются четыре обученных

CRF-классификатора, последовательно отсекающих омонимичные варианты. Первым работает CRF-классификатор для части речи, используемые им признаки — словоформа и возможные части речи (в виде бинарного вектора). Затем применяется классификатор для рода (признаки: словоформа, уже определённая часть речи, возможные варианты рода: мужской, женский, средний). После этого аналогичным образом работают CRF-классификаторы числа и падежа. На рисунке 2.12 показан пример набора признаков для классификатора рода.

технику → ТЕХНИКУ NOUN

1	1	0
masc	femn	neut

→ masc (0.65)

Рис. 2.12. Признаки для классификатора рода

Классификаторы применяются последовательно, накапливая ошибку предыдущих этапов. Частичный пример такого применения показан на рисунке 2.13. Здесь классифицируемое слово *мыла* проходит через все четыре классификатора: на первом этапе определяется часть речи (существительное), затем классификатор рода выбирает средний род, потом классификатор числа определяет, что слово относится к единственному числу, а классификатор падежа выбирает единственный возможный вариант.

Даша купила ***мыла*** *и пошла домой*

мыла	мыло	NOUN|inan|neut|nomn|plur
мыла	**мыло**	**NOUN|gent|inan|neut|sing**
мыла	мыло	NOUN|accs|inan|neut|plur
мыла	мыть	VERB|femn|impf|indc|past|sing|tran

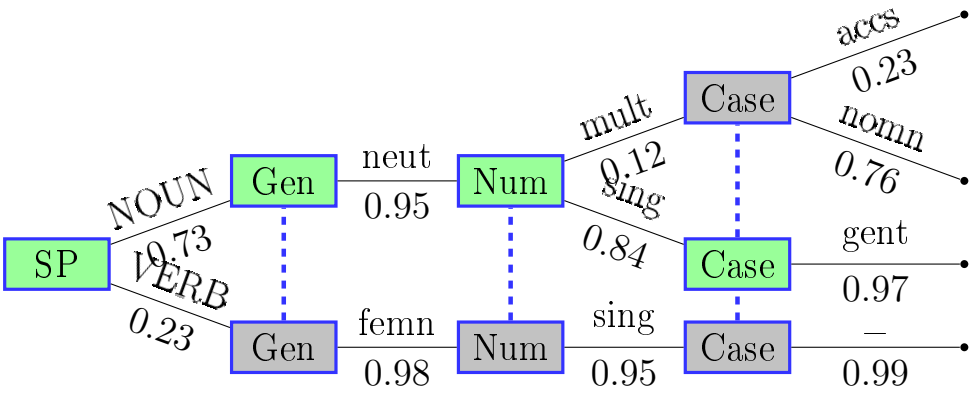


Рис. 2.13. Классификация всех тегов слова *мыла*

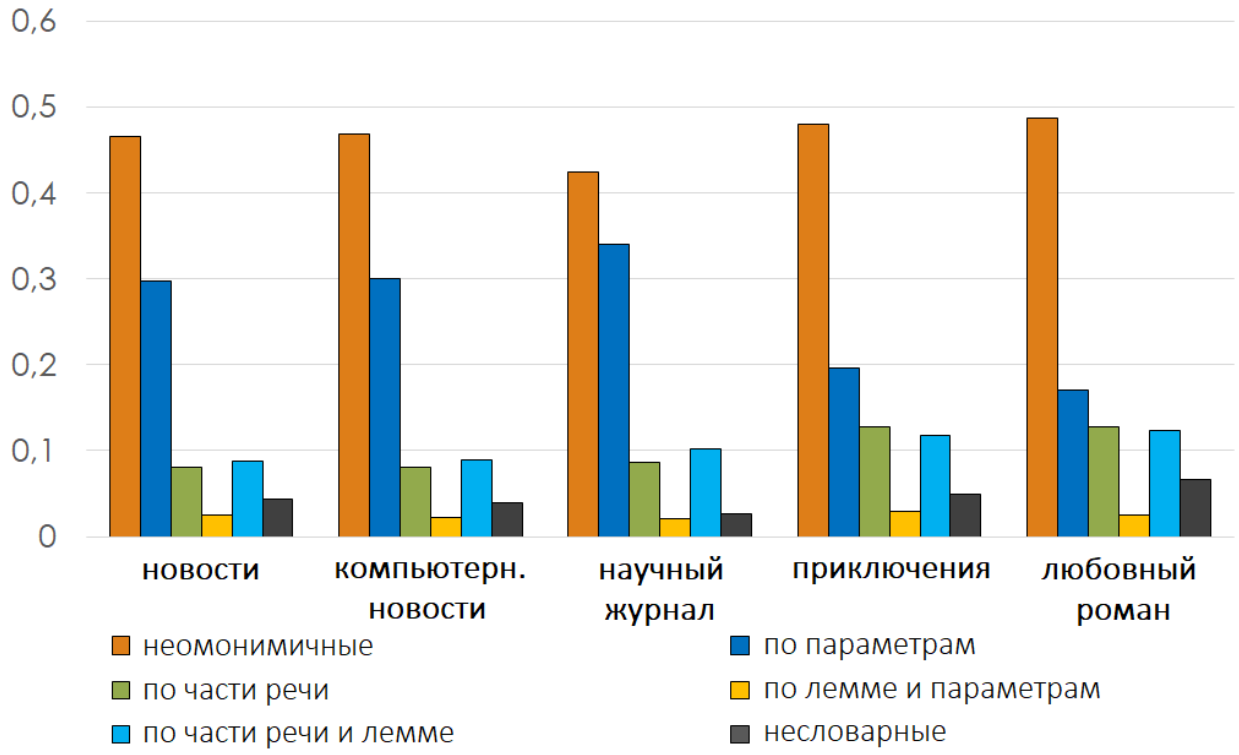


Рис. 2.14. Распределение слов по классам омонимии в текстах различных жанров

Отметим, что встречаются ситуации, когда после обработки словоформы всеми четырьмя классификаторами всё равно сохраняется морфологическая омонимия, например, по лемме или одушевлённости. В таком случае производится бесконтекстное снятие на основе статистики по корпусу. В итоге остаётся единственный вариант разбора.

2.6 Особенности омонимии в разных языках

Выше мы уже ввели четыре класса омонимии для слов: по параметрам, по части речи, по лемме и одновременно по части речи и лемме. Добавим к этим классам ещё два: неомонимичные и несловарные (отсутствующие в словаре). А теперь посмотрим, как распределены слова по классам омонимии в разных текстах на русском языке. Для экспериментов возьмём тексты новостной ленты СМИ, новости по околокомпьютерной тематике, научный журнал, любовные истории и приключения в стиле «меча и магии». Получившийся результат можно увидеть на рис. 2.14.

Как видно из диаграмм, в беллетристике авторы в среднем больше стараются избегать слов, омонимичных только по грамматическим параметрам, за счёт использования большего количества слов, омонимичных по части речи. Помимо этого, в беллетристике больше незнакомых слов — тех самых имён собственных.

Различия текстов разных стилей простираются много дальше. Разные авторы и разные стили эксплуатируют разные синтаксические конструкции. В зависимости от того, какую часть действия описывает автор, меняется соотношения частей речи. В работе [4] вводятся следующие параметры текста:

$$\begin{aligned}
\text{Предметность} &= \frac{P_n + P_p}{P_a + P_v}, \\
\text{Качественность} &= \frac{P_a + P_{adv}}{P_n + P_v}, \\
\text{Активность} &= \frac{P_v}{N}, \\
\text{Динамизм} &= \frac{P_v}{P_n + P_a + P_p}, \\
\text{Связность} &= \frac{P_c}{P_s},
\end{aligned}$$

где P_n — это количество существительных в анализируемом тексте, P_a — количество прилагательных, P_v — количество глаголов и глагольных форм (причастие, деепричастие), P_p — количество местоимений, P_{adv} — количество наречий, P_c — количество предлогов и союзов, P_s — количество самостоятельных предложений в тексте и N — количество слов в тексте.

Таким образом, можно утверждать, что статистику употребления слов лучше собирать по корпусу того стиля, который планируется анализировать. То же самое можно сказать про статистику употребления триграмм.

Но давайте вернёмся к нашим шести классам омонимии. Возьмём теперь текстовые коллекции для разных языков. На рисунке 2.15 показаны результаты для польского, русского, французского, немецкого, английского, итальянского и испанского языков. Для получения этого рисунка использовались такие системы морфологического анализа, как Morphalu[6],

[6] `http://www.cnrtl.fr/lexiques/morphalou/`

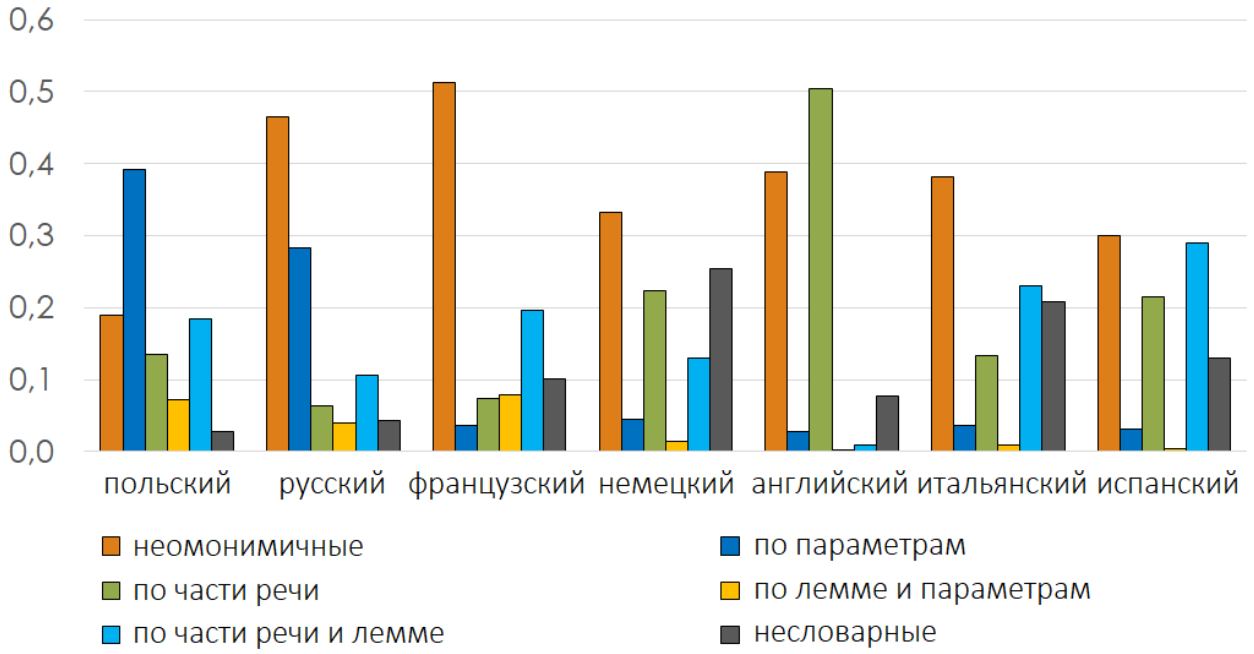


Рис. 2.15. Распределение слов по классам омонимии в текстах различных языков

FreeLing[7], TreeTagger[8] и Morfologik[9]. Как видно из рисунка, каждый язык приведённый обладает собственным «профилем» распределения слов по классам омонимии. Английский язык предсказуемо показывает половину слов, омонимичных по части речи; в русском и польском языке много слов, омонимичных по грамматическим параметрам; во французском языке очень много неомонимичных слов. Заодно диаграммы показывают различия словарей по объёму — немецкий язык показал самое большое количество слов, отсутствующих в словаре, тогда как польский словарь (и в самом деле самый большой по объёму в данном эксперименте — больше 400 000 лексем) «знает» почти все слова в тексте.

Теперь давайте проанализируем распределение слов по классам омонимии в зависимости от частоты их встречаемости в тексте. Отсортируем все слова по частоте и выделим первые десять групп по тысяче слов. Для каждой тысячи рассчитаем распределение слов по классам омонимии. Полученный результат показан на рис. 2.16.

[7] http://devel.cpl.upc.edu/freeling/downloads?order=time&desc=1
[8] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
[9] http://morfologik.blogspot.ru/2013/02/morfologik-20-rc2.html

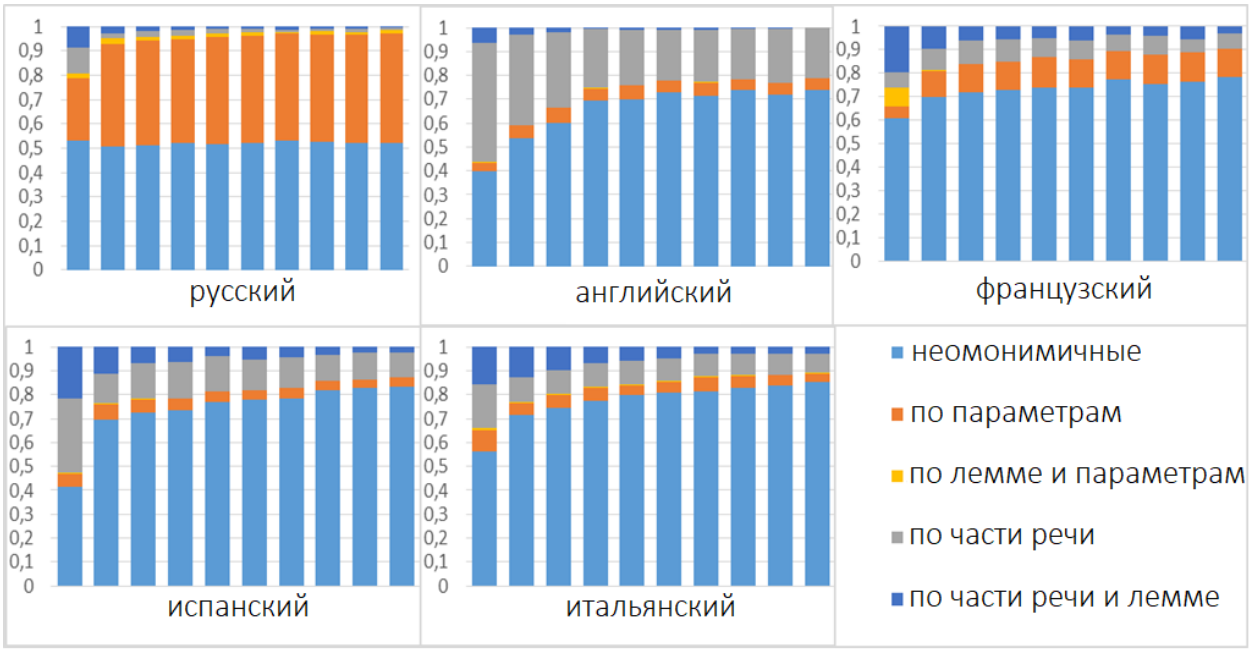


Рис. 2.16. Распределение слов по классам омонимии в зависимости от частотности слова

Как видно из диаграмм, наиболее частотные слова принадлежат несколько иным классам омонимии. Это связано с тем, что наиболее частотными словами является более старая лексика, в состав которой входят предлоги и местоимения.

Полученные диаграммы можно проанализировать с другой точки зрения. Слова, омонимичные по параметрам или параметрам и лемме, обладают известной частью речи. За счёт этого в некоторых случаях может упроститься анализ синтаксической структуры предложения. Получается, что в русском языке около 80% слов не омонимичны по части речи (сравним с примерно половиной слов, омонимичных по части речи, в английском языке).

Такие отличия диктуют два пути для анализа текстов на разных языках: либо выбранные методы должны подгоняться под особенности того или иного языка (возможно вручную), либо при начальной настройке метод должен обучаться на особенности языка и учитывать их в своей работе. Именно поэтому методы машинного обучения получают всё большее развитие в автоматической обработке текстов.

2.7 Список литературы

[1] АОТ [Электронный ресурс]. URL: http://aot.ru/docs/rusmorph.html (дата обращения 16.05.2017).

[2] Большакова Е.И. Компьютерная лингвистика: методы, ресурсы, приложения // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие — М.: МИЭМ, 2011.

[3] Большакова Е.И., Иванов К.М., Сапин А.С., Шариков Г.Ф. Система для извлечения информации из текстов на базе лексико-синтаксических шаблонов // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2016). 3-7 октября 2016 г.: Труды конференции. Том. 1 — Смоленск, Универсум, 2016, с.14-22.

[4] Горошко Е.И. Особенности мужского и женского стиля письма // Преображение — 1998. — №6. с. 48-64.

[5] Епифанов М.Е., Антонова А.Ю., Баталина А.М. и др. Итеративное применение алгоритмов снятия частеречной омонимии в русском тексте // КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ — труды Международной конференции «Диалог-2010», Том. 9(16), сс. 119-123.

[6] Зализняк А.А. Грамматический словарь русского языка. — М., Русский язык, 1980.

[7] Кнут Д. Э. Искусство программирования. Том 3. Сортировка и поиск — М.: Вильямс, 2014. 824 с.

[8] Международные стандарты в области корпусной лингвистики. // Структурная и прикладная лингвистика. Выпуск 9. СПб., 2012 С. 201-221.

[9] Клышинский Э.С. Начальные этапы анализа текста // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие — М.: МИЭМ, 2011.

[10] Ляшевская О.Н. и др. 2010. Оценка методов автоматического анализа текста: морфологические парсеры русского языка // Компьютерная лингвистика и интеллект. технологии: По материалам ежегодной Международной конференции «Диалог». Вып. 9 (16) — М.: Изд-во РГГУ.

[11] Морфологический анализатор Mystem 3.0 [Электронный ресурс]. URL: https://events.yandex.ru/lib/talks/2427/ (дата обращения 16.05.2017).

[12] Национальный корпус русского языка [Электронный ресурс]. URL: http://ruscorpora.ru/ (дата обращения 16.05.2017).

[13] Открытый корпус OpenCorpora [Электронный ресурс]. URL: http://opencorpora.org/ (дата обращения 16.05.2017).

[14] Плунгян В.А. Общая морфология. Введение в проблематику. — М.: Едиториал УРСС. — 2003. 384 с.

[15] Прикладная и компьютерная лингвистика / Под. ред. Николаева И.С., Митрениной О.В., Ландо Т.М. — М.: ЛЕНАНД, 2016. — 320 с.

[16] Рысаков С.В., Клышинский Э.С. Статистические методы снятия омонимии //Новые информационные технологии в автоматизированных системах: материалы восемнадцатого научно-практического семинара. — М: МИЭМ НИУ ВШЭ — 2015. — №. 18.

[17] Сокирко А.В. Морфологические модули на сайте www.aot.ru // Труды международной конференции «Диалог-2004. Компьютерная лингвистика и интеллектуальные технологии». М.: Наука, 2004. С. 559.

[18] Сокирко А.В. Быстрословарь: предсказание морфологии русских слов с использованием больших лингвистических ресурсов // Труды международной конференции «Диалог-2010. Компьютерная лингвистика и интеллектуальные технологии». М.: Издательский центр РГГУ, 2010. С. 450.

[19] Baum L. E., Petrie T. Statistical inference for probabilistic functions of finite state Markov chains // The annals of mathematical statistics. — 1966. — Т. 37. — №. 6. — С. 1554-1563.

[20] Bernhard D. Simple morpheme labelling in unsupervised morpheme analysis //Workshop of the Cross-Language Evaluation Forum for European Languages. — Springer Berlin Heidelberg, 2007. — С. 873-880.

[21] Bocharov V., Bichineva S., Granovsky D., Ostapuk N., Stepanova M. Quality assurance tools in the OpenCorpora project // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 25–29 мая 2011 г.). Вып. 10 (17). — М.: РГГУ, 2011.

[22] Buchholz, Sabine and Marsi, Erwin (2006). CoNLL-X shared task on multilingual dependency parsing, Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL), New York, NY, pp. 149–164.

[23] Creutz M., Lagus K. Morfessor in the morpho challenge // Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes. — 2006. — С. 12-17.

[24] Daciuk J. et al. Incremental construction of minimal acyclic finite-state automata //Computational linguistics. — 2000. — Т. 26. — №. 1. — С. 3-16.

[25] dawgdic [Электронный ресурс]. URL: `https://code.google.com/archive/p/dawgdic/` (дата обращения 16.05.2017).

[26] Elman J. L. Finding structure in time // Cognitive science. — 1990. — Т. 14. — №. 2. — С. 179-211.

[27] Fredkin E. Trie memory // Communications of the ACM. — 1960. — Т. 3. — №. 9. — С. 490-499.

[28] Grünwald P. D. The minimum description length principle. — MIT press, 2007.

[29] Hervé Déjean. Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In D.M.W. Powers (ed.) NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Language Learning, ACL, 1998, pp 295–298.

[30] Lafferty J. et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data // Proceedings of the eighteenth international conference on machine learning, ICML. — 2001. — Т. 1. — С. 282-289.

[31] Harris S. Zellig. Morpheme boundaries within words: Report on a computer test, Transformations and Discourse Analysis Papers 73, 1970, pp 68–77.

[32] Porter, M.F., An algorithm for suffix stripping // Program, 14(3), p. 130-137.

[33] Pymorphy2 [Электронный ресурс]. URL: `https://pymorphy2.readthedocs.io/en/latest/` (дата обращения 16.05.2017).

[34] Rayson, P., and Garside, R. The CLAWS Web Tagger. // ICAME Journal Vol 22, p. 121-123.

[35] Quinlan J. R. Induction of decision trees // Machine learning. — 1986. — Т. 1. — №. 1. — С. 81-106.

[36] Sha F., Pereira F. Shallow parsing with conditional random fields // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. — Association for Computational Linguistics, 2003. — С. 134-141.

[37] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In.: Proceedings of the international conference on new methods in language processing. (1994) 44-49

[38] Schütze H. Introduction to Information Retrieval // Proceedings of the international communication of association for computing machinery conference. — 2008.

[39] Segalovich I. A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine // MLMTA. — 2003. — С. 273-280.

[40] Text Encoding Initiative [Электронный ресурс]. URL:`http://www.tei-c.org/index.xml` (дата обращения 10.07.2017).

[41] Virpioja S. et al. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. — 2013.

[42] Universal Dependencies [Электронный ресурс]. URL: `http://universaldependencies.org/` (дата обращения 10.07.2017).

Глава 3

Извлечение информации из текстов: портрет направления

Большакова Е.И., Ефремова Н.Э.

Постоянно растущий объем текстов на естественном языке, находящихся в свободном доступе, значительно затруднил процесс поиска необходимой информации, а также отделение значимой информации от незначимой. При всем желании человек не в состоянии охватить всю интересующую его и содержащуюся в текстах информацию за приемлемое время. Попытки справиться с этой проблемой привели к развитию направления в компьютерной лингвистике, получившего название **Information Extraction** (IE). Основной задачей этого направления является автоматическое экстрагирование значимых для человека данных (например, о каком-либо событии), как правило, из большого массива текстов, и преобразование их в структурированную форму, что облегчает их последующую обработку и анализ. IE-системы охватывают многие сферы производственной, экономической и научной деятельности. В частности, они применяются в финансовой аналитике, в правоохранительной деятельности, для выявления новых научных трендов.

3.1 Специфика задач, подходы к решению, извлекаемая информация

Задачи извлечения информации из текстов на естественном языке можно отнести к **информационному поиску** (**Information Retrieval**) в его самом широком понимании, предполагающем поиск релевантной информации. Однако у направления Information Extraction есть принципиальные особенности. В отличие от классического поиска, выполняемого поисковыми машинами сети Интернет и выдающего пользователю список отранжированных сниппетов, на выходе IE-систем — структурированная информация, извлечённая из коллекции текстов (или одного большого текста), что так или иначе предполагает преобразование извлечённой информации. В целом, в рамках IE решается задача автоматического извлечения из текстов данных, релевантных определённой проблеме/вопросу/теме, из неструктурированных текстов [19]. Такие тексты не имеют никакой разметки или метаданных, помогающих идентифицировать искомую информацию. Для удобства дальнейшей обработки и применения извлечённые данные структурируются: в простейшем случае с помощью тегов XML, в более сложных они преобразуются и сохраняются в формальном виде: в реляционных базах данных, таблицах, сетевых базах знаний. Структурированные данные передаются средствам аналитической обработки (OLAP, Data Mining) или же визуализируются для человека-аналитика в виде семантических сетей, когнитивных карт и т.п. [5].

Первые прикладные исследования в области извлечения информации относятся к началу 1980-х годов, они касались обработки новостных и военных текстов с целью выделения в них определённых событий [19]. В настоящее время различные данные извлекаются также из художественных произведений, научно-технических статей, текстов сети Интернет (журналов, блогов и др.). Таким образом, к рассматриваемой области стали относить извлечение любых семантически значимых данных из текстов различных функциональных стилей, в том числе:

- имён персонажей и связанных с ними событий из художественных произведений;

- названий белков, генов, болезней, лекарств из текстов по медицине;
- терминов и их смысловых связей из специализированных текстов;
- ключевых слов и словосочетаний из индексируемых текстов;
- мнений по поводу продуктов и услуг из интернет-текстов отзывов.

Кроме того, ведутся исследования в области обработки мультимедийных документов: изображения, аудио- и видеофайлы автоматически обрабатываются, из них извлекается содержимое, на основании которого затем для файлов составляется описание [25].

В качестве извлекаемых из текстов данных обычно выступают [16]:

- значимый объект: имя персоналии, название компании и пр. для новостных сообщений, термин предметной области специального текста, ссылка на литературу для научно-технических документов и т. д.;
- атрибуты объекта, дополнительно характеризующие его, например, для компании это юридический адрес, телефон, имя руководителя и т.п.;
- отношение между объектами: к примеру, отношение «*быть владельцем*» связывает компанию и персону-владельца, «*быть частью*» соединяет факультет и университет;
- событие/факт, связывающее несколько объектов, например, событие «*прошла встреча*» включает участников встречи, а также место и время ее проведения.

Согласно видам извлекаемой информации общая задача извлечения информации из текстов включает следующие основные подзадачи:

- распознавание и извлечение **именованных сущностей** (**named entities**): *А.П. Чехов*, *Нижний Тагил*, *ПКО «Картография»* и т.п.;
- выделение **атрибутов** (**attributes**) объектов и семантических **отношений** (**relations**) между ними: даты рождения персоны, отношения «*работать в*» и т. д.;
- извлечение **фактов** и **событий** (**events**), охватывающих несколько их параметров (атрибутов), например, событие «*кораблекрушение*» с атрибутами *дата*, *время*, *место* и т.п.).

В нижеследующем предложении

В октябре 2005 года компания eBay за 2,6 миллиарда $ купила 30% акций Skype

содержатся несколько видов извлекаемой информации: событие «*купить*» и его параметры (атрибуты): покупатель, объект покупки, время, цена, причём параметры представлены именованными сущностями.

На данный момент наиболее исследованной подзадачей является распознавание именованных сущностей (**NER: Named Entity Recognition**); достаточно исследована и вторая. Наиболее сложна и требует дальнейшего изучения подзадача выявления событий, она позволяет отвечать на вопросы о том, что произошло, кто это сделал, когда, где, как и почему («*who did what to whom, when, where, through what methods (instruments), and why*») [40].

Для решения задач распознавания и извлечения информации из текстов используются два главных подхода: **основанный на правилах** (**rule-based**), или **инженерный**, и основанный на **машинном обучении** (**machine learning**). Отметим, что появляется все больше гибридных методов, учитывающих достоинства обоих подходов.

Инженерный подход опирается на тот факт, что извлекаемая информация употребляется в рамках определённых языковых конструкций. Например, название города пишется с большой буквы и нередко предваряется словами *город*, *гор.* или *г.* Подобная лингвистическая информация обычно вручную описывается в виде формальных шаблонов распознаваемых конструкций и правил их обработки. Затем правила применяются IE-системой к анализируемому тексту: в нем ищутся описанные шаблонами фрагменты, из которых извлекается искомая информация. К примеру, по правилу

ЕСЛИ за словом *город*, *город-курорт*, *город-музей*, *город-герой*, *гор.* или *г.* следует слово с большой буквы,

ТО извлечь это слово как название города.

из текста *...в Российской империи появился город Пятигорск...* будет выявлено название города: *Пятигорск*.

В рамках подхода, основанного на машинном обучении, применяются методы обучения **с учителем** (**supervised**), методы обучения **без**

учителя (**unsupervised**), методы **частичного обучения с учителем** (**bootstrapping**).

Чаще всего применяется обучение с учителем, которое подразумевает построение математической и программной модели, которая умеет отличать искомые данные от всех остальных. Построение такого **машинного классификатора** (т. е. обучение модели) происходит на специально размеченном вручную текстовом корпусе (**обучающей выборке**), в котором значимым объектам, их атрибутам, отношениям, фактам приписаны соответствующие метки. Метки кодируют **признаки** для распознавания этих данных. Для вышеприведенного примера для извлечения названия города в качестве признаков могут выступать: регистр (верхний) первой буквы слова, конкретные слова, стоящие перед ним (*город, город-курорт, город-музей, город-герой, гор.* или *г.*), а также признаки последующих слов (для выявления многословных названий, таких как *Нижний Тагил*).

По сути, обучение модели заключается в выявлении на основе частных данных, вошедших в обучающую выборку, общих закономерностей и зависимостей, которые присущи реальным данным. После обучения полученный классификатор применяется к текстам, при этом каждому извлечённому слову или словосочетанию может ставиться в соответствие вероятность того, являются они искомыми данными или нет.

Несмотря на то, что исследования в области извлечения информации ведутся уже около 40 лет, остаются нерешенные проблемы. В первую очередь это связано с тем, что почти всегда используемым при извлечении информации критериям удовлетворяют не только искомые данные, но и другие слова и словосочетания текста. К примеру, одно и то же слово может быть именем персоны (*Роза, Лилия*) и названием растения (*роза, лилия*), так что однозначно решить вопрос, в каком значении оно используется, не всегда возможно. Так, в предложении *Роза не любит жару* слово *Роза* стоит в начале предложения, а в предложении *ЛИЛИЯ, ТЫ ПРЕКРАСНА!* в слове *ЛИЛИЯ* все буквы большие, что делает бессмысленным использование признака регистра буквы для различения имени персоны и названия растения. Даже для уже хорошо исследованных видов объектов, например, названий городов, нет строгих правил их именования: название может со-

стоять из одного или нескольких слов (*Клин, Нижний Новгород*), содержать тире (*Алма-Ата*), часть слов или все слова могут быть написаны с большой буквы (*Франкфурт-на-Майне, Десерт Хот Спрингс*).

Кроме того, постоянно возникают новые приложения задачи извлечения информации со своей спецификой обрабатываемых текстов и распознаваемых данных. В качестве извлекаемых значимых объектов могут выступать не только имена и названия («*Мона Лиза*», *КАМАЗ*), но и буквенно-цифровые комплексы (*15 февраля 1990 года, 5000 рублей*), а также обычные слова и словосочетания текста (*заместитель директора, научный сотрудник*). Все эти объекты разнородны, правила их извлечения различны, поэтому современные прикладные системы извлечения информации, как правило, ориентированы на обработку текстов в узких предметных областях, что делает практически невозможным их применение в рамках другой прикладной задачи и/или к данным из другой области.

Первые IE-системы были построены в рамках инженерного подхода, наиболее известной из них была AutoSlog [33]. Среди первых отечественных разработок стоит упомянуть семейство мультиязычных систем извлечения информации из деловых текстов OntosMiner [5], которые обеспечивали переход от неструктурированной информации к ее семантическому представлению в формате онтологий предметных областей, заложенных в систему (бизнес-события, судебная тематика и полицейские отчёты).

Разработка прикладных IE-систем является сложным и трудоемким процессом, существенную помощь в котором могут оказать инструментальные системы, включающие стандартные модули анализа текста и даже средства сборки и отладки приложений.

Инструментальные системы, предназначенные для разработки приложений в рамках инженерного подхода, имеют обычно встроенный формальный язык для задания **лингвистических правил и шаблонов** — с их помощью стандартные программные модули настраиваются на решение конкретной прикладной задачи.

Инструментальные системы, опирающиеся на машинное обучение, позволяют использовать уже построенные программные модели (классификаторы), а также обучать новые. Ясно, что применение готовых моделей

ограничивается той проблемной областью, для которой эти модели были построены. Например, в пакете OpenNLP [28] — это извлечение имён персоналий, географических объектов, дат, времени и пр. из новостных статей. Для получения новой модели необходима обучающая выборка, т. е. размеченный по определенным правилам текстовый корпус.

3.2 Методы оценки качества извлечения

Важную роль в развитии направления сыграли конференции MUC (Message Understanding Conferences) [20], проводившиеся в 1987-1998 гг. (IE — одно из первых направлений компьютерной лингвистики, где стали проводиться открытые тестирования автоматических систем на одних и тех же задачах и данных). Интерес, проявленный к различным задачам извлечения информации во время проведения MUC 1-7, послужил импульсом к стремительному развитию этой области.

Одним из наиболее весомых вкладов конференций было утверждение стандартов оценивания систем и методов, а именно метрик полноты и точности. При извлечении сущностей, атрибутов и отношений эти показатели рассчитываются общепринятым способом: **точность** (**Precision**) как количество правильных ответов, делённое на количество всех найденных ответов, а **полнота** (**Recall**) — как количество правильных ответов, делённое на общее число правильных ответов (указанных экспертом). Дополнительной метрикой оценки качества извлечения служит F**-мера** — соотношение между точностью и полнотой, чаще всего определяющееся как гармоническое среднее:

$$F = \frac{2RP}{R+P},$$

где R — полнота, а P — точность.

Для оценки качества извлечения событий и фактов были выработаны специальные способы подсчёта полноты и точности, учитывающие количество извлечённых атрибутов события и степень точности их извлечения [13]. Экспериментальная оценка методов и систем проводилась также в рамках конференций CONLL, IREX, ACE [15].

В рамках направления IE разработано немало систем, имеющих высокие оценки эффективности. К примеру, для решения задачи извлечения именованных сущностей на MUC-7 были предложены высокоэффективные IE-системы, как основанные на правилах, так и основанные на машинном обучении, имеющие F-меру, равную 94% и 90% соответственно (эти показатели для современных систем могут быть выше). Однако каждая из представленных систем обрабатывала тексты из одной довольно ограниченной предметной области: военно-морские операции, террористические атаки в Южной Америке, служебные перемещения в организациях, объединение корпораций в товарищества, авиакрушения и запуск космических ракет, производство микроэлектроники. К тому же эти системы в большинстве своём были разработаны для текстов на английском языке.

Важно, что современные системы извлечения информации при решении определённых задач (в частности, при обработке новостных сообщений на английском языке) показывают результаты, сходные с результатами, достигаемыми в ходе ручной разметки текстов экспертами. Например, лучшая система, участвовавшая в соревнованиях MUC-7 по теме извлечения информации о запусках спутников, достигла значения F-меры 93,39%, в то время как результаты ручной разметки находились в районе 97% [24].

3.3 Именованные сущности и особенности их извлечения

Типичными представителями именованных сущностей являются:

- имена персоналий, персоны (person): *Наташа Ростова, В.О. Кот*;
- названия географических объектов и мест (location): *Ока, Эверест, Латвия*;
- названия фирм, компаний, предприятий, организаций (organization): *Открытое акционерное общество «Я», ООО «А7 — Внедренческий центр»*.

Перечисленные сущности имеют имя/название и **референт**, т. е. объект внешнего мира с данным именем. Однако в текстах встречается огром-

ное число и других значимых объектов, распознавание которых необходимо в ряде приложений. Поэтому к именованным сущностям также относят:

- Торговые марки: *Nokia, Land Rover, «Ушастый нянъ»*;
- Даты: *02.03.1913, 29 июля, 1937-1985*;
- Время: *12:19, 2 p.m, с 12.00 до 22.00*;
- Номера телефонов: *+7(123)456-78-90, +86 10 6532 1381*;
- Адреса: *3-я улица Строителей, д. 25, кв.12*;
- Денежные единицы и денежные суммы: *руб, GBP, 25 $*;
- Числа: *1, 5000000, 4,25*;
- Ссылки на литературу: *[2], [Иванов, 1995]*;
- Обозначения белков, генов, химических веществ: $H_2N\text{-}CH(R)\text{-}COOH$;
- Род занятий: *художник, политический деятель, директор школы.*

В настоящее время нет общепринятого списка категорий (видов) именованных сущностей, но известны работы, в которых были попытки их создать — см. расширенную иерархию, состоящую из 200 видов [35].

Полное решение задачи извлечения именованной сущности в общем случае включает:

- нахождение наименований сущности в тексте;
- определение категории сущности;
- связывание сущности с референтом (называемым лицом/объектом), если именованная сущность является именем собственным.

Последнее важно, поскольку одинаковые слова/словосочетания, по факту могут отсылать к совершенно разным объектам. Например, слово *Лена* может быть:

- женским именем;
- названием реки (причём, не единственной);
- названием автомобильной дороги;
- названием железнодорожной станции;
- названием населённого пункта (опять же, не единственного) и т. д.

Для распознавания наименований сущностей в тексте используются как особенности их записи, так и словарные ресурсы: словари имен, географических названий, химических веществ, денежных единиц, родов занятий и т.п. Однако, если все обозначения денежных единиц еще можно

перечислить в словаре, то, например, с названиями компаний или торговыми марками дело обстоит намного сложнее. Действительно, одни названия со временем заменяются на другие (после объединения компаний «*Рамзай*» и «*Айс-фили*» новая компания получила название «*Айсберри*»), старые названия исчезают (авиакомпания *Трансаэро* прекратила свою деятельность в 2015 г.) и постоянно возникают новые (в 2016 г. появилась игра *Pokémon Go*). Таким образом, во многих случаях составление полного словаря именованных сущностей определённой категории не представляется возможным, поэтому приходится учитывать особенности их написания.

Для имён собственных очевидным признаком является регистр составляющих их букв. В большинстве языков мира имена собственные пишутся с заглавной буквы, что позволяет их достаточно легко идентифицировать в тексте, за исключением случаев, когда имя стоит в начале предложения или если все буквы имени заглавные (или строчные). Кроме того, возможны ситуации, когда слово написанное с большой буквы, является именем человека, но употребляется в рамках фразеологизма и не имеет референта:

Что тебе горько? Что ты выросла, как Иван, не помнящий родства?

Особенности написания наименований часто касаются их внутреннего состава и структуры. Например, русские фамилии обычно оканчиваются на *-ов/-ова*, *-ев/-ева* или *-ин/-ина*, а отчества на *-ич/-на*. Определённую структуру может иметь и целое словосочетание, к примеру, наименование российской компании, как правило, начинается с указания ее организационно-правовой формы (*ООО, УП, ГБОУ*), за которой следует название в кавычках: *ПАО «Аэрофлот»*, *акционерное общество «ВИММ-БИЛЛЬ-ДАННН»* и пр.

Для выявления именованных сущностей нередко привлекают контекст: как локальный, т. е. соседние слова (такие как *город*, *улица* и др.,), так и глобальный, т. е. общую информацию об анализируемом тексте (его тематика, структура). Учет локального контекста нередко позволяет определить категорию именованной сущности и ее референт, что особо важно для многозначных наименований, подобных имени *Лена*. Так, на основе контекста, при обработке предложения

Скала Три Сестры расположена к востоку от Уральских гор.

названию *Три сестры* приписывается категория *географический объект* на основе предваряющего ее слова *скала*, а при обработке предложения

А.П. Чехов в 1900 году приступил к работе над пьесой «Три сестры».

будет выявлена именованная сущность *Три сестры* с категорией *художественное произведение*, согласно слову *пьеса*.

Сложности распознавания именованных сущностей могут возникнуть, когда они состоят из нескольких слов, включаются в другие имена/названия или стоят рядом с ними. Обычно считается, что одно наименование не может содержать внутри себя другое, поэтому *ПАО «Сбербанк России»* является единым наименованием несмотря на то, что внутри него присутствует название географического объекта *Россия*. Другая сложность — в том, что наименование объекта может не употребляться в тексте полностью. Например, в тексте

Сбербанк поддержал акцию «Красная гвоздика», которую по всей стране проводит благотворительный фонд «Память поколений». В крупных отделениях банка каждый клиент, совершивший любую операцию, получает значок в форме гвоздики.

в первом предложении встречается только часть официального названия — *Сбербанк*, а во втором вообще только слово *банк*. В идеале, система извлечения именованных сущностей должна уметь разрешать такие ситуации и связывать различные наименования одной сущности между собой.

Подобные ситуации, когда разные наименования указывают по сути на один и тот же референт, называются **кореферентностью**. Разрешение таких ситуаций предполагает выявление всех кореферентных выражений, обозначающих одну и ту же сущность (*Сбербанк, банк*), их отождествление и приведение к каноническому виду (например, к полному названию — *ПАО «Сбербанк России»*).

Для решения проблемы кореферентности и определения соответствующего референта, нередко требуется привлечение внешних источников знаний о существующих в мире персонах и объектах. В качестве такого источника часто выступает Википедия, в которой каждая страница рассматривается как отдельный референт. Рассмотрим в качестве примера следующий фрагмент текста:

Глава МИД РФ рассчитывает, что встреча президентов России и США 7-8 июля внесет ясность в перспективы отношений двух стран. Об этом Лавров заявил на международном форуме «Примаковские чтения», ... «Я надеюсь, что возобладает прагматизм, ...», — заключил министр.

В этом тексте персона *С.В. Лавров*, являющийся министром иностранных дел России, упоминается как *глава МИД РФ*, *Лавров* и *министр*. При этом *глава МИД РФ* (или *министр иностранных дел России*) является самостоятельной сущностью и имеет отдельную страницу в Википедии (заметим, что за всю историю было 4 главы МИД РФ), также в Википедии насчитывается около 70 мужчин по фамилии *Лавров*, а слово *министр* вообще не несёт в себе информации о какой-либо персоне. В данном случае для корректного связывания различных наименований сущности между собой система должна обладать некоторыми знаниями об анализируемых текстах. Поскольку обрабатывается новостной текст, скорее всего речь в нем идёт о настоящем времени (дополнительно можно проанализировать дату новости). В настоящее время министром иностранных дел России является С.В. Лавров. Эта информация должна присутствовать в системе или ее можно получить при анализе соответствующей страницы Википедии. Статья про С.В. Лаврова присутствует в Википедии, следовательно, с этой персоной можно связать два первых упоминания о ней в тексте. А поскольку никакие другие министры в тексте не фигурируют, слово *министр* будет отнесено к той же персоне.

В заключение заметим, что для высокофлективных языков отдельной подзадачей извлечения именованных сущностей может быть нормализация выявленного наименования, т. е. приведение его к начальной форме: например, для встретившегося слова *Сбербанком* это может быть приведение к форме именительного падежа — *Сбербанк*.

3.4 Особенности извлечения атрибутов, отношений и фактов

Если в тексте распознаны значимые сущности (объекты), можно пытаться устанавливать связи между ними: извлекать атрибуты объектов, отношения и факты/события с их предопределёнными заранее параметрами-атрибутами.

В простейшем случае, для извлечения **атрибутов** объектов системе достаточно знать, какая категория сущностей является основной, а какие категории выступают в качестве ее атрибутов. Например, при обработке объявления вида

> *Продам 2-комнатную квартиру по ул. Молодежная. Площадь 45,1 кв.м. Цена договорная.*

для объекта *квартира* могут быть извлечены следующие атрибуты (и их значения):

- количество комнат: *2*;
- адрес: *ул. Молодежная*;
- общая площадь (кв.м.): *45,1*;
- цена: *договорная.*

Ясно, что это не все возможные атрибуты для объекта квартира при ее продаже: в объявлении также может присутствовать информация о наличии лифта, газа, балкона, метраж комнат, тип дома и т. д.

Более сложной задачей является извлечение **отношений** между объектами; обычно рассматриваются отношения только между двумя объектами. Типы извлекаемых отношений опять же зависят от прикладной задачи и предметной области текстов. Так, из новостных текстов можно получить информацию о том, кто какую должность занимает (отношение «*занимать должность*»), из научных статей по химии извлечь информацию о взаимодействии веществ (отношение «*вступать в реакцию*») и др.

Для выявления в тексте отношений так или иначе требуется привлечение информации о типичных конструкциях (контекстах) их выражения. К примеру, отношение «*быть режиссёром*», связывающее имя режиссё-

ра (ИМЯ) и название фильма (НАЗВАНИЕ), употребляется в следующих контекстах:

фильм НАЗВАНИЕ режиссёра ИМЯ

(*Фильм «Хохлатый ибис» режиссёра Ляна Цяо получил Гран-при 39-го ММКФ*)

фильм НАЗВАНИЕ ИМЯ

(*Специальный приз жюри получил фильм «Мешок без дна» Рустама Хамдамова*)

фильм режиссёра ИМЯ НАЗВАНИЕ

(*Ранее приз зрительских симпатий ММКФ получил фильм режиссёра Владимира Котта «Карп отмороженный»*)

Такие конструкции-контексты могут быть зафиксированы и использованы IE-системой либо в виде лингвистических шаблонов (в рамках инженерного подхода), либо в разметке обучающей текстовой выборки (в рамках подхода на основе машинного обучения).

Отдельную сложность при извлечении отношений могут составлять слова, меняющие суть высказывания. Например, для того чтобы извлечь отношение «*быть лауреатом*», связывающее имя персоналии (ИМЯ) и название премии (НАЗВАНИЕ), из текстов вида

В прошлом году президент Сантос стал лауреатом Нобелевской премии мира.

В 1973 году Марлон Брандо во второй раз стал лауреатом премии «Оскар».

необходимо знать, что это отношение выражается конструкцией вида

ИМЯ ... стал лауреатом НАЗВАНИЕ

причём, между ИМЕНЕМ и словом *стал* могут встречаться другие слова (это обозначено многоточием). Но тогда из предложения

Бенедикт Камбербэтч, более известный как исполнитель главной роли в сериале «Шерлок», не стал лауреатом премии «Оскар».

отношение «*быть лауреатом*» будет выявлено ошибочно, поскольку так описанная конструкция не учитывает, что частица *не* меняет смысл глагола *стал* на противоположный.

Отметим, что некоторые виды отношений истинны только в течение определённого промежутка времени. К примеру, если отношение «*сыграть главную роль*», связывает актёра и фильм постоянно, то когда речь идёт о спектакле, главную роль в разное время в нем могут играть разные актёры. Для таких отношений необходимо дополнительно фиксировать момент их извлечения и затем отслеживать возможные изменения связываемых ими объектов.

Наиболее сложной задачей является извлечение информации о **фактах и событиях**, которая актуальна для новостных текстов. В событие обычно вовлечено несколько именованных сущностей, которые связаны определенным набором отношений. Например, событие «*выдача кредита*» фиксирует кто, кому, когда и в каком размере выдал кредит:

Северо-Западный банк Сбербанка выдал в 2016 году оборонным предприятиям кредиты на 15,3 млрд рублей.

Можно заметить, что событие описывается определенным набором атрибутов (параметров) и их значений, в данном примере: заемщик, кредитор, сумма займа, время займа (может быть также добавлен атрибут срок отдачи кредита). Такой набор образует так называемый *семантический фрейм* события [16]; значениями атрибутов выступают именованные сущности.

Это событие связывает компанию-заёмщика (*оборонные предприятия*), компанию-кредитора (*Северо-Западный банк Сбербанка*), время (*2016 год*) и денежную сумму (*15,3 млрд рублей*).

Для извлечения фактов и событий, так же как и при выявления отношений, используется информация о типичных конструкциях их выражения. Распознавание события не представляет проблемы, когда все обязательные его атрибуты содержатся в одном предложении, как в вышеприведенном примере. Однако извлечение событий осложняется, особенно в случае событий со многими атрибутами, тем, что образующие их атрибуты могут располагаться в разных предложениях текста, иногда даже не соседних. К примеру, в тексте

Мировые СМИ обсуждают предстоящую встречу Дональда Трампа с Владимиром Путиным. Уже официально подтверждено, что она пройдет в кулуарах саммита «двадцатки» 7-8 июля.

содержатся данные о событии «*пройдёт встреча*», которое связывает участников встречи, место и время ее проведения. При этом информация об участниках встречи (*Дональд Трамп* и *Владимир Путин*) находится в первом предложении, а информация о месте (*саммит «двадцатки»*) и времени (*7-8 июля*) во втором. В таком случае, после выявления сущностей необходимо проводить слияние данных, полученных из разных предложений в единый фрейм события.

Для общей оценки эффективности методов извлечения событий обычно используется способ, предложенный на конференциях MUC-5 и MUC-6 [13, 20]. Для каждого извлечённого факта проверяется, все ли атрибуты заполнены верно. Если это так, то он считается **корректно извлечённым**. При этом если значение какого-либо атрибута не выявлено (например, место события) и в тексте про это действительно не говорится, то в этом случае пропуск не считается ошибочным (т. к. его вообще нельзя заполнить). Таким образом, факт извлечён корректно, если правильно заполнены значения всех атрибутов, которые представляется возможным заполнить.

Заполненный атрибут считается **частично корректным**, если извлеклась не вся информация, к нему относящаяся, или же извлечена лишняя. Например, значение атрибута времени — *4 квартал*, в то время как в тексте стоит *4 квартал 2010 года*, считается извлечённым частично корректно. Извлечённый факт, имеющий кроме корректных ещё и частично корректные атрибуты, является частично корректным. Точность и полнота рассчитываются в это случае по следующим формулам [13]:

$$P = \frac{correct + 0.5 * partial}{actual}$$

$$R = \frac{correct + 0.5 * partial}{possible},$$

где $correct$ — число корректно извлечённых фактов; $partial$ — количество частично корректных фактов; $actual$ — число всех выявленных фактов;

possible — число фактов, которые можно извлечь из текстов (размеченных экспертом).

3.5 Лингвистические шаблоны и правила

Системы извлечения информации, использующие инженерный подход, опираются в своей работе на лингвистические шаблоны и правила, а также словарные ресурсы.

Лингвистический шаблон представляет собой формальное описание (образец) языковой конструкции, которую необходимо найти в тексте, чтобы извлечь нужную информацию. Шаблон может быть записан регулярным выражением (они встроены во многие языки программирования, а также в инструментальные IE-системы). К примеру, шаблон

[А-Я]\.[А-Я]\. ([А-Я][а-я]*),

где [А-Я] — это заглавная буква, \. — точка, ([А-Я][а-я]*) — слово с заглавной буквы, позволит распознать в обрабатываемом тексте конструкции вида *Н.В. Ушаков*; а шаблон

([А-Я][а-я]*) [А-Я]\.[А-Я]\.

поможет выявить такие фрагменты, как *Ушаков Н.В.*

Лингвистические правила обычно состоят из двух частей. Левая часть правила (часть ЕСЛИ) содержит шаблон (образец) искомой языковой конструкции, а правая (часть ТО), в свою очередь, описывает действия, которые необходимо совершить, например, извлечь составляющие ее элементы (слова), приписать им определённую категорию и т.п. Например:

ЕСЛИ встречается [А-Я]\.[А-Я]\. ([А-Я][а-я]*) или ([А-Я][а-я]*) [А-Я]\.[А-Я]\.,

ТО извлечь этот фрагмент и приписать ему категорию *имя персоналии*

Это правило (в различных системах оно может записываться в разном синтаксисе) позволяет извлечь имя персоналии *А.С. Пушкин* из предложения

Своей родословной А.С. Пушкин очень дорожил и гордился.

В общем случае, процесс извлечения из текста информации по шаблонам и правилам предполагает несколько этапов обработки текста.

Сначала проводится графематический анализ текста, в ходе которого выполняется выделение токенов (последовательностей символов от разделителя до разделителя) и разбиение текста на предложения. Токены классифицируются: слова естественного языка (*около, Петров*), знаки препинания (*!, —*), буквенно-цифровые комплексы (*ВАЗ-21106, 10.05.2003*) и пр. Распознавание различных видов токенов происходит, как правило, на основе регулярных выражений. Аналогичным образом, с дополнительным использованием словарных ресурсов (словарей фамилий, списков дней недели и месяцев и т. д.), распознаются и отдельные категории именованных сущностей (и их атрибуты): фамилии, телефоны, даты, имена файлов и т.п. Поэтому возможно извлечение указанных категорий сущностей уже на этапе графематического анализа, и некоторые графематические анализаторы выделяют их как виды токенов. К примеру, графематический анализатор, входящий в состав проекта АОТ [3], позволяет распознавать ФИО, целые числа, имена файлов и электронные адреса.

Следующим этапом обработки текста является морфологический анализ. В общем случае, в результате его работы каждой словоформе текста приписывается ее начальная форма (лемма), часть речи и значения морфологических признаков: рода, числа, времени и пр. После этого этапа уже возможно извлечение именованных сущностей по шаблонам, в которых указываются части речи слов и их морфологические признаки. Например, шаблон

N<фильм> PERS NAME

где N<фильм> обозначает существительное *фильм* в любой форме (N — Noun), PERS — шаблон имени персоны, а NAME — шаблон названия фильма, позволит выявить имя режиссёра и название снятого им фильма из предложений:

В онлайн-кинотеатре представлен новый фильм Андрея Звягинцева «Нелюбовь».

Он начал карьеру ассистентом монтажёра и режиссёра, в том числе на культовом фильме Сэма Раймми «Зловещие мертвецы».

Данный шаблон можно также использовать в составе набора других лингвистических шаблонов для извлечения бинарного отношения «*быть режиссёром*».

Другой шаблон

PERS<падеж=именит> *работает в* ORG<падеж=предлож>

где PERS<падеж=именит> — шаблон имени персоны в именительном падеже, "*работает в*" — фрагмент теста с определёнными словами, ORG<падеж=предлож> — шаблон названия организации в предложном падеже, позволит выявить из предложения

Петров А.В. работает в ЗАО «Актив»

отношение "*работать в*", связывающее имя персоны (*Петров А.В.*) и название организации (*ЗАО «Актив»*). Заметим, что этот шаблон не сработает для ситуаций, когда глагол *работать* употребляется в прошедшем времени (*Петров А.В. работал в...*), но шаблоны многих инструментальных систем позволяют учесть и такие случаи.

Ещё одним этапом обработки текстов является синтаксический анализ, в ходе которого в общем случае выявляются синтаксические связи слов и строится синтаксическая структура (синтаксическое дерево) предложения. Данный этап является достаточно сложным и ресурсозатратным, поэтому во многих IE-системах проводится лишь частичный синтаксический анализ, при котором распознаются определённые синтаксические конструкции, в первую очередь, грамматически связанные словосочетания. Для учёта таких конструкций в шаблонах записываются условия согласования (равенства) морфологических характеристик слов.

Например, для выявления названий должностей возможно применение правила вида:

ЕСЛИ встречается PERS V<стать, время=прош> NP<падеж=творит>,

PERS=V,

ТО извлечь NP, приписав категорию *название должности*.

Здесь V<стать, время=прош> — глагол (V — Verb) *стать* в прошедшем времени (любого числа и лица), NP — это именная группа (NP — Noun Phrase), структура (шаблон) которой задается отдельно (как и для PERS),

PERS=V обозначает грамматическое согласование, т. е. что число и род для PERS совпадают с числом и родом для глагола.

Предположим, что NG описывается следующими альтернативами:

N | A N (A=N) | A A N (A=A=N) | N N<падеж=родит> | N N<падеж=родит> "и" N<падеж=родит>

где A — это прилагательное (Adjective), N — существительное, а знак равенства означает согласование. Тогда вышеуказанным правилом в предложениях

Сергей Шойгу стал министром обороны.

Маргарет Тэтчер стала министром просвещения и науки.

будут выявлены и извлечены как названия должностей словосочетания *министром обороны* и *министром просвещения и науки.*

Таким образом, в лингвистических шаблонах может записываться:

- информация о составе и структуре конструкции: вхождение в нее конкретных слов (*работает, в*), особенности написания слов (регистр букв, наличие цифр и пр.);
- морфологическая информация (род, число, падеж и т.п.) отдельных слов, входящих в состав конструкции;
- синтаксические свойства: грамматическое согласование слов в словосочетаниях (прилагательного и существительного, подлежащего и сказуемого и др.).

Шаблоны и правила извлечения для IE-систем составляются лингвистами-экспертами (или экспертами в проблемной области) по текстам конкретной, как правило, узкой предметной области, что позволяет достигать достаточно высокой точности извлечения. Для записи шаблонов и правил обычно используются специальные формальные языки, и получающиеся правила обозримы и понятны, что позволяет, путём их корректировки, сравнительно легко исправлять ошибки, влияющие на качество работы систем. В то же время для текстов из более широких предметных областей шаблоны становятся слишком многочисленны и сложны даже для экспертов. Создание полного набора шаблонов, охватывающего искомые языковые конструкции, достаточно трудоёмко, порой очень сложно учесть всевозможные правила именования значимых объектов и разнооб-

разные языковые конструкции, выражающие определённое отношение/событие. Особенно это касается языков с богатой морфологией и относительно свободным порядком слов, как русский язык. Как следствие, показатели полноты извлечения информации получают не очень высокие значения.

Таким образом, системы, основанные на правилах, хорошо применимы к узким предметным областям с чёткими правилами именования значимых объектов и небольшим разнообразием искомых языковых конструкций. Точность извлечения информации в таких системах обычно выше полноты. Заметим, что при переходе к текстам другой предметной области необходимо строить новый набор шаблонов и корректировать состав используемых словарных средств.

Для повышения полноты извлечения информации в рамках инженерного подхода применяются методы автоматического построения лингвистических шаблонов по неразмеченным текстам некоторой предметной области.

Наиболее распространенным на сегодняшний день подходом к автоматизированному построению является итеративный подход (**bootstrapping**), относящийся к методам частичного обучения с учителем и описанный в [34]. Классическими итеративными методами построения шаблонов являются метод DIPRE [10] и метод Snowball [6], аналогичные методы используются в [30, 39].

Наиболее известен метод Snowball [6], в основу которого легла ключевая идея более раннего метода DIPRE, строящего шаблоны для извлечения атрибутов и отношений, например, для поиска местоположения компаний. Основные шаги метода таковы.

1. Составление набора примеров извлекаемого бинарного отношения, т. е. пар конкретных именованных сущностей, связанных этим отношением, например, пар (компания, местоположение штаб-квартиры): (*Microsoft, Redmond*).
2. Поиск составленных пар в заданной коллекции текстов и их извлечение вместе с контекстами, в которых они встречаются.
3. Синтаксический разбор найденных контекстов.

4. Генерация по контекстам новых шаблонов и пополнение ими набора уже имеющихся.
5. Извлечение информации с помощью пополненного набора шаблонов и добавление новых найденных примеров пар к используемому набору.
6. Последовательное повторение шагов 2-5, пока извлекаются новые пары примеров (процесс прекращается, когда новые шаблоны перестают появляться).

Рассмотренный метод может быть применён даже без начального набора шаблонов/примеров (если нет эксперта, который их составит), в этом случае он относится к задаче **open information extraction** [7].

3.6 Машинное обучение в задачах извлечения информации

Наибольшей популярностью пользуются методы **обучения с учителем**, когда по корпусу размеченных данных (обучающей выборке) строится модель (машинный классификатор), которая затем применяется к новым, неразмеченным текстам. Наряду с традиционными методами машинного обучения этого типа, такими как наивный байесовский классификатор, деревья решений, метод опорных векторов, логистическая регрессия, все чаще применяются **скрытые модели Маркова** (**Hidden Markov Models, HMM**), метод **условных случайных полей** (**condition random fields, CRF**) и **нейронные сети**.

Данные, извлечению которых необходимо обучиться, размечены в текстах обучающей выборки определенным образом: для них записаны их лингвистические и структурные признаки, в ряде случаев размечается также их ближайший контекст. Для упрощения работы эксперта, размечающего тексты, нередко предварительно проводится их графематический и морфологический анализ, реже — синтаксический. Используемая разметка и учитываемые признаки, а также методы, применяемые для обучения, во многом зависят от вида извлекаемой информации.

Для разметки категорий именованных сущностей предложены различные схемы [32, 38]. Самой простой является схема IO (I — *inside,* O —

outside), при применении которой токены, относящиеся к имени (*inside*), размечаются его категорией, а токены вне имени (*outside*) размечаются тегом O. Получается, что если система обучается для распознавания имён персоналий (PERS), то все токены текста делятся на два класса: относящиеся к PERS и относящиеся к O. В случае, если дополнительно учитываются географические названия (LOC), то токены будут делиться уже на три класса: PERS, LOC или O. Предложение

Владислав Сурков встретится с президентом Абхазии Раулем Хаджимба

в разметке IO с учетом PERS, LOC и O будет выглядеть следующим образом:

$[Владислав]_{PERS}$ $[Сурков]_{PERS}$ $[встретится]_{O}$ $[с]_{O}$ $[президентом]_{O}$ $[Абхазии]_{LOC}$ $[Раулем]_{PERS}$ $[Хаджимба]_{PERS}$

Основной проблемой схемы IO является то, что она не позволяет отличить ситуацию, когда сущность (например, имя персоналии) состоит из нескольких слов и их надо объединять для получения полного имени:

Савва Морозов происходил из старообрядческой купеческой семьи

от ситуации, когда за одним именем следует другое:

Ведь Путин Медведева и назначал.

Более сложная схема BIO предполагает разметку начала сущности (B — *beginning*) и ее продолжения (I), и так же помечает токены вне имён (O). Если система должна научиться различать категории PERS и LOC, то при обучении все токены текста делятся на классы B-PERS, I-PERS, B-LOC, I-LOC и O, т. е. классификатор для каждого токена учится распознавать, является этот токен началом или продолжением именованной сущности конкретной категории (PERS или LOC), а также случай, когда токен вообще не является частью наименования. В разметке BIO приведенное выше предложение будет выглядеть так:

$[Владислав]_{B-PERS}$ $[Сурков]_{I-PERS}$ $[встретится]_{O}$ $[с]_{O}$ $[президентом]_{O}$ $[Абхазии]_{B-LOC}$ $[Раулем]_{B-PERS}$ $[Хаджимба]_{I-PERS}$

Существуют и более сложные схемы разметки, позволяющие различать однословные и многословные названия; для многословных названий

предполагается разметка не только их начала и продолжения, но и окончания.

Для применения машинного обучения данные после разметки должны быть преобразованы в наборы признаков для каждого токена. Набор признаков обычно включает признаки самого токена, а также признаки, основанные на знаниях, получаемых из внешних источников, в частности, из словарных ресурсов. Причём признаки указываются не только для значимых (извлекаемых) токенов, но и для соседних, обычно берутся два токена слева и справа. К признакам токена обычно относят:

- собственно токен (*Москва, улица* и т.п.);
- вид токена: слово, знак препинания, цифро-буквенный комплекс и пр.;
- длину токена;
- является ли началом/концом предложения;
- тег токена, полученный им при разметке.

Для токенов-слов дополнительно учитывается:

- способ написания токена: только прописными буквами, только строчными, первая буква заглавная и т. д.;
- лемма, часть речи, значения морфологических признаков;
- состав слова: корни, суффиксы и окончания, типичные для фамилий, названий организаций и других категорий сущностей (ов/ова, ин/ина, *гос*, *гор*, *мос* и др.).

Что касается признаков, получаемых из словарных ресурсов, то они, как правило, отвечают на вопрос, входит ли токен в определённый словарь. Используемые словари могут включать:

- частотные имена, отчества и фамилии, названия компаний, фирм и организаций, географические названия;
- слова, являющиеся частями наименований, например, типы организаций (*ОАО, ГП*);
- слова-маркеры, за которыми обычно располагаются именованные сущности определённых категорий (*город, улица, река*).

Использование при извлечении именованных сущностей большого количества словарных ресурсов является особенностью современных систем,

основанных на машинном обучении (а также систем, основанных на инженерном подходе).

Таким образом, задача извлечения именованных сущностей рассматривается как классическая задача классификации токенов на несколько классов. При этом для обучения и использования классификаторов могут применяться разные стратегии. Например, можно научить классификатор одновременно распознавать сущности разных категорий, а можно построить отдельные классификаторы для каждой категории и потом объединять результаты их работы.

В то же время, поскольку при решении задачи извлечения сущностей активно используется локальный контекст классифицируемого токена, то эту задачу можно рассматривать как задачу предсказания последовательности. В этом случае логичнее использовать не классические методы обучения (байесовский классификатор, деревья решений др.), а скрытые марковские модели (HMM) и метод условных случайных полей (CRF), рассматривая категории именованных сущностей как скрытые состояния, а токены — как наблюдаемые.

Современной тенденцией в решении задачи извлечения именованных сущностей является также применение методов **обучения без учителя** (**методов кластеризации**), позволяющих автоматически кластеризовать слова по похожим контекстам их употребления [18, 32]. Важно, что работа этих методов происходит с неразмеченным текстовым корпусом, что позволяет преодолевать ограниченность имеющейся размеченной текстовой коллекции. Заметим, что результаты кластеризации именованных сущностей могут использоваться также как дополнительный признак, основанный на знаниях, при применении методов (частичного) обучения с учителем. Данный признак отвечает на вопрос, входит ли токен в определённый кластер.

В последние годы появились работы [12, 14], в которых применяются нейронные сети и используются подходы на основе глубокого обучения (deep learning), например, технология Word2vec, но в целом они не дали существенного прироста качества извлечения. Особенностью именно методов, использующих нейронные сети, является то, что они позволяют достичь качества, сравнимого с наилучшими современными методами (примерно

91% F-меры), но с минимальным набором дополнительной информации: признаков токенов, словарных ресурсов и пр.

В задачах распознавания отношений и фактов из-за сложностей разметки данных методы обучения с учителем используются крайне редко. Наиболее типично использование методов на основе частичного обучения [11, 26].

В работе [26] был предложен подход, называемый **distant supervision**, в котором для обучения берется большое число примеров сущностей (сотни и тысячи), связанных определенным отношением/фактом. Источником этой информации может быть внешняя база знаний типа Freebase или Википедия. Для формирования обучающей выборки в этом случае делается довольно грубое предположение, что все предложения, содержащие связанные определенным отношением сущности, являются положительными примерами, а предложения, содержащие сущности, не связанные целевыми отношениями, относятся к отрицательным примерам. Таким образом, автоматически готовится обучающая выборка предложений, к которой можно применить машинное обучение с учителем.

Признаки, применяемые при извлечении отношений и фактов, связывающих именованные сущности, в основном учитывают контекст вокруг сущностей:

- список лемм слов, стоящих между сущностями, и их части речи;
- слова и их часть речи слева от левой и справа от правой сущности;
- синтаксический путь между сущностями и его длину;
- категории именованных сущностей.

Отметим, что для корректной работы систем, основанных на машинном обучении с учителем, размеченный корпус (обучающая выборка) должен иметь достаточно большой объем, а также высокое качество разметки. Разметка же текста является непростым и трудоёмким процессом, который порождает достаточно высокий процент ошибок. Дополнительной сложностью может стать выбор подходящего метода обучения. Ещё одно слабое место подхода на основе машинного обучения связано с тем, что результаты работы методов машинного обучения обычно плохо объяснимы, локализовать и исправить возникающие ошибки практически невозможно.

При переходе на другую задачу и предметную область системы, использующие машинное обучение, сталкиваются с теми же проблемами, что и системы, основанные на правилах: систему необходимо настраивать заново. В зависимости от используемого метода может понадобиться обучение системы на новом корпусе и/или корректировка множества учитываемых ею признаков.

Однако уже размеченный корпус и созданный набор различительных признаков можно использовать многократно, пробуя на нем разные методы и стратегии обучения, не привлекая лингвиста для кропотливой работы по анализу текстов предметной области и написанию правил и шаблонов. Это обстоятельство объясняет широкое использование машинного обучения в исследовательских работах по извлечению информации в последние годы.

3.7 Инструментальные системы для извлечения информации

Для построения прикладных систем извлечения информации из текстов может применяться различный программный инструментарий, и число инструментов, так или иначе поддерживающих создание прикладных систем, достаточно велико. В данном разделе рассмотрим наиболее популярные и свободно доступные инструментальные системы, ориентированные в первую очередь на задачи извлечения и позволяющие строить приложения для извлечения информации из текстов на русском языке. Среди них есть системы, поддерживающие инженерный подход, а также системы, поддерживающие машинное обучение.

Инструментальные системы, поддерживающие инженерный подход, обычно имеют встроенный формальный язык описания лингвистических шаблонов распознаваемых конструкций и правил их извлечения. Практически все системы выполняют графематический, морфологический, лексический анализ (в том числе распознавание словосочетаний по словарю), в некоторых системах реализовано разрешение кореферентных связей. Распознавание и извлечение соответствующих правилам и шаблонам конструкций выполняется на базе **частичного синтаксического анали-**

за (**shallow analysis**), предполагающего выделение именных и глагольных групп.

Самой известной и одной из самых старых инструментальных систем является **GATE**, которая создавалась как языково-независимая среда разработки приложений по обработке текстов [8]. Конкретное IE-приложение собирается из предоставляемых системой стандартных программных модулей. Можно включать и свои, дополнительные модули. Обработка текста заключается в последовательном применении выбранных модулей и приписывании ими *аннотаций* определенным фрагментам текста. Аннотации, приписанные одними модулями, используются в своей работе другими (последующими) модулями.

Для записи правил в системе *GATE* используется язык *Jape*. Правило на языке *Jape* состоит из двух частей: **левая часть** задаёт шаблон, определяющий некоторый фрагмент текста (языковую конструкцию) по его аннотациям, а **правая часть** описывает действия с аннотациями найденного фрагмента. К примеру, следующее *Jape*-правило служит для выявления в тексте конструкций вида *Иван родился в Самаре*, т. е. определения места (города) рождения:

Rule: BornPlace
(({Token.kind == word, Token.orth == upperInitial}): person
{Token.string == "родился"} {Token.string == "в"}
({Lookup.majorType == "City"}): city)
–> person.Name ={BirthCity = city.Token.string}

Левая часть этого правила описывает фрагмент текста, первое слово которого начинается с заглавной буквы (*Token.kind == word, Token.orth == upperInitial*), второе и третье — конкретные словоформы *родился* и *в*, а четвертое — название города, описанное в словаре названий городов (аннотацией *Lookup* помечаются слова, найденные в словаре). Правая часть правила создаёт новую аннотацию *Name* с атрибутом *BirthCity*, значением которого берется *Token.string* (строка-название города). Метки *person* и *city* ссылаются на части выявленной конструкции.

Главным достоинством системы *GATE* является ее гибкость и универсальность, которая в то же время требует существенной настройки ее

модулей для построения систем извлечения информации из текстов на высокофлективных языках, например, русском.

Язык **LSPL** и поддерживающий его программный комплекс [1] создавались именно для обработки русскоязычных текстов и учитывают его специфику. Распознаваемая конструкция специфицируется на этом языке в виде *лексико-синтаксического шаблона*, а лингвистическое правило включает кроме *шаблона распознавания* еще *шаблон извлечения* и является, по сути, правилом преобразования текстового фрагмента, найденного по шаблону распознавания, в извлекаемый из него текст.

В шаблоне распознавания указывается последовательность элементов-слов, для каждого из которых могут быть конкретизированы лексема и морфологические характеристики (часть речи, падеж, род, число и т.п.), и задаются условия их грамматического согласования. Шаблон извлечения текста в правой части *LSPL*-правила позволяет выделить фрагменты распознанной конструкции и сформировать из них нужную текстовую строку. К примеру, для извлечения места (города) рождения из конструкций вида *Иван родился в Самаре* или *Нина родилась в Москве* служит *LSPL*-правило:

BornPlace = N V<родиться,t=pres> "в" City <N=V> =text> City

Оно описывает последовательность из существительного (*N*), следующего за ним глагола *родиться* (*V*) в прошедшем времени (*t=pres*), причём существительное и глагол должны быть грамматически согласованы, а также лексемы *в* и шаблона с именем *City*, который задаёт перечень всех названий городов.

Важной особенностью языка шаблонов *LSPL* является возможность указывать условия грамматического согласования, что необходимо для выделения именных групп. Выразительная мощность языка позволяет записать правила, с помощью которых генерируются новые шаблоны, что позволяет проводить многоэтапный анализ текста по шаблонам. В поддерживающий язык программный комплекс входит среда с графическим пользовательским интерфейсом для отладки лингвистических шаблонов.

Система **Томита-парсер** [5] также разработана для русскоязычных текстов и предназначена для извлечения фактов. Для задания распозна-

ваемых в тексте конструкций используются правила, записанные на языке расширенных КС-грамматик. В правилах парсера кроме различных условий на элементы конструкции задаются действия по извлечению распознанных атрибутов факта (**интерпретации**).

Правила Томита-парсера имеют вид $S \to S_1, \ldots, S_n\ \{Q\};$. В левой части (до знака $\to$) указывается один нетерминальный символ S, в правой — список описаний терминальных и нетерминальных символов $S_1, \ldots, S_n$, за которым в общем случае записываются условия Q, применяемые ко всему правилу в целом. В свою очередь, описания S_i состоят из трех частей:

$$N\langle P_1, ..., P_n\rangle interp(I_1; ...; I_n),$$

где N — имя терминала или нетерминала, P_i — пометы-ограничения на свойства N, а I_i — имена **полей** (атрибутов) фактов, куда записывается фрагмент выявленной конструкции. Пометы могут задавать как ограничения на регистр букв, так и грамматические ограничения ($gram$).

Приведем пример грамматики для распознавания в тексте фактов вида *Иван родился в Самаре* и извлечения информации о городе рождения:

Person –> AnyWord <gram="имя">;

Born –> "родиться" <gram="praet, sg">;

City –> Noun<kwtype=city>;

S -> Person<gn–agr[1]> interp(BornFact.Person)

Born<gn–agr[1]> "в" City interp(BornFact.City);

Правило *Person* в грамматике служит для распознавания имени (любая словоформа текста с признаком "*имя*"). Второе правило описывает возможные словоформы глагола родиться в прошедшем времени (*praet*) и единственном числе (*sg*). Правило *City* позволяет распознавать название города (существительное *Noun*, тип которого — *city*). Последнее правило грамматики описывает всю распознаваемую конструкцию-факт: помета *<gn–agr[1]>* указывает на необходимость согласования *Person* и *Born* в роде и числе, а *interp (BornFact.Person)* и *interp (BornFact.City)* задают извлечение распознанных *Person* и *City* как атрибутов факта.

Среди инструментальных систем для разработки приложений по извлечению информации на базе машинного обучения, большинство исполь-

зуют методы обучения с учителем, позволяя как использовать уже обученные модели, так и создавать новые.

OpenNLP [28] — набор инструментов на языке Java для создания приложений извлечения именованных сущностей, определения языка документа, классификации текстов и т. д. Для его использования предоставляется интерфейс командной строки (*Command line interface, CLI*) и программный интерфейс приложения (*Application Programming Interface, API*). В состав *OpenNLP* входят модули, обученные для распознавания имен собственных и числительных и классификацию их на 7 категорий: имена персон, географические объекты, названия организаций, даты, время, процентные величины, денежные суммы. Для применения модулей распознавания необходимо предварительно выполнить графематический анализ текста. В зависимости от настроек создаваемое приложение может извлекать не все, а только выбранное количество категорий.

В *OpenNLP* существует несколько обученных моделей извлечения, которые зависят от языка (испанский, английский, нидерландский) и категории сущности, для которых они были получены. Построение моделей происходило по корпусу новостных текстов, размеченных с помощью схемы BIO.

Инструменты *OpenNLP* позволяют обучить собственную модель для извлечения именованных сущностей; можно обучать модель для нового языка (в том числе, и для русского) и/или для категорий сущностей, не входящих в состав *OpenNLP*. Для обучения требуется подать на вход приложению размеченные по определенным правилам тексты; для получения хорошей модели требуется не менее 15000 предложений. В качестве признаков предлагается использовать заранее определенный набор из 22 признаков токена, но можно сгенерировать и свои дополнительные признаки, включая признаки на основе кластеризации. Для обучения используется метод максимальной энтропии.

Программные средства **Stanford CoreNLP** [37] также предназначены для создания приложений анализа текстов на естественном языке. *Stanford CoreNLP* включает:

- Модуль *Stanford Named Entity Recognizer* (*NER*) для извлечения именованных сущностей.
- Модули *Stanford Relation Extractor* и *Stanford OpenIE* для выявления отношений.
- Модуль *Stanford Pattern-based Information Extraction and Diagnostics* (*SPIED*) для итеративного построения набора шаблонов для извлечения информации.

Все эти модули написаны на языке Java и также предоставляют для своего использования CLI и API.

Модуль *NER* позволяет использовать как уже существующие, так и создавать новые модели на основе метода CRF (теоретически, возможно построение модели и для русского языка). Среди уже обученных есть модели для английского, немецкого, испанского и китайского языков. Модели для английского языка позволяют распознавать от 3 до 7 категорий сущностей; список категорий такой же, что и в *OpenNLP*. Построение моделей производилось по размеченным текстам различных конференций-соревнований, в частности, по текстам MUC-6 и MUC-7; тексты были размечены с использованием схемы IO. При построении своих моделей есть возможность для настройки наборов учитываемых признаков.

Модуль *Stanford Relation Extractor* ориентирован на поиск отношений между двумя сущностями. Обученная для английского языка модель позволяет извлекать 4 типа семантических отношений:

- «*жить в*», связывающее имя персоны и географический объект;
- «*находится в*», связывающее два географических объекта;
- «*располагаться в*», связывающее название организации и географический объект;
- «*работать в*», связывающее имя персоны и название организации.

Для обучения собственной модели необходимо подать на вход данные в предопределенном модулем формате.

Модуль *Stanford OpenIE* служит для выявления отношений в неразмеченных текстах любых предметных областей. Суть применяемого в нем метода [7] заключается в том, что сначала проводится синтаксический анализ текста, затем каждое предложение разбивается на простые. Эти про-

стые предложения без потери их основного смысла сокращаются до более коротких фрагментов (так называемых атомарных предложений), а уже из них выявляются отношения и связываемые ими сущности. Например, предложение

Born in a small town, she took the midnight train going anywhere

будет разбито на атомарные предложения *she took midnight train* и *she born in small town*, из которых будут получены следующие отношения:

(she; took; midnight train) и *(she; born in; small town)*

Извлечение отношений в данном методе опирается не на размеченные тексты или начальное множество шаблонов/примеров, а на основные типы синтаксических зависимостей. Для данного примера это тройки (субъект; *took*; объект) и (субъект; *born in*; объект после предлога *in*).

3.8 Извлечение терминологической информации

Для научно-технических и других специализированных текстов значимыми объектами являются **термины** — слова и словосочетания, называющие понятия определённой предметной области (*коммунальные сооружения, спектральный коэффициент излучения, прерывание от внешнего устройства* и т.п.). Термины, как правило, входят в число наиболее частотных единиц научно-технического текста и достаточно точно отображают его содержание.

Извлечение терминов и их семантических связей из текстов необходимо при решении ряда прикладных задач. В первую очередь — для разработки и пополнения различных терминологических ресурсов, таких как терминологические словари, тезаурусы и онтологии [2]. Указанные словарные ресурсы создаются в результате обработки коллекций текстов определённой предметной области. Кроме того, термины и различные варианты наименования соответствующего понятия в тексте используются для расширения запроса при предметно-ориентированном информационном поиске: для этого применяются уже упомянутые тезаурусы. При автоматической обработке отдельного текстового документа извлечение терминов необхо-

димо для извлечения **ключевых терминов**, построения **глоссария** (перечня основных терминов текста с их определениями) или **предметного указателя**.

Среди терминологических слов и словосочетаний различают:

- общепринятые термины, которые, как правило, зафиксированы в существующих терминологических словарях и извлекаются на основе компьютерных словарей;
- новые термины, возникшие в ходе научных исследований — они, как правило, отсутствуют в словарях, и для их извлечения применяются эвристические методы, опирающиеся на лингвистические и/или статистические критерии [23].

Лингвистические критерии в первую очередь учитывают грамматическую структуру терминологических словосочетаний. Термины преимущественно представляют собой одно-, двух- и трехсловные именные словосочетания, и их структура может быть представлена в виде **грамматического (синтаксического) образца**, который задаёт части речи составляющих термин слов, некоторые синтаксические связи между ними и легко записывается в виде лингвистического шаблона. К примеру, A N — образец двухсловных терминов, состоящих из прилагательного и следующего за ним существительного (*понятийная операция, существенный пример*), а N Ngen — образец терминологических словосочетаний из существительного и существительного в родительном падеже (*квантор общности, анафора рекурсии*) и др.

Распознанные по соответствующему образцу словосочетания могут рассматриваться только в качестве потенциальных терминов (**терминов-кандидатов**). В действительности, среди извлеченных по образцу словосочетаний оказываются не только термины, но и словосочетания общеупотребительной лексики. Так, типичный для терминов образец A N позволяет выявить в обрабатываемом тексте как термины: *вероятностная модель, временной ряд*, так и словосочетания, не являющиеся таковыми: *малая часть, главный недостаток, важная задача.*

Для фильтрации списка терминов-кандидатов, выявленных по образцам, применяется ряд способов, один из самых распространённых — ис-

пользование стоп-слов, к которым относят служебные слова (союзы, местоимения, частицы и т. д.), слова общей и оценочной лексики: *каждый, другой, плохой* и т.п., заведомо не являющиеся терминами в рассматриваемой области. Из множества терминов-кандидатов исключаются сами стоп-слова и словосочетания, которые полностью состоят из них.

Дополнительно отметим, что лингвистические критерии в ряде случаев учитывают особенности морфемной структуры терминов. Так, медицинские термины обычно образуются с помощью греческих или латинских корней и аффиксов: *стенокардия* (*стенос — узкий, кардия — сердце*), *кардиология* (*кардио* (от *кардия*) — *относящийся к сердцу*, *логия* (от *логос*) — *учение*).

Для извлечения терминов целесообразно также учитывать контексты, в которых они употребляются, это особенно действенно для новых терминов. В частности, в научно-технических текстах регулярно используются конструкции, в рамках которых термины определяются или вводятся в употребление. Например, фраза вида

Такая последовательность называется временным рядом

вводит термин *временной ряд*, а фраза

Под адресом возврата понимается адрес...

объясняет термин *адрес возврата*. Последнюю фразу можно формализовать в виде шаблона следующим образом:

под T *<падеж=творит> понимается* D,

где *под* и *понимается* — фиксированные словоформы, T — термин в творительном падеже, D — определение (объяснение) термина. С помощью данного шаблона возможно выявление в тексте не только термина, но и его определения, что необходимо при создании, например, глоссария обрабатываемого документа.

Лингвистические критерии хорошо работают вне зависимости от размера текста и частоты употребления в них терминов, однако обычно они учитывают только типичную грамматическую структуру терминов и типичные контексты (все возможные структуры и контексты описать в шаблонах очень сложно). Для более надёжного извлечения терминов дополнительно используются статистические критерии, которые опираются на

предположение, что наиболее информативные единицы текста имеют тенденцию к многократному употреблению в нем.

Статистические критерии учитывают как частоту встречаемости слов в обрабатываемом тексте или коллекции текстов, так и вычисляемые на базе этих частот статистические величины. К статистическим критериям относится широко применяемая в информационном поиске мера TF-IDF [36], а также применяемый для извлечения длинных терминов показатель C-Value [17]. В среднем, статистические критерии работают тем лучше, чем больше размер обрабатываемого текста или коллекции текстов, поэтому они широко применяются при построении терминологических словарей и тезаурусов по текстам предметной области. Среди статистических мер, используемых для извлечения терминологических словосочетаний, особую роль занимают **меры ассоциации**, оценивающие устойчивость многословных терминов.

Устойчивость словосочетания, т. е. его повторяемость в речи, а также степень связанности входящих в словосочетание слов можно измерить статистически: чем чаще слова встречаются рядом друг с другом на расстоянии 3-5 слов (такое расстояние свидетельствует о наличии синтаксической и/или смысловой связи между словами), тем с большей вероятностью они образуют словосочетание. В компьютерной лингвистике синтаксически правильные словосочетания, устойчивые в статистическом смысле, обычно называются **коллокациями**. Большинство многословных терминов являются коллокациями.

Для выявления коллокаций в ходе обработки текста для каждой пары слов собирается информация о частоте их встречаемости по отдельности и вместе, затем вычисляются значения выбранной меры ассоциации, и пары слов упорядочиваются согласно значениям используемый меры. В общем случае, чем выше значение меры, тем сильнее связаны слова и устойчиво их сочетание. При извлечении терминов берутся пары слов с бóльшим значением меры, порог отсечения определяется эмпирически.

Наиболее часто для выявления терминов как коллокаций используются мера *MI* и ее модификации, а также *t-score*, *Dice*, *log-likelihood* [31]. Например, мера MI:

$$MI = \log_2 \frac{f(a,b)N}{f(a)f(b)}$$

учитывает N — размер корпуса в словах, $f(a)$ — частоту встречаемости слова a, $f(b)$ — частоту встречаемости слова b, $f(a,b)$ — частоту совместной встречаемости слов a и b и оценивает степень зависимости появления двух слов в корпусе друг от друга.

Если выявленные двусловные коллокации рассматривать как единое целое, то с помощью указанных мер можно распознавать в тексте и более длинные словосочетания (трехсловные, четырехсловные и т. д.), что позволяет извлекать с помощью статистических критериев длинные термины с произвольной синтаксической структурой.

Применяемые статистические и лингвистические критерии в полной мере не могут учесть всех особенностей извлекаемых терминов: в текстах достаточно часто остаются нераспознанными малочастотные термины или термины с нестандартной синтаксической структурой, и в то же время извлекается много нетерминологических единиц.

В современных системах извлечения терминов основным способом повышения полноты и точности распознавания терминов является подбор нужной комбинации статистических и лингвистических критериев. Как правило, сначала отбираются слова и словосочетания, удовлетворяющие определенным лингвистическим критериям, а затем полученный список сокращается с помощью статистических критериев. В последнее время для определения наилучшей комбинации признаков, используемых для извлечения терминов из коллекции текстов определённой предметной области, стали привлекать методы машинного обучения [27]. При обучении машинного классификатора используется широкий набор лингвистических и статистических признаков термина, включая различные статистические меры, лингвистические особенности (часть речи и др.), особенности записи слов (регистр букв), что особенно важно для распознавания однословных терминов.

При извлечении терминологической информации отдельной проблемой является распознавание всех вхождений терминов в анализируемый текст с сопутствующим подсчётом частоты употребления, что необходимо

в задачах обработки отдельного текста: для извлечения ключевых слов, построения предметных указателей. Сложности выявления различных вхождений терминов в текст в первую очередь связаны с тем, что термины при употреблении достаточно часто видоизменяются — усекаются, сокращаются, заменяются синонимами, соединяются и т. д. [21, 22]: *коммуникативная многозначность запроса — коммуникативная многозначность*, *синтаксическое представление — СинП*, *вложенный файл — вложение*. Подобные текстовые варианты представляют собой различные формы выражения одного и того же понятия и по возможности должны быть распознаны.

Для выявления текстовых вариантов терминов обычно используются правила их образования, записываемые по отдельности для каждого грамматического образца термина [21]. Например, правило вида A N $\longrightarrow$ A A N описывает варьирование английских терминов вида A N (прилагательное и следующее за ним существительное), и позволяет, в частности, для термина *acidic protein* (*кислый белок*) распознать в тексте его вариант *acidic epidermal protein* (*кислый белок эпидермиса*). В целом правила варьирования терминов зависят от конкретного естественного языка.

Важной задачей извлечения терминологической информации является выявление семантических связей терминов, к которым относятся:

- синонимическая связь (*компьютер — ЭВМ*);
- род-вид (*регистр — регистр общего назначения*);
- часть-целое (*автомобиль — двигатель*);
- причина-следствие (*кипение жидкости — испарение жидкости*) и т. д.

Выявление связей терминов опирается на распознавание в тексте типичных языковых конструкций [29]: учитывается, что каждый вид связи употребляется в рамках своих типичных конструкций. Синонимы нередко вводятся в конструкциях определения термина, например, фраза

Такие операции будем называть понятийными операциями (понятийными функциями)

вводит новый термин *понятийная операция* и его синоним *понятийная функция*. Конструкция вида

such T1 *as* T2 ,

где T1 и T2, — термины, позволяет выявить из фразы

such crimes as money laundering (*такие преступления, как отмывание денег*)

термины *crimes* и *money laundering* и связать их отношением род-вид.

Отметим, что методы, разработанные в рамках других задач извлечения информации нередко успешно применяются и для извлечения терминологической информации, например, частичное обучение с учителем [9].

3.9 Заключение

Извлечение информации из текстов — достаточно развитое направление компьютерной лингвистики и автоматической обработки текстов, предлагающее широкий спектр методов и соответствующих инструментальных средств для построения различных прикладных систем, а также демонстрирующее достаточно эффективное решение задач извлечения разнотипной информации.

Актуальность задач направления сохраняется: ясно, что построение эффективной IE-системы может значительно облегчить последующую обработку извлечённых структурированных данных, что является ключевым моментом в жизненном цикле накопления и использований новых знаний (**Knowledge Discovery**) [5].

Современными тенденциями развития данного направления являются:

- расширение использования разных факторов и ресурсов, в частности, больших внешних ресурсов знаний (Википедия, DBPedia, WordNet, графы знаний и др.);
- учёт при извлечении нелокальных зависимостей текстовых единиц;
- проведение более глубокого синтаксического анализа и использование синтаксических признаков при машинном обучении;
- сдвиг фокуса от структуризации извлечённой информации к ее визуализации, удобной для человека-аналитика, с предоставлением ему инструментов для просмотра и редактирования данных.

3.10 Список литературы

[1] Большакова Е.И., Носков А.А. Программные средства анализа текста на основе лексико-синтаксических шаблонов языка LSPL // Программные системы и инструменты: Тематический сборник, № 11 / Под ред. Королева Л.Н. — М.: МАКС Пресс, 2010, с. 61-73.

[2] Лукашевич Н. В. Тезаурусы в задачах информационного поиска. — М.: МГУ, 2011.

[3] Сокирко А.В. Морфологические модули на сайте www.aot.ru // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции Диалог'2004 / Под ред. И.М.Кобозевой, А.С. Нариньяни, В.П. Селегея. М.: Наука, 2004. с. 559-564.

[4] Томита-парсер. Руководство разработчика. URL: `https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/`.

[5] Хорошевский В.Ф., OntosMiner: Семейство систем извлечения информации из мультиязычных коллекций документов // Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004: Труды конференции. В 3-х т. — М.: Физматлит, 2004, т. 2, с. 573-581.

[6] Agichtein E., Gravano L. Snowball: extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Int. Conference on Digital Libraries, New York, 2000, pp. 85-94.

[7] Angeli G. et al. Manning. Leveraging Linguistic Structure For Open Domain Information Extraction. In Proceedings of the Association of Computational Linguistics (ACL), 2015.

[8] Bontcheva K., Maynard D., Tablan V., and Cunningham H. GATE: A Unicode-based infrastructure supporting multilingual information extraction. In: Proceedings of Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL'03), Borovets, 2003.

[9] Bosma W., Vossen P. Bootstrapping Language Neutral Term Extraction. In: Proceedings of the 7th Language Resources and Evaluation Conference, LREC, Valetta, 2010, pp. 2277-2282.

[10] Brin, S. Extracting patterns and relations from the World-Wide Web. In: Proceedings of International Workshop on the World Wide Web and Databases (WebDB'98), LNCS N 1590, Springer, 1998, pp. 172–183.

[11] Bunescu R., Mooney R. Learning to extract relations from the web using minimal supervision. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Prague, 2007, pp. 576–583.

[12] Chiu, Jason P. C. and Eric Nichols. Named Entity Recognition with Bidirectional LSTM-CNNs. TACL 4 (2016): 357-370.

[13] Chinchor N. MUC-5 Evaluation Metrics. In: Fifth Messages Understanding Conference (MUC-5), Morgan Kaufman, 1993.

[14] Collobert R. et al. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 2011, 12:2493– 2537.

[15] Doddington G. R. et al. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In: Proceedings of the 7th Language Resources and Evaluation Conference, LREC, 2004.

[16] Feldman R., Sanger J. (ed.). The text mining handbook: advanced approaches in analyzing unstructured data. — Cambridge University Press, 2007.

[17] Frantzi K., Ananiadou S., Mima H. Automatic Recognition of Multi-Word Terms: The Cvalue/NC-value method // C. Nikolau et al. (eds.): International Journal on Digital Libraries. — 2000. — Vol. 3(2). — P. 115-130.

[18] Gareev R., Tkachenko M., Solovyev V., Simanovsky A., Ivanov V. Introducing baselines for Russian named entity recognition, Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, 2013, pp. 329–342.

[19] Grishman R. Information Extraction. In: The Handbook of Computational Linguistics and Natural Language Processing. A. Clark, C. Fox, and S. Lappin (Eds), Wiley-Blackwell, 2010, pp. 515-530.

[20] Grishman R., Sundheim B. Message Understanding Conference — 6: A Brief History. In: Proceedings of COLING-1996, NY, 1996, pp. 466-471.

[21] Jacquemin C., Tsoukermann E. NLP for term variant extraction: synergy between morphology, lexicon, and syntax // Strzalkowski T. (ed.): Natural Language Information Retrieval. — Dordrecht: Kluwer Academic Publishers, 1999. — P. 25-74.

[22] Justeson J., Katz S. Technical terminology: some linguistic properties and an algorithm for identification in text // Natural Language Engineering. — 1995. — Vol. 1(1). — P. 9-27.

[23] Korkontzelos I., Ananiadou S. Term Extraction. In: Oxford Handbook of Computational Linguistics (2nd Ed.). Oxford University Press, Oxford, 2014.

[24] Marsh E., Perzanowski D. MUC-7 evaluation of IE technology: Overview of results. In MUC-7, volume 20, 1998.

[25] Maybury M. Multimedia Information Extraction: Advances in Video, Audio, and Imagery Analysis for Search, Data Mining, Surveillance and Authoring. Wiley-IEEE Computer Society Press. 496 pp.

[26] Mintz M., Bills S., Snow R., Jurafsky D. Distant supervision for relation extraction without labeled data. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th Int. Joint Conf. on Natural Language Processing, 2009, pp. 1003–1011.

[27] Nokel M.A., Bolshakova E.I., Loukachevich N.V. Combining Multiple Features for Single-Word Term Extraction. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" (2012). Issue 11. Vol. 1 of 2. Main conference program. Moscow, RGGU, p.490-501.

[28] OpenNLP. URL: `http://opennlp.apache.org`.

[29] Paice C., Jones P. The Identification of Important Concepts in Highly Structured Technical Papers // Proceeding of 16th Annual International Conference on Research and Development in Information Retrieval. — 1993. — P. 69-78.

[30] Pasca M., Lin D., Bigham J., Lifchits A., Jain A. Names and Similarities On The Web: Fact Extraction In The Fast Lane. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL), 2006, p. 809–816.

[31] Pazienza, M. T., Pennacchiotti, M. and Zanzotto, F. M. Terminology extraction: An analysis of linguistic and statistical approaches. In S. Sirmakessis (eds.), Knowledge mining: Proceedings of the NEMIS 2004 final conference, p. 255–279, 2005, Berlin Heidelberg. Springer.

[32] Ratinov L., Roth D. Design Challenges and Misconceptions in Named Entity Recognition, Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Boulder, 2009, pp. 147–155.

[33] Riloff E.: Automatically constructing a dictionary for information extraction tasks. In: Proceedings of Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington, DC, 1993, pp. 811–816.

[34] Riloff E., Jones R. Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999.

[35] Sekine's Extended Named Entity Hierarchy. URL: http://nlp.cs.nyu.edu/ene/.

[36] Sparck Jones K. A Statistical Interpretation of Term Specificity and its application in retrieval // Journal of Documentation. — 1972. — № 28. — P. 11-21.

[37] Stanford CoreNLP. URL: https://stanfordnlp.github.io/CoreNLP/.

[38] TimeML. URL: http://www.timeml.org.

[39] Trampus M., Mladenic D. Learning Event Patterns from Text. Informatica, Vol. 35, 2011.

[40] Wiil U. Counterterrorism and Open Source Intelligence. Lecture Notes in Social Networks. Springer, 2011.

Глава 4

Автоматические методы анализа тональности

Лукашевич Н.В.

4.1 Введение

Автоматический анализ тональности текстов, т. е. выявление мнения автора текста по поводу предмета, обсуждаемого в тексте, является одной из активно развиваемых технологий в сфере автоматической обработки текстов в последнее десятилетие. Актуальность этого приложения во многом связана с развитием социальных сетей, онлайновых рекомендательных сервисов, содержащих большое количество мнений людей по разным вопросам, в частности, о разных товарах, услугах.

Большое количество работ посвящено анализу тональности отзывов пользователей, которые они оставляют в рекомендательных сервисах [96, 63]. Важное направление анализа тональности связано с так называемым мониторингом репутации компании, такой мониторинг состоит в отслеживании позитивных и негативных отзывов о компании и ее деятельности, и формирование стратегии реагирования на поступающие негативные отзывы [12].

Анализ тональности финансовых отчетов и финансовых новостей используется в задачах определения трендов на фондовом и валютных рын-

ках [24, 85, 86]. Тональность упоминания терминов в научных статьях используется для предсказания наиболее важных понятий и научных трендов [73]. Оценочная направленность текстов может быть использована для определения личностных характеристик автора текста [102, 124].

Растет роль автоматических методов анализа тональности сообщений в социальных сетях для политических и социальных исследований, включая выявление политических предпочтений [123], предсказание результатов выборов [121, 120], выявление отношения к различным политическим решениям. Также автоматический анализ тональности может использоваться для выявления высказываний, содержащих ненависть и призывающих к розни, фейковых новостей и др. [125].

Задачей первых подходов к анализу тональности текстов было определить общую тональность документа или его фрагмента [96]. Такой уровень анализа предполагает, что каждый документ выражает единое мнение по поводу некоторой единичной сущности, как например в отзыве о некотором товаре.

Поскольку в документе может быть выражена разная тональность по отношению к разным упомянутым в нем сущностям, то на следующем этапе стали решаться задачи анализа тональности по отношению к заданным сущностям, упомянутым в тексте [12, 46].

Наконец, еще более детальным уровнем анализа тональности текстов является анализ мнения по конкретным свойствам или частям (так называемым аспектам) сущности, по которым автор текста может высказывать разную тональность мнения [18, 42, 62, 63, 101].

В [62, 63] мнение определяется как пятерка $\langle e_i, a_{ij}, s_{ijkl}, h_k, t_l \rangle$, где e_i — это сущность, к которой относится мнение, a_{ij} — это аспект (часть или характеристика) сущности, s_{ijkl} — это тональность мнения относительно этой сущности и данного аспекта, h_k — это автор мнения, t_l — это время, в которое мнение высказано. При этом мнение s_{ijkl} может быть положительным, отрицательным или нейтральным, или может выражаться с разной степенью интенсивности, измеряемой, например, по шкале 1–5.

4.2 Сложности анализа тональности текстов

4.2.1 Жанры текстов по тональности

Подходы к извлечению основных компонентов мнения в значительной мере зависят от жанра анализируемого текста. Так, одним из наиболее изученных жанров текста в задаче анализа тональности являются отзывы пользователей о товарах или услугах. Такие тексты чаще всего характеризуются тем, что во всем тексте рассматривается одна сущность (но, возможно, в ее разных аспектах), а мнение выражается одним автором, а именно автором отзыва.

Хотя и в отзывах встречаются отклонения от этого основного принципа, осложняющие процесс извлечения мнений. Например, в отзыве о фильме может встретиться предложение *Книга была лучше*, что является негативным мнением о фильме. Посещение ресторана может сравниваться с прошлым посещением, или с другим рестораном, например, *Очень расстроена, в прошлый раз еда была вкуснее* [66]. Также может упоминаться мнение других людей. Но в целом, отзывы — это тексты, выражающие мнение одного автора по отношению к одной сущности.

Другой тип оценочных текстов, в которых чаще всего имеется один автор мнения, но большое количество разных оцениваемых сущностей, представляют собой тексты личных блогов, которые также могут осложняться упоминанием мнений других людей.

В таких жанрах документов, как новостные тексты, или особенно аналитические тексты, может одновременно упоминаться множество мнений с разными авторами и разными объектами оценки. Например, аналитический текст может рассматривать отношения между странами, в которых выражено оценочное отношение стран друг к другу, кроме того, упоминать мнение третьих лиц по поводу каких-либо субъектов или ситуаций, а также ещё и содержать мнение автора по поводу упомянутых субъектов и/или ситуаций. Понятно, что в текстах с множественными авторами или объектами мнения сложность качественного автоматического анализа тональности многократно возрастает.

Большое влияние на особенности анализа тональности текстов имеет также длина анализируемого текста. Короткие тексты, например, сообщения Твиттера, краткие комментарии, требуют очень точного анализа.

В текстах большей длины высказываемое мнение может быть повторено несколько раз в разных вариантах, что облегчает анализ. Однако в длинных текстах нарастает разнообразие объектов, которые подвергаются оценке. Длинные тексты могут включать мнения других людей. Если задача состоит в том, чтобы найти оценку по отношению к упоминаемым сущностям, то возникает проблема определения сферы действия оценок. Например, часто оценку связывают с сущностью, упоминаемой в том же предложении. Но автор может сослаться на объект с помощью средств референции, например, местоимений. Кроме того, если весь текст посвящен обсуждению одной сущности, то она может быть явным образом упомянута достаточно далеко от места расположения оценки [19].

4.2.2 Эксплицитные и имплицитные оценки

Обычно предполагается, что тональность выражается с помощью оценочной лексики, что представляет собой эксплицитный способ выражения оценок. Вместе с тем оценка может выражаться и имплицитным способом с использованием оценочных фактов [63, 66] или слов с коннотациями [11, 39].

Например, в отзывах о ресторанах могут встретиться предложения *долго ждали* или *в супе плавает муха*, что, с одной стороны, описывает происходящее (сообщает реальные факты), с другой стороны сообщает и оценку этому происходящему.

Согласно определению [62], имплицитное мнение (оценка) — это объективное высказывание, из которого следует оценка, т. е. имплицитное мнение сообщает желательный или нежелательный факт. При подготовке размеченных коллекций для тестирования систем анализа тональности такие оценочные факты могут специально размечаться [66, 91].

Коннотации — это оценочные ассоциации слов [11, 39], появление в тексте слов с положительными или отрицательными коннотациями коррелирует с соответствующими оценками, выражаемыми в тексте. Так, в

отзывах о фильмах словами с положительными коннотациями обычно являются имена известных актеров. В отзывах о ресторанах на русском языке отрицательными коннотациями обладают такие слова, как *майонез* и *клеенка*. Если эти слова появляются в отзыве, обычно в этом месте выражается негативная оценка, например,

Вместо скатерти может быть клеенка.

Ассортимент в салат-баре снизился до 2-х салатов и 2-х соусов (один из которых — майонез).

Очевидно, что анализ таких имплицитных видов тональности особенно сложен, поскольку в значительной мере зависит от предметной области, т. е. оценочные факты невозможно заранее собрать для множества возможных областей; нахождение и извлечение этих фактов из текстов также достаточно сложно из-за вариативности их выражения.

4.2.3 Многозначность оценочной лексики. Зависимость тональности слова от контекста

Однако и с трактовкой явной оценочной лексики могут возникать сложности. Слова могут быть многозначными, при этом в одном значении они могут быть нейтральными, а в других значениях негативными или позитивными. Например, слово *пресный* в словосочетании *пресная вода* является нейтральным, возможно с некоторой положительной коннотацией. В то время как в других значениях *пресный на вкус*, и *пресный как неинтересный* данное слово несет негативную оценку [68].

Слово может менять свою полярность или терять полярность в зависимости от предметной области или текущего контекста. Например, слова *подлый* и *предательство* не являются оценочными в области отзывов о фильмах, поскольку не могут использоваться в качестве оценивания чего-либо в фильмах. А если эти слова встречаются в отзывах зрителей, то относятся к пересказу содержания фильма.

Слово *смешной*, скорее всего, окажется негативным в сфере политики, и выражает положительную тональность, если речь идет о комедиях.

При характеризации других жанров фильмов это слово может быть как положительным, так и отрицательным.

Внутри предметной области оценочные слова могут нести положительную или отрицательную тональность в зависимости от аспекта (характеристики) объекта, к которому они применяются. Например, слово *долго* может быть как отрицательным, так и положительным в предметной области цифровых камер: если говорят, что батарейка живет долго, то это хорошо; если говорят, что нужно долго настраивать фокус, то это плохо [35].

4.2.4 Модификаторы полярности: отрицание, интенсификаторы и др.

Появление оценочных слов в тексте может сопровождаться словами-модификаторами, которые усиливают (например, *очень, более*), снижают (*слишком, менее*) или преобразуют в обратную исходную тональность (например, отрицание: частицы *не, нет*), которая ассоциируется с данным словом. Таким образом, при анализе тональности нужно учитывать такие модификаторы и иметь некоторую численную модель, которая модифицирует исходные полярности слова [114, 126, 128]. Одна из распространенных моделей трактовки модификаторов тональности приписывает им некоторые коэффициенты, которые рассматриваются как множители относительно априорной полярности слов, к которым относятся эти модификаторы.

Другой важной проблемой является определение сферы действия модификатора полярности в конкретном предложении, например, отрицания. Например, в предложении *Мне не нравится дизайн новой модели, но в ней есть некоторые интересные функции*, частица *не* относится только к слову *нравится*, и не модифицирует полярность слова *интересный*.

4.2.5 Факторы «нереального» контекста в анализе тональности

При анализе тональности важно учитывать, насколько то, что оценивается, соответствует реальности. Например, в предложении *Мы надея-*

лись, что фильм нам понравится употребляется слово *понравится* с положительной тональностью, однако здесь ничего не говорится о том, понравился ли нам фильм на самом деле, т. е. в процессе автоматического анализа текста данное слово не должно учитываться, как свидетельство позитивного отношения к фильму.

В лингвистике имеется понятие ирреализа или ирреального наклонения [99], которое определяется как группа грамматических средств, используемая для обозначения того, что сообщаемое в предложении не относится к тому, что реально происходит.

Для русского языка в работе [58] отдельно тестируется система правил для обработки тональности предложений, в которых встретились маркеры ирреализа, включая вопросительные знаки, условные обороты со словом *если*, частицы *ли* и *бы*. При подборе параметров на обучающем наборе цитат для тестирования систем анализа тональности РОМИП–2013 [30] было выявлено, что оценочные слова, найденные в предложениях, в которых встречаются данные маркеры, оптимально учитывать со снижением их априорной оценки тональности.

4.2.6 Сравнения

Сравнения усложняют процесс определения тональности, поскольку вводят в текст некоторые дополнительные сущности, и часть упоминаемых оценок относится именно к ним. Такие дополнительные сущности иногда очень трудно выделить самих по себе, а также отделить относящиеся к ним оценки, например, в отзыве про фильм «Левиафан» упоминается еще два фильма:

Фильм замечательный, он получил множество наград. Но я бы не сказала, что он лучше более ранних работ на подобную тему. Мне, например, гораздо больше понравился фильм 2004 года «Именины» — режиссера Валерия Наумова. А в восторг привел фильм еще более раннего выпуска 2001 года с очень плохим названием «Механическая сюита» режиссера Дмитрия Месхиева.

Впрочем, для автоматической системы данный пример не самый сложный, поскольку тональность по отношению ко всем упоминаемым фильмам положительная. Более сложная ситуация возникает в следующем фрагменте отзыва о ресторанах:

Зимой довольно часто посещала это место и была в восторге, все было на высоте — атмосфера дружеская, обслуживание супер... Была на выходных, и разочаровалась.

Здесь мы видим большое количество позитивных слов, которые, однако, не относятся к текущему посещению ресторана. Кроме того, достаточно трудно автоматически определить, что в данном отзыве содержится сравнение, поскольку речь идёт не о сравнительных оборотах, а именно смысловом сравнении разных сущностей в тексте [63, 90, 129].

4.2.7 Ирония и сарказм

Обработка иронии и сарказма являются серьёзными проблемами в работе автоматических систем анализа тональности, поскольку тональность ироничного (саркастичного) высказывания отличается от его буквальной тональности [2].

В различных работах встречаются различные определения иронии [2, 127]. В работе [20] предлагается обобщающее понимание иронии как несоответствие между буквальным значением высказывания и его подразумеваемым значением. Чаще всего, за положительно выглядящим высказыванием (содержащим больше положительных оценочных слов или равное количество положительных и отрицательных слов) скрывается отрицательное мнение, например, *Сбербанк — самая крупная сеть неработающих банкоматов в России* (пример из [5]). Сарказм рассматривается как более резкая, агрессивная, возможно унижающая форма высказывания [20].

Разметка текстовых данных для изучения иронии и сарказма представляет собой сложную задачу. Интересным ресурсом для анализа этих явлений являются сообщения Твиттера, которые пользователь может разметить специализированными хештегами: #ирония, #сарказм и некоторыми другими [104, 113]. Однако последние исследования иронии в Твиттере

показывают, что ироничные твиты, отмеченные хэштегами и не отмеченные, имеют разные характеристики [57].

4.3 Словарные ресурсы для анализа тональности

Поскольку, по большей мере, тональность в тексте выражена лексическими средствами (словами и выражениями), то для разных языков существуют опубликованные словари оценочной лексики. Такие словари могут быть созданы вручную или автоматически. Несмотря на то, что в каждой конкретной предметной области нужны специализированные словари, общие словари также полезны, поскольку могут служить исходным материалом, который в процессе работы может быть уточнён и дополнен.

Созданные словари оценочной лексики могут быть представлены в виде простых списков слов с некоторыми атрибутами. Также разметка тональности слов может быть выполнена с учётом значений слов, так, что каждое значение слова получает свою отдельную оценку тональности.

Кроме того, из-за того, что имеется высокая зависимость словарей оценочной лексики от конкретной предметной области, то имеется значительное число работ, которые посвящены извлечению оценочной лексики из текстов заданной предметной области [51, 59, 103, 47, 48].

4.3.1 Словари оценочной лексики для английского языка

Больше всего словарей оценочной лексики создано для английского языка.

Наиболее ранним из известных словарей оценочной лексики английского языка является словарь General Inquirer [111], который был создан для автоматизированного контент-анализа текстов. Словарь содержит списки слов по категориям тональности (позитивная и негативная), по категории силы тональности (сильная и слабая), по категориям ощуще-

ний (удовольствие, боль, моральные оценки) и др. Словарь используется во многих современных работах по анализу тональности [128, 88, 62].

Созданный в 1999 году словарь ANEW [25] описывает около 1000 слов английского языка по трём 9-балльным шкалам: удовольствие — неудовольствие, возбуждённость — спокойствие, контролирующий (например, *авторитарный*) — контролируемый (например, *послушный*). Оценки собирались у студентов, носителей английского языка. Каждое слово должно быть оценено по всем шкалам. Так, слово *afraid* (*бояться*) получило низкие средние баллы по шкале удовольствия (2.00) (мало удовольствия), достаточно высокие по шкале возбужденности (6.67) (возбужденность присутствует) и относительно низкие по шкале контролирумости (3.98) (контролируемость имеется).

Один из известных словарей оценочной лексики английского языка MPQA [128, 34] был составлен из нескольких источников (ручных и автоматически порожденных словарей оценочных слов) и содержит свыше 8000 отдельных слов. Слова в словаре размечены метками полярности (позитивный, негативный или нейтральный), и оценочные слова снабжены пометами силы оценочного содержания (сильный или слабый). Приведем пример нескольких строк из этого словаря:

type=weaksubj len=1 word1=abandonment pos1=noun stemmed1=n priorpolarity=negative
type=weaksubj len=1 word1=abandon pos1=verb stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abase pos1=verb stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abasement pos1=anypos stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abash pos1=verb stemmed1=y priorpolarity=negative

В работах [17, 37] описывается словарь SentiWordNet, который основан на тезаурусе английского языка WordNet. Он получен в результате автоматической разметки синсетов (=наборов синонимов) тезауруса WordNet [38], в результате чего каждому синсету поставлено в соответствие три числа, которые обозначают долю позитивности (P), негативности (N) и нейтральности (=объективности O) слов из данного синсета. Подход основан на использовании толкований слов. Предполагается, что слова с одинаковой оценочной ориентацией имеют «похожие» толкования, например, для слова *отличный* толкование будет: *очень хороший, высшего качества.*

Таким образом, разные значения одного и того же слова могут иметь различные оценки тональности. Например, прилагательное *happy* имеет четыре значения: первые два значения выражают радость и имеют высокие положительные значения (P=0.875 и P= 0.75) соответственно. Третье значение, как в словосочетании *happy to help*, неоднозначно (P=0.5 и O=0.5). Наконец, последнее значение было размечено как наиболее объективное (P=0.125 и O=0.875).

Словарь WordNet-Affect [112] представляет собой разметку синсетов тезауруса WordNet специализированными метками типов эмоций, черт характера, физического состояния, эмоционального состояния и др. Кроме того, сделана разметка по полярности, в которой использовано четыре значения: позитивное, негативное, неоднозначное (например, удивление), нейтральное.

В словаре SenticNet [27] слова и выражения размечены по четырем измерениям: приятность (pleasantness), внимание (attention), восприимчивость (sensitivity), склонность (aptitude). Для получения числовых оценок авторы использовали оценочные слова и соответствующие веса, определенные в Hourglass of Emotions [26] как начальное множество для получения оценок тональности для остальных понятий. Авторы данного словаря уделяют особое внимание выражениям, в состав которых входят градуальные прилагательные, которые не имеют априорной тональности (*большой* и др.). Последняя версия SenticNet содержит около 30 тысяч слов и выражений.

Словарь оценочных слов AFINN [88] был специально создан для анализа постов в социальных сетях, включает ругательные и сленговые слова. Он содержит около 2400 слов, помеченных числовым весом полярности, изменяющегося от -5 (очень негативный) до $+5$ (очень позитивный):

abandon -2, abduction -2, abhor -3, abusive -3, accept 1 ..

Лексикон оценочных ассоциаций (Word-Emotion Association) Исследовательского центра Канады (NRC Canada) был создан с помощью краудсорсинга, т. е. путем опроса обычных людей, и содержит слова и выражения, которые имеют ассоциации с тональностью и определенными эмоциями [79]. Эмоциональная разметка осуществлялась по категориям: ра-

дость, грусть, страх, гнев (anger), доверие, отвращение, удивление, ожидание (anticipation).

Таким образом, для английского языка имеется набор разных словарей с информации о тональности слов. Применение конкретных словарей для анализа тональности дает разные результаты. Например, в работе [82] сравниваются словари SentiWordNet, WordNet-Affect, MPQA, и SenticNet для анализа тональности сообщений Твиттера. Лучшими оказались лексикон среднего размера MPQA и большой лексикон SentiWordNet. В работе [92] также проводилось сравнение нескольких лексиконов оценочных слов на материале классификации сообщений Твиттера. Набор словарей включал: словарь MPQA, словарь AFINN и словарь из работы [50]. Лучшим оказался небольшой по величине словарь AFINN.

Таким образом, словари оценочной лексики, созданные для одного языка, в значительной мере различаются между собой по покрытию, а также могут различаться по приписанным оценкам тональности для конкретных слов [20].

4.3.2 Автоматическое порождение оценочных словарей

Большое внимание уделяется автоматизации построения словарей оценочной лексики для конкретных языков или предметных областей. Словарь оценочной лексики для заданного естественного языка может быть создан посредством перевода и интеграции оценочных словарей, существующих на других языках [74, 98, 110, 41].

Отдельным направлением исследований является автоматическое извлечение из текстов оценочных слов и выражений, поскольку подчеркивается, что используемые оценочные выражения в значительной степени зависят от предметной области и от типа оцениваемой сущности.

В классической работе [49] выделение оценочных прилагательных и определение их семантической направленности основано на синтаксических шаблонах и союзах И, ИЛИ, НО. Предполагается, что если два прилагательных связаны союзами И или ИЛИ, то они оба являются или не явля-

ются оценочными, а так же одинаково семантически направлены. В случае союза НО, семантическое направление различается. В результате был построен классификатор связей, работающий с точностью 82%. На последнем шаге выполнялась кластеризация слов, в результате которой образовалось два кластера, больший из которых выбирался положительным. Точность кластеризации 92%. Реализация и тестирование похожего алгоритма для русского языка описаны в работе [36].

Для получения оценочных слов и вычисления их направленности могут использоваться словари и тезаурусы. Метод, предложенный в [50], предполагает использование тезауруса для обогащения, заданного вручную, эталонного множества оценочных слов. Основная идея состоит в том, что если слово оценочное, то его синонимы, гипонимы также будут оценочными и одинаково семантически направлены, в случае антонимов — противоположно направлены.

Важной задачей является создание словарей оценочных слов и выражений для конкретных предметных областей, поскольку такой словарь является в значительной степени зависимым от предметной области: некоторые оценочные выражения употребляются только в конкретных предметных областях, другие являются оценочными в одной области и не являются оценочными в другой.

Один из частых подходов извлечения словаря оценочных слов для заданной предметной области состоит в задании набора общезначимых оценочных слов, а затем пополнения этого набора на основе корпуса текстов [51, 59, 103, 47, 48].

В работе [101] описана система OPINE, которая служит для извлечения из отзывов атрибутов описанных продуктов, а также оценок по ним. OPINE выделяет следующие атрибуты продукта: свойства продукта, части продукта, атрибуты частей продукта, связанные сущности, свойства и части связанных сущностей. Предполагается, что оценочные фразы появляются в непосредственной близости от атрибутов объекта. Для извлечения оценочных слов используется 10 правил, основанных на синтаксической структуре предложения:

(M, NP=f) $\Rightarrow$ po=M: (expensive) scanner

(S=f, P, O) ⇒ po=O: lamp has (problems)
(S, P, O=f) ⇒ po=P: I (hate) this scanner
(S=f, P, O) ⇒ po=P: program (crashed)

где M — модификатор, NP — именная группа, S — подлежащее, P — предикат, O — объект, f — признак, po — кандидат в оценочные слова. Определение семантической ориентации слов базируется на ряде факторов, включая употребление с союзами, учет словообразования, информации о синонимах и антонимах из тезауруса WordNet [38].

В работе [28] рассматривается метод автоматического извлечения оценочных слов на основе нескольких корпусов текстов, которые существуют для многих предметных областей, а именно:

- корпус отзывов о сущностях с оценками, вручную проставленными потребителями,
- корпус нейтральных описаний сущностей, например, сюжеты фильмов,
- нейтральный контрастный корпус общезначимых новостей.

Из указанных корпусов извлекаются списки слов; для каждого слова рассчитывается набор статистических характеристик (частотности, относительные частотности между корпусами, разные корреляции появления слова и пользовательской оценки к отзыву), а также учитываются лингвистические факторы, например, наличие приставок или написание с большой буквы. Далее используются методы машинного обучения для получения качественного списка оценочных слов, характерных для данной предметной области.

Результат выглядит как упорядоченный список слов, расположенных в порядке снижения вероятности оценочности конкретного слова, предсказанного классификатором. Данный метод не дает возможности проставить оценку тональности слова, но сосредотачивает возможные оценочные слова ближе к началу списка, что облегчает их просмотр экспертами для разметки по тональности.

Классификатор для извлечения оценочных слов из таких корпусов был обучен на данных отзывов о фильмах, а затем обученная модель была применена к другим предметным областям. В [28] было показано, что модель хорошо переносится на другие предметные области. Например, оценка при-

менения модели в предметной области отзывов о книгах показала, что в первой тысяче полученного списка содержится более 85% оценочных слов.

Приведём пример начала списка предполагаемых оценочных слов, извлечённого по описанной модели для предметной области отзывов о ресторанах, с предсказанными вероятностями их оценочности:

невкусный	0.970
безвкусный	0.964
неуютный	0.956
невнимательный	0.939
непринужденный	0.937
пафосный	0.924
неторопливый	0.919...

В ряде последних работ указывается, что часто в оценочные словари включают не только слова, выражающие мнения, но и слова, которые ассоциируются у читателя с чем-то хорошим или плохим, т.е имеющие коннотации, оценочные ассоциации [39]: *очередь, налог, пробка, безработица* и др. (см., пример твита: и все-таки живая очередь одержала победу над электронной в сбербанке на чайке). Это связано с тем, что в тексте может упоминаться отрицательный или положительный факт, чья отрицательность или положительность известна (например, повышение безработицы, очередь в банке), без явного выражения собственного мнения [62].

Для автоматического выявления слов, имеющих отрицательные или положительные коннотации, используется специальный набор контекстов вида *бороться с*, *предотвратить*, *бороться за* и др. [39]. Похожий метод был использован и в работе [68] в качестве одного из источников лексики в словаре оценочной лексики русского языка РуСентиЛекс.

Другой способ выявления слов, имеющих отрицательные или положительные коннотации, обсуждается в работе [133]. Авторы заметили, что слова, имеющие коннотации, практически не могут употребляться с оценочными словами противоположной направленности. Так, практически невозможно сказать: хорошая безработица, прекрасная преступность и т.п.

Таким образом, можно выявлять коннотации слов, выявляя разница частотности их встречаемости с положительными и отрицательными оценочными словами.

Одним из известных подходов для извлечения оценочных слов из текстов является подход, предложенный в работе [118]. В этом подходе предлагалось задать некоторое множество исходных позитивных и негативных слов, а для остальных слов насчитывать совместную встречаемость с заданными позитивными и негативными словами. Для оценки оценочной ориентации слов (SO) было предложено использовать формулу поточечной взаимной информации PMI следующим образом:

$$\mathrm{SO}(w) = \mathrm{PMI}(w, Pos) - \mathrm{PMI}(w, Neg). \tag{1}$$

Поточечная взаимная информаци определяется следующим образом:

$$\mathrm{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \tag{2}$$

где $p(w_i)$ — это вероятность встретить слово в корпусе, обычно вычисляется как отношение количества вхождений слова $p(w_i)$ к общему количеству слов в корпусе.

Существенной проблемой этого простого метода являлась необходимость задания исходного множества позитивных и негативных слов, а также необходимость сбора объёмной текстовой коллекции. В дальнейшем оказалось, что данный метод хорошо применим для извлечения оценочных слов и выражений из сообщений Твиттера. При этом сами сообщения легко собрать в большом количестве, пользуясь API Твиттера. Кроме того, не нужно задавать множества исходных оценочных слов, поскольку в качестве таких слов используются позитивные и негативные хэштеги и эмотиконы, проставленные самими пользователями. Известно, что оценочная ориентация хэштега (или эмотикона) далеко не всегда соответствует оценочной ориентации сообщения, которому этот хэштег (эмотикон) приписан, однако в целом такие данные можно использовать вполне эффективно [80]. Такой же подход для русского языка описан в работах [8, 9].

В работе [108] те же самые данные используются для извлечения оценочных слов и выражений посредством обучения классификации твитов классификатором SVM. При этом эмотиконы и хэштеги используются как разметка твитов на позитивные и негативные. Эта разметка, как уже указывалось выше, является очень несовершенной, однако объем обучающей коллекции позволяет все-таки обучать автоматические классификаторы хорошего качества (такой подход называется distant supervision [77, 43]).

В качестве признаков классификации в упомянутой работе [108] рассматриваются отдельные слова и биграммы. Затем полученные внутри классификатора веса w каждого слова или биграммы, рассматриваются как их оценочные веса. Полученный словарь авторы используют в качестве признаков для выполнения заданий по анализу тональности сообщений социальных сетей на данных тестирований Semeval–2013, 2014 и показывают улучшение результатов по сравнению с результатами [80]. В работе [122] в похожей задаче извлечения оценочных слов из сообщений Твиттера используется нейронная сеть.

4.3.3 Словари оценочной лексики для русского языка

Словарь ProductSentiRus. В работе [29] описывается подход к автоматическому созданию словаря оценочной лексики в области товаров и услуг для русского языка ProductSentiRus. Словарь ProductSentiRus был получен применением обученной модели из ранее упомянутой работы [28] к наборам отзывов в нескольких предметных областях: фильмы, книги, игры, цифровые камеры и мобильные телефоны. Для получения списка оценочных слов, полезных для работы в разных предметных областях, была экспериментально выбрана формула комбинирования списков, которая учитывает факторы присутствия слова во всех предметных областях, позицию слова в списке каждой предметной области и вероятность оценочности этого слова, предсказанную классификатором.

Словарь представлен как список 5 тысяч слов, упорядоченных по мере снижения вычисленной вероятности их оценочности без указания позитивной или негативной тональности. Точность оценочных слов в первой тысяче слова списка составляет более 91%.

Наиболее вероятными оценочными словами в списке ProductSentiRus являются:

бесподобный	0.963
невнятный	0.953
отличнейший	0.935
обалденный	0.933
безумно	0.924
непонятно	0.921
неприятно	0.920
отвратный	0.920
нежный	0.916

Видно, что несмотря на то, что слова извлекались по специализированным отзывам, в верху списка расположены общеупотребительные оценочные слова, не имеющие привязки к конкретным предметным областям.

Словарь РуСентиЛекс. По своей структуре лексикон РуСентиЛекс представляет собой упорядоченный по алфавиту список слов и выражений. Он содержит следующие типы русскоязычных слов, значения которых связаны с тональностью:

- слова (выражения) литературного русского языка, для которых хотя бы одно значение имеет оценочный компонент, что означает, что слово в этом значении либо явно выражает отношение к обсуждаемому объекту (отличный), либо передается через выражаемую эмоцию (грустно);
- слова (выражения), не передающие оценочное отношения автора, но имеющие положительную или отрицательную коннотацию [39], например, *безработица, терроризм, болезнь, спам* и др.;
- сленговые и ругательные слова из Твиттера.

Все лексические единицы, описанные в РуСентиЛекс, и их значения, рассматриваются с трех точек зрения. Во-первых, указывается полярность слова: позитивная, негативная или нейтральная; возможно также приписывание пар полярностей. Во-вторых, проставляется источник тональности: прямо выраженная оценка, эмоция или коннотация.

В-третьих, представлены тональные различия между значениями многозначного слова. Если все значения многозначного слова имеют одну и ту же тональность во всех значениях, то указывается просто тональность слова. Если слово имеет различные характеристики тональности в своих разных значениях, то описываются особенности каждого значения. Для идентификации значений устанавливается ссылка на понятия тезауруса русского языка РуТез[1] [65].

При подготовке словаря РуСентиЛекс было замечено, что в русском языке имеется значимое количество многозначных слов, которые во всех своих зафиксированных в тезаурусе значениях имеют одну и ту же тональность (например, *грязный*). Поэтому такие многозначные слова не расписываются подробно по значениям, им указывается общая тональность слова. Таким образом, значения таких слов в тезаурусе могут пополняться, но тональность у них уже приписана.

Словарь РуСентиЛекс хранится в простом текстовом формате, подобном формату словаря MPQA [128]. Каждой единице словаря, которая может быть словом, словосочетанием или лексической единицей (т. е. парой слово-понятие тезауруса РуТез) приписываются следующие атрибуты:

- слово или фраза,
- часть речи,
- слово или фраза, в которой каждое слово стоит в лемматизированной форме, что необходимо для сопоставления фраз с текстами, в которых фраза может стоять в разных словоизменительных формах,
- тональность. Тональность может быть позитивная (positive), негативная (negative), нейтральная (neutral) или двойная, например, positive/negative. В последнем случае такая отметка означает, что слово (фраза) обычно употребляется с какой-то оценкой, но эта оценка в значительной степени зависит от контекста употребления слова;
- источник тональности (явно выраженная оценка, эмоция, или факт);
- отсылки к понятиям тезауруса РуТез для значений тех слов, которые имеют различающуюся тональность в разных значениях. Для этого производится указание на имя соответствующего понятия в тезауру-

[1] http://www.labinform.ru/pub/ruthes/index.htm

се. Отметим, что в таких случаях описывается тональность для всех значений многозначного слова.

Например, слово *пресный* имеет три различных значения в тезаурусе РуТез. Два из них (значение как *безвкусный о еде* и значение *неинтересный*) имеют негативную тональность. Еще одно значение слова, связанное с пресной водой, имеет позитивную коннотацию, поскольку обладание пресной водой — это хорошо, ее истощение — это плохо и т.п.

Таким образом, описание значений слова *пресный* выглядит следующим образом (метки в кавычках соответствуют именам понятий в тезаурусе РуТез):

пресный, Adj, пресный, negative, emotion, «НЕВКУСНЫЙ»
пресный, Adj, пресный, negative, opinion, «НЕИНТЕРЕСНЫЙ»
пресный, Adj, пресный, positive, fact, «ПРЕСНАЯ ВОДА»

Другое оценочное русское слово *грязный* имеет два значения в тезаурусе РуТез, но оно описывается в РуСентиЛекс без ссылок на значения, поскольку оба этих значения являются негативными:

грязный, Adj, грязный, negative, opinion.

Слова-кандидаты на включение в лексикон были извлечены автоматически. Далее эксперты-лингвисты анализировали употребление каждого слова в современных новостных текстах. Новостные тексты выбраны потому, что они адресованы максимальной аудитории современных русскоязычных людей и поэтому в среднем передают норму современного русского языка.

Например, при анализе слова *аккуратист* (*аккуратный человек*) некоторым экспертам казалось, что употребление этого слова несет негативную оценку. Но проанализированные контексты употребления этого слова показали, что оценка скорее позитивная, например:

Страховщики заплатят штраф за отсутствие скидок водителям-аккуратистам (Вести-ФМ 29.10.2015);

Шамардин: Это интеллигентнейший человек, который показал себя с первых дней своего обучения. Аккуратист во всем, в одежде, в поведении (Спорт FM, 13.10.2015).

Словарь РуСентиЛекс опубликован[2].

Словарь Linis-crowd. Словаря Linis-Crowd [1, 52] создавался для анализа тональности текстов социальных сетей. Для создания словаря сначала из нескольких ресурсов были собраны слова-кандидаты, включая:

- наиболее частотные прилагательные русского языка, употребляемые в текстах социальных сетей,
- образованные от отобранных прилагательных наречия,
- словарь ProductSentiRus [29], из которого были выбраны слова, подходящие для анализа сообщений в социальных сетях и др.

Далее была произведена контекстная оценка тональности собранных слов, а также текстов, в которых они употреблялись, методом краудсорсинга по шкале от -2 (сильно негативный) до $+2$ (сильно позитивный). Оценки различных разметчиков усреднялись. В настоящее время результаты разметки слов и текстов опубликованы.

Вопросы создания словарей оценочной лексики для русского языка обсуждаются также в работах [53, 109, 41, 79, 110].

4.4 Анализ тональности документов в целом

Автоматический анализ тональности осуществляется с помощью следующих основных двух типов подходов [63, 97]: подходы, основанные на словарях и правилах (инженерно-лингвистический подход), и подходы на основе машинного обучения.

Методы, основанные на словарях и правилах, заключаются в использовании специально создаваемых словарей оценочных слов и выражений и применении лингвистических правил, с помощью которых учитывается контекст употребления слов.

Методы машинного обучения, используемые для систем автоматического анализа тональности, делятся на два типа: метод машинного обучения с учителем и методы машинного обучения без учителя или с частичным обучением.

[2] http://www.labinform.ru/pub/rusentilex/index.htm

Метод машинного обучения с учителем состоит в том, что система «обучается» на коллекции размеченных текстов, с которыми анализируемый текст сравнивается на предмет близости к некоторому классу. В зависимости от задачи могут быть выбраны разные алгоритмы классификации, разные признаки представления текста, разные способы подсчета весов признаков.

Отметим также, что в настоящее время предложены подходы, которые интегрируют имеющиеся словари оценочной лексики (как созданные вручную, так и порожденные автоматически) в методы машинного обучения [80, 10].

4.4.1 Анализ тональности на основе словарей и правил

При применении инженерно-лингвистического подхода сначала должен быть составлен **словарь** оценочных слов и выражений, в котором каждой единице словаря должна быть приписана априорная оценка тональности. Источниками словаря могут служить:

— существующие словари оценочной лексики (общие или сделанные для других предметных областей), которые можно автоматизированно сопоставить с имеющимся корпусом текстов и отобрать относящуюся к данной предметной области лексику,

— автоматизированная обработка имеющегося корпуса текстов для отбора оценочной лексики,

— автоматизированный перевод общих или специализированных словарей оценочной лексики с других языков,

— экспертный просмотр имеющихся текстов и пополнение словаря оценочной лексики вручную.

Далее анализируемый текст сопоставляется с имеющимся словарем. Тональность текста складывается из тональности предложений, а тональность предложений — из тональности слов. Также могут применяться лингвистические правила, с помощью которых учитывается контекст употреб-

ления слов: наличие в предложении отрицаний, слов, усиливающих или понижающих тональность соседних слов, и другие.

Рассмотрим несколько примеров создания словарей в разных проектах.

Для создания тонального словаря [114] из корпуса, который был взят для разработки системы (коллекция из 400 текстов отзывов о книгах, автомобилях, компьютерах, кухонной утвари, отелях, фильмах, музыке и телефонах, размеченных пользователями, по 50 отзывов на каждую тематику (25 позитивных и 25 негативных), всего 279 761 слов) были извлечены все прилагательные. Они были размечены вручную по шкале от -5 до $+5$ в зависимости от тональности, где оценка 0 присваивалась нейтральным словам, которые исключались из словаря. Слова оценивались по их тональности вне контекста. Всего словарь прилагательных составил 2252 слова.

Также были созданы отдельные словари существительных (1142 слова), глаголов (903 слова) и наречий (745 слов), которые были размечены по той же шкале. Для создания этих словарей были использованы также выборка из 100 текстов из 2000 отзывов о фильмах и позитивные и негативные слова из словаря General Inquirer. Авторы системы отмечают, что это обеспечило некоторый баланс между словами разных стилей (в отзывах используется неформальный язык, General Inquirer представляет более формальную лексику).

В системе [7] для создания словаря был использован корпус текстов СМИ, специально созданный на основе информационных русскоязычных порталов Интернета. Корпус содержал около 100 млн. словоупотреблений. Общий словарь, насчитывающий более 15000 единиц, состоял из таких частей, как существительные, глаголы, прилагательные, наречия, глагольные и неглагольные коллокации. Под коллокациями в данной работе понимались любые устойчивые и достаточно часто встречающиеся сочетания слов, как с идиоматическим значением, так и с неидиоматическим.

Все единицы словаря размечались вручную и включали только единицы, несущие тональность или усиливающие ее. Слова, тональность которых зависит от тематики текста, размечались согласно тому, в какой тонально-

сти они чаще употребляются в корпусе СМИ. Тональность приписывалась словам и выражениям по бинарной шкале (позитив/негатив).

Для вычисления тональности высказывания или документа необходимо использование **правил**, которые суммируют или модифицируют тональность исходных слов в зависимости от контекста.

Наиболее известными правилами, применяемыми в системах анализа тональности текстов, являются следующие:

- применение слов-операторов, усиливающих исходную оценку оценочного слова (*очень*) или меняющих оценку на противоположную (*не, нет*),
- суммирование оценок слов, входящих в состав высказывания или текста,
- выставление негативной оценки для словосочетания, в котором употреблено хотя бы одно негативное выражение [58, 4].

В работе [114] описана и протестирована система правил для системы анализа тональности текстов (отзывы) на английском языке. В частности, в этой работе описываются правила, в результате действия которых тональность слов, находящихся в одной клаузе с маркерами ирреализа (слов, являющихся индикаторами ситуаций, которые не принадлежат к произошедшим или происходящим в реальной действительности), не принимается во внимание. Список таких маркеров включает модальные глаголы, индикаторы условного наклонения (*if*), такие слова с негативной полярностью, как *any, anything*, некоторые глаголы (например, *expect* и *doubt*), вопросы и слова, заключенные в кавычки.

В работе [84] исследуется подход к улучшению анализа условных предложений типа *Если у кого-то есть надежный автомобиль, то я бы купил его*, некоторые из которых даже при наличии явных оценочных слов не несут никакой оценки по отношению к каким-либо сущностям. Для отличения условных предложений с тональностью и без нее в данной работе предлагается подход машинного обучения с учителем на основе ряда лингвистических и позиционных признаков, наличие служебных слов, расположения оценочных слов в предложении и т.п.

В работе [87] используется шесть правил композиции оценок для определения тональности: конверсия тональности, агрегация, распространение, доминирование, нейтрализация и интенсификация.

Конверсия — это применение отрицаний и перевод в противоположную тональность. *Агрегация* применяется для синтаксических групп вида прилагательное-существительное, существительное-существительное, наречие-прилагательное, наречие-глагол, имеющих противоположную тональность, например, *beautiful fight (прекрасная битва)*. В таком случае, этой фразе приписывается доминирующая тональность модификатора:

POS('beautiful') & NEG('fight') ⇒ POS('beautiful fight').

Правило *распространения* применяется, когда в предложении употребляется глагол распространения или передачи:

PROP-POS('to admire') & 'his behavior' ⇒ POS('his behavior');

'Mr.X' & TRANS('supports') & NEG('crime business') ⇒ NEG('Mr. X').

Правило *доминирования* заключается в том, что если полярности глагола и его объекта различны, то полярность глагола преобладает:

NEG('to deceive') & POS('hopes') ⇒ NEG('to deceive hopes'));

Если в сложном предложении фразы соединены союзом «но», то тональность второй части предложения доминирует:

NEG ('It was hard to climb a mountain all night long'), но POS('a magnificent view rewarded the traveler at the morning') ⇒ POS(предложение))

Правило *нейтрализации* применяется, когда предлог-модификатор или оператор условия относится к тональному выражению:

'despite' & NEG('worries') ⇒ NEUT('despite worries').

Правило *интенсификации* усиливает или ослабляет вес тональности:

Pos_score('happy') < Pos_score('extremely happy')).

В целом, основными достоинствами подхода, основанного на словарях и правилах, являются следующие:

- Предсказуемые результаты: результаты работы системы зависят от качества применяемых правил и словаря. По результатам можно судить о том, как можно улучшить систему.

- Возможность более глубокого и тонкого анализа тональности на уровне высказывания.

К недостаткам инженерно-лингвистического подхода относятся следующие проблемы.

- Составление словаря и правил вручную является дорогостоящей задачей. Но, как указывается в работе [20], создание обучающей коллекции для методов машинного обучения ещё более сложно и дорого.
- Системы, созданные с помощью данного подхода, могут показывать низкие результаты по полноте, если словарь не настроен на определённую предметную область.

4.4.2 Анализ тональности на основе машинного обучения

Для создания системы автоматического анализа тональности с помощью машинного обучения с учителем отбирается некоторое количество текстов для обучения классификатора, которые вручную размечаются по тональности, затем выбирается алгоритм классификации и признаки, в виде которых будет представлен текст для этого алгоритма. После этого производится обучение классификатора.

Эффективность системы анализа тональности, построенной с помощью методов машинного обучения, определяется выбором некоторого количества решений [63, 97].

Во-первых, нужно выбрать алгоритм классификации. До последнего времени, чаще всего применялись и показывали наилучшие результаты такие алгоритмы, как метод опорных векторов SVM и наивный байесовский классификатор [63, 97]. В последнее время лучшие результаты на многих задачах анализа показывают нейронные сети [115, 108].

Во-вторых, нужно выбрать признаки представления текста. Текст может быть представлен в виде:

- набора слов (bag-of-words);
- набора n-грамм (обычно это униграммы и биграммы, кроме того, могут использоваться символьные n-граммы);

- также используются дополнительные признаки: части речи; пунктуация (например, наличие восклицательных знаков); наличие отрицаний; наличие ключевых слов и фраз, обычно выражающих оценку (например, «плохой», «хороший», «нравиться»);
- могут учитываться семантические отношения между словами с помощью таких ресурсов, как различные тезаурусы, онтологии, семантические сети (например, каждое слово в тексте может проходить проверку на то, является ли оно синонимом и насколько близким, к ключевому слову);
- могут учитываться также синтаксические зависимости, дискурсивная структура текста.

В-третьих, нужно выбрать способ подсчета веса признака. Признаки могут рассматриваться в булевой форме (бинарный вес, есть или нет). Может быть учтена их частотность. Также часто используется известный подход взвешивания на основе веса tf.idf [71], где каждый признак получает вес, пропорциональный частоте появления признака в тексте (tf) и обратно пропорциональный количеству документов коллекции, в которых есть этот признак.

В работах [72, 94] была предложена так называемая delta-схема взвешивания весов признаков. Основная идея заключается в том, чтобы учесть распространенность признака не во всей текстовой коллекции, как это делает взвешивание tf.idf, а насколько неравномерно распределено слово в двух классах тональности. Слово должно иметь тем больший вес, чем неравномернее оно распределено по этим классам:

$$w_i = tf_i \cdot \log_2\left(\frac{N_1}{df_{i,1}}\right) - tf_i \cdot \log_2\left(\frac{N_2}{df_{i,2}}\right) = tf_i \cdot \log_2\left(\frac{N_1 \cdot df_{i,2}}{N_2 \cdot df_{i,1}}\right), \qquad (3)$$

гдеN_1 — количество документов в положительном классе, $df_{i,1}$ — количество документов положительного класса, в которых слово встречалось, N_2 — количество документов класса с отрицательной тональностью, $df_{i,2}$ — количество документов с отрицательной тональностью, в которых встречалось слово.

Также важным фактором является выбор предобработки, например лемматизации или стемминг. Существенным является определение стоп-слов, которые будут удалены при обработке текстов. Такими стоп-словами могут являться служебные части речи, числительные, даты и др. Здесь нужно обратить внимание, что некоторые стандартные стоп-слова, которые не очень важны при задачах тематической классификации или тематического поиска, могут играть большое значение в классификации текстов по тональности, например, *не, нет, очень* и др. Поэтому, используя какой-то стандартный список стоп-слов, важно не удалить такого рода слова.

Системы анализа тональности, созданные с помощью методов машинного обучения, обычно достигают высоких по сравнению с другими методами показателей эффективности в том случае, если для обучения классификатора используется достаточно большая размеченная коллекция документов.

Недостатками методов машинного обучении являются следующие.

- Создание обучающей коллекции текстов вручную является достаточно трудоемкой и дорогостоящей задачей.
- Построенный алгоритм плохо переносится на другую предметную область. Хотя методы машинного обучения с учителем показывают очень хорошие результаты в предметной области, на которой обучается классификатор, эффективность сильно снижается при его использовании в другой области. Это требует специальных техник настройки систем на новые предметные области [95, 16, 31].
- Результаты обработки могут быть трудно объяснимы, что затрудняет процедуру выявления и исправления проблем.

4.5 Анализ тональности по аспектам

В задаче анализа тональности по аспектам обычно предполагается, что текст содержит в себе оценку одного объекта, но этот объект рассматривается относительно его различных частей и/или атрибутов, которые могут быть оценены автором текста по-разному. Типичным текстом для анализа

тональности по аспектам является отзыв пользователя о некотором продукте или услуге.

Аспекты могут быть сгруппированы в категории (далее аспектные категории). Для ресторанов — это обычно кухня, обслуживание, интерьер (обстановка). Также в текстах отзывов можно встретить оценку объекта в целом: *прекрасный ресторан*. Эту категорию также можно рассматривать как аспектную (аспект *Объект_в_целом*). Слова и выражения, посредством которых можно сослаться в тексте на аспект сущности, называются *аспектными терминами*.

Таким образом, задача анализа тональности по аспектам включает следующие подзадачи:

- выделение аспектных терминов,
- классификация аспектных терминов в аспектные категории,
- автоматическое определение аспектов по отношению к выделенным категориям.

4.5.1 Классы аспектных терминов

Аспектные термины в предметной области могут быть классифицированы по нескольким основаниям.

Наиболее частым видом аспектных терминов являются явные аспектные термины, которые явно называют объект, его части или характеристики, которые оцениваются автором текста, например, *суп, обслуживание, зал*, в отзывах о ресторанах.

Явные аспектные термины чаще всего выражаются существительными или группами существительного, но некоторые аспекты могут выражаться и глаголами, например, *встретить (хорошо, неприветливо), ждать (слишком долго, не пришлось)* при оценке качества сервиса в ресторанах.

Вторым видом аспектных терминов являются так называемые неявные аспектные термины, которые представляют собой слова с явно выраженным оценочным компонентом значения, которые одновременно указывают и на обсуждаемый аспект (обычно достаточно обобщенную аспектную категорию), например, *вкусный* (*положительный+еда* в отзывах о

ресторанах), *комфортный* (*положительный+комфорт* в отзывах об автомобилях).

Как и другие оценочные слова, неявные аспектные термины могут сочетаться с т.н. оценочными операторами, которые меняют или усиливают их оценку *не очень вкусный, не слишком комфортный.* Важность таких аспектных терминов для словарей автоматических систем анализа тональности заключается в том, что в ситуациях нераспознавания упомянутых автором эксплицитных терминов (из-за опечаток, новой лексики, сложной референции) неявные аспектные термины дают возможность извлечь позицию пользователя по отношению к некоторой аспектной категории.

Третьим видом выражения своего мнения по поводу некоторой характеристики заданной сущности является сообщение некоторого произошедшего негативного или позитивного факта, который одновременно указывает как на аспектную категорию, так и на его оценку пользователем (далее оценочные факты).

Одним из видов оценочных фактов являются технические проблемы, упоминаемые в отзывах [44, 45, 119]. В [44] указывается, что упоминание технических проблем на английском языке часто включает в себя:

- набор специального вида глаголов, обозначающих, что что-то случилось (*fail, crash, overload, trip, fix, mess, break, overcharge, disrupt*);
- набор глаголов, обозначающих, что что-то не случилось, и часто эти глаголы упоминаются с отрицаниями, а также с глаголами операторами вида (*stop, refuse, cease* — прекратить, прекратиться, остановиться и др.),
- некоторыми глаголами с частицами (*knock off, knock out, hang up*),
- а также существительными и словосочетаниями.

Вместе с тем оценочные факты могут включать и значительно более широкий спектр ситуаций, чем технические проблемы, как, например, обнаружение чего-то нежелательного: *Два раза был в этом ресторане, и оба раза нашел в своей тарелке волос.* В работе [62] приводится следующий пример оценочного факта: *I bought the mattress a week ago, and a valley has formed* (Я купил матрас неделю назад, и уже образовалась впадина).

Близкие по смыслу оценочные факты могут выражаться в тексте разнообразными способами, что затрудняет их обнаружение. Однако частым признаком такого факта является появление в тексте неоценочных слов, имеющих отрицательные или положительные коннотации. Примерами таких слов с отрицательными коннотациями в общественно-политических текстах являются слова *безработица, инфляция, стагнация.*

В области отзывов о ресторанах слова *волос, майонез* несут в себе отрицательные коннотации, т. е. уже появление таких слов в текстах является признаком того, что тональность текста будет скорее отрицательной. В технической области такими словами являются слова, обозначающие поломки (*fail, crash, overload, trip, fix, mess, break*), как это указывалось в работах [44, 45].

Также в [62] указывается, что есть еще одна категория неявных оценок и аспектов, которые называются авторами «ресурсная проблема». Приводится пример: This washer uses a lot of water (Эта посудомоечная машина расходует много воды). Таким образом, расходование воды является здесь аспектом, а вода — ресурсным термином, чрезмерное расходование которого является отрицательным фактом.

В [133] указывается, что ресурсные термины должны извлекаться на основе употребления с квантификаторами *много-мало*, а также рядом с глаголами потребления. В работе рассматривается итеративный алгоритм, в котором вначале задаются некоторое количество известных глаголов потребления, а также несколько известных ресурсов: *газ, вода, электричество, деньги, чернила, моющее средство (detergent), мыло, шампунь.*

В работе [67] рассматриваются наиболее частотные типы оценочных фактов в области отзывов о ресторанах и об автомобилях. Ресурсы в отзывах о ресторанах включают:

- время гостей ресторана;
- внимание официантов;
- три вида ресурсов, связанных с едой: количество пищи на тарелке, выбор в меню и наличие блюд;
- место в помещении ресторана и свободные столики;
- деньги клиентов.

В обеих областях встречаются факты, связанные с отклонением от нормального порядка вещей, что обычно указывается словами *отсутствуют, отсутствие*. Также существенную частотность имеют факты, связанные с разными звуками (*звучать, хрустеть*).

4.5.2 Автоматизация выявления признаков/свойств для товаров или услуг

В качестве аспектных терминов, чаще всего, рассматриваются существительные и группы существительного [18, 21, 50]. Длина группы существительного предполагается не больше, чем 3-4 слова. При этом указывается, что если извлекать только отдельные существительные как аспектные термины, то они часто могут быть неоднозначными, что, например, приводит к низкому согласию между экспертами [6].

Согласно [62], существует четыре основных подхода к автоматизации извлечения аспектных терминов из текстов:

- подход, основанный на частотных существительных и группах существительного;
- подход, использующий отношения между оценочными выражениями и аспектными терминами;
- подход, основанный на машинном обучении с учителем;
- подход, основанный на статистических тематических моделях.

Извлечение аспектных терминов на основе частотных характеристик слов. Для извлечения кандидатов в аспекты большое значение имеет частотность их упоминания в анализируемой текстовой коллекции [101, 50]. В [78] подчеркивается, что частотные признаки работают удивительно хорошо для таких простых признаков. Вместе с тем все-таки среди частотных существительных встречается достаточно много не-аспектов, например, общелитературной лексики, кроме того, плохо улавливаются малочастотные аспектные термины.

В работе [56] для извлечения аспектных терминов используется известный в информационном поиске признак tf.idf [71], который вычисляется как на уровне документов, так и на уровне абзацев. Scaffidi et al. [106] использу-

ют для извлечения аспектных терминов сравнение частот именных групп в коллекции отзывов с частотами этих групп в контрастной коллекции — Национальном британском корпусе.

Если в качестве аспектных терминов извлекаются не только отдельные существительные, но и группы существительного, то необходимо использовать дополнительные признаки для более точного определения длины именной группы. Чаще всего используются так называемые контекстные признаки, которые оценивают частоту встречаемости словосочетания с частотой контекста. Такие признаки позволяют определить границы именной группы. Например, в [18] используется так называемая мера FLR:

$$FLR(a) = f(a) \cdot LR(a), \tag{4}$$

$$LR(a) = \sqrt{l(a) \cdot r(a)}, \tag{5}$$

где $f(a)$ — частота аспектного термина, $l(a)$ — количество разных слов, находящихся слева от a, $r(a)$ — количество разных слов, находящихся справа от a.

Далее отбираются группы существительного с данной мерой, большей, чем в среднем для словосочетаний. Таким образом, данная мера в первую очередь отбирает группы существительного, которые имеют большое разнообразие слов на своих границах, что показывает, что анализирумый термин a не является фрагментом более длинного словосочетания.

Другим критерием, направленным к этой же цели, является известный признак C-value [40], который снижает вес данного слова или словосочетания, если оно входит в частотное словосочетание большей длины. Тем самым предполагается, что это более длинное словосочетание может рассматриваться как кандидат на аспект, а текущее представляет его фрагмент. Такой признак для отбора аспектов используется в работе [135].

В работе [48] предлагается считать аспектными терминами только те именные группы, которые появляются в виде подлежащих или объектов глаголов, или в составе предложных групп.

В работе [101] алгоритм исключает из списка потенциальных аспектных терминов те из них, которые не встречаются достаточно часто в за-

данных шаблонах, обозначающих часть-целое (меронимию) с целевым объектом. Для этого на основе поиска в Интернет считается показатель PMI (pointwise mutual information) встречаемости предполагаемого аспектного термина с целевым объектом. Например, для цифровых камер проверяется встречаемость кандидатов в термины в образцах вида *of camera, camera has*.

Кроме того, в этой работе используется иерархия тезауруса WordNet для выявления названий компонентов/частей, а также словообразовательные суффиксы типа (-iness, -ity). Отметим, что в какой-то мере использование WordNet, фиксированных суффиксов предполагает применение алгоритма именно к техническим областям. Подобный подход (WordNet, суффиксы) представляется неприменимым к фильмам, программному обеспечению, ресторанам. В работе [103] при обзоре работ указывается, что подход [101] является затратным по времени, поскольку идет интенсивное обращение к Интернет-поиску.

Отметим, что этот набор характеристик для извлечения аспектов (за исключением проверки на отношение меронимии) в работе [101] очень похож на характеристики, используемые для извлечения терминов в заданной предметной области [64].

Отношения аспектов с оценочными словами. Итеративные методы для извлечения аспектных терминов. Во многих работах указывается, что аспектный термин должен входить в шаблоны с оценочными словами [21] или хотя бы употребляться в одном и том же предложении с оценочными словами [18, 21]; также могут использоваться меры, учитывающие оба эти фактора [21].

В работе [136] для извлечения отношений между аспектными терминами и оценочными словами используется синтаксический анализатор. Отношения между аспектом и оценочным словом извлекаются на основе заданных путей синтаксической зависимости. Так, например, в предложении «This movie is not a masterpiece» слова *movie* и *masterpiece* будут размечены соответственно аспектом и оценочным словом, поскольку между ними существует путь в синтаксическом дереве «NN – nsubj – VB – dobj – NN»

(именная группа – подлежащее – глагольная группа – прямое дополнение – именная группа).

Для извлечения аспектных терминов с учетом их отношений с оценочными словами часто используются итеративные методы (bootstrapping). В качестве начального множества могут использоваться частотные именные группы, которые предполагаются аспектами, либо задаются вручную.

В известной работе [50] начальное множество аспектных терминов (частотные слова и именные группы) используется для выявления ассоциативных правил, т. е. шаблонов, посредством которых аспекты обычно связаны с оценочными словами. После получения таких правил извлекаются менее частотные аспектные термины, т. е. те именные группы, которые появлялись именно в таких шаблонах с оценочными словами.

В работе [103] рассматривается подход двойного распространения (double propagation) к извлечению аспектных терминов и расширению словаря оценочных слов. В качестве исходного множества задается небольшой словарь оценочных слов, также задаются синтаксические шаблоны, в которые обычно входят оценочные слова и аспектные термины. В итоге, вхождение известного оценочного слова в такой шаблон помогает извлекать аспект, а известный аспект, входящий в такой шаблон, помогает извлекать оценочное слово.

Для очистки полученного множества аспектов применяется ряд правил. Например, предполагается, что в одном фрагменте предложения без запятых содержится только один аспектный термин, а другой кандидат должен быть удален, удаляется менее частотный в коллекции.

Оценка этого метода проводилась на пяти областях; была получена средняя F-мера — 85%. Отметим, что эксперименты проводились на небольшом числе отзывов — в среднем 62.8 отзыва из каждой области [103].

В работе [133] для оценки значимости аспектных терминов вводятся ещё два фактора. Первый фактор рассматривает, насколько разнообразны оценочные слова, применяемые к аспекту-кандидату — разнообразие обычно свидетельствует о значимости аспектного термина. Во-вторых, в коллекции ищется подтверждение связи аспектного термина с сущностью посредством заданных шаблонов.

Например, в области автомобилей можно найти такие фразы, как *the engine of the car (двигатель автомобиля), the car has a big engine (автомобиль имеет большой двигатель)*, которые свидетельствуют об отношении часть-целое между engine и car. Если слово одновременно встречается и с оценочным словом, и в отношениях с заданной сущностью, то это дает этому аспекту-кандидату сразу высокий вес: например, *there is a bad hole in the mattress (в матрасе имелась большая дыра).*

В работе [18] для итеративного поиска аспектных терминов используется некоторое начальное множество аспектов, которое пополняется на основе:

- учета меры взаимной информации нахождения аспекта кандидата в одних и тех же предложениях, что и аспекты из начального множества аспектов и частотности аспекта-кандидата;
- при пополнении аспектов полезна очистка избыточных аспектов, например, если в множество аспектов входит и более короткий аспект.

Число вручную выделяемых аспектных терминов товара в данной работе может достигать до 200 аспектов в технических областях. F-мера выделяемых аспектов в данной работе порядка 72.9%. Обучение проводилось на 45-100 текстов для отдельного объекта [18].

В [78] указывается, что итеративные методы, основанные на отношениях с оценочными словами, могут находить низкочастотные аспекты. Вместе с тем извлекается достаточно много не-аспектов, которые подошли под заданные шаблоны. При создании комбинированных методов, сочетающих шаблоны и частотность, начинают теряться низкочастотные аспекты и возрастает число параметров для настройки.

В [133] указывается, что метод двойного распространения (double propagation) для одновременного извлечения аспектов и оценочных слов, основанный на синтаксическом пути между ними, хорошо работает для коллекций среднего размера: для маленьких коллекций метод дает пониженную полноту, в то время как для больших коллекций — в заданные синтаксические шаблоны проникает много шума.

Использование методов машинного обучения для выявления аспектных терминов. Имеется два направления использования методов машинного обучения с учителем для выявления аспектных терминов:

- методы, основанные на предварительном составлении списка аспектных терминов в некоторой предметной области, и обучение модели, использующей перечисленные в предыдущих разделах признаки, присущие аспектам;
- методы, основанные на разметке последовательности слов в отзывах (разметка аспектных терминов, оценочных слов).

В работе [55] для извлечения аспектных терминов, помимо частотности аспектов-кандидатов в отзывах, используется сопоставление кандидатов с заголовками словарных статей в Википедии, семантическая близость кандидатов, рассчитанная на основе совокупностей ссылок соответствующих статей Википедии (в итоге 2 признака), а также ассоциирование кандидата в аспекты с именем сущности при поиске в Интернет. Результат извлечения аспектов для нескольких объектов оценивается как 72.7% F-меры.

В работе [6] в качестве набора признаков для извлечения аспектных терминов в виде отдельных существительных из отзывов о лаптопах на русском языке рассматривается следующий набор признаков:

- частотность в коллекции отзывов;
- близость к оценочным словам (окно величиной p), в данном случае рассматривалась не близость к оценочным словам в коллекции, а близость на расстоянии 3 к словам хороший/плохой в выдаче результатов поиска Яндекса;
- признак странности, вычисляющий относительную частоту слова по сравнению с контрастной коллекцией;
- признак tf.idf;
- мера взаимной информации pmi, которая учитывает совместную встречаемость между существительными кандидатами и заявленным типом товара (лаптоп).

На основе различных вариантов каждой из мер, авторами работы было получено 23 признака. Указывается, что результат извлечения близок к

результатам англоязычных работ, которые заявляют о F-мере 76–86% для разных областей.

Однако наиболее популярными в области извлечения аспектных терминов на основе методов машинного обучения являются подходы, основанные на последовательной разметке, при которой аспекты размечаются в корпусе. К размеченным данным применяются методы вида HMM (Hidden Markov models) и CRF (Conditional Random Fields) [32, 89]. В качестве признаков используются такие характеристики, как собственно слова, части речи, синтаксические зависимости, расстояния, предложения с оценочными словами и др. Эти же модели могут применяться и для совместного извлечения аспектов и оценочной лексики.

В [78] указывается, что методы, основанные на машинном обучении, могут выявлять и низкочастотные аспекты, но требуют разметки данных. Особенно большие трудозатраты требуются для разметки данных для последовательных методов машинного обучения.

Использование тематических моделей для извлечения аспектных терминов. Извлечение аспектов может выполняться на основе применения вероятностных тематических моделей, т. е. методами, которые предполагают, что каждый текст состоит из набора скрытых тем, а каждая скрытая тема представляет собой вероятностное распределение слов. Обычно рассматриваются два типа тематических моделей: pLSA (probalistic Latent Semantic Analysis) и LDA (Latent Dirichlet Allocation) [22, 3].

В результате применения тематических моделей к коллекции текстов порождается совокупность тем, каждая из которых представляет собой список слов с вероятностями их отнесения к этой теме. Предполагается, что при применении подходящей тематической модели выделенные темы будут содержать аспектные термины, относящиеся к конкретной аспектной категории.

Для извлечения аспектов необходима модификация базовых тематических моделей, направленная на то, чтобы отделить оценочные слова и аспекты в отдельные темы. При успешном применении таких моделей происходит два одновременных действия: извлечение аспектов и их группирование в обобщенные категории аспектов.

Одна из известных тематических моделей на основе LDA для извлечения аспектных терминов описана в работе [117], в которой показано, что применение базовой модели LDA, которая строится на информации о взаимной встречаемости слов в одних и тех же текстах, не является эффективной для извлечения аспектов, поскольку во множестве разных отзывов может содержаться один и тот же набор аспектов. Авторы работы применяют глобальную модель для извлечения именований сущностей, а для извлечения аспектных терминов используют скользящее окно из слов или предложений (например, 3 предложения). Собственно, встречаемость слов в таких фрагментах используется для выявления аспектов, при этом они не различают аспектные термины и оценочные слова. В статье приводится следующий пример темы «Обслуживание»: *staff, friendly, helpful, service, desk, concierge, excellent, extremely, hotel, great, reception, English, pleasant, help.*

В работе [134] предложена гибридная модель MaxEnt-LDA (комбинация моделей Maximum Entropy и LDA), в которой производится совместное извлечение аспектных и оценочных слов на основе синтаксических признаков, помогающих разделить аспектные и оценочные слова. Метод Maximum Entropy используется для подбора параметров на размеченных данных.

В [62] указываются следующие проблемы применения тематических моделей для извлечения и группирования аспектных терминов:

- требуются большие объемы данных и тщательная настройка параметров моделей для получения достаточно качественных результатов;
- методы основаны на семплировании Гиббса и поэтому каждый раз дают несколько иной результат;
- тематические модели легко выявляют частотные аспекты, которые выявляются и многими другими методами.

4.5.3 Группирование аспектных терминов

Выделенные аспектные термины могут быть достаточно разнообразными, и для удобства пользователя они обычно группируются в обобщенные категории. Такими категориями для ресторана могут быть: Кухня,

Интерьер, Обслуживание, Местоположение. При этом аспектная категория «Кухня» объединяет множество блюд и продуктов питания, которые могут предлагаться в том или ином ресторане.

В [62] указывается, что автоматизация группировки аспектов является критической для многих приложений анализа тональности отзывов. Использование общезначимых словарей синонимов и тезаурусов имеет в этой задаче ограниченное применение, поскольку такие группировки аспектных терминов существенно зависят от предметной области. Кроме того, часто аспектные термины выражаются словосочетаниями, которые обычно не описываются в словарях.

В работах [131, 132] предложен алгоритм частичного обучения, который разбивает аспектные термины на предопределенные категории аспектов. При этом предполагается, что сами по себе аспектные термины уже выделены каким-то методом. Сначала авторы вручную относят небольшое количество аспектных терминов к категориям. Затем применяют Expectation Maximization (EM) алгоритм для работы с размеченными и неразмеченными примерами. Кластеризация проводится на базе сходства контекстов упоминания аспектных терминов в окне 15 слов налево и направо. Если в окне встречается другой аспектный термин, то он не включается в окно. Также исключаются стоп-слова.

В методе также применяются два вида дополнительной информации для лучшей инициализации EM-алгоритма: аспектные термины в виде именных групп, имеющие общие слова, обычно относятся к одной категории аспектов (battery life и battery power), и аспектные термины, являющиеся синонимами в словаре, также чаще всего будут принадлежать одной группе. Эти две эвристики позволяют EM-алгоритму достигать лучших результатов. Данный алгоритм и различные другие варианты кластеризации аспектных терминов тестируются на нескольких предметных областях. Лучший результат, полученный на основе EM алгоритма в этой работе, достигает качества кластеризации, измеряемого мерой Purity, — 0.55. Purity — мера в кластеризации, замеряющая долю максимального эталонного кластера в автоматическом кластере, которая затем усредняется по всем автоматическим кластерам. Таким образом, на текущий момент лучший метод

кластеризации в состоянии лишь приблизительно наполовину повторить эталонную кластеризацию.

В работе [130] ставится задача выстроить иерархическую классификацию аспектных терминов, подобно экспертной классификации. Иерархия аспектов строится на основе нескольких признаков сходства:

- контекстный признак: два слова влево и вправо;
- признак совместной встречаемости аспектных терминов, вычисляемый на основе меры взаимной информации PMI;
- длина синтаксического пути между аспектными терминами в предложении, а также синтаксические роли в предложениях (подлежащее, объект, модификатор и т.п.);
- лексические признаки, включая извлеченное из интернета определение аспектного термина.

Иерархия строится итеративно, на основе минимизации нескольких критериев (minimum Hierarchy Evolution, minimum Hierarchy Discrepancy, minimum Semantic Inconsistency), веса признаков подбираются на основе 50 иерархий WordNet и ODP (Open Directory Project).

Результаты показывают, что если начальная иерархия совсем не задана, то качество получаемой иерархии в среднем 30-40% F-меры. Если задано 20% иерархии, то качество составляет 40-50% F-меры. Среди признаков максимальный вклад у меры совместной встречаемости.

Ранее обсуждалось, что статистические тематические модели могут одновременно извлекать и группировать аспекты. Для учета в этих моделях знаний о предметной области, в работе [14] было предложено использовать дополнительные ограничения, извлекаемые из онтологии предметной области, которые могут улучшить качество создаваемых кластеров. Ограничения носят форму must-links и cannot-links. Must-links определяют, что два слова должны быть в одном кластере, cannot-links задают, что два слова не могут быть в одном кластере. Однако предложенный метод приводит к экспоненциальному росту в кодировании cannot-links и имеет сложности в обработке большого количества ограничений.

В работе [81] знание о предметной области сообщается в виде тематической модели в виде исходных (seed) слов для каждой категории аспектов.

Кроме того, модель разделяет аспекты и оценочные слова. Приводятся следующие примеры исходных слов:

- Staff (staff, service, waiter, hospitality, upkeep)
- Cleanliness (curtains, restroom, floor, beds, cleanliness)
- Comfort (comfort, mattress, furniture, couch, pillows)

Оценка подхода показывает, что 2 заданных слова в аспекте приводит в среднем к качеству извлечения аспектных слов, измеряемых мерой точности на заданном уровне 30 слов: P@30=70%, 5 заданных слов — P@30=77%.

4.5.4 Определение тональности по отношению к аспектам

Как и в общей задаче анализа тональности по документам и предложениям, в задаче определения тональности по отношению к аспектам возможно использование двух основных методов: методов машинного обучения и инженерно-лингвистических методов.

Ключевой вопрос при проставлении оценок тональности аспектов заключается в том, как определить диапазон действия каждого оценочного выражения, относится ли оценочное выражение к аспекту, упомянутому в этом предложении [62]. Одно из основных направлений решения этой проблемы базируется на использовании синтаксической структуры предложений в форме деревьев зависимости [46, 42].

Методы машинного обучения для определения тональности по отношению к аспектам. В качестве примера использования метода машинного обучения для определения тональности по отношению к аспекту рассмотрим работу [42]. В данной работе на основе заранее собранных и вычитанных оценочных слов и аспектов задача проставления оценок аспектам рассматривается как задача классификации, т. е. для заданного предложения классификатор должен проставить, к какому именно аспектному термину относится данное оценочное слово, что может быть достаточно сложным для длинного предложения, в котором упомянуто несколько оценок и несколько аспектов (*хорошая пицца, но лазанья была ужасная*). В качестве признаков рассматриваются следующие:

- признаки расположения: расстояние между аспектным термином и оценочным словом, число аспектов и оценочных слов в предложении, длина предложения, пунктуация, наличие одних аспектов между другими аспектами и оценочными словами, порядок расположения аспекта и оценочного слова,
- лексические признаки: набор слов между аспектным термином и оценочным словом, наличие союзов и др.;
- части речи оценочного слова и аспектного термина, набор тегов частей речи между аспектом и оценочным словом, части речи соседних слов;
- признаки, основанные на синтаксической структуре: набор тегов по пути между аспектом и оценочным словом, близость по синтаксическому дереву.

В экспериментах было показано, что все четыре типа признаков существенны для выделения пары аспектный термин — оценочное слово, достигнутая F-мера составила 82.2%. Базовый уровень для сравнения, состоявший в том, что оценочное слово приписывается к ближайшему аспекту, составил — 76.6% F-меры. Авторы подчеркивают, что они ожидали, что прирост будет больше.

Лингвистико-инженерные методы проставления оценок аспектам. В лингвистико-инженерных методах предполагается, что на момент классификации известны:

- названия сущностей, их аспектов;
- имеется словарь оценочных слов и выражений, а также правила их преобразования в зависимости от контекста и правила суммирования.

Обработка идет обычно по предложениям и включает в себя несколько этапов [62].

Сначала производится проставление в предложении известных аспектных терминов и оценочных слов; оценочные слова имеют проставленную в словаре оценку тональности — в простейшем случае $\{1, -1\}$. К оценочным словам применяются операторы, которые могут менять тональность оценочного слова на противоположную.

Далее необходимо учесть структуру предложения для возможной модификации базовых оценок. В частности, в работе [35] указывается на важ-

ность обработки союзов типа *но, однако*. Если во второй части предложения не обнаружено оценочных слов, но присутствуют союзы *но* или *однако*, то второй части предложения должна быть приписана оценка, противоположная оценке первой части предложения.

В результате должно быть проведено агрегирование оценок по каждой аспектной категории. В работе [35] предлагается следующая процедура проставления оценок аспектов в отдельном предложении. Пусть в предложении s содержится набор аспектных терминов $a_1, \ldots, a_n$ и оценочных выражений $sw_1, \ldots, sw_n$, для которых оценки из словаря уже модифицированы с учетом операторов и контекста. Тогда оценки тональности каждого аспектного термина вычисляются по следующей формуле:

$$score(a_i, s) = \sum_{sw_j \in S} \frac{sw_j so}{dist(sw_j, a_i)}, \tag{6}$$

где sw_j — оценочное слово или выражение, $sw_j so$ — числовая оценка тональности sw_j , $dist(sw_j, a_j)$ — расстояние между оценочным словом и аспектом. Таким образом, к каждому аспектному термину в предложении приписываются все оценки, упомянутые в этом предложении, однако их вес падает в зависимости от расстояния между аспектом и оценкой. Если окончательный вес — положительный, то и оценка аспекта положительная, отрицательный вес означает отрицательную оценку, вес 0 — нейтральную оценку.

Результаты, представленные в [35], использующие вышеуказанную формулу, учет операторов, обработку союза *но* и учет контекстнозависимых оценочных слов, достигает F-меры 91% на 5 предметных областях. Система Opine на этих же данных получает 87% [101], алгоритм [50] — 83%.

4.6 Тестирование систем анализа тональности текстов

Задача автоматического анализа тональности текстов является сложной комплексной проблемой. Поэтому организуются различные открытые

тестирования подходов к анализу тональности текстов. Примерами таких тестирований являются Blog Track, проводимый в рамках конференции TREC, в котором нужно по запросу найти мнение пользователя о сущности, упомянутой в запросе [70]; задания конференции TAC под названием Opinion QA Tasks [33], включающие нахождение ответов на вопросы, содержащие мнения; задания анализа мнений на конференции NTCIR, посвященной обработке текстов на восточных языках [107], анализ сообщений из Твиттера с целью мониторинга репутации заданного объекта [12] и др.

4.6.1 Тестирование систем анализа общей тональности текстов для русского языка

В 2011–2013 годах было организовано тестирование систем анализа тональности для русского языка, которое включало определение общей тональности отзыва и определение общей тональности новостных цитат [30].

Для работы с **отзывами** были выбраны три предметные области: фильмы, книги и цифровые камеры. Участники тестирования должны были классифицировать отзывы на два класса (положительный/отрицательный), три класса (положительный/отрицательный/нейтральный) или по пятибалльной шкале.

Обучающая коллекция для этого тестирования была основана на двух источниках. Во-первых, использовались отзывы с портала Imhonet (imhonet.ru), эти отзывы были снабжены оценкой пользователей по 10-балльной шкале. Во-вторых, обучающая коллекция отзывов о цифровых камерах с оценкой пользователей по 5-балльной шкале была получена с сайта Яндекс-маркет.

Для тестирования систем была собрана другая коллекция отзывов, которая изначально не имела проставленных оценок пользователей. Данная коллекция состояла из отзывов пользователей в блогах и была получена посредствам исполнения запросов в поисковой машине Яндекс-блоги. Таким образом, в данной задаче была попытка моделировать одну из существующих практических постановок задач, когда имеющиеся данные для обучения несколько отличаются от реальных данных, на которых долж-

на работать система. Кроме того, такая постановка задачи ставила всех участников в равные условия.

Основными мерами качества в данной задаче были правильность классификации (accuracy) и F-мера в варианте макро-усреднения. Макро-усреднение здесь означает, что сначала точность и полнота вычисляются для каждого класса в отдельности, затем находится среднее для значения каждой меры. Макро-меры позволяют лучше оценить, насколько хорошо системы различают объекты разных классов в условиях несбалансированной коллекции [71].

Участники применяли как подходы, основанные на различных методах машинного обучения, так и инженерно-лингвистические подходы. Однако подавляющее большинство лучших подходов в задачах классификации отзывов базируется на применении метода опорных векторов SVM [30].

Еще одним заданием тестирования систем анализа общей тональности была задача классификации коротких (в среднем, 1-2 предложения) фрагментов прямой или косвенной речи, извлеченных из новостных сообщений (далее **цитаты**), например,

По мнению эксперта, глава белорусского государства больше всего боится, что страну все-таки лишат права провести чемпионат мира по хоккею в 2014 году.

Цитаты нужно было классифицировать на три класса: позитивный, негативный, нейтральный. В качестве обучающей коллекции было размечено и выдано участникам 4260 цитат. Для тестирования было разослано более 120 тысяч цитат, но реальное оценивание производилось на 5500 цитатах. Для оценки подходов также применялись ранее использованные меры: макро F-мера и правильности классификации.

В противоположность задаче классификации отзывов все лучшие подходы были основаны на лингвистических знаниях (словарь + правила), что, видимо, связано с отсутствием большой обучающей коллекции и широтой тематик цитат [30].

4.6.2 Тестирование систем анализа тональности по аспектам

С 2014 года в рамках международной конференции SemEval организуется тестирование систем анализа тональности по отношению к аспектам сущности [100]. Данные для обучения и тестирования включали изолированные предложения, извлеченные из отзывов в двух предметных областях: ресторанах и лаптопах. Для обучения в каждой из областей было подготовлено около 3 тысяч предложений. Множество аспектных категорий по ресторанам включали: food (еда), service (обслуживание), price (цена), ambience (обстановка, атмосфера), anecdotes/miscellaneous (другое).

В 2015 г. тестирование обработки отзывов в рамках SemEval[3] направлено на обработку полных отзывов. Аспектные категории усложняются и теперь состоят из пар сущность-характеристика (Entity-Atribute pairs — E#A). Набор пар E#A включает в области ресторанов шесть типов сущностей (RESTAURANT, FOOD, DRINKS, SERVICE, AMBIENCE, LOCATION) и 5 типов атрибутов (GENERAL, PRICES, QUALITY, STYLE_OPTIONS, MISCELLANEOUS). Область лаптопов содержит 22 типа сущностей and 9 типов атрибутов (GENERAL, PRICE, QUALITY, OPERATION_PERFORMANCE и др.). Примеры аннотирования предложений в области отзывов о ресторанах выглядят следующим образом:

Great for a romantic evening, but over-priced. (AMBIENCE#GENERAL), (RESTAURANT#PRICES)

The fajitas were delicious, but expensive. (FOOD#QUALITY), (FOOD# PRICES)

Мероприятие по оценке систем анализа тональности для текстов на русском языке SentiRuEval, которое было организовано в 2014–2015 гг., было направлено на исследование методов анализа текстов по отношению к некоторому заданному объекту или его характеристикам [66].

В 2014–2015 годах в рамках SentiRuEval было два типа задания: объектно-ориентированный анализ твитов для двух типов организаций (банки и телекоммуникационные компании) и аспектно-ориентированный

[3]http://alt.qcri.org/semeval2015/task12/

анализ отзывов пользователей в двух предметных областях (рестораны и автомобили). Рассмотрим задачу аспектно-ориентированного анализа тональности отзывов в рамках SentiRuEval.

Каждый отзыв содержит мнения пользователя о конкретном объекте. Такие мнения структурируются по заранее заданному набору целевых аспектов, т. е. составных частей, либо характеристик оцениваемого объекта. Для ресторанной тематики такими аспектами являются: кухня, интерьер, сервис, цена. Для автомобилей список аспектов включает в себя: безопасность, комфорт, надежность, внешний вид, цены, ходовые качества. Набор целевых аспектов дополнен аспектом «объект в целом», представляющим общее мнение об объекте.

Для создания обучающей коллекции была осуществлена разметка отзывов, при которой в тексты вносилась следующая информация:

- выделяются аспектные термины, включая эксплицитные, имплицитные и тональные факты;
- выделенным аспектным терминам приписывается их тональность: позитивный, негативный, противоречивый (both) и нейтральный;
- выделенные аспектные термины относятся к аспектной категории;
- отмечается статус выделенного аспектного термина относительно текущего мнения: релевантный (REL), относится к прошлому мнению автора или других людей (PREV), относится к другому объекту (CMPR), относится к гипотетической ситуации (IRR), ирония (IRN). Такая разметка помогает выявить аспектные термины, учет которых может ухудшить качество анализа, поскольку они не относятся к текущему мнению автора;
- приписывается оценка аспектной категории в целом по отзыву: нейтральный, положительный, отрицательный, противоречивый, оценка отсутствует.

Участники могли решать следующие задачи на выбор.

- **Задача А**. Выделение релевантных отзыву эксплицитных аспектных терминов. При этом не должны размечаться как эксплицитные аспектные термины упоминания, относящихся к другим объектам или ситуациям, упоминаемым в отзывах.

- **Задача Б**. Выделение релевантных отзыву всех аспектных терминов, включая неявные аспектные термины и тональные факты.
- **Задача В**. Присваивание оценки тональности эксплицитным аспектным терминам.
- **Задача Г**. Присвоение аспектной категории эксплицитным аспектным терминам.
- **Задача Д**. Заполнение оценок аспектных категорий по отзывам в целом.

Для каждой задачи организаторами были подготовлены прогоны, представляющие базовые уровни (baseline) для сравнения, т.е представляющие собой очень простые решения поставленных задач.

Базовая система для задач А и Б извлекает список размеченных терминов из обучающей коллекции, лемматизирует их и размечает их в тестовой коллекции на основе ее лемматизированного представления. Если к некоторой последовательности слов применимо более одного термина, то предпочитается более длинный термин.

Базовая система задачи В приписывает аспектному термину его наиболее частотную аспектную категорию, на основе информации из обучающей коллекции. Если термин отсутствует в обучающей коллекции, то приписывается наиболее частотная аспектная категория. Базовая система задачи Г приписывает аспектным терминам тональности на основе таких же принципов. Базовый уровень для задачи Е представляет собой наиболее частую категорию тональности для каждой аспектной категории (во всех случаях это была положительная тональность).

В тестировании приняли участие 11 участников, причем задача анализа отзывов о ресторанах привлекла значительно больше внимания, чем отзывы об автомобилях. Как указывается в [66], лучшие результаты, полученные участниками для задач А и Б по извлечению аспектных терминов, пока не намного превзошли базовый метод извлечения аспектных терминов, переносящий разметку из обучающего множества в тестовое. Например, при точном сопоставлении эксплицитных аспектов по ресторанам лучший результат составил — 63.2% F-меры , а baseline результат — 60.8%. Многие участники не смогли превзойти результат базового прогона. Задачи

В и Г являются задачами классификации аспектных терминов, и лучшие результаты были получены на основе методов машинного обучения SVM и Gradient Boosting.

Среди особенностей применяемых подходов для решения разных типов задач можно назвать использование новых, недавно появившихся типов учитываемых факторов, заключающихся в использовании нейронных сетей для представления слов коллекции в виде векторов, т.н. word embedding [76], такие факторы использовались в работах [23, 75, 116].

Обучающие и тестовые данные, результаты участников, а также скрипты для подсчета результатов доступны по адресу: http://goo.gl/Wqsqit.

4.6.3 Тестирование систем анализ тональности сообщений Твиттера

Одним из популярных объектов анализа тональности являются сообщения Твиттера. Эти сообщения представляют собой тексты до 140 символов и являются очень удобным средством для опубликования своего мнения о самых разных объектах и темах [93, 54].

С 2013 года, в рамках конференции SemEval проходили тестирования систем автоматического распознавания тональности. Участникам было предложено две задачи: классификация сообщений на уровне фраз и классификация сообщений на уровне целого сообщения. В первой задаче требовалось определить, является ли данная фраза позитивно, негативно или нейтрально окрашенной. Во второй задаче требовалось определить, выражает ли данное сообщение позитивное или негативное мнение автора в противоположность объективной информации [83, 105].

В 2012–2014 годах, в рамках конференции CLEF, проходило соревнование систем оценки репутации (online reputation management systems) RepLab [12, 13]. Цель тестирования состояла в том, чтобы определить, положительно или отрицательно твит влияет на репутацию компании. Организаторы RepLab делают акцент на том, что определение тональности репутации существенно отличается от обычного определения тональности, в котором требуется отличить субъективную информацию от объективной.

При определении тональности репутации должны приниматься во внимание как факты, так и субъективные мнения. Совокупность фактов и мнений помогает определить, имеет ли текст позитивные или негативные последствия для репутации выбранного объекта [12, 13].

В качестве наборов данных RepLab были представлены твиты из следующих предметных областей: автомобили, банки, университеты и музыка. Для каждой предметной области было собрано как минимум 2200 твитов: первые 700 твитов формировали обучающую коллекцию, оставшиеся 1500 — тестовую. Обучающие и тестовые коллекции собирались с интервалом в несколько месяцев. Оценка систем была представлена на единой тестовой коллекции, без деления на предметные области.

В 2015–2016 годах тестирования, похожие на тестирование RepLab, были организованы и для русского языка в рамках тестирования SentiRuEval [5, 69]. Целью задачи объектно-ориентированного анализа твитов по тональности для русского языка заключались в выявлении сообщений, которые оказывают влияние на репутацию организации, упомянутой в твите. Такие твиты могут содержать как положительное или отрицательное мнение автора, так и положительный или отрицательный факт относительно упомянутой организации.

В качестве предметных областей были выбраны твиты о телекоммуникационных компаниях (ТКК) и твиты о банках. Важно понимать, что исследуется задача оценки отзыва по отношению к компании, а не текста сообщения в целом. Отличие постановка задачи в рамках SentiRuEval от тестирования RepLab, состоит в том, что были выбраны твиты из двух предметных областей, и результат работы систем участников для этих предметных областей считался по отдельности, что дает возможность изучить зависимость результатов репутационного анализа твитов от конкретной предметной области. Для проведения тестирования были взяты твиты о восьми банках и семи телекоммуникационных компаниях.

Подходы участников тестирования анализа тональности твитов в 2015 и 2016 годах существенно различались [5, 69]. В 2015 году основным подходом был метод машинного обучения SVM, и в качестве данных использовалась только обучающая коллекция. Это привело к тому, что системы

ошибались в классификации твитов тестовой выборки, если они содержали оценочные слова и выражения, которые отсутствовали в обучающей выборке [5].

В 2016 году лучшим подходом стал подход на основе нейронных сетей, который использовал векторные представления слов, насчитанные на большой коллекции комментариев пользователей [15]. Такие представления позволяют лучше преодолевать различия в обучающей и тестовой выборке за счет того, что сходные по смыслу слова имеют сходные векторные представления. Конкретное слово, возможно, не встречалось в обучающей выборке, но в ней могли встречаться сходные слова.

Следующие по качеству результатов подходы комбинировали машинное обучение и имеющиеся словари русского языка [69]. Для этого создавался дополнительный набор признаков для представления сообщений, включающий такие признаки, как количество положительных слов из словаря, отрицательных слов, средняя оценка твита по словарю [10, 68]. Таким образом, происходило обучение классификатора не только на основе каких-то конкретных слов, встретившихся в обучающей коллекции, но и на основе информации из словаря в целом. Таким образом, слово из словаря, отсутствующее в обучающей коллекции, может быть теперь учтено при классификации твитов.

4.7 Заключение

В данной главе были представлены различные задачи анализа тональности и основные подходы их решения. Показано разнообразие текстов, используемых для анализа, и приложений.

Первые результаты автоматического анализа текстов могут быть получены достаточно быстро, как на основе инженерно-линвистического подхода, так и на основе методов машинного обучения. Однако попытки улучшить качество систем анализа тональности упираются в большое количество разнообразных проблем, которые были перечислены в данной работе: оценочные факты, учет сферы действия отрицаний, учет нереального контекста, обработка сравнений и многое другое.

Благодарности

Работа частично поддержана грантом РФФИ, проект 15-07-09306.

4.8 Список литературы

[1] Алексеева С. В., Кольцов С. Н., Кольцова О. Ю. Linis-crowd. org: лексический ресурс для анализа тональности социально-политических текстов на русском языке //Компьютерная лингвистика и вычислительные онтологии: сборник научных статей. Труды XVIII объединенной конференции «Интернет и современное общество» (IMS-2015), Санкт-Петербург, 2015. С. 25-34.

[2] Борисова Е. Г., Пирогова Ю. К. Моделирование нетривиальных условий понимания сообщения (на примере иронии) //Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» 2013. С. 148-162.

[3] Воронцов К.В., Потапенко А.А. Модификации EM-алгоритма для вероятностного тематического моделирования // Машинное обучение и анализ данных, 2013. V.1. № 6. С. 657–686.

[4] Ермаков А.Е., Киселев С.Л. Лингвистическая модель для компьютерного анализа тональности публикаций СМИ //Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог. 2005. С. 282-285.

[5] Лукашевич Н. В., Рубцова Ю. В. Объектно-ориентированный анализ твитов по тональности: результаты и проблемы //Аналитика и управление данными в областях с интенсивным использованием данных. 2015. С. 499-507.

[6] Марчук А. А., Уланов А. В., Макеев И. В., Чугреев, А. А. Автоматическое извлечение параметров продуктов из текстов отзывов при помощи интернет-статистик // Труды Международной конференции

Компьютерная лингвистика и информационные технологии Диалог-2013. 2013. т.2. С. 81–91.

[7] Пазельская А. Г., Соловьев А. Н. Метод определения эмоций в текстах на русском языке //Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог»(Бекасово, 25–29 мая 2011 г.). М.: Изд-во РГГУ. 2011. С. 17.

[8] Рубцова Ю. В. Метод построения и анализа корпуса коротких текстов для задачи классификации отзывов //Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XV Всероссийской научной конференции RCDL. 2013. С. 269-275.

[9] Рубцова Ю. В. Построение корпуса текстов для настройки тонового классификатора //Программные продукты и системы. 2015. №. 1 (109).

[10] Русначенко Н., Лукашевич Н. Методы интеграции лексиконов в машинное обучение для систем анализа тональности // Искусственный интеллект и принятие решений, N 2. 2017. С. 78-89.

[11] Телия В.Н. Коннотативный аспект семантики номинативных единиц. — М.: Наука, 1986. — 143 с.

[12] Amigo E., Corujo A., Gonzalo J., Meij E., Rijke M. Overview of RepLab 2012: Evaluating Online Reputation Management Systems // CLEF-2012. 2012.

[13] Amigo E., Albornoz J.C., Chugur I., Corujo A., Gonzalo J., Martin T., Meij E., de Rijke M, Spina D. Overview of RepLab 2013: Evaluating online reputation monitoring systems //Information Access Evaluation. Multilinguality, Multimodality, and Visualization. Springer Berlin Heidelberg, 2013. P. 333

[14] Andrzejewski D., Zhu X., Craven M. Incorporating domain knowledge into topic modeling via Dirichlet forest priors // Proceedings of ICML. 2009. P. 25–32.

[15] Arkhipenko K., Kozlov I., Trofimovich J., Skorniakov K., Gomzin A., Turdakov D. Comparison of Neural Network Architectures for Sentiment Analysis of Russian Tweets // In Proceedings of International Conference on computational linguistics and intellectual technologies Dialog-2016. 2016. P. 50-58.

[16] Aue A., Gamon M. Customizing sentiment classifiers to new domains: A case study // In Proceedings of International Conference on Recent Advances in Natural Language Processing, Borovets, BG, 2005.

[17] Baccianella, S., Esuli, A., Sebastiani F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining // Proceedings of LREC-2010, V. 10, 2010. P. 2200-2204.

[18] Bagheri A., Saraee M., de Jong F. An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews // Natural Language Processing and Information Systems. Springer: Berlin Heidelberg, 2013. P. 140–151.

[19] Ben-Ami Z., Feldman R., Rosenfeld B. Entities' Sentiment Relevance //In Proceedings of ACL-2014. 2014. P. 87-92.

[20] Benamara F., Taboada M., Mathieu Y. Evaluative language beyond bags of words: Linguistic insights and computational applications //Computational Linguistics, V.43, 2017. P. 201-264.

[21] Blair-Goldensohn S., Hannan K., McDonald R., Neylon T., Reis G. A., Reynar J. Building a sentiment summarizer for local service reviews // Proceedings of WWW Workshop on NLP in the Information Explosion Era. 2008.

[22] Blei D., Ng A., Jordan M. Latent dirichlet allocation // The Journal of Machine Learning Research, 2003. № 3. P. 993–1022.

[23] Blinov P.D., Kotelnikov E.V. Semantic Similarity for Aspect-Based Sentiment Analysis // Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2015. 2015. V. 2. P. 23–33.

[24] Bollen J., Mao H., Zeng X. Twitter mood predicts the stock market //Journal of computational science. 2011. T. 2. N. 1. P. 1-8.

[25] Bradley M.M., Lang P.J. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report C-1, the center for research in psychophysiology, University of Florida, 1999. P. 1-45.

[26] Cambria E., Livingstone A., and A. Hussain. The hourglass of emotions // Cognitive Behavioural Systems, Lecture Notes in Computer Science, vol. 7403, Springer, 2012, P. 144–157.

[27] Cambria E., Olsher D., Rajagopal D. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis //Twenty-eighth AAAI conference on artificial intelligence. 2014.

[28] Chetviorkin I., Loukachevitch N. Extraction of domain-specific opinion words for similar domains //Proceedings of the Workshop on Information Extraction and Knowledge Acquisition held in conjunction with RANLP-2011. 2011. P. 7-12.

[29] Chetviorkin, I., Loukachevitch N. Extraction of Russian Sentiment Lexicon for Product Meta-Domain // Proceedings of COLING-2012, 2012. P. 593-610.

[30] Chetviorkin I., Loukachevitch N. Evaluating Sentiment Analysis Systems in Russian // Proceedings of BSNLP Workshop, ACL 2013. 2013. P. 12–16.

[31] Choi Y., Cardie C. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification // In Proceedings of EMNLP-'09, 2009. P. 590–598.

[32] Choi Y., Cardie C. Hierarchical sequential learning for extracting opinions and their attributes // Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010). 2010. P. 269–274.

[33] Dang H. T., Owczarzak K. Overview of the tac 2008 opinion question answering and summarization tasks // Proceedings of the First Text Analysis Conference. 2008.

[34] Deng L., Wiebe J. MPQA 3.0: An Entity/Event-Level Sentiment Corpus //HLT-NAACL. 2015. P. 1323-1328.

[35] Ding X., Liu B., Yu Ph. A holistic lexicon-based approach to opinion mining // Proceedings of the Conference on Web Search and Web Data Mining (WSDM-2008). 2008. P. 231–240.

[36] Dubatovka A., Kurochkin Yu., Mikhailova E. Automatic Generation of the domain-specific Russian dictionaries // In proceedings of International Conference on Computational Linguistics and Intellectual Technologies Dialog-2016. 2016. P. 146-158.

[37] Esuli A., Sebastiani F. SENTIWORDNET: A high-coverage lexical resource for opinion mining //Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR). — 2006.

[38] Fellbaum C. WordNet. — John Wiley and Sons, Inc., 1998.

[39] Feng S., Kang J. S., Kuznetsova P., Choi Y. Connotation Lexicon: A Dash of Sentiment Beneath the Surface Meaning // Proceedings of ACL. 2013. P. 1774–1784.

[40] Frantzi K., Ananiadou S., Mima H. Automatic recognition of multi-word terms: the C-value/NC-value method // International Journal on Digital Libraries, 2000. V. 3. № 2. 115–130.

[41] Gao, D., Wei, F., Li, W., Liu, X., Zhou, M. Cross-lingual sentiment lexicon learning with bilingual word graph label propagation. Computational Linguistics. 2015.

[42] Glavas G., Korencic D., Snajder J. Aspect-Oriented Opinion Mining from User Reviews in Croatian // Proceedings of BSNLP workshop, ACL-2013. 2013. P. 18–23.

[43] Go A., Bhayani R., Huang L. Twitter sentiment classification using distant supervision //CS224N Project Report, Stanford. 2009. V. 1. №. 2009. P. 12.

[44] Gupta N. K. Extracting phrases describing problems with products and services from twitter messages // Computacion y Sistemas. 2013. V. 17, №2. P. 197–206.

[45] Ivanov V., Tutubalina E. Clause-based approach to extracting problem phrases from user reviews of products // Analysis of Images, Social Networks and Texts. Springer International Publishing, 2014. P. 229–236.

[46] Jiang, Long, Mo Yu, Ming Zhou, Xiaohua Liu, Tiejun Zhao. Target dependent twitter sentiment classification // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011). 2011. P. 151–160.

[47] Jijkoun V., de Rijke M., and Weerkamp W. Generating focused topic-specific sentiment lexicons // In Proceedings of ACL '10. 2010. P. 585–594.

[48] Hai Z., Chang K., Cong G. One seed to find them all: mining opinion features via association // Proceedings of the 21st ACM international conference on Information and knowledge management. 2012. ACM. P. 255–264.

[49] Hatzivassiloglou V., McKeown K. R. Predicting the semantic orientation of adjectives //Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. 1997. P. 174-181.

[50] Hu M., Liu B. Mining and summarizing customer reviews // Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004. P. 168–177.

[51] Kanayama H., Nasukawa T. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis // In Proceedings of EMNLP-2006. 2006. P. 355–363.

[52] Koltsova O. Y., Alexeeva S. V., Kolcov S. N. An Opinion Word Lexicon and a Training Dataset for Russian Sentiment Analysis of Social Media //In Proceedings of International conference on computational linguistics and intellectual technologies Dialog-2016. 2016. P. 277-287.

[53] Kotelnikov E. V., Bushmeleva N. A., Razova E. V., Peskisheva T. A., Pletneva M. V. Manually created sentiment lexicons: Research and Development // In Proceedings of International conference on computational linguistics and intellectual technologies Dialog-2016. 2016. P. 300-314.

[54] Kouloumpis E, Wilson T, Moore JD. Twitter sentiment analysis: The good the bad and the omg! // In Proceedings of Icwsm-2011. 2011. P.538-541.

[55] Kovelamudi S., Ramalingam S., Sood A., Varma V. Domain Independent Model for Product Attribute Extraction from User Reviews using Wikipedia // Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP-2010). 2011. P. 1408–1412.

[56] Ku Lun-Wei, Yu-Ting Liang, Hsin-Hsi Chen. Opinion extraction, summarization and tracking in news and blog corpora // Proceedings of AAAI-CAAW'06. 2006.

[57] Kunneman, Florian, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. Signaling sarcasm: From hyperbole to hashtag // Information Processing and Management, 51. 2015. P. 500–509.

[58] Kuznetsova E. S., Loukachevitch N. V., Chetviorkin I. I. Testing rules for a sentiment analysis system // In Proceedings of International conference on computational linguistics and intellectual technologies Dialog-2013. v.2, 2013. P. 71-81.

[59] Lau, Raymond, Lai, Chun-Lam, Bruza, Peter D., Wong, Kam-Fa. Pseudo Labeling for Scalable Semi-supervised Learning of Domain-specific Sentiment Lexicons // In 20th ACM Conference on Information and Knoweledge Management. 2011.

[60] Zhang L., Liu B. Identifying noun product features that imply opinions // Proceedings of the Annual Meeting of the Association for Computational Linguistics (short paper) (ACL-2011). 2011. P. 575–580.

[61] Zhang L., Liu B. Extracting Resource Terms for Sentiment Analysis // Proceedings of IJCNLP-2011. 2011. P.1171–1179.

[62] Liu B., Zhang L. A survey of opinion mining and sentiment analysis // Mining Text Data. Springer: US, 2012. P. 415–463.

[63] Liu B. Sentiment analysis and Subjectivity // Handbook of Natural Language Processing. CRC Press, Taylor and Francis Group, Boca Raton, 2010. P. 1–38.

[64] Loukachevitch N., Nokel M. An Experimental Study of Term Extraction for Real Information-Retrieval Thesauri // Proceedings of Terminology and Artificial Intelligence Conference TIA-2013. 2013. P. 69–78.

[65] Loukachevitch N., Dobrov B. RuThes linguistic ontology vs. Russian wordnets //Proceedings of Global WordNet Conference GWC-2014. 2014.

[66] Loukachevitch N., Blinov P., Kotelnikov E., Rubtsova Y., Ivanov V., Tutubalina E. SentiRuEval: Testing Object-oriented Sentiment Analysis Systems in Russian // Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2015. 2015. V. 2. P. 2–13.

[67] Loukachevitch N., Kotelnikov E., Blinov P. Types of Aspect Terms in Aspect-Oriented Sentiment Labeling //BSNLP 2015. 2015. P. 90.

[68] Loukachevitch N. V., Levchik A. Creating a General Russian Sentiment Lexicon // In Proceedings of LREC-2016. 2016.

[69] Loukachevitch N., Rubtsova Y. V. SentiRuEval-2016: Overcoming Time Gap and Data Sparsity in Tweet Sentiment Analysis //Computational Linguistics and Intellectual Technologies Proceedings of the Annual International Conference Dialogue, Moscow, RGGU. 2016. P. 416-427.

[70] Macdonald C., Santos R. L., Ounis I., Soboroff I. Blog track research at TREC // SIGIR Forum 44(1), 2010. P. 58–75.

[71] Manning C. D., Raghavan P., Schutze H. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008.

[72] Martineau J., Finin T. Delta TFIDF: An Improved Feature Space for Sentiment Analysis // In Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media , San Jose, CA, May. AAAI Press. 2009.

[73] McKeown, K., Daume, H., Chaturvedi, S., Paparrizos, J., Thadani, K., Barrio, P., Gravano, L. Predicting the impact of scientific concepts using full text features // Journal of the Association for Information Science and Technology, 67(11). 2016. P. 2684-2696.

[74] Mihalcea R., Banea, C., Wiebe J. Learning multilingual subjective language via cross-lingual projections // In Proceedings of the 45th AnnualMeeting of the Association of Computational Linguistics ACL-2007. 2007. P. 976–983.

[75] Mayorov V., Andrianov I., Astrakhantsev N., Avanesov V., Kozlov I., Turdakov D. A High Precision Method for Aspect Extraction in Russian // Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2015. V. 2. 2015. P. 34–43.

[76] Mikolov T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. Distributed representations of words and phrases and their compositionality // Advances in neural information processing systems. 2013. P. 3111–3119.

[77] Mintz, M., Bills, S., Snow, R., Jurafsky, D. Distant supervision for relation extraction without labeled data // In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, V. 2. 2009. P. 1003-1011.

[78] Moghaddam S., Ester M. Aspect-based Opinion Mining from Online Reviews. Tutorial at SIGIR-2012. 2012.

[79] Mohammad, S. M., Turney, P. D. Crowdsourcing a word–emotion association lexicon // Computational Intelligence. 29(3). 2013. P. 436-465.

[80] Mohammad S. M., Kiritchenko S., Zhu X. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets //Second Joint Conference on Lexical and Computational Semantics (* SEM). V. 2. 2013. P. 321-327.

[81] Mukherjee A., Liu B. Aspect Extraction through Semi-Supervised Modeling // Proceedings of 50th Annual Meeting of Association for Computational Linguistics (ACL-2012). 2012. P. 339–348.

[82] Musto, C., Semeraro G., Polignano M. A comparison of lexicon-based approaches for sentiment analysis of microblog posts // In Proceedings of the 8th International Workshop on Information Filtering and Retrieval Co-located with XIII AI/*IA Symposium on Artificial Intelligence (AIIA 2014). 2014. P. 59–68.

[83] Nakov P. et al. Semeval-2013 task 2: Sentiment analysis in twitter // Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013). 2013. P. 312–320.

[84] Narayanan, R., Liu B., Choudhary A. Sentiment analysis of conditional sentences // In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2009). 2009.

[85] Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., Ngo, D. C. L. Text mining for market prediction: A systematic review. Expert Systems with Applications, 41(16). 2014. P. 7653-7670.

[86] Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., Ngo, D. C. L. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment // Expert Systems with Applications, 42(1). 2015. P. 306-324.

[87] Neviarouskaya A., Prendinger H., Ishizuka M. Recognition of affect, judgment, and appreciation in text // Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010). 2010. P. 806–814.

[88] Nielsen F. A new ANEW: evaluation of a word list for sentiment analysis in microblogs // Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages. 2011. P. 93–98.

[89] Niklas J., Gurevych I. Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields // Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010). 2010. P. 1035–1045.

[90] Nitin Jindal and Bing Liu. Mining Comparative Sentences and Relations // Proceedings of AAAI-2006, 2006.

[91] Nozza D., Fersini E., Messina E. A Multi-View Sentiment Corpus //In Proceedings of EACL-2017. 2017. P. 273-280.

[92] Ozdemir C., Bergler S. A Comparative Study of Different Sentiment Lexica for Sentiment Analysis of Tweets // Proceedings of RANLP-2015. 2015. P. 488-496.

[93] Pak A, Paroubek P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining // In proceedings of LREC-2010. 2010. P. 1320-1326.

[94] Paltoglou G., Thelwall M. A study of information retrieval weighting schemes for sentiment analysis //Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics ACL-2010. 2010. P. 1386-1395.

[95] Pan S.J., Ni, X., Sun, J. T., Yang, Q., Chen, Z. Cross-domain sentiment classification via spectral feature alignment //Proceedings of the 19th international conference on World wide web. ACM, 2010. P. 751-760.

[96] Pang B., Lee L., Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques // Proceedings of the ACL-02

conference on Empirical methods in natural language processing, V. 10. 2002. P. 79–86.

[97] Pang B., Lee L. Opinion mining and sentiment analysis //Foundations and Trends in Information Retrieval. 2008. V. 2. №. 1–2. P. 1-135.

[98] Perez-Rosas V., Banea C., Mihalcea R. Learning Sentiment Lexicons in Spanish // Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), 2012.

[99] Plungyan V. A. Irrealis and modality in Russian and in typological perspective //Modality in Slavonic languages. 2005. P. 187-198.

[100] Pontiki M., Galanis D., Pavlopoulos J., Papageorgiou H., Androutsopoulos I., Manandhar S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis // Proceedings of International Workshop on Semantic Evaluations SemEval-2014. 2014. P. 27–35.

[101] Popescu, A., Etzioni O. Extracting product features and opinions from reviews // Natural language processing and text mining. Springer: London. 2007. P. 9–28.

[102] Poria, S., Gelbukh, A. F., Agarwal, B., Cambria, E., Howard, N. Common Sense Knowledge Based Personality Recognition from Text // In Proceedigns of MICAI-2013. 2013. P. 484-496.

[103] Qiu G., Liu B, Bu J, Chen C. Opinion word expansion and target extraction through double propagation // Computational Linguistics. 2011. V. 1. № 1. P. 1–18.

[104] Reyes A., Rosso P., Veale T. A multidimensional approach for detecting irony in twitter //Language resources and evaluation. 2013. P. 1-30.

[105] Rosenthal S., Ritter A., Nakov P., Stoyanov V. SemEval-2014 Task 9: Sentiment Analysis in Twitter // Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 2014. P. 73–80.

[106] Scaffidi Ch., Bierhoff K., Chang E., Felker M., Ng H., Jin Ch.. Red Opal: product-feature scoring from reviews // Proceedings of Twelfth ACM Conference on Electronic Commerce (EC-2007). 2007. P. 182–191.

[107] Seki Y. et al. Overview of multilingual opinion analysis task at NTCIR-7 // Proceedings of the Seventh NTCIR Workshop. 2008. P. 185–203.

[108] Severyn A., Moschitti A. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification //SemEval@ NAACL-HLT. 2015. P. 464-469.

[109] Shalunts G., Backfried G. SentiSAIL: sentiment analysis in English, German and Russian //International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, Cham, 2015. P. 87-97.

[110] Steinberger J., Lenkova P., Ebrahim M., Ehrmann M., Hurriyetogly A., Kabadjov M., Steinberger R., Tanev H., Zavarella V. Vazquez S. Creating Sentiment Dictionaries via Triangulation //In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT-2011, 2011. P. 28–36.

[111] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. The General Inquirer: A computer approach to content analysis. MIT Press, 1966.

[112] Strapparava, Carlo and Alessandro Valitutti. WordNet-Affect: An affective extension of WordNet // In Proceedings of the International Conference on Language Resources and Evaluation. 2004. P. 1083–1086.

[113] Sulis E., Farias, D. I. H., Rosso, P., Patti, V., Ruffo. Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not //Knowledge-Based Systems. 2016. V. 108. P. 132-143.

[114] Taboada M. Brooke, J., Tofiloski, M., Voll, K., Stede, M.. Lexicon-based methods for sentiment analysis //Computational linguistics. 2011. V. 37. №. 2. P. 267-307.

[115] Tang, D., Qin, B., Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification// In Proceedings EMNLP-2015. 2015. P. 1422-1432.

[116] Tarasov D.S. Deep Recurrent Neural Networks for Multiple Language Aspect based Sentiment Analysis of User Reviews // Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2015. V. 2. 2015. P. 53–64.

[117] Titov I., McDonald R. A joint model of text and aspect ratings for sentiment summarization // Urbana, 51, 61801. 2008.

[118] Turney P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews // In proceedings of ACL-2002, 2002.

[119] Tutubalina E., Ivanov V. Unsupervised Approach to Extracting Problem Phrases from User Reviews of Products // Proceedings of the Aha! workshop on Information Discovery in Texts, Coling-2014. 2014. P. 48–53.

[120] Vepsäläinen T., Li H., Suomi R. Facebook likes and public opinion: Predicting the 2015 Finnish parliamentary elections //Government Information Quarterly. 2017.

[121] Vilares D., Thelwall M., Alonso M. A. The megaphone of the people? Spanish SentiStrength for real-time analysis of political tweets //Journal of Information Science. 2015. V. 41. №. 6. P. 799-813.

[122] Vo D.T., Zhang Y. Don't count, predict! an automatic approach to learning sentiment lexicons for short text //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. V. 2. P. 219-224.

[123] Volkova S., Coppersmith G., Van Durme B. Inferring User Political Preferences from Streaming Communications //ACL (1). 2014. P. 186-196.

[124] Volkova, S., Bachrach, Y., Armstrong, M., Sharma, V. Inferring Latent User Properties from Texts Published in Social Media // In Proceedings AAAI-2015. 2015. P. 4296-4297.

[125] Volkova S., Bell E. Account Deletion Prediction on RuNet: A Case Study of Suspicious Twitter Accounts Active During the Russian-Ukrainian Crisis //Proceedings of NAACL-HLT. 2016. P. 1-6.

[126] Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A. A survey on the role of negation in sentiment analysis // In Proceedings of the workshop on negation and speculation in natural language processing. 2010. P. 60-68.

[127] Wilson T., Sperber D. 2007. On verbal irony // Irony in language and thought. 2007. P. 35–56.

[128] Wilson, T., Wiebe, J., Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis // Proceedings of the conference on human language technology and empirical methods in natural language processing. 2005. P. 347-354.

[129] Yang S., Ko Y. Extracting Comparative Entities and Predicates from Texts Using Comparative Type Classification // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011. P. 1636-1644.

[130] Yu J., Zha Z. J., Wang M., Wang K., Chua T. S. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews // Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. 2011. P. 140–150.

[131] Zhai Z., Liu B., Xu H., Jia P. Grouping Product Features Using Semi-Supervised Learning with Soft-Constraints // Proceedings of Coling-2010. 2010. P. 1272–1280.

[132] Zhai Z., Liu B., Xu H., Jia P. Clustering product features for opinion mining // Proceedings of the fourth ACM international conference on Web search and data mining. ACM. 2011. P. 347–354.

[133] Zhang L., Liu B. Aspect and Entity Extraction for Opinion Mining // Data Mining and Knowledge Discovery for Big Data. Springer: Berlin Heidelberg, 2014. P. 1–40.

[134] Zhao Wayne Xin, Jing Jiang, Hongfei Yan, Xiaoming Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid // Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010). 2010. P. 56–65.

[135] Zhu J., Wang H., Tsou B., Zhu M. Multiaspect opinion polling from textual reviews // Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2009). 2009. P. 1799–1802.

[136] Zhuang L., Jing F., Zhu X. Movie review mining and summarization // Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2006), 2006. P. 43–50.

Глава 5

Обзор вероятностных тематических моделей

Воронцов К.В.

5.1 Введение

Тематическое моделирование — это одно из современных направлений статистического анализа текстов, активно развивающееся с конца 90-х годов. *Вероятностная тематическая модель* (probabilistic topic model) выявляет тематику коллекции документов, представляя каждую тему дискретным распределением вероятностей терминов, а каждый документ — дискретным распределением вероятностей тем.

Тематическое моделирование похоже на кластеризацию документов. Отличие в том, что при кластеризации документ целиком относится к одному кластеру, тогда как тематическая модель осуществляет «мягкую кластеризацию» (soft clustering), относя документ к нескольким кластерам-темам с некоторыми вероятностями. Тематические модели называют также моделями мягкой би-кластеризации, поскольку каждый термин также распределяется по нескольким темам.

Последнее свойство позволяет обходить проблемы синонимии и полисемии слов. Синонимы, взаимозаменяемые в схожих контекстах, группируются в одних и тех же темах. Многозначные слова и омонимы, наоборот,

распределяют свои вероятности по нескольким семантически не связанным темам. Например, значение слова «ядро» может быть понято из того, какая тема доминирует в контексте данного слова — математика, физика, биология или военная история.

В роли терминов могут выступать как отдельные слова, так и словосочетания. Тема образуется семантически связанными, часто совместно встречающимися терминами. Такое определение «темы» допускает точную математическую формализацию, но может отличаться от принятых в лингвистике или литературоведении.

Сжатое описание документа в виде вектора вероятностей тем содержит важнейшую информацию о семантике документа и может использоваться для решения многих задач текстовой аналитики. Тематические модели применяются для выявления трендов в новостных потоках, патентных базах, архивах научных публикаций [152, 121], многоязычного информационного поиска [131, 130], поиска тематических сообществ в социальных сетях [154, 123, 97, 27], классификации и категоризации документов [106, 155], тематической сегментации текстов [139], анализа изображений и видеопотоков [49, 66, 42, 122], тегирования веб-страниц [58], обнаружения текстового спама [10], в рекомендательных системах [146, 134, 62, 149, 148], в биоинформатике для анализа нуклеотидных [59] и аминокислотных последовательностей [111, 57], в задачах популяционной генетики [99]. Многие другие разновидности и приложения тематических моделей упоминаются в обзорах [35, 22].

Построение тематической модели по коллекции документов является некорректно поставленной оптимизационной задачей, которая может иметь бесконечное множество решений. Согласно теории регуляризации А. Н. Тихонова [11], решение такой задачи возможно доопределить и сделать устойчивым, добавив к основному критерию *регуляризатор* — дополнительный критерий, учитывающий какие-либо специфические особенности прикладной задачи или знания предметной области. В сложных приложениях дополнительных критериев может быть несколько.

Аддитивная регуляризация тематических моделей (additive regularization of topic models, ARTM) — это многокритериальный подход, в кото-

ром оптимизируется взвешенная сумма критериев [3, 126]. Большинство известных тематических моделей либо изначально формулируются в терминах регуляризации, либо допускают такую переформулировку. ARTM позволяет отделять регуляризаторы от одних моделей и использовать в других. Это приводит к модульной технологии моделирования. Собрав библиотеку часто используемых регуляризаторов, можно затем их комбинировать, чтобы строить тематические модели с требуемыми свойствами. Оптимизация любых моделей и их комбинаций производится одним и тем же обобщённым EM-алгоритмом. Эти идеи реализованы в библиотеке с открытым кодом `BigARTM`, доступной по адресу `http://bigartm.org`.

ARTM не является ещё одной моделью или ещё одним методом. Это общий подход к построению и комбинированию тематических моделей.

В литературе по тематическому моделированию в настоящее время доминируют методы байесовского обучения. Из-за сложности математического аппарата в статьях часто опускаются важные для понимания детали. Иногда авторы ограничиваются упрощённым описанием модели в виде порождающего процесса (generative story) или графической плоской нотации (plate notation). Последующий переход к алгоритму и его программной реализации остаётся неоднозначным и неочевидным.

Цель данного теоретического обзора — показать разнообразие задач и подходов тематического моделирования, сосредоточившись на первом и очень важном этапе моделирования — как от исходных требований и предположений перейти к формальной постановке оптимизационной задачи. Дальнейшие шаги в ARTM намного проще байесовского вывода, что позволяет сократить изложение, не скрывая математических выкладок. Сопоставимый по охвату и обстоятельности обзор байесовских тематических моделей занял бы сотни страниц.

В разделах 5.2 и 5.3 вводятся основные понятия. В разделах 5.4–5.11 в терминах регуляризации описываются разновидности тематических моделей. Эти разделы практически не связаны друг с другом, их можно читать в произвольном порядке или использовать как путеводитель по ссылкам на литературу. Раздел 5.12 посвящён оцениванию качества. В разде-

ле 5.13 обсуждается применение тематического моделирования для разведочного информационного поиска. В разделе 5.14 — краткое заключение.

5.2 Основы тематического моделирования

Пусть D — множество (коллекция) текстовых документов, W — множество (словарь) употребляемых в них терминов. Терминами могут быть как отдельные слова, так и словосочетания. Каждый документ $d \in D$ представляет собой последовательность n_d терминов $w_1, \ldots, w_{n_d}$ из словаря W.

Гипотеза о существовании тем. Предполагается, что каждое вхождение термина w в документ d связано с некоторой темой t из заданного конечного множества T. Коллекция документов представляет собой последовательность троек (w_i, d_i, t_i), $i = 1, \ldots, n$. Термины w_i и документы d_i являются наблюдаемыми переменными, темы t_i не известны и являются *латентными* (скрытыми) переменными.

Гипотеза «мешка слов». Предполагается, что порядок терминов в документах не важен для выявления тематики, то есть тематику документа можно узнать даже после произвольной перестановки терминов, хотя для человека такой текст потеряет смысл. Это предположение называют гипотезой «мешка слов» (bag of words). Порядок документов в коллекции также не имеет значения; это предположение называют гипотезой «мешка документов». Гипотеза «мешка слов» позволяет перейти к компактному представлению документа как *мультимножества* — подмножества $d \subset W$, в котором каждый элемент $w \in d$ повторён n_{dw} раз.

Вероятностные гипотезы. Предполагается, что конечное множество $D \times W \times T$ является *вероятностным пространством* с неизвестной функцией вероятности $p(d, w, t)$. Предполагается, что выборка троек (d_i, w_i, t_i) порождена из распределения $p(d, w, t)$ случайно и независимо. Это вероятностное уточнение гипотезы «мешка слов», из которого, в частности, следует, что все n троек в выборке равновероятны.

Гипотеза условной независимости. Предполагается, что появление слов в документе d по теме t зависит от темы, но не зависит от документа d,

и описывается общим для всех документов распределением $p(w\,|\,t)$:

$$p(w\,|\,d,t) = p(w\,|\,t). \tag{1}$$

Вероятностная порождающая модель. Согласно формуле полной вероятности и гипотезе условной независимости, распределение терминов в документе $p(w\,|\,d)$ описывается *вероятностной смесью* распределений терминов в темах $\varphi_{wt} = p(w\,|\,t)$ с весами $\theta_{td} = p(t\,|\,d)$:

$$p(w\,|\,d) = \sum_{t\in T} p(w\,|\,t,d)\,p(t\,|\,d) = \sum_{t\in T} p(w\,|\,t)\,p(t\,|\,d) = \sum_{t\in T} \varphi_{wt}\theta_{td}. \tag{2}$$

Вероятностная модель (2) описывает процесс порождения коллекции по известным распределениям $p(w\,|\,t)$ и $p(t\,|\,d)$, см. алгоритм 1 и рис. 5.1.

Построение тематической модели — это обратная задача: по заданной коллекции D требуется найти параметры модели φ_{wt} и θ_{td}, хорошо приближающей частотные оценки условных вероятностей $\hat{p}(w\,|\,d) = \frac{n_{dw}}{n_d}$.

Распределение вида $p(t\,|\,x)$ будем называть *тематикой* объекта x. Например, можно говорить о тематике документа $p(t\,|\,d)$, тематике термина $p(t\,|\,w)$, тематике термина в документе $p(t\,|\,d,w)$.

Низкоранговое матричное разложение. Обычно число тем $|T|$ много меньше $|D|$ и $|W|$, и задача сводится к поиску приближённого представления заданной матрицы частот терминов в документах $P = \big(\hat{p}(w\,|\,d)\big)_{W\times D}$ в виде произведения $P = \Phi\Theta$ двух неизвестных матриц меньшего размера — *матрицы терминов тем* $\Phi = (\varphi_{wt})_{W\times T}$ и *матрицы тем документов* $\Theta = (\theta_{td})_{T\times D}$. Все три матрицы P, Φ, Θ являются *стохастическими*, то есть имеют неотрицательные нормированные столбцы p_d, φ_t, θ_d, представляющие дискретные распределения.

Принцип максимума правдоподобия используется в математической статистике для оценивания параметров вероятностных моделей по наблюдаемым данным. Согласно этому принципу, выбираются такие значения параметров, при которых наблюдаемая выборка наиболее правдоподобна. *Функция правдоподобия* определяется как зависимость вероятности выборки от параметров модели. Благодаря предположению о независимости на-

Алгоритм 1. Вероятностная модель порождения коллекции.

Вход: распределения $p(w\,|\,t)$, $p(t\,|\,d)$;
Выход: выборка пар (d_i, w_i), $i = 1, \ldots, n$;

для всех $d \in D$
  задать длину n_d документа d;
  для всех $i = 1, \ldots, n_d$
    $d_i := d$;
    выбрать случайную тему t_i из распределения $p(t\,|\,d_i)$;
    выбрать случайный термин w_i из распределения $p(w\,|\,t_i)$;

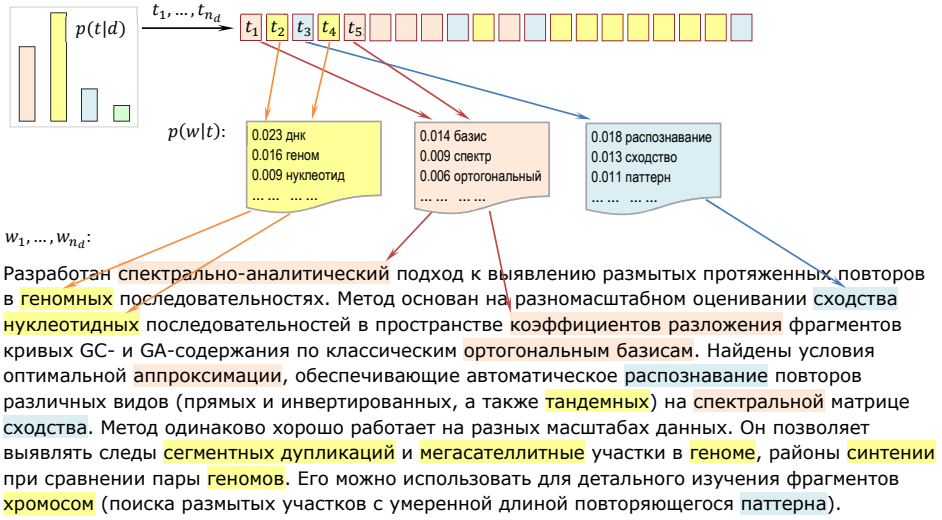


Рис. 5.1. Процесс порождения текстовой коллекции вероятностной тематической моделью (2): в каждой позиции i документа d_i сначала порождается тема $t_i \sim p(t\,|\,d_i)$, затем термин $w_i \sim p(w\,|\,t_i)$

блюдений, она равна произведению вероятностей слов в документах:

$$p\big((d_i, w_i)_{i=1}^n; \Phi, \Theta\big) = \prod_{i=1}^{n} p(d_i, w_i) = \prod_{d \in D} \prod_{w \in d} p(w\,|\,d)^{n_{dw}} \underbrace{p(d)^{n_{dw}}}_{\text{const}} \to \max_{\Phi,\Theta}.$$

Прологарифмировав правдоподобие, перейдём от произведения к сумме и отбросим слагаемые, не зависящие от параметров модели. Получим задачу максимизации логарифма правдоподобия (log-likelihood)

$$\sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \to \max_{\Phi,\Theta} \tag{3}$$

при ограничениях неотрицательности и нормировки всех столбцов φ_t, θ_d:

$$\sum_{w \in W} \varphi_{wt} = 1; \quad \varphi_{wt} \geqslant 0; \qquad \sum_{t \in T} \theta_{td} = 1; \quad \theta_{td} \geqslant 0. \tag{4}$$

Это задача *вероятностного латентного семантического анализа* (probabilistic latent semantic analysis, PLSA) [48]. Для её решения используется EM-алгоритм, который приводится ниже в более общей постановке.

Предварительная обработка текста перед построением тематических моделей обычно состоит из следующей серии преобразований.

Лемматизация — это приведение каждого слова в документе к его нормальной форме. В русском языке нормальными формами считаются: для существительных — именительный падеж, единственное число; для прилагательных — именительный падеж, единственное число, мужской род; для глаголов, причастий, деепричастий — глагол в инфинитиве. Хорошими лемматизаторами для русского языка считаются последние версии `mystem` и `pymorphy`.

Стемминг — это отбрасывание окончаний и других изменяемых частей слов. Он подходит для английского языка, для русского предпочтительна лемматизация.

Стоп-слова — это частые слова, встречающиеся в текстах любой тематики. Они бесполезны для тематического моделирования и могут быть отброшены. К ним относятся предлоги, союзы, числительные, местоимения, некоторые глаголы, прилагательные и наречия. Число таких слов обычно варьируется в пределах нескольких сотен. Их отбрасывание почти не влияет на объём словаря, но может приводить к заметному сокращению длины некоторых текстов.

Редкие слова также рекомендуется отбрасывать, поскольку они не могут повлиять на тематику коллекции. Отбрасывание редких слов, а также строк, не являющихся словами естественного языка (например, чисел), помогает во много раз сокращать объём словаря, снижая затраты времени и памяти на построение моделей.

Ключевые фразы — это словосочетания, характерные для предметной области. Их использование вместо отдельных слов или наряду с ними улуч-

шает интерпретируемость тем. Для их выделения можно использовать тезаурусы [8] или методы автоматического выделения терминов (automatic term extraction, ATE), не требующие привлечения экспертов [40, 67, 108].

Именованные сущности — это названия объектов реального мира, относящихся к определённым категориям: персоны, организации, геолокации, события, даты, и т. д. Для распознавания именованных сущностей (named entities recognition, NER) используются различные методы машинного обучения [83, 60, 89].

5.3 Регуляризация

Задача стохастического матричного разложения является некорректно поставленной, поскольку в общем случае множество её решений бесконечно. Если имеется решение $\Phi\Theta$, то $(\Phi S)(S^{-1}\Theta)$ также является решением для всех невырожденных матриц S, при условии, что матрицы ΦS и $S^{-1}\Theta$ также стохастические. Существует общий подход к решению некорректно поставленных обратных задач, называемый *регуляризацией* [11]. Когда оптимизационная задача недоопределена, к основному критерию добавляют дополнительный критерий — регуляризатор, учитывающий специфику решаемой задачи и знания предметной области. В практических задачах автоматической обработки текстов дополнительных критериев и ограничений на решение может быть много.

Аддитивная регуляризация тематических моделей (ARTM) [3] основана на максимизации линейной комбинации логарифма правдоподобия и нескольких *регуляризаторов* $R_i(\Phi, \Theta),\ i = 1, \ldots, k$:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + \underbrace{\sum_{i=1}^{k} \tau_i R_i(\Phi, \Theta)}_{R(\Phi,\Theta)} \to \max_{\Phi,\Theta}; \tag{5}$$

при прежних ограничениях (4), где τ_i — неотрицательные *коэффициенты регуляризации*. Преобразование вектора критериев в один скалярный кри-

терий — это приём, широко используемый в многокритериальной оптимизации и называемый *скаляризацией*.

Задача (5), (4) относится к классу невыпуклых задач математического программирования. Для неё возможно найти лишь локальный экстремум, качество которого зависит от начального приближения. На практике поиск глобального экстремума не столь важен, как адекватная формализация дополнительных критериев и поиск компромисса между этими критериями.

Необходимые условия максимума. Введём оператор norm, который преобразует произвольный заданный вектор $(x_i)_{i\in I}$ в вектор вероятностей $(p_i)_{i\in I}$ дискретного распределения путём обнуления отрицательных элементов и нормировки:

$$p_i = \underset{i\in I}{\operatorname{norm}}(x_i) = \frac{\max\{0, x_i\}}{\sum\limits_{j\in I}\max\{0, x_j\}}, \text{ для всех } i \in I.$$

Если $x_i \leqslant 0$ для всех $i \in I$, то результатом norm является нулевой вектор.

Теорема 1. *Пусть функция $R(\Phi, \Theta)$ непрерывно дифференцируема. Точка (Φ, Θ) локального экстремума задачи (5), (4) удовлетворяет системе уравнений со вспомогательными переменными p_{tdw} для всех невырожденных тем t и документов d:*

$$p_{tdw} = \underset{t\in T}{\operatorname{norm}}\big(\varphi_{wt}\theta_{td}\big); \tag{6}$$

$$\varphi_{wt} = \underset{w\in W}{\operatorname{norm}}\left(n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}}\right); \qquad n_{wt} = \sum_{d\in D} n_{dw}p_{tdw}; \tag{7}$$

$$\theta_{td} = \underset{t\in T}{\operatorname{norm}}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \qquad n_{td} = \sum_{w\in d} n_{dw}p_{tdw}. \tag{8}$$

Доказательство этой теоремы можно найти в [125]. Оно следует из необходимых условий Каруша–Куна–Таккера для локального экстремума задачи (5), (4).

Система уравнений (6)–(8) имеет элементарную вероятностную интерпретацию. Переменная p_{tdw} выражает тематику слова w в документе d

по формуле Байеса:

$$p(t\,|\,d,w) = \frac{p(w,t\,|\,d)}{p(w\,|\,d)} = \frac{p(w\,|\,t)p(t\,|\,d)}{p(w\,|\,d)} = \frac{\varphi_{wt}\theta_{td}}{\sum_s \varphi_{ws}\theta_{sd}} = p_{tdw}.$$

Следовательно, $n_{dw}p_{tdw}$ есть оценка числа употреблений термина w в документе d по теме t. Тогда n_{dt} оценивает число терминов темы t в документе d, n_{wt} — число употреблений термина w по теме t во всей коллекции. При $R = 0$ формулы (7)–(8) переходят в частотные оценки условных вероятностей $\varphi_{wt} = \frac{n_{wt}}{n_t}$ и $\theta_{td} = \frac{n_{td}}{n_d}$.

Условия вырожденности, упомянутые в теореме, возникают в тех редких на практике случаях, когда регуляризатор R оказывает чрезмерное разреживающее воздействие на параметры модели. Формально они определяются следующим образом:

тема t *вырождена*, если $n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}} \leqslant 0$ для всех терминов $w \in W$;
документ d *вырожден*, если $n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}} \leqslant 0$ для всех тем $t \in T$.

Вырожденные темы и документы исключаются из модели. Сокращение числа тем может быть желательным побочным эффектом регуляризации. Вырожденность документа может означать, что модель не в состоянии его описать, например, если он слишком короткий или не соответствует тематике коллекции.

EM-алгоритм. Для решения системы (6)–(8) удобно применять метод простых итераций. Сначала выбираются начальные приближения параметров $\varphi_{wt}, \theta_{td}$, затем в цикле чередуются два шага до сходимости. Вычисление переменных p_{tdw} по формуле (6) называется *E-шагом* (expectation). Оценивание параметров $\varphi_{wt}, \theta_{td}$ по формулам (7) и (8) называется *M-шагом* (maximization). Этот итерационный процесс является частным случаем EM-алгоритма [37]. Известно, что он сходится в слабом смысле: на каждой итерации правдоподобие увеличивается. Разновидности EM-алгоритма для тематического моделирования рассматриваются в [18, 5].

Онлайновый EM-алгоритм считается наиболее быстрым и хорошо распараллеливается [47, 20]. Основная его идея в том, что на больших коллекциях матрица Φ сходится после обработки относительно небольшой

доли документов. В таких случаях одного прохода по коллекции достаточно для построения модели. Поэтому онлайновый EM-алгоритм хорошо подходит для анализа потоковых данных.

В онлайновом алгоритме вся коллекция разбивается на пакеты документов. Каждый пакет обрабатывается при фиксированной матрице Φ. Для каждого документа d из пакета итерационно повторяются E-шаг, часть M-шага для вычисления вектора θ_d и накапливаются счётчики n_{wt}. Матрица Φ обновляется по накопленным счётчикам по окончании обработки пакета или нескольких пакетов.

В онлайновом алгоритме можно хранить матрицу Φ в оперативной памяти, а матрицу Θ вообще не хранить. Тематическую модель документа можно получать «на лету» и сразу использовать. Детали параллельной реализации онлайнового EM-алгоритма в библиотеке `BigARTM` описаны в [43].

BigARTM. В библиотеке реализованы оба варианта EM-алгоритма, оффлайновый и онлайновый. В обоих вариантах можно добавлять любое число регуляризаторов. Поддерживается набор стандартных регуляризаторов и имеются механизмы создания новых регуляризаторов пользователем. Коэффициенты регуляризации задаются в момент создания модели, но потом могут быть в любой момент изменены, даже в ходе EM-итераций.

Байесовская регуляризация. До сих пор мы предполагали, что данные порождаются вероятностной моделью с параметрами (Φ, Θ), которые не известны и не случайны. В байесовском подходе предполагается, что параметры случайны и подчиняются *априорному* распределению $p(\Phi, \Theta; \gamma)$ с неслучайным *гиперпараметром* γ. В этом случае максимизация совместного правдоподобия данных и модели приводит к принципу *максимума апостериорной вероятности* (maximum a posteriori probability, MAP):

$$p(D, \Phi, \Theta; \gamma) = p(D \,|\, \Phi, \Theta)\, p(\Phi, \Theta; \gamma) = p(\Phi, \Theta; \gamma) \prod_{i=1}^{n} p(d_i, w_i \,|\, \Phi, \Theta) \to \max_{\Phi, \Theta, \gamma}.$$

После логарифмирования получаем модификацию задачи (3), в которой логарифм априорного распределения является регуляризатором:

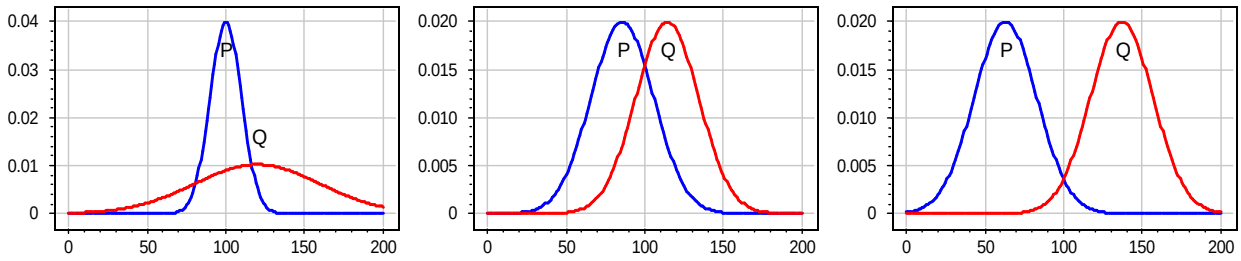
$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + \underbrace{\ln p(\Phi,\Theta;\gamma)}_{R(\Phi,\Theta)} \to \max_{\Phi,\Theta,\gamma}. \tag{9}$$

В байесовском подходе применяется также принцип максимизации неполного правдоподобия, в котором по случайным параметрам (Φ,Θ) производится интегрирование и оптимизируются гиперпараметры γ. Считается, что этот приём снижает размерность задачи и риск переобучения. Действительно, размерность вектора γ, как правило, много меньше размеров матриц Φ,Θ и не зависит от объёма коллекции. Однако для решения прикладных задач всё равно нужны именно эти матрицы. Формулы для них выводятся громоздкими приближёнными методами, но в итоге мало отличаются от MAP-оценок [18].

В байесовском подходе оцениваются не сами параметры Φ,Θ, а их апостериорное распределение $p(\Phi,\Theta\,|\,D;\gamma)$. Для задач тематического моделирования в этом нет особого смысла. На практике полученное распределение используется исключительно для того, чтобы вернуться к точечным оценкам математического ожидания. Другие оценки используются крайне редко, даже точечные оценки медианы или моды.

Техники приближённого байесовского вывода (вариационный вывод [120], сэмплирование Гиббса [116], распространение ожидания) не позволяют легко комбинировать модели и добавлять регуляризаторы, не имеющие вероятностной интерпретации. Для каждой новой модели приходится заново выполнять математические выкладки и программную реализацию. В прикладных проектах сроки, стоимость и риски таких разработок становятся непреодолимым барьером. Поэтому на практике пользуются простой устаревшей моделью LDA, а байесовское тематическое моделирование редко выходит за рамки академических исследований. Тем не менее, в литературе по тематическому моделированию байесовский подход доминирует.

Многокритериальный не-байесовский подход ARTM — это попытка изменить ситуацию. Байесовские тематические модели в большинстве случаев удаётся переформулировать в терминах регуляризации, записав поста-



$\mathrm{KL}(P\|Q) = 0.44$ $\mathrm{KL}(Q\|P) = 2.97$ | $\mathrm{KL}(P\|Q) = 0.44$ $\mathrm{KL}(Q\|P) = 0.44$ | $\mathrm{KL}(P\|Q) = 2.97$ $\mathrm{KL}(Q\|P) = 2.97$

Рис. 5.2. Дивергенция $\mathrm{KL}(P\|Q)$ является мерой вложенности распределения $P = (p_i)_{i=1}^n$ в распределение $Q = (q_i)_{i=1}^n$. Вложенность P в Q приблизительно одинакова на левом и среднем графиках, вложенность Q в P — на левом и правом графиках

новку задачи в виде (9). С этого момента регуляризатор отделяется от модели и может быть использован в других моделях. Это приводит к модульной технологии тематического моделирования, которая реализована и развивается в проекте `BigARTM`.

Дивергенция Кульбака–Лейблера (*KL-дивергенция*, относительная энтропия) далее будет одним из важнейших инструментов конструирования регуляризаторов. Это несимметричная функция расстояния между дискретными распределениями $P = (p_i)_{i=1}^n$ и $Q = (q_i)_{i=1}^n$ с совпадающими носителями, $\{i\colon p_i > 0\} = \{i\colon q_i > 0\}$:

$$\mathrm{KL}(P\|Q) \equiv \mathrm{KL}_i(p_i\|q_i) = \sum_{i=1}^n p_i \ln \frac{p_i}{q_i} = H(P,Q) - H(P),$$

где $H(P) = -\sum_i p_i \ln p_i$ и $H(P,Q) = -\sum_i p_i \ln q_i$ — соответственно *энтропия* распределения P и *кросс-энтропия* пары распределений (P,Q).

Перечислим наиболее важные свойства KL-дивергенции.

1. KL-дивергенция неотрицательна и равна нулю тогда и только тогда, когда распределения совпадают, $p_i \equiv q_i$.

2. Если $\mathrm{KL}(P\|Q) < \mathrm{KL}(Q\|P)$, то распределение P сильнее вложено в Q, чем Q в P, см. рис. 5.2. Таким образом, KL-дивергенция является мерой вложенности двух распределений.

3. Если P — эмпирическое распределение, а $Q(\alpha)$ — параметрическая модель, то минимизация KL-дивергенции эквивалентна минимизации

кросс-энтропии и максимизации правдоподобия:

$$\mathrm{KL}(P\|Q(\alpha)) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i(\alpha)} \to \min_{\alpha} \quad \Leftrightarrow \quad \sum_{i=1}^{n} p_i \ln q_i(\alpha) \to \max_{\alpha}.$$

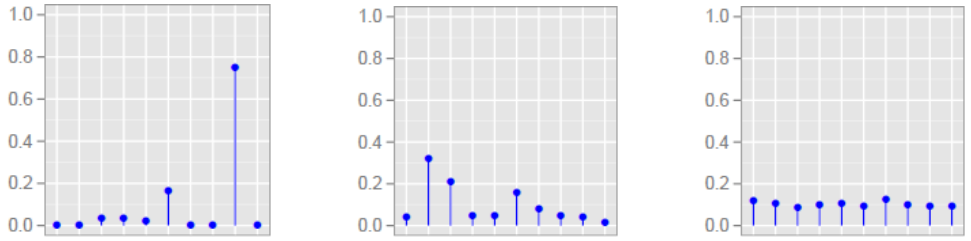
4. Максимизация правдоподобия (3) эквивалентна минимизации взвешенной суммы KL-дивергенций между эмпирическими распределениями $\hat{p}(w\,|\,d) = \frac{n_{dw}}{n_d}$ и модельными $p(w\,|\,d)$, по всем документам d из D:

$$\sum_{d\in D} n_d \,\mathrm{KL}_w\Big(\frac{n_{dw}}{n_d} \,\Big\|\, \sum_{t\in T} \varphi_{wt}\theta_{td}\Big) \;\to\; \min_{\Phi,\Theta},$$

где весом документа d является его длина n_d. Если веса n_d убрать, то все документы будут искусственно приведены к одинаковой длине. Такая модификация может быть полезна при моделировании коллекций, содержащих документы одинаковой важности, но существенно разной длины.

Модель PLSA — это первая вероятностная тематическая модель, предложенная Томасом Хофманном в 1999 году [48]. В ARTM она соответствует частному случаю, когда регуляризатор отсутствует, $R(\Phi,\Theta) = 0$.

Латентное размещение Дирихле. Дэвид Блэй, Эндрю Ын и Майкл Джордан предложили модель LDA (latent Dirichlet allocation) [26] для решения проблемы переобучения в PLSA, которая предсказывала вероятности слов $p(w\,|\,d)$ на новых документах заметно хуже, чем на обучающей коллекции. Позже выяснилось, что на больших коллекциях обе модели почти не переобучаются, а их правдоподобия отличаются незначительно [73, 143, 69]. Различия проявляются только на низкочастотных терминах, которые не важны для образования тем. В робастных вариантах PLSA и LDA такие термины игнорируются, что резко снижает как переобучение, так и различие в правдоподобии моделей [98]. Сам вопрос о переобучении поставлен не вполне корректно. Во-первых, тематические модели строятся не ради предсказания слов в документах, а для выявления латентной кластерной структуры коллекции. Во-вторых, переобучение зависит не столько от самой модели, сколько от того, как мы договоримся измерять её качество. Для измерения обычно используется перплексия, которая сильно



$\mathrm{Dir}(\varphi; 0.1)$ $\mathrm{Dir}(\varphi; 1)$ (равномерное) $\mathrm{Dir}(\varphi; 100)$

Рис. 5.3. Пример неотрицательных нормированных векторов $\varphi_t \in \mathbb{R}^{10}$, порождённых симметричными распределениями Дирихле с параметрами, соответственно, 0.1, 1, 100

штрафует заниженные вероятности низкочастотных терминов. Тем не менее, LDA до сих пор считается моделью №1 в тематическом моделировании, а про PLSA вспоминают всё реже.

Модель LDA основана на предположении, что столбцы θ_d и φ_t являются случайными векторами, которые порождаются распределениями Дирихле с параметрами $\alpha \in \mathbb{R}^{|T|}$ и $\beta \in \mathbb{R}^{|W|}$ соответственно:

$$\mathrm{Dir}(\theta_d; \alpha) = \frac{\Gamma(\alpha_0)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_{td}^{\alpha_t - 1}, \quad \alpha_t > 0, \;\; \alpha_0 = \sum_t \alpha_t, \quad \theta_{td} > 0, \quad \sum_t \theta_{td} = 1;$$

$$\mathrm{Dir}(\varphi_t; \beta) = \frac{\Gamma(\beta_0)}{\prod_w \Gamma(\beta_w)} \prod_w \varphi_{wt}^{\beta_w - 1}, \;\; \beta_w > 0, \;\; \beta_0 = \sum_w \beta_w, \;\; \varphi_{wt} > 0, \;\; \sum_w \varphi_{wt} = 1;$$

где $\Gamma(z)$ — гамма-функция. Параметры распределений Dir связаны с математическим ожиданием порождаемых векторов: $\mathsf{E}\theta_{td} = \frac{\alpha_t}{\alpha_0}$, $\mathsf{E}\varphi_{wt} = \frac{\beta_w}{\beta_0}$.

Распределения Дирихле способны порождать как разреженные, так и плотные векторы дискретных распределений, рис. 5.3. Чем меньше β_w, тем более разрежена компонента φ_{wt} в порождаемых векторах φ_t. Если вектор параметров состоит из равных значений β_w, то распределение Дирихле называется *симметричным*. При $\beta_w \equiv 1$ оно совпадает с равномерным распределением на единичном симплексе.

Вероятностная модель порождения данных является двухуровневой: сначала из распределения Дирихле порождаются вектор-столбцы φ_t. Они задают распределения $p(w\,|\,t) = \varphi_{wt}$, из которых порождаются монотематичные части документов d, описываемые эмпирическими распре-

делениями $\hat{p}(w\,|\,t,d)$. Таким образом, двухуровневая модель порождения текста способна описывать кластерные структуры в текстовых коллекциях. Векторы распределений $p(w\,|\,t)$ интерпретируются как центроиды кластеров, а распределения $\hat{p}(w\,|\,t,d)$ являются точками этих кластеров.

Более убедительных лингвистических обоснований распределение Дирихле не имеет. Его широкое распространение в тематическом моделировании объясняется скорее математическим удобством и популярностью байесовского обучения. Распределение Дирихле является сопряжённым к мультиномиальному распределению, что существенно упрощает байесовский вывод. Благодаря этому свойству оно оказывается «на особом положении» в байесовском тематическом моделировании, и большинство моделей строятся с использованием распределений Дирихле.

Согласно (9), модели LDA соответствует регуляризатор, с точностью до константы равный логарифму априорного распределения Дирихле:

$$\begin{aligned} R(\Phi,\Theta) &= \ln\prod_{t\in T}\mathrm{Dir}(\varphi_t;\beta)\prod_{d\in D}\mathrm{Dir}(\theta_d;\alpha) + \mathrm{const} = \\ &= \sum_{t\in T}\sum_{w\in W}(\beta_w-1)\ln\varphi_{wt} + \sum_{d\in D}\sum_{t\in T}(\alpha_t-1)\ln\theta_{td}. \end{aligned} \tag{10}$$

Применение уравнений (7)–(8) к регуляризатору Дирихле (10) приводит к следующим формулам M-шага:

$$\varphi_{wt} = \underset{w\in W}{\mathrm{norm}}\big(n_{wt}+\beta_w-1\big); \qquad \theta_{td} = \underset{t\in T}{\mathrm{norm}}\big(n_{td}+\alpha_t-1\big).$$

При $\beta_w = 1,\ \alpha_t = 1$ распределение Дирихле совпадает с равномерным распределением на симплексе, формулы M-шага переходят в частотные оценки условных вероятностей, а модель LDA переходит в PLSA [44].

При $\beta_w > 1,\ \alpha_t > 1$ регуляризатор имеет сглаживающий эффект: он делает большие вероятности ещё больше, при этом малые вероятности за счёт нормировки становятся меньше, однако никогда не достигают нуля.

При $0 < \beta_w < 1,\ 0 < \alpha_t < 1$ регуляризатор имеет разреживающий эффект и способен обнулять малые вероятности.

Не-вероятностная интерпретация модели LDA. Регуляризатор (10) можно эквивалентным образом записать через KL-дивергенции:

$$R(\Phi,\Theta) = |W|\sum_{t\in T}\mathrm{KL}_w\big(\tfrac{1}{|W|}\,\big\|\,\varphi_{wt}\big) - \beta_0\sum_{t\in T}\mathrm{KL}_w\big(\tfrac{\beta_w}{\beta_0}\,\big\|\,\varphi_{wt}\big) + \\ + |T|\sum_{d\in D}\mathrm{KL}_t\big(\tfrac{1}{|T|}\,\big\|\,\theta_{td}\big) - \alpha_0\sum_{d\in D}\mathrm{KL}_t\big(\tfrac{\alpha_t}{\alpha_0}\,\big\|\,\theta_{td}\big).$$

Отсюда следует, что модель LDA оказывает сглаживающие и разреживающие воздействия на матрицы Φ, Θ. Все столбцы матрицы Φ должны быть близки к одному и тому же распределению $\frac{\beta_w}{\beta_0}$, причём параметр β_0 становится коэффициентом регуляризации. Аналогично, все столбцы матрицы Θ должны быть близки к распределению $\frac{\alpha_t}{\alpha_0}$, и этим требованием управляет коэффициент регуляризации α_0. Кроме этих сглаживающих воздействий имеются слабые неуправляемые разреживающие воздействия: столбцы обеих матриц должны быть далеки от равномерного распределения. Дальше всего от равномерного распределения находятся вырожденные распределения, в которых единичная вероятность сконцентрирована в единственном элементе. Поэтому разреживание приводит к обнулению малых вероятностей в матрицах Φ, Θ.

5.4 Интерпретируемость тем

Отказ от априорных распределений Дирихле позволяет обобщить модель LDA: снять ограничения на знаки гиперпараметров в (10) и свободнее обращаться со сглаживанием и разреживанием для улучшения интерпретируемости тематических моделей.

Гипотеза разреженности является одним из естественных необходимых условий интерпретируемости. Предполагается, что каждая тема характеризуется небольшим числом терминов, и каждый документ относится к небольшому числу тем. В таком случае значительная часть вероятностей φ_{wt} и θ_{td} должны быть равны нулю.

Многократные попытки разреживания модели LDA приводили к чрезмерно сложным конструкциям [110, 39, 133, 61, 32] из-за внутреннего проти-

воречия между требованиями разреженности и ограничениями строгой положительности параметров в распределении Дирихле. Проблема решается неожиданно просто, если оставить кросс-энтропийный регуляризатор (10) и разрешить гиперпараметрам α_t, β_w принимать любые значения, включая отрицательные. По всей видимости, впервые она была предложена в динамической модели PLSA для обработки видеопотоков [122], где документами являлись короткие видеофрагменты, терминами — признаки на изображениях, темами — появление определённого объекта в течение определённого времени, например, проезд автомобиля. Сильно разреженные распределения потребовались для описания тем с кратким «временем жизни».

Сглаживание и разреживание. По аналогии с (10) введём обобщённый регуляризатор сглаживания и разреживания:

$$R(\Phi, \Theta) = \sum_{t \in T} \sum_{w \in W} \beta_{wt} \ln \varphi_{wt} + \sum_{d \in D} \sum_{t \in T} \alpha_{td} \ln \theta_{td}.$$

Подставив этот регуляризатор в (7)–(8), получим формулы M-шага:

$$\varphi_{wt} = \underset{w \in W}{\operatorname{norm}}\big(n_{wt} + \beta_{wt}\big); \qquad \theta_{td} = \underset{t \in T}{\operatorname{norm}}\big(n_{td} + \alpha_{td}\big).$$

Положительное значение параметра α_{td} или β_{wt} соответствует сглаживанию, отрицательное — разреживанию.

Частичное обучение. В процессе создания, использования или оценивания тематической модели эксперты, пользователи или асессоры могут отмечать в темах релевантные или нерелевантные термины и документы. Размеченные данные позволяют фиксировать интерпретации тем и повышают устойчивость модели. Разметка может затрагивать лишь часть документов и тем, поэтому её использование относится к задачам *частичного обучения* (semi-supervised learning).

Пусть для каждой темы $t \in T$ заданы четыре подмножества:

W_t^+ — «белый список» релевантных терминов;

W_t^- — «чёрный список» нерелевантных терминов;

D_t^+ — «белый список» релевантных документов;

D_t^- — «чёрный список» нерелевантных документов.

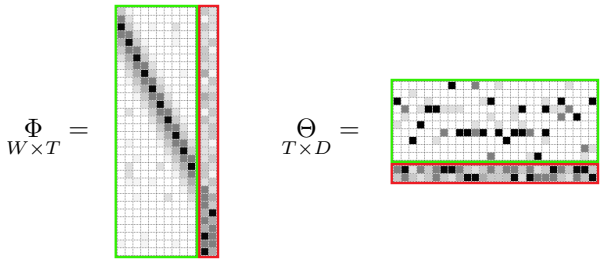


Рис. 5.4. Структура разреженности матриц Φ и Θ с предметными и фоновыми темами

Частичное обучение по релевантности является частным случаем регуляризатора сглаживания и разреживания при

$$\beta_{wt} = \beta_+[w \in W_t^+] - \beta_-[w \in W_t^-],$$
$$\alpha_{td} = \alpha_+[d \in D_t^+] - \alpha_-[d \in D_t^-],$$

где $\beta_\pm$ и $\alpha_\pm$ — коэффициенты регуляризации.

Предметные и фоновые темы. Чтобы модель была интерпретируемой, каждая тема должна иметь *семантическое ядро* — множество слов, характеризующих определённую предметную область и редко употребляемых в других темах. Для этого матрицы Φ и Θ должны иметь структуру разреженности, аналогичную показанной на рис. 5.4. Множество тем разбивается на два подмножества, $T = S \sqcup B$.

Предметные темы $t \in S$ содержат термины предметных областей. Их распределения $p(w|t)$ разрежены и существенно различны (декоррелированы). Распределения $p(d|t)$ также разрежены, так как каждая предметная тема присутствует в относительно небольшой доле документов.

Фоновые темы $t \in B$ образуются из слов общей лексики, которых не должно быть в предметных темах. Их распределения $p(w|t)$ и $p(d|t)$ сглажены, так как эти слова присутствуют в большинстве документов. Тематическую модель с фоновыми темами можно рассматривать как обобщение робастных моделей [30, 98], в которых использовалось только одно фоновое распределение.

Сфокусированный тематический поиск. Частичное обучение тем можно рассматривать как разновидность тематического информационно-

го поиска. В качестве запроса задаётся *семантическое ядро* одной или нескольких тем. Это может быть любой фрагмент текста, «белый список» терминов (seed words) или *z-метки* — темы, приписанные отдельным словам или фрагментам в документах [15]. Тематическая поисковая система должна не только найти и ранжировать релевантные документы, но и разложить поисковую выдачу по темам. В типичных приложениях релевантный контент составляет ничтожно малую долю коллекции. Тем не менее, именно этот контент должен быть тщательно систематизирован. Образно говоря, требуется «классифицировать иголки в стоге сена» [27]. Темы становятся элементом графического интерфейса пользователя, инструментом навигации и понимания текстовой коллекции. Отсюда важность требования интерпретируемости каждой темы.

Частичное обучение использовалось для поиска и кластеризации новостей [52], поиска в социальных медиа информации, связанной с болезнями, симптомами и методами лечения [92, 93], с преступностью и экстремизмом [70, 109], с национальностями и межнациональными отношениями [27, 56, 91].

В модели ATAM (ailment topic aspects model) сглаживающее распределение β_{wt} формировалось по большой коллекции медицинских статей [93].

В моделях SSLDA (semi-supervised LDA) и ISLDA (interval semi-supervised LDA) для поиска этнорелевантных тем использовалось сглаживание по словарю из нескольких сотен этнонимов [27]. В модели SSLDA для каждой этнорелевантной темы задаётся свой словарь этнонимов, связанных с одним определённым этносом. В модели ISLDA множество тем разбивается на интервалы, и для всех тем каждого интервала задаётся общий словарь этнонимов. Преимущество этих моделей в том, что интерпретация каждой темы известна заранее. Недостатки в том, что трудно предугадывать число тем для каждой этничности и строить полиэтничные темы для выявления межэтнических конфликтов. Альтернативный подход заключается в том, чтобы задать число этно-тем и применить к ним общее сглаживание по словарю этнонимов. Тематическая модель сама определит, как разделить их по этничностям [16, 17]. Недостаток этого подхода в том, что интерпретируемость найденных тем приходится проверять вручную.

Декоррелирование. Тематическая модель не должна содержать дублирующихся или похожих тем. Чем различнее темы, тем информативнее модель. Для повышения различности тем будем минимизировать сумму попарных скалярных произведений $\langle \varphi_t, \varphi_s \rangle = \sum_w \varphi_{wt}\varphi_{ws}$ между столбцами матрицы Φ. Получим регуляризатор:

$$R(\Phi) = -\frac{\tau}{2} \sum_{t \in T} \sum_{s \in T \setminus t} \sum_{w \in W} \varphi_{wt}\varphi_{ws}.$$

Формула М-шага, согласно (7), имеет вид

$$\varphi_{wt} = \underset{w \in W}{\operatorname{norm}} \Big(n_{wt} - \tau \varphi_{wt} \sum_{s \in T \setminus t} \varphi_{ws} \Big).$$

Этот регуляризатор контрастирует строки матрицы Φ. В каждой строке, независимо от остальных, вероятности φ_{wt} наиболее значимых тем термина w увеличиваются, вероятности остальных тем уменьшаются и могут обращаться в нуль. Разреживание — это сопутствующий эффект декоррелирования. В [118] был замечен ещё один полезный эффект: слова общей лексики группируются в отдельные темы. Эксперименты с комбинированием регуляризаторов сглаживания, разреживания и декоррелирования в ARTM подтверждают это наблюдение [6, 128, 127].

Декоррелирование впервые было предложено в модели TWC-LDA (topic-weak-correlated LDA) в рамках байесовского подхода [118]. Соответствующее априорное распределение не является сопряжённым к мультиномиальному, поэтому байесовский вывод сталкивается с техническими трудностями. В ARTM расчётные формулы выводятся в одну строку.

Комбинация регуляризаторов сглаживания фоновых тем, разреживания предметных тем в матрице Θ и декоррелирования столбцов матрицы Φ использовалась уже во многих работах для улучшения интерпретируемости тем [6, 127, 128, 129, 12]. Подобрав коэффициенты регуляризации, можно одновременно значительно улучшить разреженность, контрастность, чистоту и когерентность тем при незначительной потере правдоподобия модели [128]. Были выработаны основные рекомендации: декоррелирование и

сглаживание включать сразу, разреживание — после 10–20 итераций, когда образуется тенденция к сходимости параметров модели.

Та же комбинация регуляризаторов была использована для тематического разведочного поиска в [12]. Оказалось, что она существенно улучшает качество поиска, хотя никакие критерии качества поиска непосредственно не оптимизировались.

5.5 Определение числа тем

Регуляризатор отбора тем предложен в [127] для удаления незначимых тем из тематической модели. Он основан на идее кросс-энтропийного разреживания распределения $p(t)$, которое легко выражается через параметры тематической модели:

$$R(\Theta) = \tau n \sum_{t \in T} \frac{1}{|T|} \ln p(t), \qquad p(t) = \sum_d p(d)\theta_{td}.$$

Подставим этот регуляризатор в формулу М-шага (8):

$$\theta_{td} = \underset{t \in T}{\text{norm}}\Big(n_{td} - \tau \frac{n}{|T|} \frac{p(d)}{p(t)} \theta_{td}\Big).$$

Заменим θ_{td} в правой части равенства несмещённой оценкой $\frac{n_{td}}{n_d}$:

$$\theta_{td} = \underset{t \in T}{\text{norm}}\Big(n_{td}\Big(1 - \tau \frac{n}{n_t |T|}\Big)\Big).$$

Этот регуляризатор разреживает целиком строки матрицы Θ. Если значение счётчика n_t в знаменателе достаточно мало, то все элементы t-й строки оказываются равными нулю, и тема t полностью исключается из модели. При использовании данного регуляризатора сначала устанавливается заведомо избыточное число тем $|T|$. В ходе итераций число нулевых строк матрицы Θ постепенно увеличивается.

Отбор тем в ARTM намного проще, чем в байесовских моделях иерархического процесса Дирихле (hierarchical Dirichlet process, HDP) [119] или процесса китайского ресторана (Chinese restaurant process, CRP) [24].

В обоих подходах, ARTM и HDP, имеется управляющий параметр, выбирая который, можно получать модели с числом тем, различающимся на порядки (в ARTM это коэффициент регуляризации τ, в HDP — гиперпараметр γ). Поэтому про оба подхода нельзя сказать, что они определяют оптимальное число тем.

В [129] были проведены эксперименты на полусинтетических данных, представляющих собой смесь двух распределений $p(w|d)$ — реальной коллекции, для которой истинное число тем неизвестно, и синтетической коллекции с заданным числом тем. Оказалось, что HDP и ARTM способны определять истинное число тем на синтетических и полусинтетических данных. При этом ARTM определяет его более точно и устойчиво. Однако чем ближе полусинтетические данные к реальным, тем менее чётко различим момент, когда модель достигает истинного числа тем. На реальных данных он неразличим вовсе, причём для обоих подходов. Отсюда можно сделать вывод, что в реальных текстовых коллекциях никакого «истинного числа тем» просто не существует. Чем больше коллекция, тем более мелкие семантические различия в темах возможно уловить. Эти соображения подтверждаются опытом построения иерархических тематических моделей и рубрикаторов. Темы можно дробить на более мелкие подтемы вплоть до порога статистической значимости. Выбор этого порога также является эвристикой, и от него зависит итоговое число тем.

В ходе экспериментов [129] также выяснилось, что регуляризатор отбора тем имеет полезный сопутствующий эффект: он удаляет из модели дублирующие, расщеплённые и линейно зависимые темы.

По скорости вычислений `BigARTM` с регуляризатором отбора тем оказался в 100 раз быстрее свободно доступной реализации HDP.

5.6 Модальности

Мультимодальная тематическая модель описывает документы, содержащие метаданные наряду с основным текстом. Метаданные помогают более точно определять тематику документов, и, наоборот, тематическая

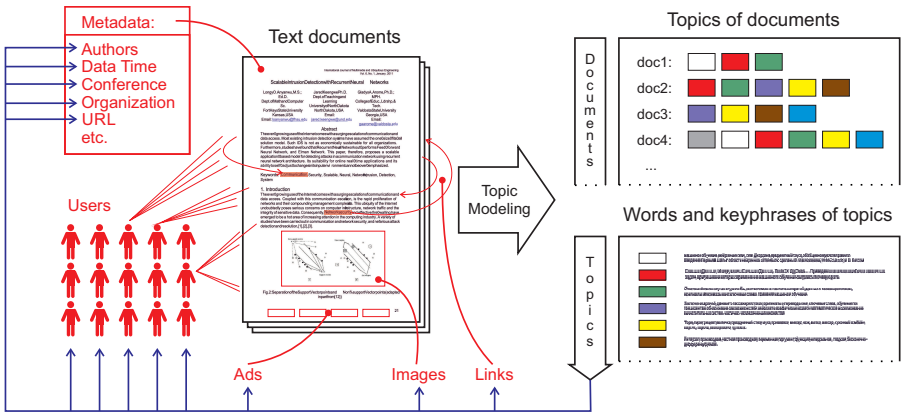


Рис. 5.5. Обычная тематическая модель определяет распределения тем в документах $p(t\,|\,d)$ и терминов в темах $p(w\,|\,t)$. Мультимодальная модель распространяет семантику тем на элементы всех остальных модальностей, в том числе нетекстовые

модель может использоваться для выявления семантики метаданных или предсказания пропущенных метаданных.

Каждый тип метаданных образует отдельную *модальность* со своим словарём. Слова естественного языка, словосочетания [132, 141], теги [58], именованные сущности [85] — это примеры текстовых модальностей. Для анализа коротких текстов с опечатками используют модальность буквенных n-грамм, что позволяет улучшать качество информационного поиска [50]. Примерами нетекстовых модальностей являются (рис. 5.5): авторы [105], моменты времени [121, 152, 122], классы, жанры или категории [106, 155], цитируемые или цитирующие документы [38] или авторы [55], пользователи электронных библиотек, социальных сетей или рекомендательных систем [62, 113, 134, 148, 149], графические элементы изображений [25, 49, 66], рекламные объявления на веб-страницах [96].

Все перечисленные случаи, несмотря на разнообразие интерпретаций, описываются единым формализмом модальностей в ARTM. Каждый документ рассматривается как универсальный контейнер, содержащий токены различных модальностей, включая обычные слова.

Пусть M — множество модальностей. Каждая модальность имеет свой словарь токенов W_m, $m \in M$. Эти множества попарно не пересекаются. Их объединение будем обозначать через W. Модальность токена $w \in W$ будем обозначать через $m(w)$.

Тематическая модель модальности m аналогична модели (2):

$$p(w\,|\,d) = \sum_{t\in T} p(w\,|\,t)\,p(t\,|\,d) = \sum_{t\in T} \varphi_{wt}\theta_{td}, \quad w \in W_m, \quad d \in D. \tag{11}$$

Каждой модальности m из M соответствует стохастическая матрица $\Phi_m = \big(\varphi_{wt}\big)_{W_m\times T}$. Совокупность матриц Φ_m, если их записать в столбец, образует $W\times T$-матрицу Φ. Распределение тем в каждом документе является общим для всех модальностей.

Мультимодальная модель строится путём максимизации взвешенной суммы логарифмов правдоподобия модальностей и регуляризаторов. Веса τ_m позволяют сбалансировать модальности по их важности и с учётом их частотности в документах:

$$\sum_{m\in M}\tau_m \sum_{d\in D}\sum_{w\in W_m} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + R(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}; \tag{12}$$

$$\sum_{w\in W_m} \varphi_{wt} = 1; \quad \varphi_{wt} \geqslant 0; \qquad \sum_{t\in T} \theta_{td} = 1; \quad \theta_{td} \geqslant 0. \tag{13}$$

Теорема 2. *Пусть функция $R(\Phi,\Theta)$ непрерывно дифференцируема. Точка (Φ,Θ) локального экстремума задачи (12)–(13) удовлетворяет системе уравнений со вспомогательными переменными p_{tdw} для всех невырожденных тем t и документов d:*

$$p_{tdw} = \underset{t\in T}{\operatorname{norm}}\big(\varphi_{wt}\theta_{td}\big); \tag{14}$$

$$\varphi_{wt} = \underset{w\in W_m}{\operatorname{norm}}\left(n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}}\right); \qquad n_{wt} = \sum_{d\in D} \tau_{m(w)} n_{dw} p_{tdw}; \tag{15}$$

$$\theta_{td} = \underset{t\in T}{\operatorname{norm}}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \qquad n_{td} = \sum_{m\in M}\sum_{w\in W^m} \tau_m n_{dw} p_{tdw}. \tag{16}$$

Теорема 1 является частным случаем теоремы 2 в случае, когда модальность только одна, $|M| = 1$ и $\tau_m = 1$. Переход от одной модальности к произвольному числу модальностей сводится к двум поправкам: (а) матрица Φ разбивается на блоки Φ_m, которые нормируются по-отдельности; (б) исходные данные n_{dw} домножаются на веса модальностей $\tau_{m(w)}$.

В проекте BigARTM реализована возможность комбинировать любое число модальностей с любыми регуляризаторами [16].

Модальность языков. Мультиязычные текстовые коллекции используются для кросс-язычного информационного поиска, когда по запросу на одном языке требуется найти семантически близкие документы на другом языке. Для связывания языков используются параллельные тексты или двуязычные словари. Первые мультиязычные тематические модели появились почти одновременно [36, 80, 90] и представляли собой мультимодальную модель, в которой модальностями являются языки, и каждая связка параллельных текстов объединяется в один документ. Оказалось, что связывания документов достаточно для синхронизации тем в двух языках и кросс-язычного поиска. Попытки более точного и трудоёмкого выравнивания по предложениям или по словам практически не улучшают качество поиска. Обстоятельный обзор мультиязычных тематических моделей можно найти в [130].

Для использования двуязычного словаря в [7] был предложен регуляризатор сглаживания. Он формализует предположение, что если слово u в языке k является переводом слова w из языка ℓ, то тематики этих слов $p(t\,|\,u)$ и $p(t\,|\,w)$ должны быть близки в смысле KL-дивергенции:

$$R(\Phi) = \sum_{w,u} \sum_{t \in T} n_{ut} \ln \varphi_{wt}.$$

Согласно формуле М-шага, вероятность слова в теме увеличивается, если оно имеет переводы, имеющие высокую вероятность в данной теме:

$$\varphi_{wt} = \underset{w \in W^\ell}{\operatorname{norm}}\Big(n_{wt} + \tau \sum_u n_{ut}\Big).$$

Этот регуляризатор не учитывал, что перевод слова может зависеть от темы, и что среди переводов слова могут находиться переводы его омонимов. Поэтому в той же работе был предложен второй регуляризатор, который вводил в модель новые параметры $\pi_{uwt} = p(u\,|\,w,t)$ — вероятности того, что слово u является переводом слова w в теме t. Предполагается, что тема t, как распределение $\hat{p}(u\,|\,t) = \frac{n_{ut}}{n_t}$ над словами языка k, должна

быть близка в смысле KL-дивергенции к вероятностной модели той же темы $p(u\,|\,t) = \sum_w \pi_{uwt}\varphi_{wt}$, построенной по переводам слов из языка ℓ:

$$R(\Phi,\Pi) = \tau \sum_{u\in W^k}\sum_{t\in T} n_{ut} \ln \sum_{w\in W^\ell} \pi_{uwt}\varphi_{wt} \;\to\; \max_{\Phi,\Pi}.$$

Формула M-шага теперь учитывает вероятности переводов π_{uwt}, и ещё добавляется рекуррентная формула для оценивания этих вероятностей:

$$\varphi_{wt} = \underset{w\in W^\ell}{\operatorname{norm}}\Big(n_{wt} + \tau \sum_u \pi_{wut} n_{ut}\Big);$$
$$\pi_{uwt} = \underset{u\in W^k}{\operatorname{norm}}\big(\pi_{wut} n_{ut}\big).$$

Эксперименты показали, что связывание параллельных текстов сильнее улучшает качество поиска, чем оба способа учёта словарей. Второй способ немного лучше первого. Кроме того, он позволяет выбирать варианты перевода в зависимости от контекста, что может быть полезно для статистического машинного перевода.

Модальности категорий и авторов. Допустим, что распределения тем в документах $p(t\,|\,d)$ порождаются одной из модальностей, например, авторами, рубриками или категориями. Будем считать, что с каждым термином w в каждом документе d связана не только тема $t \in T$, но и категория c из заданного множества категорий C. Расширим вероятностное пространство до множества $D \times W \times T \times C$. Пусть известно подмножество категорий $C_d \subseteq C$, к которым может относиться документ d.

Рассмотрим мультимодальную модель (11), в которой распределение вероятности тем документов $\theta_{td} = p(t\,|\,d)$ описывается смесью распределений тем категорий $\psi_{tc} = p(t\,|\,c)$ и категорий документов $\pi_{cd} = p(c\,|\,d)$:

$$p(w\,|\,d) = \sum_{t\in T} p(w\,|\,t) \sum_{c\in C_d} p(t\,|\,c)p(c\,|\,d) = \sum_{t\in T}\sum_{c\in C_d} \varphi_{wt}\psi_{tc}\pi_{cd}. \qquad (17)$$

Это также задача стохастического матричного разложения, только теперь требуется найти три матрицы: Φ — матрица терминов тем, $\Psi =$

$(\psi_{tc})_{T\times C}$ — *матрица тем категорий*, $\Pi = (\pi_{cd})_{C\times D}$ — *матрица категорий документов*.

Модель основана на двух гипотезах условной независимости:

$p(t\,|\,c,d) = p(t\,|\,c)$ — тематика документа d зависит не от самого документа, а только от того, каким категориям он принадлежит;

$p(w\,|\,t,c,d) = p(w\,|\,t)$ — распределение терминов определяется тематикой документа и не зависит от самого документа и его категорий.

Кроме того, предполагается, что $\pi_{cd} = p(c\,|\,d) = 0$ для всех $c \notin C_d$.

Задача максимизации регуляризованного правдоподобия:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T}\sum_{c\in C_d} \varphi_{wt}\psi_{tc}\pi_{cd} + R(\Phi,\Psi,\Pi) \;\to\; \max_{\Phi,\Psi,\Pi}; \tag{18}$$

$$\sum_{w\in W} \varphi_{wt} = 1,\ \varphi_{wt} \geqslant 0; \quad \sum_{t\in T} \psi_{tc} = 1,\ \psi_{tc} \geqslant 0; \quad \sum_{c\in C_d} \pi_{cd} = 1,\ \pi_{cd} \geqslant 0. \tag{19}$$

Теорема 3. *Пусть функция $R(\Phi,\Psi,\Pi)$ непрерывно дифференцируема. Локальный экстремум (Φ,Ψ,Π) задачи (18), (19) удовлетворяет системе уравнений со вспомогательными переменными $p_{tcdw} = p(t,c\,|\,d,w)$:*

$$p_{tcdw} = \underset{(t,c)\in T\times C_d}{\operatorname{norm}} \varphi_{wt}\psi_{tc}\pi_{cd};$$

$$\varphi_{wt} = \underset{w\in W}{\operatorname{norm}}\left(n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}}\right); \qquad n_{wt} = \sum_{d\in D}\sum_{c\in C_d} n_{dw}p_{tcdw};$$

$$\psi_{tc} = \underset{t\in T}{\operatorname{norm}}\left(n_{tc} + \psi_{tc}\frac{\partial R}{\partial \psi_{tc}}\right); \qquad n_{tc} = \sum_{d\in D}\sum_{w\in d} n_{dw}p_{tcdw};$$

$$\pi_{cd} = \underset{c\in C_d}{\operatorname{norm}}\left(n_{cd} + \pi_{cd}\frac{\partial R}{\partial \pi_{cd}}\right); \qquad n_{cd} = \sum_{w\in d}\sum_{t\in T} n_{dw}p_{tcdw}.$$

Данная модель, основанная на трёхматричном разложении, наиболее известна как *автор-тематическая модель* ATM (author-topic model), в которой порождающей модальностью являются авторы документов [105]. В *тематической модели тегирования документов* TWTM (tag weighted topic model) порождающей модальностью являются теги документа [64]. Аналогичная модель использовалась для обработки видеопотоков в [49]. Документы d соответствовали последовательным 1-секундным видеокли-

пам, термины w — элементарным визуальным событиям, темы t — простым действиям, состоящим из сочетания событий, категории c — более сложным поведениям, состоящим из сочетания действий, причём ставилась задача выделить в каждом клипе одно основное поведение.

Модель (17) можно свести к двухматричному разложению, если отождествить темы с категориями, $C \equiv T$, и взять единичную матрицу Ψ. Данная модель известна в литературе как Flat-LDA [106] и Labeled-LDA [102]. Её выразительные возможности беднее, чем у PLSA и LDA, так как значительная доля элементов матрицы $\Pi \equiv \Theta$ фиксирована и равна нулю.

Трёхматричные разложения пока не реализованы в проекте `BigARTM`.

Темпоральные модели. Время создания документов важно при анализе новостных потоков, научных публикаций, патентных баз, данных социальных сетей. Тематические модели, учитывающие время, называются *темпоральными*. Они позволяют выделять событийные и перманентные темы, детектировать новые темы, прослеживать сюжеты, выделять тренды.

Пусть I — конечное множество интервалов времени, и каждый документ относится к одному или нескольким интервалам, D_i — подмножество документов, относящихся к интервалу i. Будем полагать, что темы как распределения $p(w\,|\,t)$ не меняются во времени. Требуется найти распределение каждой темы во времени $p(i\,|\,t)$.

Тривиальный подход заключается в том, чтобы построить тематическую модель без учёта времени, затем найти распределение тем в каждом интервале $p(t\,|\,i)$ как среднее θ_{td} по всем документам $d \in D_i$ и перенормировать условные вероятности: $p(i\,|\,t) = p(t\,|\,i)\frac{p(i)}{p(t)}$. Недостаток данного подхода в том, что информация о времени никак не используется при обучении модели и не влияет на формирование тем.

В ARTM эта проблема решается введением модальности времени I. Искомое распределение $p(i\,|\,t) = \varphi_{it}$ получается в столбце матрицы Φ. Дополнительные ограничения на поведение тем во времени можно вводить с помощью регуляризации.

В одной из первых темпоральных тематических моделей TOT (topics over time) [140] каждая тема моделировалась параметрическим β-распределением во времени. Это семейство монотонных и унимодальных непре-

рывных функций, с помощью которого можно описывать узкие пики событийных тем и ограниченный набор трендов. Темы, имеющие спорадические всплески, данная модель описывает плохо.

Непараметрические темпоральные модели способны описывать произвольные изменения тем во времени. Рассмотрим два естественных предположения и формализуем их с помощью регуляризации.

Во-первых, предположим, что многие темы являются событийными и имеют относительно небольшое «время жизни», поэтому в каждом интервале времени i присутствуют не все темы. Потребуем разреженности распределений $p(t\,|\,i)$ с помощью кросс-энтропийного регуляризатора:

$$R_1(\Phi \text{ или } \Theta) = -\tau_1 \sum_{i\in I} \sum_{t\in T} \ln p(t\,|\,i).$$

Во-вторых, предположим, что распределения $p(i\,|\,t)$ как функции времени меняются не слишком быстро и введём регуляризатор сглаживания:

$$R_2(\Phi \text{ или } \Theta) = -\tau_2 \sum_{i\in I} \sum_{t\in T} \big|p(i\,|\,t) - p(i{-}1\,|\,t)\big|.$$

Оба регуляризатора можно записать и как функцию от Φ, и как функцию от Θ. В случае регуляризатора $R_2(\Phi)$ формула М-шага имеет вид[1]

$$\varphi_{it} = \underset{i\in I}{\text{norm}}\big(n_{it} + \tau_2\varphi_{it}\,\text{sign}(\varphi_{i-1,t} - \varphi_{it}) + \tau_2\varphi_{it}\,\text{sign}(\varphi_{i+1,t} - \varphi_{it})\big),$$

где функция sign возвращает $+1$ для положительного аргумента и -1 для отрицательного. Регуляризатор сглаживает значения в каждой точке временного ряда $p(i\,|\,t)$ по отношению к соседним точкам слева и справа.

5.7 Зависимости

Классификация. *Тематическая модель классификации* Dependency LDA [106] является байесовским аналогом модели (11) с модальностями

[1]Никита Дойков. Адаптивная регуляризация вероятностных тематических моделей. Бакалаврская диссертация, ВМК МГУ, 2015.
http://www.MachineLearning.ru/wiki/images/9/9f/2015_417_DoykovNV.pdf

терминов W и классов C. Имеется обучающая выборка документов d, для каждого из которых известно подмножество классов $C_d \subset C$. Требуется классифицировать новые документы с неизвестным C_d. Для этого будем использовать *линейную вероятностную модель классификации*, в которой объектами являются документы d, признаки соответствуют темам t и принимают значения $\theta_{td} = p(t\,|\,d)$:

$$\hat{C}_d = \Big\{ c \in C \;\Big|\; p(c\,|\,d) = \sum_{t \in T} \varphi_{ct}\theta_{td} \geqslant \gamma_c \Big\}.$$

Коэффициенты линейной модели $\varphi_{ct} = p(c\,|\,t)$ и пороги γ_c обучаются по выборке документов с известными C_d. Признаковое описание нового документа θ_d вычисляется тематической моделью только по его терминам.

Эксперименты в [106] показали, что тематические модели превосходят обычные методы многоклассовой классификации на больших текстовых коллекциях с большим числом несбалансированных, пересекающихся, взаимозависимых классов. В [125] те же выводы на тех же коллекциях были воспроизведены для мультимодальной ARTM. *Несбалансированность* означает, что классы могут содержать как малое, так и очень большое число документов. В случае *пересекающихся* классов документ может относиться как к одному классу, так и к большому числу классов. *Взаимозависимые* классы имеют общие термины и темы, поэтому при классификации документа могут вступать в конкуренцию.

В некоторых задачах классификации имеется информация о том, что документ d из обучающей выборки не принадлежит подмножеству классов $C'_d \subset C$. Для этого случая запишем правдоподобие вероятностной модели бинарных данных:

$$L(\Phi, \Theta) = \tau \sum_{d \in D} \sum_{c \in C_d} \ln \sum_{t \in T} \varphi_{ct}\theta_{td} + \tau \sum_{d \in D} \sum_{c \in C'_d} \ln\Big(1 - \sum_{t \in T} \varphi_{ct}\theta_{td}\Big) \;\to\; \max.$$

Первое слагаемое равняется log-правдоподобию модальности классов (11), если положить $n_{dc} = [c \in C_d]$. Второе слагаемое можно рассматривать как регуляризатор не-принадлежности документов классам.

Регрессия. Задачи предсказания числовой величины как функции от текста возникают во многих приложениях электронной коммерции: предсказание рейтинга товара, фильма или книги по тексту отзыва; предсказание числа кликов по тексту рекламного объявления; предсказание зарплаты по описанию вакансии; предсказание полезности (числа лайков) отзыва на отель, ресторан, сервис. Для восстановления числовых функций по конечной обучающей выборке пар «объект–ответ» используются регрессионные модели, однако все они принимают на входе векторные описания объектов. Тематическая модель позволяет заменить текст документа d его векторным представлением θ_d. С другой стороны, критерий оптимизации регрессионной модели можно использовать в качестве регуляризатора, чтобы найти темы, наиболее информативные с точки зрения точности предсказаний [74, 115].

Пусть для каждого документа d обучающей выборки D задано целевое значение $y_d \in \mathbb{R}$. Рассмотрим *линейную модель регрессии*, которая предсказывает математическое ожидание целевой величины:

$$\mathsf{E}(y\,|\,d) = \sum_{t\in T} v_t \theta_{td},$$

где $v \in \mathbb{R}^T$ — вектор коэффициентов. Применим метод наименьших квадратов для обучения вектора v по выборке документов:

$$R(\Theta, v) = -\tau \sum_{d\in D} \Big(y_d - \sum_{t\in T} v_t \theta_{td} \Big)^2.$$

Подставляя этот регуляризатор в (8) и приравнивая нулю его производную по вектору v, получим формулы М-шага:

$$\theta_{td} = \operatorname*{norm}_t \Big(n_{td} + \tau v_t \theta_{td} \Big(y_d - \sum_{s\in T} v_s \theta_{sd} \Big) \Big);$$
$$v = (\Theta\Theta^T)^{-1}\Theta y.$$

Заметим, что формула для вектора v является стандартным решением задачи наименьших квадратов при фиксированной матрице Θ. Вектор v

можно обновлять по окончании каждого прохода коллекции, либо после обработки каждого пакета документов в онлайновом EM-алгоритме.

В [115] показано, что качество восстановления регрессии на текстах может существенно зависеть от инициализации тематической модели, там же предложено несколько стратегий инициализации.

Корреляции тем. *Модель коррелированных тем* CTM (correlated topic model) предназначена для выявления связей между темами [21]. Например, статья по геологии более вероятно связана с археологией, чем с генетикой. Знание о том, какие темы чаще совместно встречаются в документах коллекции, позволяет точнее моделировать тематику отдельных документов в мультидисциплинарных коллекциях.

Для описания корреляций удобно использовать многомерное нормальное распределение. Оно не подходит для описания неотрицательных нормированных вектор-столбцов θ_d, но неплохо описывает векторы их логарифмов $\eta_{td} = \ln \theta_{td}$. Поэтому в модель вводится многомерное лог-нормальное распределение (logistic normal) с двумя параметрами: вектором математического ожидания μ и ковариационной матрицей Σ:

$$p(\eta_d \,|\, \mu, \Sigma) = \frac{\exp\bigl(-\frac{1}{2}(\eta_d - \mu)^\intercal \Sigma^{-1}(\eta_d - \mu)\bigr)}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}}.$$

Изначально модель CTM была разработана в рамках байесовского подхода, где возникали технические трудности из-за того, что лог-нормальное распределение не является сопряжённым к мультиномиальному. В рамках ARTM идея CTM формализуется и реализуется намного проще.

Определим регуляризатор как логарифм правдоподобия лог-нормальной модели для выборки векторов документов η_d:

$$R(\Theta, \mu, \Sigma) = \tau \sum_{d \in D} \ln p(\eta_d \,|\, \mu, \Sigma) = -\frac{\tau}{2} \sum_{d \in D} (\ln \theta_d - \mu)^\intercal \Sigma^{-1} (\ln \theta_d - \mu) + \text{const}.$$

Согласно (8), формула M-шага для θ_{td} принимает вид

$$\theta_{td} = \underset{t \in T}{\text{norm}} \Bigl(n_{td} - \tau \sum_{s \in T} \Sigma^{-1}_{ts} \bigl(\ln \theta_{sd} - \mu_s \bigr) \Bigr), \tag{20}$$

где Σ_{ts}^{-1} — элементы обратной ковариационной матрицы. Параметры Σ, μ нормального распределения обновляются после каждого прохода коллекции, либо после каждого пакета документов в онлайновом EM-алгоритме:

$$\mu = \frac{1}{|D|}\sum_{d\in D} \ln \theta_d;$$
$$\Sigma = \frac{1}{|D|}\sum_{d\in D} \big(\ln \theta_d - \mu\big)\big(\ln \theta_d - \mu\big)^{\intercal}.$$

Таким образом, трудоёмкая операция обращения ковариационной матрицы выполняется относительно редко. В [21] использовалась LASSO-регрессия, чтобы получать разреженную ковариационную матрицу.

5.8 Связи между документами

Ссылки и цитирование. Иногда имеется дополнительная информация о связях между документами и предполагается, что связанные документы имеют схожую тематику. Связь может означать, что два документа относятся к одной рубрике, совместно упоминаются или ссылаются друг на друга. Формализуем это предположение с помощью регуляризатора:

$$R(\Theta) = \tau \sum_{d,c} n_{dc} \sum_{t\in T} \theta_{td}\theta_{tc},$$

где n_{dc} — вес связи между документами, например, число ссылок из d на c. В [38] предложена похожая модель LDA-JS, в которой вместо максимизации ковариации минимизируется дивергенция Йенсена-Шеннона между распределениями θ_d и θ_c. Формула М-шага для θ_{td}, согласно (8), имеет вид

$$\theta_{td} = \underset{t\in T}{\operatorname{norm}}\Big(n_{td} + \ \tau\theta_{td}\sum_{c\in D} n_{dc}\theta_{tc}\Big).$$

Это ещё одна разновидность сглаживания. Вероятности θ_{td} в ходе итераций приближаются к вероятностям θ_{tc} документов, связанных с d.

Регуляризатор матрицы Θ становится неэффективным при пакетной обработке больших коллекций, когда документы c, на которые ссылает-

ся данный документ d, находятся в других пакетах. Проблема решается введением модальности документов, на которые есть ссылки из других документов. Этот способ порождает новую проблему: если мощность этой модальности окажется равной числу документов, то матрица Φ может не поместиться в оперативную память. Можно сократить эту модальность, оставив только наиболее влиятельные документы c, число ссылок на которые $n_c = \sum_d n_{dc}$ превышает выбранный порог.

Данная идея пришла из модели влияния научных публикаций LDA-post [38]. В ней используются две модальности: слова W_1 и цитируемые документы $W_2 \subseteq D$. Модель выявляет наиболее влиятельные документы внутри каждой темы. Ненулевые элементы в строке c матрицы Φ_2 показывают, на какие темы повлиял документ $c \in W_2$. Также модель позволяет различать, какие из ссылок существенно повлияли на научную статью, а какие являются второстепенными, чисто формальными или «данью вежливости». Считается, что документ c повлиял на документ d, если d ссылается на c и они имеют значительную долю общей тематики.

Геолокации. Информация о географическом положении используется при анализе данных социальных сетей. Географическая привязка документа d или его автора задаётся либо модальностями *геотегов* (названиями страны, региона, населённого пункта), либо *геолокацией* — парой географических координат $\ell_d = (x_d, y_d)$. В первом случае можно использовать обычную мультимодальную модель, во втором случае нужен дополнительный регуляризатор. ARTM позволяет совмещать в модели оба типа географических данных.

Целью моделирования может быть выделение региональных тем, определение «ареала обитания» каждой темы, поиск похожих тем в других регионах. Например, в качестве одной из иллюстраций в [150] определяются регионы популярности национальной кухни по постам пользователей Flickr. Другая иллюстрация из [75] показывает, что тематическая модель, учитывающая, из какого штата США пришло сообщение, точнее прослеживает путь урагана «Катрина».

Квадратичный регуляризатор матрицы Θ, предложенный в [150], формализует предположение, что документы со схожими геолокациями имеют

схожую тематику:

$$R(\Theta) = -\frac{\tau}{2}\sum_{(c,d)} w_{cd} \sum_{t\in T} \big(\theta_{td} - \theta_{tc}\big)^2,$$

где w_{cd} — вес пары документов (c, d), выражающий близость геолокаций. Например, $w_{cd} = \exp(-\gamma r_{cd}^2)$, где $r_{cd}^2 = (x_c - x_d)^2 + (y_c - y_d)^2$ — квадрат евклидова расстояния.

Этот регуляризатор требует при обработке каждого документа d доступа к векторам θ_c других документов, что затрудняет пакетную обработку больших коллекций. Альтернативный способ сглаживания тематики географически близких сообщений основан на регуляризации матрицы Φ.

Пусть G — модальность геотегов, $\varphi_{gt} = p(g\,|\,t)$. Тематика геотега g выражается по формуле Байеса: $p(t\,|\,g) = \varphi_{gt}\frac{n_t}{n_g}$, где n_g — частота геотега g в исходных данных, $n_t = \sum_g n_{gt}$ — частота темы t в модальности геотегов, вычисляемая EM-алгоритмом.

Квадратичный регуляризатор матрицы Φ по модальности геотегов формализует предположение, что географически близкие геотеги имеют схожую тематику:

$$R(\Phi) = -\frac{\tau}{2}\sum_{g,g'\in G} w_{gg'} \sum_{t\in T} n_t^2 \Big(\frac{\varphi_{gt}}{n_g} - \frac{\varphi_{g't}}{n_{g'}}\Big)^2,$$

где $w_{gg'}$ — вес пары геотегов (g, g'), выражающий их географическую близость. Ниже мы рассмотрим обобщение этого регуляризатора на более широкий класс задач.

Графы и социальные сети. В [75] предложена более общая тематическая модель NetPLSA, учитывающая произвольные графовые (сетевые) структуры на множестве документов. Пусть задан граф $\langle V, E\rangle$ с множеством вершин V и множеством рёбер E. Каждой его вершине $v \in V$ соответствует подмножество документов $D_v \subset D$. Например, в роли D_v может выступать отдельный документ, все статьи одного автора v, все посты из одного географического региона v, и т. д.

Тематика вершины $v \in V$ выражается через параметры модели Θ:

$$p(t\,|\,v) = \sum_{d \in D_v} p(t\,|\,d)\,p(d\,|\,v) = \frac{1}{|D_v|} \sum_{d \in D_v} \theta_{td}.$$

В модели NetPLSA используется квадратичный регуляризатор:

$$R(\Theta) = -\frac{\tau}{2} \sum_{(u,v) \in E} w_{uv} \sum_{t \in T} \big(p(t\,|\,v) - p(t\,|\,u)\big)^2,$$

где веса w_{uv} рёбер графа (u, v) задаются естественным образом, когда в задаче есть соответствующая дополнительная информация. Например, если D_v — все статьи автора v, то в качестве веса ребра w_{uv} естественно взять число статей, написанных авторами u и v в соавторстве. Если подобной информации нет, то вес полагается равным единице.

Этот регуляризатор требует при обработке каждого документа d доступа к векторам θ_c других документов, что затрудняет эффективную пакетную обработку больших коллекций. Альтернативный путь состоит в том, чтобы множество вершин графа V объявить модальностью и перейти к регуляризации матрицы Φ. В каждый документ $d \in D_v$ добавим токен v модальности V. Выразим тематику вершины v через параметры Φ по формуле Байеса: $p(t\,|\,v) = p(v\,|\,t)\frac{p(t)}{p(v)} = \varphi_{vt}\frac{n_t}{|D_v|}$, где $n_t = \sum_v n_{vt}$ — частота темы t в модальности V, вычисляемая EM-алгоритмом.

Регуляризатор сохраняет прежний вид, но становится функцией от Φ:

$$R(\Phi) = -\frac{\tau}{2} \sum_{(u,v) \in E} w_{uv} \sum_{t \in T} n_t^2 \Big(\frac{\varphi_{vt}}{|D_v|} - \frac{\varphi_{ut}}{|D_u|}\Big)^2.$$

Во многих приложениях важны направленности связей, которые квадратичный регуляризатор не учитывает. Например, связь (u, v) может означать ссылку из документа u на документ v. В модели iTopicModel [117] предполагается, что если $(u, v) \in E$, то тематика $p(t\,|\,u)$ шире тематики $p(t\,|\,v)$. Поэтому минимизируется сумма дивергенций $\mathrm{KL}\big(p(t\,|\,v)\,\|\,p(t\,|\,u)\big)$,

причём $p(t\,|\,v)$ можно выразить как через Θ, так и через Φ:

$$R(\Theta \text{ или } \Phi) = \frac{\tau}{2} \sum_{(u,v)\in E} w_{uv} \sum_{t\in T} p(t\,|\,v) \ln p(t\,|\,u).$$

Как показали эксперименты[2], регуляризация матрицы Φ приводит практически к тем же результатам, что и регуляризация Θ для моделей NetPLSA и iTopicModels.

5.9 Иерархии тем

Иерархические тематические модели рекурсивно делят темы на подтемы. Они применяются для построения рубрикаторов, систематизации больших объёмов текстовой информации, информационного поиска и навигации по большим мультидисциплинарным коллекциям.

Задача автоматической рубрикации текстов сложна своей неоднозначностью и субъективностью. Различия во мнениях экспертов относительно рубрикации документов могут достигать 40% [1]. Несмотря на обилие работ по иерархическим тематическим моделям [23, 65, 79, 151, 100, 135, 136, 137, 138], оптимизация размера и структуры иерархии остаётся открытой проблемой; более того, оценивание качества иерархий — также открытая проблема [151].

Стратегии построения тематических иерархий весьма разнообразны: нисходящие (дивизимные) и восходящие (агломеративные), представляющие иерархию деревом или многодольным графом, наращивающие граф по уровням или по вершинам, основанные на кластеризации документов или терминов. Нельзя назвать какую-то из стратегий предпочтительной; у каждой есть свои достоинства и недостатки.

В [33] предложена нисходящая стратегия на основе ARTM. Иерархия представляется многодольным графом с увеличивающимся числом тем на каждом уровне. Модель строится по уровням сверху вниз. Число уровней и число тем каждого уровня задаётся вручную. Каждый уровень пред-

[2]Виктор Булатов. Использование графовой структуры в тематическом моделировании. Магистерская диссертация, ФИВТ МФТИ, 2016.
http://www.MachineLearning.ru/wiki/images/4/4d/Bulatov-2016-ms.pdf

ставляет собой обычную «плоскую» тематическую модель, поэтому время построения модели остаётся линейным по объёму коллекции.

Для моделирования связей между уровнями в модель вводятся параметры $\psi_{st} = p(s\,|\,t)$ — условные вероятности подтем в темах. В случае мультидисциплинарных коллекций подтемам разрешается иметь по несколько родительских тем. ARTM позволяет управлять разреженностью этого распределения с помощью дополнительного кросс-энтропийного регуляризатора. Можно усиливать разреженность распределений $p(t\,|\,s) = \psi_{st}\frac{n_t}{n_s}$ вплоть до вырожденности, тогда каждая подтема будет иметь ровно одну родительскую тему, а вся иерархия будет иметь вид дерева.

Регуляризатор подтем. На верхнем уровне иерархии строится обычная плоская тематическая модель. Пусть модель ℓ-го уровня с множеством тем T уже построена, и требуется построить модель уровня $\ell{+}1$ с множеством дочерних тем S (subtopics) и большим числом тем, $|S| > |T|$. Потребуем, чтобы родительские темы t хорошо приближались вероятностными смесями дочерних тем s:

$$\sum_{t\in T} n_t \operatorname{KL}_w\Big(p(w\,|\,t) \,\Big\|\, \sum_{s\in S} p(w\,|\,s)\,p(s\,|\,t)\Big) =$$
$$= \sum_{t\in T} n_t \operatorname{KL}_w\Big(\tfrac{n_{wt}}{n_t} \,\Big\|\, \sum_{s\in S} \varphi_{ws}\psi_{st}\Big) \;\to\; \min_{\Phi,\Psi},$$

где $\Psi = (\psi_{st})_{S\times T}$ — *матрица связей*, которая становится дополнительной матрицей параметров для тематической модели дочернего уровня.

Это задача матричного разложения $\Phi^\ell = \Phi\Psi$ для матрицы Φ^ℓ родительского уровня. Обычно мы используем низкоранговые разложения, приближая матрицу высокого ранга произведением матриц более низкого ранга. Однако в данном случае всё наоборот: предполагается, что матрицы Φ и Ψ имеют полный ранг $|S|$, заведомо превышающий $\operatorname{rank}\Phi^\ell = |T|$. Среди матричных разложений обязательно имеются точные решения, но они нам не подходят. Матрице Φ выгодно иметь полный ранг, чтобы описывать коллекцию точнее, чем это делает матрица Φ^ℓ. Требование, чтобы она

заодно приближала матрицу Φ^ℓ, вводится через регуляризатор:

$$R(\Phi, \Psi) = \tau \sum_{t \in T} \sum_{w \in W} n_{wt} \ln \sum_{s \in S} \varphi_{ws} \psi_{st}.$$

Задача максимизации $R(\Phi, \Psi)$ с точностью до обозначений совпадает с основной задачей тематического моделирования (3), если считать родительские темы t *псевдодокументами* с частотами слов $\tau n_{wt} = \tau n_t \varphi_{wt}$. Это означает, что вместо добавления слагаемого в формулы М-шага данный регуляризатор можно реализовать ещё проще. Построив родительский уровень, надо добавить в коллекцию ровно $|T|$ псевдодокументов, задав им в качестве частот терминов значения τn_{wt}. Матрица Ψ получится в столбцах матрицы Θ, соответствующих псевдодокументам.

В `BigARTM` этот подход реализован в виде отдельного класса `hARTM`.

5.10 Совстречаемость слов

Гипотеза «мешка слов» является одним из самых критикуемых постулатов тематического моделирования. Поэтому многие исследования направлены на создание более адекватных моделей, учитывающих порядок слов. Из них наиболее важными представляются три направления.

Первое направление связано с выделением *коллокаций* — статистически устойчивых *n-грамм* (последовательностей подряд идущих n слов). Темы, построенные на n-граммах, намного лучше интерпретируются, чем построенные на униграммах (отдельных словах). Проблема в том, что число *n-грамм* катастрофически быстро растёт с ростом объёма коллекции.

Второе направление связано с анализом совместной встречаемости слов. Появление программы `word2vec` [76] стимулировало развитие *векторных представлений слов* (word embedding). Они находят массу применений благодаря свойству *дистрибутивности* — семантически близким словам соответствуют близкие векторы. Тематические модели способны строить векторные представления слов, обладающие свойствами интерпретируемости, разреженности и дистрибутивности.

Третье направление связано с *тематической сегментацией* и гипотезой, что текст на естественном языке состоит из последовательности монотематичных сообщений, и каждое предложение чаще всего относится только к одной теме. Задачи сегментации рассматриваются в разделе 5.11.

Коллокации. Использование словосочетаний заметно улучшает интерпретируемость тем, что демонстрируется практически в каждой публикации по n-граммным тематическим моделям, см. например [53]. Первая биграммная тематическая модель BTM (bigram topic model) [132] представляла собой по сути мультимодальную модель, в которой каждому слову v соответствовала отдельная модальность со словарём $W_v \subseteq W$, составленным из всех слов, встречающихся непосредственно после слова v. Запишем log-правдоподобие этой модели в виде регуляризатора:

$$R(\Phi, \Theta) = \sum_{d \in D} \sum_{v \in d} \sum_{w \in W_v} n_{dvw} \ln \sum_{t \in T} \varphi^v_{wt} \theta_{td},$$

где $\varphi^v_{wt} = p(w|v,t)$ — условная вероятность слов w после слова v в теме t; n_{dvw} — частота биграммы «vw» в документе d. Главный недостаток модели BTM в том, что она учитывает только биграммы. Вторая проблема в том, что число всех биграмм быстро увеличивается с ростом коллекции, и использовать модель BTM на больших коллекциях затруднительно.

Модель TNG (topical n-grams) [141] устраняет эти недостатки. Условное распределение слов описывается вероятностной смесью $p(w|v,t) = \xi_{vwt}\varphi^v_{wt} + (1 - \xi_{vwt})\varphi_{wt}$, где ξ_{vwt} — переменная, равная вероятности того, что пара слов «vw» является биграммой в теме t. При некоторых не особо жёстких предположениях log-правдоподобие этой модели оценивается снизу взвешенной суммой log-правдоподобий модальностей униграмм и биграмм в модели ARTM. Другими словами, мультимодальная ARTM может быть использована для поиска приближённого решения в модели TNG.

В ARTM n-граммная модель естественным образом определяется как мультимодальная, в которой для каждого n выделяется отдельная модальность. Для предварительного сокращения словарей n-грамм подходит метод поиска коллокаций TopMine [40]. Он линейно масштабируется на большие коллекции и позволяет формировать словарь, в котором

каждая n-грамма обладает тремя свойствами: (а) имеет высокую частоту в коллекции; (б) состоит из слов, неслучайно часто образующих n-грамму; (в) не содержится ни в какой $(n{+}1)$-грамме, обладающей свойствами (а) и (б). В последующих работах были предложены методы SegPhrase [67] и AutoPhrase [108], демонстрирующие ещё лучшие результаты.

Битермы. *Короткими текстами* (short text) называют документы, длина которых не достаточна для надёжного определения их тематики. Примерами коротких текстов являются сообщения Твиттера, заголовки новостных сообщений, рекламные объявления, реплики в записях диалогов контакт-центра и т. д. Известны простые подходы к проблеме, но они не всегда применимы: объединять сообщения по какому-либо признаку (автору, времени, региону и т. д.); считать каждое сообщение отдельным документом, разреживая $p(t\,|\,d)$ вплоть до единственной темы; дополнять коллекцию длинными текстами (например, статьями Википедии). Одним из наиболее успешных и универсальных подходов к проблеме коротких текстов считается *тематическая модель битермов* (biterm topic model, BTM) [144].

Битермом называется пара слов, встречающихся рядом — в одном коротком сообщении или в одном предложении или в окне $\pm h$ слов. В отличие от биграммы, между двумя словами битерма могут находиться другие слова. Конкретизация понятия «рядом» зависит от постановки задачи и особенностей коллекции.

Модель BTM описывает вероятность совместного появления слов (u, v). Исходными данными являются частоты n_{uv} битермов (u, v) в коллекции, или матрица вероятностей $P = (p_{uv})_{W\times W}$, где $p_{uv} = \underset{(u,v)\in W^2}{\operatorname{norm}}(n_{uv})$.

Примем гипотезу условной независимости $p(u, v\,|\,t) = p(u\,|\,t)\,p(v\,|\,t)$, то есть допустим, что слова битермов порождаются независимо друг от друга из одной и той же темы. Тогда, по формуле полной вероятности,

$$p(u,v) = \sum_{t\in T} p(u\,|\,t)\,p(v\,|\,t)\,p(t) = \sum_{t\in T} \varphi_{ut}\varphi_{vt}\pi_t,$$

где $\varphi_{wt} = p(w\,|\,t)$ и $\pi_t = p(t)$ — параметры тематической модели. Это трёхматричное разложение $P = \Phi\Pi\Phi^{\mathrm{T}}$, где $\Pi = \operatorname{diag}(\pi_1, \ldots, \pi_T)$ — диагональ-

ная матрица. Эта модель не определяет тематику документов Θ, поэтому менее подвержена влиянию эффектов, вызванных короткими текстами.

ARTM позволяет объединить модель битермов с обычной тематической моделью, чтобы всё-таки получить матрицу Θ. Возьмём log-правдоподобие модели битермов в качестве регуляризатора с коэффициентом τ:

$$R(\Phi, \Pi) = \tau \sum_{u,v} n_{uv} \ln \sum_{t} \varphi_{ut}\varphi_{vt}\pi_t.$$

Подставляя этот регуляризатор в (7)–(8), получаем формулы M-шага:

$$\varphi_{wt} = \underset{w \in W}{\operatorname{norm}}\Big(n_{wt} + \tau \sum_{u \in W} n_{uw} p_{tuw}\Big);$$
$$p_{tuw} = \underset{t \in T}{\operatorname{norm}}\big(n_t \varphi_{wt} \varphi_{ut}\big).$$

Эти формулы интерпретируются как добавление *псевдо-документов*. Каждому слову $u \in W$ ставится в соответствие псевдо-документ d_u, объединяющий все контексты слова u, то есть это мешок слов, встретившихся рядом со словом u по всей коллекции. Число вхождений слова w в псевдо-документ d_u равно τn_{uw}. Вспомогательные переменные $p_{tuw} = p(t \,|\, u, w)$ соответствуют формуле E-шага для псевдо-документа d_u, если доопределить его тематику как $\theta_{tu} = \underset{t}{\operatorname{norm}}(n_t \varphi_{ut})$. Другими словами, в модели битермов столбцы матрицы Θ, соответствующие псевдо-документам, образуются путём перенормировки строк матрицы Φ по формуле Байеса.

Увеличивая коэффициент τ, можно добиться того, чтобы матрица Φ формировалась практически только по битермам. В таком случае модель ARTM переходит в модель битермов, которая строится по коллекции псевдодокументов, без использования исходных документов.

Сеть слов. Идея моделировать не документы, а связи между словами, была положена в основу тематических моделей совстречаемости слов WTM (word topic model) [31] и WNTM (word network topic model) [156]. Любопытно, что более ранняя публикация модели WTM осталась незамеченной (видимо, как не-байесовская), и во второй статье даже нет ссылки на неё. Модели WTM и WNTM сводятся к применению PLSA и LDA соответствен-

но к коллекции псевдо-документов d_u:

$$p(w\,|\,d_u) = \sum_{t\in T} p(w\,|\,t)\,p(t\,|\,d_u) = \sum_{t\in T} \varphi_{wt}\theta_{tu}.$$

Запишем log-правдоподобие модели $p(w\,|\,d_u)$ в виде регуляризатора:

$$R(\Phi,\Theta) = \tau \sum_{u,w\in W} n_{uw} \ln \sum_{t\in T} \varphi_{wt}\theta_{tu},$$

где n_{uw} — совстречаемость слов u, w (кстати, $n_{uw} = n_{wu}$).

Основное отличие этих моделей от модели битермов в том, что здесь в явном виде строится матрица Θ для псевдоколлекции, тогда как в модели битермов $\Theta = \operatorname{diag}(\pi_1,\ldots,\pi_t)\Phi^{\mathsf{T}}$ и количество параметров вдвое меньше. Как показали эксперименты на коллекциях коротких текстов, модель WNTM немного превосходит модель битермов и существенно превосходит обычные тематические модели [156]. На коллекциях длинных документов тематические модели совстречаемости слов не дают значимых преимуществ перед обычными тематическими моделями.

Когерентность. Тема называется *когерентной* (согласованной), если наиболее частые термины данной темы часто встречаются рядом в документах коллекции [87]. Совстречаемость терминов может оцениваться по самой коллекции D [81], или по сторонней коллекции, например, по Википедии [84]. Средняя когерентность тем считается хорошей мерой интерпретируемости тематической модели [88].

Пусть заданы оценки совстречаемости $C_{wv} = \hat{p}(w\,|\,v)$ для пар терминов $(w,v) \in W^2$. Обычно C_{wv} оценивают как долю документов, содержащих термин v, в которых термин w встречается не далее чем через 10 слов от v.

Запишем формулу полной вероятности $p(w\,|\,t) = \sum_v C_{wv}\varphi_{vt}$ и заменим в ней условную вероятность φ_{vt} частотной оценкой: $\hat{p}(w\,|\,t) = \sum_v C_{wv}\frac{n_{vt}}{n_t}$. Введём регуляризатор, требующий, чтобы параметры φ_{wt} тематической модели были согласованы с оценками $\hat{p}(w\,|\,t)$ в смысле кросс-энтропии:

$$R(\Phi) = \tau \sum_{t\in T} n_t \sum_{w\in W} \hat{p}(w\,|\,t) \ln \varphi_{wt}.$$

Формула М-шага, согласно (7), принимает вид

$$\varphi_{wt} = \underset{w \in W}{\operatorname{norm}}\Big(n_{wt} + \tau \sum_{v \in W \setminus w} C_{wv} n_{vt}\Big).$$

Этот сглаживающий регуляризатор увеличивает вероятность термина в теме, если он часто совместно встречается с другими терминами данной темы. Формула была получена в [81] для модели LDA и алгоритма сэмплирования Гиббса, но с более сложным обоснованием через обобщённую урновую схему Пойя и с более сложной эвристической оценкой C_{wv}.

В работе [84] предложен другой регуляризатор когерентности:

$$R(\Phi) = \tau \sum_{t \in T} \ln \sum_{u,v \in W} C_{uv} \varphi_{ut} \varphi_{vt},$$

в котором оценка совстречаемости $C_{uv} = N_{uv}\big[\mathrm{PMI}(u,v) > 0\big]$ определяется через *поточечную взаимную информацию* (pointwise mutual information)

$$\mathrm{PMI}(u,v) = \ln \frac{|D| N_{uv}}{N_u N_v}, \tag{21}$$

где N_{uv} — число документов, в которых термины u, v хотя бы один раз встречаются рядом (не далее, чем через 10 слов), N_u — число документов, в которых термин u встречается хотя бы один раз.

Таким образом, в литературе пока отсутствует единый подход к оптимизации когерентности. Предлагаемые критерии похожи на модели битермов и сети слов. Все они формализуют общую идею, что если слова часто совместно встречаются, то они имеют схожую тематику.

Модели векторных представлений слов ставят в соответствие каждому слову w вектор ν_w фиксированной размерности. Основное требование к этому отображению — чтобы близким по смыслу словам соответствовали близкие векторы. Согласно *дистрибутивной гипотезе* (distributional hypothesis) смысл слова определяется распределением слов, в окружении которых оно встречается [46]. Слова, встречающиеся в схожих контекстах, имеют схожую семантику и, соответственно, должны иметь близкие век-

торы. Для формализации этого принципа в [76, 77] предлагается несколько вероятностных моделей, и все они реализованы в программе word2vec. В частности, модель skip-gram предсказывает появление слова w в контексте слова u, то есть при условии, что слово u находится рядом:

$$p(w\,|\,u) = \underset{w\in W}{\mathrm{SoftMax}}\,\langle \nu_w, \nu_u\rangle = \underset{w\in W}{\mathrm{norm}}\big(\exp\langle \nu_w, \nu_u\rangle\big) = \frac{\exp\langle \nu_w, \nu_u\rangle}{\sum_v \exp\langle \nu_v, \nu_u\rangle},$$

где $\langle \nu_w, \nu_u\rangle = \sum_t \nu_{wt}\nu_{ut}$ — скалярное произведение векторов. В отличие от тематических моделей, нормировка вероятностей производится нелинейным преобразованием SoftMax, а сами векторы ν_w не нормируются.

Для обучения модели решается задача максимизации log-правдоподобия, как правило, градиентными методами:

$$\sum_{u,w\in W} n_{uw} \ln p(w\,|\,u) \;\to\; \max_{\{\nu_w\}}.$$

Постановка задачи очень похожа на тематические модели BTM и WNTM. Модели семейства word2vec и другие модели векторных представлений слов также являются матричными разложениями [63, 95, 68]. Главное отличие заключается в том, что в этих векторных представлениях координаты не интерпретируемы, не нормированы и не разрежены, тогда как в тематических моделях словам соответствуют разреженные дискретные распределения тем $p(t\,|\,w)$. С другой стороны, тематические модели изначально не предназначались для определения семантической близости слов, поэтому делают они это плохо.

В работе А. С. Попова[3] предложен способ построения *тематических векторных представлений слов* по псевдоколлекции документов, аналогичный моделям BTM и WNTM. В задачах семантической близости слов они конкурируют с моделями word2vec и существенно превосходят обычные тематические модели. При этом тематические векторные представления являются интерпретируемыми и разреженными. Используя кросс-энтропий-

[3] Артём Попов. Регуляризация тематических моделей для векторных представлений слов. Бакалаврская диссертация, ВМК МГУ, 2017.
http://www.MachineLearning.ru/wiki/images/4/45/2017PopovBsc.pdf

ные регуляризаторы, разреженность векторов удаётся доводить до 93% без потери качества.

Кроме того, АРТМ позволяет обобщить тематические модели дистрибутивной семантики для мультимодальных коллекций. Используя данные о совстречаемости токенов различных модальностей, возможно строить интерпретируемые тематические векторные представления для всех модальностей. В то же время привлечение дополнительной информации о других модальностях повышает качество решения задачи близости слов.

5.11 Тематическая сегментация

Гипотеза «мешка слов» и предположение о статистической независимости соседних слов приводят к слишком частой хаотичной смене тематики между соседними словами. Если проследить, к каким темам относятся последовательные слова в тексте, то тематическая модель в целом покажется не настолько хорошо интерпретируемой, как ранжированные списки наиболее частотных слов в темах.

Тематические модели сегментации основаны на более реалистичных гипотезах о связном тексте. Каждое предложение относится к одной теме, иногда к небольшому числу тем. Следующее предложение часто продолжает тематику предыдущего. Смена темы чаще происходит между абзацами, ещё чаще между секциями документа. Каждое предложение можно считать «мешком терминов».

Тематическая модель предложений. Допустим, что каждый документ d разбит на множество сегментов S_d. Это могут быть предложения, абзацы или *фразы* — синтаксически корректные части предложений. Обозначим через n_s длину сегмента s, через n_{sw} — число вхождений термина w в сегмент s.

Предположим, что все слова сегмента относятся к одной теме и запишем функцию вероятности сегмента $s \in S_d$ через параметры тематической

модели φ_{wt}, θ_{td}:

$$p(s\,|\,d) = \sum_{t\in T} p(t\,|\,d) \prod_{w\in s} p(w\,|\,t)^{n_{sw}} = \sum_{t\in T} \theta_{td} \prod_{w\in s} \varphi_{wt}^{n_{sw}}.$$

Будем считать каждый документ «мешком сегментов». Тогда функция вероятности выборки будет равна произведению функций вероятности сегментов. Поставим задачу максимизации суммы log-правдоподобия и регуляризатора R:

$$\sum_{d\in D} \sum_{s\in S_d} \ln \sum_{t\in T} \theta_{td} \prod_{w\in s} \varphi_{wt}^{n_{sw}} + R(\Phi,\Theta) \to \max_{\Phi,\Theta}, \tag{22}$$

при обычных ограничениях (4). В частном случае, когда каждый сегмент состоит только из одного слова, данная задача переходит в (5).

Теорема 4. *Пусть функция $R(\Phi,\Theta)$ непрерывно дифференцируема. Точка (Φ,Θ) локального экстремума задачи (22), (4) удовлетворяет системе уравнений со вспомогательными переменными $p_{tds} \equiv p(t\,|\,d,s)$:*

$$p_{tds} = \underset{t\in T}{\operatorname{norm}}\big(\theta_{td} \prod_{w\in s} \varphi_{wt}^{n_{sw}}\big);$$

$$\varphi_{wt} = \underset{w\in W}{\operatorname{norm}}\left(n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}}\right); \qquad n_{wt} = \sum_{d\in D}\sum_{s\in S_d} [w\in s]\, p_{tds}$$

$$\theta_{td} = \underset{t\in T}{\operatorname{norm}}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \qquad n_{td} = \sum_{s\in S_d} p_{tds}.$$

Аналогичная задача ставилась для модели коротких сообщений Twitter-LDA [153], только в роли документов выступали авторы, в роли сегментов — все сообщения данного автора.

Тематическая модель предложений senLDA [19] имеет более важное структурное отличие: вместо матрицы параметров $\theta_{td} = p(t\,|\,d)$ в senLDA используется вектор параметров $\pi_t = p(t)$. Тем самым игнорируется разделение множества всех предложений коллекции по документам, что позволяет уменьшить число параметров модели. Если в senLDA нужно узнать тематику документа, то её нетрудно вычислить, усреднив тематику всех его предложений.

Тематическая модель сегментации. Теперь рассмотрим более сложный случай, когда текст состоит из предложений, и требуется объединить их в более крупные тематические сегменты, границы которых заранее не определены.

Метод TopicTiling [103] основан на пост-обработке распределений $p(t\,|\,d, w_i)$, $i = 1, \ldots, n$, получаемых какой-либо тематической моделью, например, LDA. Определим тематику предложения s как среднюю тематику $p(t\,|\,d, w)$ всех его слов w. Посчитаем косинусную близость тематики для всех пар соседних предложений. Чем глубже локальный минимум близости, тем выше уверенность, что между данной парой предложений проходит граница сегментов. Метод TopicTiling использует набор эвристик для подбора числа предложений слева и справа от локального минимума близости, определения числа сегментов, подбора числа тем и числа итераций, игнорирования стоп-слов, фоновых тем и коротких предложений. Аккуратная настройка параметров этих эвристик позволяет достичь высокого качества сегментации [103]. TopicTiling не является полноценной тематической моделью сегментации текста, поскольку пост-обработка никак не влияет на сами темы. Чтобы найти темы, наиболее выгодные для сегментации, требуется специальный регуляризатор.

Регуляризатор E-шага. Некоторые требования к тематической модели удобнее выражать через распределения $p_{tdw} = p(t\,|\,d, w)$, а не φ_{wt} и θ_{td}. Например, требования сходства тематики терминов внутри предложений или соседних предложений внутри документа. Таким способом можно учитывать порядок слов внутри документов в обход гипотезы «мешка слов».

Рассмотрим регуляризатор $R(\Pi)$ как функцию от трёхмерной матрицы вспомогательных переменных $\Pi = (p_{tdw})_{T\times D\times W}$. Согласно уравнению (6), матрица Π является функцией от Φ и Θ. Поэтому к регуляризатору $R(\Pi(\Phi, \Theta))$ применима теорема 1.

Рассмотрим задачу максимизации регуляризованного log-правдоподобия с двумя регуляризаторами, один из которых зависит от Π:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + R(\Pi(\Phi, \Theta)) + R'(\Phi, \Theta) \;\to\; \max_{\Phi,\Theta}, \qquad (23)$$

при ограничениях неотрицательности и нормировки (4).

Теорема 5. *Пусть функции* $R(\Pi(\Phi,\Theta))$ *и* $R'(\Phi,\Theta)$ *непрерывно дифференцируемы и функция* $R(\Pi)$ *не зависит от переменных* p_{tdw} *в случае* $n_{dw}=0$. *Тогда точка* (Φ,Θ) *локального экстремума задачи* (23), (4) *удовлетворяет системе уравнений со вспомогательными переменными* $p_{tdw}\equiv p(t\,|\,d,w)$ *и* $\tilde{p}_{tdw}$*:*

$$p_{tdw} = \underset{t\in T}{\operatorname{norm}}\big(\varphi_{wt}\theta_{td}\big); \tag{24}$$

$$\tilde{p}_{tdw} = p_{tdw}\left(1+\frac{1}{n_{dw}}\left(\frac{\partial R(\Pi)}{\partial p_{tdw}} - \sum_{z\in T} p_{zdw}\frac{\partial R(\Pi)}{\partial p_{zdw}}\right)\right); \tag{25}$$

$$\varphi_{wt} = \underset{w\in W}{\operatorname{norm}}\left(\sum_{d\in D} n_{dw}\tilde{p}_{tdw} + \varphi_{wt}\frac{\partial R'}{\partial \varphi_{wt}}\right); \tag{26}$$

$$\theta_{td} = \underset{t\in T}{\operatorname{norm}}\left(\sum_{w\in d} n_{dw}\tilde{p}_{tdw} + \theta_{td}\frac{\partial R'}{\partial \theta_{td}}\right). \tag{27}$$

Таким образом, в EM-алгоритме для каждого документа d сначала вычисляются вспомогательные переменные p_{tdw}, затем они преобразуются в новые переменные $\tilde{p}_{tdw}$, которые подставляются в обычные формулы M-шага (7)–(8) вместо p_{tdw}. Такой способ вычислений будем называть *регуляризацией E-шага.*

Переменные $\tilde{p}_{tdw}$ могут принимать отрицательные значения, поэтому в общем случае они не образуют вероятностных распределений. Тем не менее, условие нормировки для них выполнено всегда.

Разреживающий регуляризатор E-шага для сегментации. Применим регуляризацию E-шага для построения тематической модели сегментированного текста. Определим тематику сегмента $s\in S_d$ как среднюю тематику всех его терминов:

$$p_{tds}\equiv p(t\,|\,d,s) = \sum_{w\in s} p(t\,|\,d,w)\,p(w\,|\,s) = \frac{1}{n_s}\sum_{w\in s} n_{sw}p_{tdw}.$$

Чтобы каждый сегмент относился к небольшому числу тем, будем минимизировать кросс-энтропию между $p(t\,|\,d,s)$ и равномерным распределением,

что приведёт нас к разреживающему регуляризатору E-шага:

$$R(\Pi) = -\tau \sum_{d \in D} \sum_{s \in S_d} \sum_{t \in T} \ln \sum_{w \in s} n_{sw} p_{tdw}. \tag{28}$$

Опуская рутинные выкладки, приведём результат подстановки (28) в (25):

$$\tilde{p}_{tdw} = p_{tdw} \left(1 - \frac{\tau}{n_{dw}} \sum_{s \in S_d} \frac{n_{sw}}{n_s} \left(\frac{1}{p_{tds}} - \sum_{z \in T} \frac{p_{zdw}}{p_{zds}}\right)\right).$$

Хотя формула выглядит громоздкой, эффект применения регуляризатора понять не трудно. Если вероятность p_{tds} темы в сегменте окажется меньше некоторого порога, то вероятности p_{tdw} будут уменьшаться для всех терминов w данного сегмента. В итоге тематика каждого сегмента сконцентрируется в небольшом числе тем.

В результате разреживания тематика соседних сегментов может оказаться близкой, и их можно будет объединить в один тематический сегмент. Назовём тему t с максимальным значением $p(t \mid d, s)$ *доминирующей темой* сегмента s документа d. Если тема доминирует в соседних сегментах, то она будет доминирующей и в их объединении. Если объединить последовательные сегменты с одинаковой доминирующей темой в один более крупный сегмент, то данная тема также останется в нём доминирующей. Это простая агломеративная стратегия тематической сегментации. В отличие от TopicTiling, у неё нет эвристических параметров, которые надо настраивать, и она почти не увеличивает время пост-обработки E-шага.

5.12 Критерии качества

Количественное оценивание тематических моделей является нетривиальной проблемой. В отличие от задач классификации или регрессии здесь нет чёткого понятия «ошибки» или «потери». Критерии качества кластеризации типа средних внутрикластерных или межкластерных расстояний плохо подходят для оценивания «мягкой» совместной кластеризации документов и терминов.

Критерии качества тематических моделей принято делить на внутренние (intrinsic) и внешние (extrinsic). *Внутренние критерии* характеризуют качество модели по исходной текстовой коллекции. *Внешние критерии* оценивают полезность модели с точки зрения приложения и конечных пользователей. Иногда для этого приходится собирать дополнительные данные, например, оценки асессоров.

Внешние критерии крайне разнообразны и зависят от решаемой прикладной задачи. Практически в каждой публикации по тематическому моделированию используется какой-либо внешний критерий: качество классификации документов [106], точность и полнота информационного поиска [147, 14, 7, 12], число найденных хорошо интерпретируемых тем [17], качество сегментации текстов [103]. В [34] предлагается методика диагностики моделей, основанная на сопоставлении найденных тем с заранее известными концептами.

Перплексия. Наиболее распространённым внутренним критерием является *перплексия* (perplexity), используемая для оценивания моделей языка в компьютерной лингвистике. Это мера несоответствия или «удивлённости» модели $p(w\,|\,d)$ токенам w, наблюдаемым в документах d коллекции D. Она определяется через log-правдоподобие (3), а в случае мультимодальной модели — через log-правдоподобие (12) отдельно для каждой модальности:

$$\text{perplexity}_m(D;p) = \exp\left(-\frac{1}{n_m}\sum_{d\in D}\sum_{w\in W^m} n_{dw}\ln p(w\,|\,d)\right), \qquad (29)$$

где $n_m = \sum\limits_{d\in D}\sum\limits_{w\in W^m} n_{dw}$ — длина коллекции по m-й модальности.

Чем меньше величина перплексии, тем лучше модель p предсказывает появление токенов w в документах d коллекции D.

Перплексия имеет следующую интерпретацию. Если термины w порождаются из равномерного распределения $p(w) = 1/V$ на словаре мощности V, то перплексия модели $p(w)$ на таком тексте сходится к V с ростом его длины. Чем сильнее распределение $p(w)$ отличается от равномерного, тем меньше перплексия. В случае условных вероятностей $p(w\,|\,d)$ интерпретация немного другая: если каждый документ генерируется из V равнове-

роятных терминов (возможно, различных в разных документах), то перплексия сходится к V.

Недостатком перплексии является неочевидность её численных значений, а также её зависимость от ряда посторонних факторов — длины документов, мощности и разреженности словаря. В частности, с помощью перплексии некорректно сравнивать тематические модели одной и той же коллекции, построенные на разных словарях.

Обозначим через $p_D(w|d)$ модель, построенную по обучающей коллекции документов D. Перплексия обучающей выборки $\mathscr{P}_m(D;p_D)$ является оптимистично смещённой (заниженной) характеристикой качества модели из-за эффекта переобучения. Обобщающую способность тематических моделей принято оценивать *перплексией контрольной выборки* (hold-out perplexity) $\mathscr{P}_m(D';p_D)$. Обычно коллекцию разделяют на обучающую и контрольную случайным образом в пропорции $9:1$ [26].

Недостатком контрольной перплексии является высокая чувствительность к редким и новым словам, которые практически бесполезны для тематических моделей. В ранних экспериментах было показано, что LDA существенно превосходит PLSA по перплексии, откуда был сделан вывод, что LDA меньше переобучается [26]. В [4, 98, 5] были предложены *робастные тематические модели*, описывающие редкие слова специальным «фоновым» распределением. Перплексия робастных вариантов PLSA и LDA оказалась существенно меньшей и практически одинаковой.

Когерентность. Интерпретируемость тем является плохо формализуемым требованием. Содержательно оно означает, что по спискам наиболее частотных слов и документов темы эксперт может понять, о чём эта тема, и дать ей адекватное название [29]. Свойство интерпретируемости важно в информационно-поисковых системах для систематизации и визуализации результатов тематического поиска или категоризации документов.

Большинство существующих методов оценивания интерпретируемости основано на привлечении экспертов-асессоров. В [86] экспертам предлагалось непосредственно оценивать полезность тем по трёхбалльной шкале. В методе интрузий [29] для каждой найденной темы составляется список из 10 наиболее частотных слов, в который внедряется одно случайное слово.

Тема считается интерпретируемой, если подавляющее большинство экспертов правильно указывают лишнее слово. Экспертные подходы необходимы на стадии исследований, но они затрудняют автоматическое построение тематических моделей. В серии работ [86, 87, 87, 81] показано, что среди величин, вычисляемых по коллекции автоматически, лучше всего коррелирует с экспертными оценками интерпретируемости *когерентность* (coherence).

Тема называется *когерентной* (согласованной), если термины, наиболее частые в данной теме, неслучайно часто совместно встречаются рядом в документах коллекции [87, 88]. Численной мерой когерентности темы t является поточечная взаимная информация (21), вычисляемая по k наиболее вероятным словам темы (число k обычно полагается равным 10):

$$\mathrm{PMI}(t) = \sum_{i=1}^{k-1} \sum_{j=i}^{k} \mathrm{PMI}(w_i, w_j),$$

где w_i — i-й термин в порядке убывания φ_{wt}.

Когерентность модели определяется как средняя когерентность тем. Она может оцениваться по сторонней коллекции (например, по Википедии) [84], либо по той же коллекции, по которой строится модель [81].

Разреженность и различность тем. Разреженность модели измеряется долей нулевых элементов в матрицах Φ и Θ. В моделях, разделяющих множество тем T на предметные S и фоновые B, разреженность оценивается только по частям матриц Φ, Θ, соответствующим предметным темам.

В [127] вводятся косвенные меры интерпретируемости тем, не требующие привлечения асессоров. Предполагается, что интерпретируемая тема должна содержать *лексическое ядро* — множество слов, которые с большой вероятностью употребляются в данной теме и редко употребляются в других темах. В таком случае матрицы Φ и Θ должны обладать структурой разреженности, аналогичной рис. 5.4.

Ядро $W_t = \{w \in W \mid p(t|w) > 0.25\}$ темы t определяется как множество терминов с высокой условной вероятностью $p(t|w) = \varphi_{wt}\frac{n_t}{n_w}$. Затем по ядру определяется три показателя интерпретируемости темы t:

$\mathrm{pur}_t = \sum\limits_{w \in W_t} p(w|t)$ — *чистота темы* (чем выше, тем лучше);

$\text{con}_t = \frac{1}{|W_t|} \sum_{w \in W_t} p(t\,|\,w)$ — *контрастность темы* (выше лучше);

$\ker_t = |W_t|$ — *размер ядра* (ориентировочный оптимум $\frac{|W|}{|T|}$).

Показатели размера ядра, чистоты и контрастности для модели в целом определяются как средние по всем предметным темам $t \in S$.

Доля фоновых слов во всей коллекции

$$\text{BackRatio} = \frac{1}{n} \sum_{d \in D} \sum_{w \in d} \sum_{t \in B} n_{dw} p(t\,|\,d, w)$$

принимает значения от 0 до 1. Когда она близка к 0, модель не способна отделять слова общей лексики от специальной терминологии. Значения, близкие к 1, свидетельствуют о вырождении тематической модели.

Такие критерии, как размер ядра или доля фоновых слов, могут использоваться для контроля адекватности модели. Чрезмерная регуляризация может приводить к деградации тем или вырождению модели для слишком большой доли документов.

Образно говоря, регуляризаторы в малых дозах являются лекарствами, но в случае передозировки могут превращаться в яд. Многие критерии, включая перплексию, слабо чувствительны к некоторым типам вырождения, например, когда в предметных темах остаётся слишком мало слов.

На практике к тематическим моделям предъявляются сочетания разнообразных требований. Задачи тематического моделирования по сути являются многокритериальными, поэтому и качество модели должно оцениваться по многим критериям.

В проекте `BigARTM` поддерживается библиотека стандартных метрик качества и механизмы добавления новых пользовательских метрик.

5.13 Разведочный информационный поиск

Важным приложением тематического моделирования является *информационный поиск* (information retrieval) [147, 14]. Современные поисковые системы предназначены, главным образом, для поиска конкретных ответов на короткие текстовые запросы. Другие поисковые потребности воз-

никают у пользователей, которым необходимо разобраться в новой предметной области или пополнить свой багаж знаний. Пользователь может не владеть терминологией, слабо понимать структуру предметной области, не иметь точных формулировок запроса и не подразумевать единственный правильный ответ. В таких случаях нужен поиск не по ключевым словам, а по смыслу. Запросом может быть длинный фрагмент текста, документ или подборка документов. Результатом поиска должна быть удобно систематизированная информация, «дорожная карта» предметной области.

Для этих случаев подходит парадигма *разведочного информационного поиска* (exploratory search) [71, 142]. Его целью является получение ответов на сложные вопросы: «какие темы представлены в тексте запроса», «что читать в первую очередь по этим темам», «что находится на стыке этих тем со смежными областями», «какова тематическая структура данной предметной области», «как она развивалась во времени», «каковы последние достижения», «где находятся основные центры компетентности», «кто является экспертом по данной теме» и т. д. Пользователь обычной поисковой системы вынужден итеративно переформулировать свои короткие запросы, расширяя зону поиска по мере усвоения терминологии предметной области, периодически пересматривая и систематизируя результаты поиска. Это требует затрат времени и высокой квалификации. При отсутствии инструмента для получения «общей картины» остаётся сомнение, что какие-то важные аспекты изучаемой проблемы так и не были найдены. Если образно представить итеративный поиск как блуждание по лабиринту знаний, то разведочный поиск — это средство автоматического построения карты для любой части этого лабиринта.

Тематический разведочный поиск. Обычные (полнотекстовые) поисковые системы основаны на инвертированных индексах, в которых для каждого слова хранится список содержащих его документов [9]. Поисковая система ищет документы, содержащие все слова запроса, поэтому по длинному запросу, скорее всего, ничего не будет найдено.

Система тематического разведочного поиска сначала строит тематическую модель запроса и определяет короткий список тем запроса. Затем для поиска документов схожей тематики применяются те же ме-

ханизмы индексирования и поиска, только в роли слов выступают темы. Поскольку число тем на несколько порядков меньше объёма словаря, тематический поиск требует намного меньше памяти по сравнению с полнотекстовым поиском и может быть реализован на весьма скромной технике. Технологии информационного поиска на основе тематического моделирования в настоящее время находятся в стадии исследований и разработок [116, 21, 94, 28, 13, 134].

В литературе по разведочному поиску тематическое моделирование стали использовать относительно недавно [107, 45, 104, 124], а многие обзоры о нём вообще не упоминают [41, 101, 114, 54, 72, 51]. В недавней статье [124] важными преимуществами тематических моделей называются гибкость, возможности визуализации и навигации. В то же время, в качестве недостатков отмечаются проблемы с интерпретируемостью тем, трудности с модификацией тематической модели при поступлении новых документов и высокая вычислительная сложность. Эти проблемы относятся к устаревшим методам и успешно решены в последние годы: десятки новых моделей разработаны для улучшения интерпретируемости; онлайновые алгоритмы способны обрабатывать большие коллекции и потоки документов за линейное время [78, 20, 125]. С другой стороны, в работах по тематическому моделированию разведочный поиск часто называют одним из важнейших приложений, а оценки качества поиска используют для валидации моделей [147, 14]. Однако эти исследования пока не привели к созданию общедоступных систем разведочного поиска. Всё это говорит о разобщённости научных сообществ, разрабатывающих эти два направления. Тенденция к их сближению наметилась лишь в последние годы.

Такие приложения, как разведочный поиск, стимулируют развитие многокритериального тематического моделирования. Тематическая модель для разведочного поиска в идеале должна быть интерпретируемой, разреженной, мультиграммной, мультимодальной, мультиязычной, иерархической, динамической, сегментирующей, обучаемой по оценкам асессоров или логам пользователей. Также она должна автоматически определять число тем на каждом уровне иерархии и автоматически создавать и именовать новые темы. Наконец, она должна быть онлайновой, параллельной

и распределённой, чтобы эффективно обрабатывать большие коллекции текстов. Таким образом, многие из рассмотренных в данном обзоре моделей должны быть скомбинированы для создания полнофункционального разведочного поиска.

Качество разведочного поиска. Модель ARTM для разведочного поиска была предложена в [12] и улучшена в [145]. Для измерения качества разведочного тематического поиска использовались критерии точности и полноты на основе оценок асессоров. Для оценивания была составлена выборка запросов — заданий разведочного поиска. Каждый запрос представлял собой текст объёмом около одной страницы формата A4, описывающий тематику поиска. Каждое задание сначала выполнялось независимо несколькими асессорами, затем системой тематического поиска, затем релевантность найденных системой документов снова оценивалась асессорами. Данная методика позволяет, единожды сделав разметку результатов поиска, многократно оценивать качество различных тематических моделей и алгоритмов поиска. Эксперименты на коллекциях 175 тысяч статей русскоязычного коллективного блога `habrahabr.ru` и 760 тысяч статей англоязычного блога `techcrunch.com` показали, что тематический поиск находит больше релевантных документов, чем асессоры, сокращая среднее время поиска с получаса до секунды. Комбинирование регуляризаторов декоррелирования, разреживания и сглаживания вместе с модальностями n-грамм, авторов и категорий значимо улучшает качество поиска и позволяет достичь точности выше 80% и полноты выше 90%.

Визуализация. Систематизация результатов тематического поиска невозможна без интерактивного графического представления. В обзоре [2] описываются и сравниваются 16 средств визуализации тематических моделей на основе веб-интерфейсов. Ещё больше идей можно почерпнуть из интерактивного обзора[4], который на момент написания данной статьи насчитывал 380 средств визуализации текстов. Несмотря на такое богатство технических решений, основных идей визуализации тематических моделей не так много: это либо двумерное отображение семантической близости тем

[4] `http://textvis.lnu.se` — интерактивный обзор средств визуализации текстов.

в виде графа или «дорожной карты», либо тематическая иерархия, либо динамика развития тем во времени, либо графовая структура взаимосвязей между темами, документами, авторами или иными модальностями, либо сегментная структура отдельных документов.

Статичные визуализации практически бесполезны при графической визуализации больших данных. Это было понято более 20 лет назад и сформулировано Беном Шнейдерманом в виде *мантры визуального поиска информации*: «сначала крупный план, затем масшабирование и фильтрация, детали по требованию»[5] [112].

Отображение результатов тематического моделирования и разведочного поиска соответствует концепции дальнего чтения (distant reading) социолога литературы Франко Моретти [82]. Он противопоставляет этот способ изучения текстов нашему обычному чтению (close reading). Невозможно прочитать сотни миллионов книг или статей, но вполне возможно применить статистические методы и графическую визуализацию, чтобы понять в общих чертах, о чём вся эта литература, и научиться быстрее отыскивать нужное. «*Дальнее чтение* — это специальная форма представления знаний, в которой меньше элементов, грубее их взаимосвязи, остаются лишь формы, очертания, структуры, модели»[6].

Для библиотеки `BigARTM` в настоящее время развивается собственный инструмент визуализации на основе веб-интерфейса VisARTM[7], поддерживающий важнейшие формы представления тематических моделей. Интересной возможностью VisARTM является построение *спектра тем* — оптимальное ранжирование списка тем, при котором семантически близкие темы оказываются в списке рядом. Это помогает пользователям быстрее находить темы и группировать их по смыслу.

[5] Visual Information Seeking Mantra: «Overview first, zoom and filter, details on demand» [112].

[6] «*Distant reading* is not an obstacle but a specific form of knowledge: fewer elements, hence a sharper sense of their overall interconnection. Shapes, relations, structures. Forms. Models» [82].

[7] Дмитрий Федоряка. Технология интерактивной визуализации тематических моделей. Бакалаврская диссертация, ФУПМ МФТИ, 2017.
`http://www.MachineLearning.ru/wiki/images/d/d8/Fedoriaka17bsc.pdf`

5.14 Заключение

Данный обзор написан по материалам спецкурса «Вероятностное тематическое моделирование»[8], который автор читает на факультете ВМК Московского Государственного Университета им. М. В. Ломоносова. Обновляемая электронная версия доступна на сайте MachineLearning.ru[9],

Что не вошло в этот обзор, но может оказаться в ближайших обновлениях: доказательства пяти теорем; стратегии подбора коэффициентов регуляризации; методы суммаризации и автоматического именования тем; примеры применения тематических моделей для автоматического выделения терминов, обнаружения новых тем и отслеживания сюжетов, анализа тональности и выявления мнений, анализа записей разговоров контакт-центра, анализа банковских транзакционных данных, агрегации и категоризации научного контента.

Благодарности

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проекты 17-07-01536, 16-37-00498.

5.15 Список литературы

[1] *Агеев М. С., Добров Б. В., Лукашевич Н. В.* Автоматическая рубрикация текстов: методы и проблемы // *Учёные записки Казанского государственного университета. Серия Физико-математические науки.* — 2008. — Т. 150, № 4. — С. 25–40.

[2] *Айсина Р. М.* Обзор средств визуализации тематических моделей коллекций текстовых документов // *Машинное обучение и анализ данных (http://jmlda.org).* — 2015. — Т. 1, № 11. — С. 1584–1618.

[3] *Воронцов К. В.* Аддитивная регуляризация тематических моделей коллекций текстовых документов // *Доклады РАН.* — 2014. — Т. 456, № 3. — С. 268–271.

[4] *Воронцов К. В., Потапенко А. А.* Регуляризация, робастность и разреженность вероятностных тематических моделей // *Компьютерные исследования и моделирование.* — 2012. — Т. 4, № 4. — С. 693–706.

[8] `http://www.MachineLearning.ru/wiki?title=BTM`.

[9] `http://www.MachineLearning.ru/wiki/images/d/d5/Voron17survey-artm.pdf`.

[5] *Воронцов К. В., Потапенко А. А.* Модификации EM-алгоритма для вероятностного тематического моделирования // *Машинное обучение и анализ данных.* — 2013. — Т. 1, № 6. — С. 657–686.

[6] *Воронцов К. В., Потапенко А. А.* Регуляризация вероятностных тематических моделей для повышения интерпретируемости и определения числа тем // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2014 г.). — Вып. 13 (20). — М: Изд-во РГГУ, 2014. — С. 676–687.

[7] *Дударенко М. А.* Регуляризация многоязычных тематических моделей // *Вычислительные методы и программирование.* — 2015. — Т. 16. — С. 26–38.

[8] *Лукашевич Н. В.* Тезаурусы в задачах информационного поиска. — Издательство МГУ имени М. В. Ломоносова, 2011.

[9] *Маннинг К. Д., Рагхаван П., Шютце Х.* Введение в информационный поиск. — Вильямс, 2011.

[10] *Павлов А. С., Добров Б. В.* Метод обнаружения массово порожденных неестественных текстов на основе анализа тематической структуры // *Вычислительные методы и программирование: новые вычислительные технологии.* — 2011. — Т. 12. — С. 58–72.

[11] *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. — М.: Наука, 1986.

[12] *Янина А. О., Воронцов К. В.* Мультимодальные тематические модели для разведочного поиска в коллективном блоге // *Машинное обучение и анализ данных.* — 2016. — Т. 2, № 2. — С. 173–186.

[13] *Airoldi E. M., Erosheva E. A., Fienberg S. E., Joutard C., Love T., Shringarpure S.* Reconceptualizing the classification of PNAS articles // *Proceedings of The National Academy of Sciences.* — 2010. — Vol. 107. — Pp. 20899–20904.

[14] *Andrzejewski D., Buttler D.* Latent topic feedback for information retrieval // Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '11. — 2011. — Pp. 600–608.

[15] *Andrzejewski D., Zhu X.* Latent Dirichlet allocation with topic-in-set knowledge // Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing. — SemiSupLearn '09. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. — Pp. 43–48.

[16] *Apishev M., Koltcov S., Koltsova O., Nikolenko S., Vorontsov K.* Additive regularization for topic modeling in sociological studies of user-generated text

content // MICAI 2016, 15th Mexican International Conference on Artificial Intelligence. — Vol. 10061. — Springer, Lecture Notes in Artificial Intelligence, 2016. — P. 166–181.

[17] *Apishev M., Koltcov S., Koltsova O., Nikolenko S., Vorontsov K.* Mining ethnic content online with additively regularized topic models // *Computacion y Sistemas.* — 2016. — Vol. 20, no. 3. — P. 387–403.

[18] *Asuncion A., Welling M., Smyth P., Teh Y. W.* On smoothing and inference for topic models // Proceedings of the International Conference on Uncertainty in Artificial Intelligence. — 2009. — Pp. 27–34.

[19] *Balikas G., Amini M., Clausel M.* On a topic model for sentences // Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. — SIGIR '16. — New York, NY, USA: ACM, 2016. — Pp. 921–924.

[20] *Bassiou N., Kotropoulos C.* Online PLSA: Batch updating techniques including out-of-vocabulary words // *Neural Networks and Learning Systems, IEEE Transactions on.* — Nov 2014. — Vol. 25, no. 11. — Pp. 1953–1966.

[21] *Blei D., Lafferty J.* A correlated topic model of Science // *Annals of Applied Statistics.* — 2007. — Vol. 1. — Pp. 17–35.

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

[23] *Blei D. M., Griffiths T., Jordan M., Tenenbaum J.* Hierarchical topic models and the nested chinese restaurant process // NIPS. — 2003.

[24] *Blei D. M., Griffiths T. L., Jordan M. I.* The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies // *J. ACM.* — 2010. — Vol. 57, no. 2. — Pp. 7:1–7:30.

[25] *Blei D. M., Jordan M. I.* Modeling annotated data // Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. — New York, NY, USA: ACM, 2003. — Pp. 127–134.

[26] *Blei D. M., Ng A. Y., Jordan M. I.* Latent Dirichlet allocation // *Journal of Machine Learning Research.* — 2003. — Vol. 3. — Pp. 993–1022.

[27] *Bodrunova S., Koltsov S., Koltsova O., Nikolenko S. I., Shimorina A.* Interval semi-supervised LDA: Classifying needles in a haystack // MICAI (1) / Ed. by F. C. Espinoza, A. F. Gelbukh, M. Gonzalez-Mendoza. — Vol. 8265 of *Lecture Notes in Computer Science.* — Springer, 2013. — Pp. 265–274.

[28] *Bolelli L., Ertekin S., Giles C. L.* Topic and trend detection in text collections using latent Dirichlet allocation // ECIR. — Vol. 5478 of *Lecture Notes in Computer Science*. — Springer, 2009. — Pp. 776–780.

[29] *Chang J., Gerrish S., Wang C., Boyd-Graber J. L., Blei D. M.* Reading tea leaves: How humans interpret topic models // Neural Information Processing Systems (NIPS). — 2009. — Pp. 288–296.

[30] *Chemudugunta C., Smyth P., Steyvers M.* Modeling general and specific aspects of documents with a probabilistic topic model // Advances in Neural Information Processing Systems. — Vol. 19. — MIT Press, 2007. — Pp. 241–248.

[31] *Chen B.* Word topic models for spoken document retrieval and transcription. — 2009. — Vol. 8, no. 1. — Pp. 2:1–2:27.

[32] *Chien J.-T., Chang Y.-L.* Bayesian sparse topic model // *Journal of Signal Processessing Systems*. — 2013. — Vol. 74. — Pp. 375–389.

[33] *Chirkova N. A., Vorontsov K. V.* Additive regularization for hierarchical multimodal topic modeling // *Journal Machine Learning and Data Analysis*. — 2016. — Vol. 2, no. 2. — Pp. 187–200.

[34] *Chuang J., Gupta S., Manning C., Heer J.* Topic model diagnostics: Assessing domain relevance via topical alignment // Proceedings of the 30th International Conference on Machine Learning (ICML-13) / Ed. by S. Dasgupta, D. Mcallester. — Vol. 28. — JMLR Workshop and Conference Proceedings, 2013. — Pp. 612–620.

[35] *Daud A., Li J., Zhou L., Muhammad F.* Knowledge discovery through directed probabilistic topic models: a survey // *Frontiers of Computer Science in China*. — 2010. — Vol. 4, no. 2. — Pp. 280–301.

[36] *De Smet W., Moens M.-F.* Cross-language linking of news stories on the web using interlingual topic modelling // Proceedings of the 2Nd ACM Workshop on Social Web Search and Mining. — SWSM '09. — New York, NY, USA: ACM, 2009. — Pp. 57–64.

[37] *Dempster A. P., Laird N. M., Rubin D. B.* Maximum likelihood from incomplete data via the EM algorithm // *J. of the Royal Statistical Society, Series B*. — 1977. — no. 34. — Pp. 1–38.

[38] *Dietz L., Bickel S., Scheffer T.* Unsupervised prediction of citation influences // Proceedings of the 24th international conference on Machine learning. — ICML '07. — New York, NY, USA: ACM, 2007. — Pp. 233–240.

[39] *Eisenstein J., Ahmed A., Xing E. P.* Sparse additive generative models of text // ICML'11. — 2011. — Pp. 1041–1048.

[40] *El-Kishky A., Song Y., Wang C., Voss C. R., Han J.* Scalable topical phrase mining from text corpora // *Proc. VLDB Endowment.* — 2014. — Vol. 8, no. 3. — Pp. 305–316.

[41] *Feldman S. E.* The answer machine // Synthesis Lectures on Information Concepts, Retrieval, and Services. — Morgan & Claypool Publishers, 2012. — Vol. 4. — Pp. 1–137.

[42] *Feng Y., Lapata M.* Topic models for image annotation and text illustration // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. — Association for Computational Linguistics, 2010. — Pp. 831–839.

[43] *Frei O., Apishev M.* Parallel non-blocking deterministic algorithm for online topic modeling // AIST'2016, Analysis of Images, Social networks and Texts. — Vol. 661. — Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS), 2016. — P. 132–144.

[44] *Girolami M., Kabán A.* On an equivalence between PLSI and LDA // SIGIR'03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. — 2003. — Pp. 433–434.

[45] *Grant C. E., George C. P., Kanjilal V., Nirkhiwale S., Wilson J. N., Wang D. Z.* A topic-based search, visualization, and exploration system // FLAIRS Conference. — AAAI Press, 2015. — Pp. 43–48.

[46] *Harris Z.* Distributional structure // *Word.* — 1954. — Vol. 10, no. 23. — Pp. 146–162.

[47] *Hoffman M. D., Blei D. M., Bach F. R.* Online learning for latent Dirichlet allocation // NIPS. — Curran Associates, Inc., 2010. — Pp. 856–864.

[48] *Hofmann T.* Probabilistic latent semantic indexing // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. — New York, NY, USA: ACM, 1999. — Pp. 50–57.

[49] *Hospedales T., Gong S., Xiang T.* Video behaviour mining using a dynamic topic model // *International Journal of Computer Vision.* — 2012. — Vol. 98, no. 3. — Pp. 303–323.

[50] *Huang P.-S., He X., Gao J., Deng L., Acero A., Heck L.* Learning deep structured semantic models for web search using clickthrough data // Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management. — CIKM '13. — New York, NY, USA: ACM, 2013. — Pp. 2333–2338.

[51] *Jacksi K., Dimililer N., Zeebaree S. R. M.* A survey of exploratory search systems based on LOD resources // Proceedings of the 5th International Conference on Computing and Informatics, ICOCI 2015. — School of Computing, Universiti Utara Malaysia, 2015. — Pp. 501–509.

[52] *Jagarlamudi J., Daumé III H., Udupa R.* Incorporating lexical priors into topic models // Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. — EACL'12. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. — Pp. 204–213.

[53] *Jameel S., Lam W.* An N-gram topic model for time-stamped documents // 35th European Conference on Information Retrieval, ECIR-2013, Moscow, Russia, 24-27 March 2013. — Lecture Notes in Computer Science (LNCS), Springer Verlag-Germany, 2013. — Pp. 292–304.

[54] *Jiang T.* Exploratory Search: A Critical Analysis of the Theoretical Foundations, System Features, and Research Trends // Library and Information Sciences: Trends and Research / Ed. by C. Chen, R. Larsen. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. — Pp. 79–103.

[55] *Kataria S., Mitra P., Caragea C., Giles C. L.* Context sensitive topic models for author influence in document networks // Proceedings of the Twenty-Second international joint conference on Artificial Intelligence — Volume 3. — IJCAI'11. — AAAI Press, 2011. — Pp. 2274–2280.

[56] *Koltcov S., Koltsova O., Nikolenko S.* Latent Dirichlet allocation: Stability and applications to studies of user-generated content // Proceedings of the 2014 ACM Conference on Web Science. — WebSci'14. — New York, NY, USA: ACM, 2014. — Pp. 161–165.

[57] *Konietzny S., Dietz L., McHardy A.* Inferring functional modules of protein families with probabilistic topic models // *BMC Bioinformatics*. — 2011. — Vol. 12, no. 1. — P. 141.

[58] *Krestel R., Fankhauser P., Nejdl W.* Latent Dirichlet allocation for tag recommendation // Proceedings of the third ACM conference on Recommender systems. — ACM, 2009. — Pp. 61–68.

[59] *La Rosa M., Fiannaca A., Rizzo R., Urso A.* Probabilistic topic modeling for the analysis and classification of genomic sequences // *BMC Bioinformatics*. — 2015. — Vol. 16, no. Suppl 6. — P. S2.

[60] *Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C.* Neural architectures for named entity recognition // HLT-NAACL / Ed. by K. Knight, A. Nenkova, O. Rambow. — The Association for Computational Linguistics, 2016. — Pp. 260–270.

[61] *Larsson M. O., Ugander J.* A concave regularization technique for sparse mixture models // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger. — 2011. — Pp. 1890–1898.

[62] *Lee S. S., Chung T., McLeod D.* Dynamic item recommendation by topic modeling for social networks // Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. — IEEE, 2011. — Pp. 884–889.

[63] *Levy O., Goldberg Y.* Neural Word Embedding as Implicit Matrix Factorization // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger. — Curran Associates, Inc., 2014. — Pp. 2177–2185.

[64] *Li S., Li J., Pan R.* Tag-weighted topic model for mining semi-structured documents // IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. — AAAI Press, 2013. — Pp. 2855–2861.

[65] *Li W., McCallum A.* Pachinko allocation: Dag-structured mixture models of topic correlations // ICML. — 2006.

[66] *Li X.-X., Sun C.-B., Lu P., Wang X.-J., Zhong Y.-X.* Simultaneous image classification and annotation based on probabilistic model // *The Journal of China Universities of Posts and Telecommunications.* — 2012. — Vol. 19, no. 2. — Pp. 107–115.

[67] *Liu J., Shang J., Wang C., Ren X., Han J.* Mining quality phrases from massive text corpora // Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. — SIGMOD '15. — New York, NY, USA: ACM, 2015. — Pp. 1729–1744.

[68] *Liu Y., Liu Z., Chua T.-S., Sun M.* Topical word embeddings // Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. — AAAI'15. — AAAI Press, 2015. — Pp. 2418–2424.

[69] *Lu Y., Mei Q., Zhai C.* Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA // *Information Retrieval.* — 2011. — Vol. 14, no. 2. — Pp. 178–203.

[70] *M. A. Basher A. R., Fung B. C. M.* Analyzing topics and authors in chat logs for crime investigation // *Knowledge and Information Systems.* — 2014. — Vol. 39, no. 2. — Pp. 351–381.

[71] *Marchionini G.* Exploratory search: From finding to understanding // *Commun. ACM.* — 2006. — Vol. 49, no. 4. — Pp. 41–46.

[72] *Marie N., Gandon F.* Survey of linked data based exploration systems // Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data (IESD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014. — 2014.

[73] *Masada T., Kiyasu S., Miyahara S.* Comparing LDA with pLSI as a dimensionality reduction method in document clustering // Proceedings of the 3rd International Conference on Large-scale knowledge resources: construction and application. — LKR'08. — Springer-Verlag, 2008. — Pp. 13–26.

[74] *McAuliffe J. D., Blei D. M.* Supervised topic models // Advances in Neural Information Processing Systems 20 / Ed. by J. C. Platt, D. Koller, Y. Singer, S. T. Roweis. — Curran Associates, Inc., 2008. — Pp. 121–128.

[75] *Mei Q., Cai D., Zhang D., Zhai C.* Topic modeling with network regularization // Proceedings of the 17th International Conference on World Wide Web. — WWW'08. — New York, NY, USA: ACM, 2008. — Pp. 101–110.

[76] *Mikolov T., Chen K., Corrado G., Dean J.* Efficient estimation of word representations in vector space // *CoRR*. — 2013. — Vol. abs/1301.3781.

[77] *Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.* Distributed representations of words and phrases and their compositionality // *CoRR*. — 2013. — Vol. abs/1310.4546.

[78] *Mimno D., Hoffman M., Blei D.* Sparse stochastic inference for latent Dirichlet allocation // Proceedings of the 29th International Conference on Machine Learning (ICML-12) / Ed. by J. Langford, J. Pineau. — New York, NY, USA: Omnipress, July 2012. — Pp. 1599–1606.

[79] *Mimno D., Li W., McCallum A.* Mixtures of hierarchical topics with pachinko allocation // ICML. — 2007.

[80] *Mimno D., Wallach H. M., Naradowsky J., Smith D. A., McCallum A.* Polylingual topic models // Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2. — EMNLP '09. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. — Pp. 880–889.

[81] *Mimno D., Wallach H. M., Talley E., Leenders M., McCallum A.* Optimizing semantic coherence in topic models // Proceedings of the Conference on Empirical Methods in Natural Language Processing. — EMNLP '11. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. — Pp. 262–272.

[82] *Moretti F.* Graphs, maps, trees : abstract models for literary history. — London; New York: Verso, 2007.

[83] *Nadeau D., Sekine S.* A survey of named entity recognition and classification // *Linguisticae Investigationes*. — 2007. — Vol. 30, no. 1. — Pp. 3–26.

[84] *Newman D., Bonilla E. V., Buntine W. L.* Improving topic coherence with regularized topic models // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger. — 2011. — Pp. 496–504.

[85] *Newman D., Chemudugunta C., Smyth P.* Statistical entity-topic models // Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '06. — New York, NY, USA: ACM, 2006. — Pp. 680–686.

[86] *Newman D., Karimi S., Cavedon L.* External evaluation of topic models // Australasian Document Computing Symposium. — December 2009. — Pp. 11–18.

[87] *Newman D., Lau J. H., Grieser K., Baldwin T.* Automatic evaluation of topic coherence // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. — HLT '10. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. — Pp. 100–108.

[88] *Newman D., Noh Y., Talley E., Karimi S., Baldwin T.* Evaluating topic models for digital libraries // Proceedings of the 10th annual Joint Conference on Digital libraries. — JCDL '10. — New York, NY, USA: ACM, 2010. — Pp. 215–224.

[89] *Ni J., Dinu G., Florian R.* Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection // The 55th Annual Meeting of the Association for Computational Linguistics (ACL). — 2017.

[90] *Ni X., Sun J.-T., Hu J., Chen Z.* Mining multilingual topics from wikipedia // Proceedings of the 18th International Conference on World Wide Web. — WWW '09. — New York, NY, USA: ACM, 2009. — Pp. 1155–1156.

[91] *Nikolenko S. I., Koltcov S., Koltsova O.* Topic modelling for qualitative studies // *Journal of Information Science.* — 2017. — Vol. 43, no. 1. — Pp. 88–102.

[92] *Paul M. J., Dredze M.* Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models // Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9–14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA. — 2013. — Pp. 168–178.

[93] *Paul M. J., Dredze M.* Discovering health topics in social media using topic models // *PLoS ONE.* — 2014. — Vol. 9, no. 8.

[94] *Paul M. J., Girju R.* Topic modeling of research fields: An interdisciplinary perspective // RANLP. — RANLP 2009 Organising Committee / ACL, 2009. — Pp. 337–342.

[95] *Pennington J., Socher R., Manning C. D.* Glove: Global vectors for word representation // Empirical Methods in Natural Language Processing (EMNLP). — 2014. — Pp. 1532–1543.

[96] *Phuong D. V., Phuong T. M.* A keyword-topic model for contextual advertising // Proceedings of the Third Symposium on Information and Communication Technology. — SoICT '12. — New York, NY, USA: ACM, 2012. — Pp. 63–70.

[97] *Pinto J. C. L., Chahed T.* Modeling multi-topic information diffusion in social networks using latent Dirichlet allocation and Hawkes processes // Tenth International Conference on Signal-Image Technology & Internet-Based Systems. — 2014. — Pp. 339–346.

[98] *Potapenko A. A., Vorontsov K. V.* Robust PLSA performs better than LDA // 35th European Conference on Information Retrieval, ECIR-2013, Moscow, Russia, 24-27 March 2013. — Lecture Notes in Computer Science (LNCS), Springer Verlag-Germany, 2013. — Pp. 784–787.

[99] *Pritchard J. K., Stephens M., Donnelly P.* Inference of population structure using multilocus genotype data // *Genetics.* — 2000. — Vol. 155. — Pp. 945–959.

[100] *Pujara J., Skomoroch P.* Large-scale hierarchical topic models // NIPS Workshop on Big Learning. — 2012.

[101] *Rahman M.* Search engines going beyond keyword search: A survey // *International Journal of Computer Applications.* — August 2013. — Vol. 75, no. 17. — Pp. 1–8.

[102] *Ramage D., Hall D., Nallapati R., Manning C. D.* Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora // Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1. — EMNLP '09. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. — Pp. 248–256.

[103] *Riedl M., Biemann C.* TopicTiling: A text segmentation algorithm based on LDA // Proceedings of ACL 2012 Student Research Workshop. — ACL '12. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. — Pp. 37–42.

[104] *Rönnqvist S.* Exploratory topic modeling with distributional semantics // Advances in Intelligent Data Analysis XIV: 14th International Symposium, IDA 2015, Saint Etienne. France, October 22 -24, 2015. Proceedings / Ed. by E. Fromont, T. De Bie, M. van Leeuwen. — Springer International Publishing, 2015. — Pp. 241–252.

[105] *Rosen-Zvi M., Griffiths T., Steyvers M., Smyth P.* The author-topic model for authors and documents // Proceedings of the 20th conference on Uncertainty in artificial intelligence. — UAI '04. — Arlington, Virginia, United States: AUAI Press, 2004. — Pp. 487–494.

[106] *Rubin T. N., Chambers A., Smyth P., Steyvers M.* Statistical topic models for multi-label document classification // *Machine Learning.* — 2012. — Vol. 88, no. 1-2. — Pp. 157–208.

[107] *Scherer M., von Landesberger T., Schreck T.* Topic modeling for search and exploration in multivariate research data repositories // Research and Advanced Technology for Digital Libraries: International Conference on Theory and Practice of Digital Libraries, TPDL 2013, Valletta, Malta, September 22-26, 2013. Proceedings / Ed. by T. Aalberg, C. Papatheodorou, M. Dobreva, G. Tsakonas, C. J. Farrugia. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. — Pp. 370–373.

[108] *Shang J., Liu J., Jiang M., Ren X., Voss C. R., Han J.* Automated phrase mining from massive text corpora // *CoRR*. — 2017. — Vol. abs/1702.04457.

[109] *Sharma A., Pawar D. M.* Survey paper on topic modeling techniques to gain usefull forcasting information on violant extremist activities over cyber space // *International Journal of Advanced Research in Computer Science and Software Engineering*. — 2015. — Vol. 5, no. 12. — Pp. 429–436.

[110] *Shashanka M., Raj B., Smaragdis P.* Sparse overcomplete latent variable decomposition of counts data // Advances in Neural Information Processing Systems, NIPS-2007 / Ed. by J. C. Platt, D. Koller, Y. Singer, S. Roweis. — Cambridge, MA: MIT Press, 2008. — Pp. 1313–1320.

[111] *Shivashankar S., Srivathsan S., Ravindran B., Tendulkar A. V.* Multi-view methods for protein structure comparison using latent dirichlet allocation. // *Bioinformatics [ISMB/ECCB]*. — 2011. — Vol. 27, no. 13. — Pp. 61–68.

[112] *Shneiderman B.* The eyes have it: A task by data type taxonomy for information visualizations // Proceedings of the 1996 IEEE Symposium on Visual Languages. — VL'96. — Washington, DC, USA: IEEE Computer Society, 1996. — Pp. 336–343.

[113] *Si X., Sun M.* Tag-LDA for scalable real-time tag recommendation // *Journal of Information & Computational Science*. — 2009. — Vol. 6. — Pp. 23–31.

[114] *Singh R., Hsu Y.-W., Moon N.* Multiple perspective interactive search: a paradigm for exploratory search and information retrieval on the Web // *Multimedia Tools and Applications*. — 2013. — Vol. 62, no. 2. — Pp. 507–543.

[115] *Sokolov E., Bogolubsky L.* Topic models regularization and initialization for regression problems // Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications. — New York, NY, USA: ACM, 2015. — Pp. 21–27.

[116] *Steyvers M., Griffiths T.* Finding scientific topics // *Proceedings of the National Academy of Sciences*. — 2004. — Vol. 101, no. Suppl. 1. — Pp. 5228–5235.

[117] *Sun Y., Han J., Gao J., Yu Y.* iTopicModel: Information network-integrated topic modeling // 2009 Ninth IEEE International Conference on Data Mining. — 2009. — Pp. 493–502.

[118] *Tan Y., Ou Z.* Topic-weak-correlated latent Dirichlet allocation // 7th International Symposium Chinese Spoken Language Processing (ISCSLP). — 2010. — Pp. 224–228.

[119] *Teh Y. W., Jordan M. I., Beal M. J., Blei D. M.* Hierarchical Dirichlet processes // *Journal of the American Statistical Association.* — 2006. — Vol. 101, no. 476. — Pp. 1566–1581.

[120] *Teh Y. W., Newman D., Welling M.* A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation // NIPS. — 2006. — Pp. 1353–1360.

[121] TextFlow: Towards better understanding of evolving topics in text. / W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, X. Tong // *IEEE transactions on visualization and computer graphics.* — 2011. — Vol. 17, no. 12. — Pp. 2412–2421.

[122] *Varadarajan J., Emonet R., Odobez J.-M.* A sparsity constraint for topic models — application to temporal activity mining // NIPS-2010 Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions. — 2010.

[123] *Varshney D., Kumar S., Gupta V.* Modeling information diffusion in social networks using latent topic information // Intelligent Computing Theory / Ed. by D.-S. Huang, V. Bevilacqua, P. Premaratne. — Springer International Publishing, 2014. — Vol. 8588 of *Lecture Notes in Computer Science.* — Pp. 137–148.

[124] *Veas E. E., di Sciascio C.* Interactive topic analysis with visual analytics and recommender systems // 2nd Workshop on Cognitive Computing and Applications for Augmented Human Intelligence, CCAAHI2015, International Joint Conference on Artificial Intelligence, IJCAI, Buenos Aires, Argentina, July 2015. — Aachen, Germany, Germany: CEUR-WS.org, 2015.

[125] *Vorontsov K., Frei O., Apishev M., Romov P., Suvorova M., Yanina A.* Non-bayesian additive regularization for multimodal topic modeling of large collections // Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications. — New York, NY, USA: ACM, 2015. — Pp. 29–37.

[126] *Vorontsov K. V., Potapenko A. A.* Additive regularization of topic models // *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization.* — 2014.

[127] *Vorontsov K. V., Potapenko A. A.* Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization // AIST'2014, Analysis of Images, Social networks and Texts. — Vol. 436. — Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS), 2014. — Pp. 29–46.

[128] *Vorontsov K. V., Potapenko A. A.* Additive regularization of topic models // *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications.* — 2015. — Vol. 101, no. 1. — Pp. 303–323.

[129] *Vorontsov K. V., Potapenko A. A., Plavin A. V.* Additive regularization of topic models for topic selection and sparse factorization // The Third International Symposium On Learning And Data Sciences (SLDS 2015). April 20-22, 2015. Royal Holloway, University of London, UK. / Ed. by A. G. et al. — Springer International Publishing Switzerland 2015, 2015. — Pp. 193–202.

[130] *Vulic I., De Smet W., Tang J., Moens M.-F.* Probabilistic topic modeling in multilingual settings: an overview of its methodology and applications // *Information Processing & Management.* — 2015. — Vol. 51, no. 1. — Pp. 111–147.

[131] *Vulić I., Smet W., Moens M.-F.* Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora // *Information Retrieval.* — 2012. — Pp. 1–38.

[132] *Wallach H. M.* Topic modeling: Beyond bag-of-words // Proceedings of the 23rd International Conference on Machine Learning. — ICML '06. — New York, NY, USA: ACM, 2006. — Pp. 977–984.

[133] *Wang C., Blei D. M.* Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process // NIPS. — Curran Associates, Inc., 2009. — Pp. 1982–1989.

[134] *Wang C., Blei D. M.* Collaborative topic modeling for recommending scientific articles // Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — New York, NY, USA: ACM, 2011. — Pp. 448–456.

[135] *Wang C., Danilevsky M., Desai N., Zhang Y., Nguyen P., Taula T., Han J.* A phrase mining framework for recursive construction of a topical hierarchy // Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '13. — New York, NY, USA: ACM, 2013. — Pp. 437–445.

[136] *Wang C., Liu J., Desai N., Danilevsky M., Han J.* Constructing topical hierarchies in heterogeneous information networks // *Knowledge and Information Systems.* — 2014. — Vol. 44, no. 3. — Pp. 529–558.

[137] *Wang C., Liu X., Song Y., Han J.* Scalable and robust construction of topical hierarchies // *CoRR.* — 2014. — Vol. abs/1403.3460.

[138] *Wang C., Liu X., Song Y., Han J.* Towards interactive construction of topical hierarchy: A recursive tensor decomposition approach // Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '15. — New York, NY, USA: ACM, 2015. — Pp. 1225–1234.

[139] *Wang H., Zhang D., Zhai C.* Structural topic model for latent topical structure analysis // Proceedings of the 49th Annual Meeting of the Association for

Computational Linguistics: Human Language Technologies - Volume 1. — HLT '11. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. — Pp. 1526–1535.

[140] *Wang X., McCallum A.* Topics over time: A non-markov continuous-time model of topical trends // Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '06. — New York, NY, USA: ACM, 2006. — Pp. 424–433.

[141] *Wang X., McCallum A., Wei X.* Topical n-grams: Phrase and topic discovery, with an application to information retrieval // Proceedings of the 2007 Seventh IEEE International Conference on Data Mining. — Washington, DC, USA: IEEE Computer Society, 2007. — Pp. 697–702.

[142] *White R. W., Roth R. A.* Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. — Morgan and Claypool Publishers, 2009.

[143] *Wu Y., Ding Y., Wang X., Xu J.* A comparative study of topic models for topic clustering of Chinese web news // Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. — Vol. 5. — july 2010. — Pp. 236–240.

[144] *Yan X., Guo J., Lan Y., Cheng X.* A biterm topic model for short texts // Proceedings of the 22Nd International Conference on World Wide Web. — WWW '13. — Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013. — Pp. 1445–1456.

[145] *Yanina A., Vorontsov K.* Multi-objective topic modeling for exploratory search in tech news // AINL. — 2016 (to appear).

[146] *Yeh J.-h., Wu M.-l.* Recommendation based on latent topics and social network analysis // Proceedings of the 2010 Second International Conference on Computer Engineering and Applications. — Vol. 1. — IEEE Computer Society, 2010. — Pp. 209–213.

[147] *Yi X., Allan J.* A comparative study of utilizing topic models for information retrieval // Advances in Information Retrieval. — Springer Berlin Heidelberg, 2009. — Vol. 5478 of *Lecture Notes in Computer Science*. — Pp. 29–41.

[148] *Yin H., Cui B., Chen L., Hu Z., Zhang C.* Modeling location-based user rating profiles for personalized recommendation // *ACM Transactions of Knowledge Discovery from Data*. — 2015.

[149] *Yin H., Cui B., Sun Y., Hu Z., Chen L.* LCARS: A spatial item recommender system // *ACM Transaction on Information Systems*. — 2014.

[150] *Yin Z., Cao L., Han J., Zhai C., Huang T.* Geographical topic discovery and comparison // Proceedings of the 20th international conference on World wide web / ACM. — 2011. — Pp. 247–256.

[151] *Zavitsanos E., Paliouras G., Vouros G. A.* Non-parametric estimation of topic hierarchies from texts with hierarchical Dirichlet processes // *Journal of Machine Learning Research.* — 2011. — Vol. 12. — Pp. 2749–2775.

[152] *Zhang J., Song Y., Zhang C., Liu S.* Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora // Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. — 2010. — Pp. 1079–1088.

[153] *Zhao W. X., Jiang J., Weng J., He J., Lim E.-P., Yan H., Li X.* Comparing Twitter and traditional media using topic models // Proceedings of the 33rd European Conference on Advances in Information Retrieval. — ECIR'11. — Berlin, Heidelberg: Springer-Verlag, 2011. — Pp. 338–349.

[154] *Zhao X. W., Wang J., He Y., Nie J.-Y., Li X.* Originator or propagator?: Incorporating social role theory into topic models for Twitter content analysis // Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management. — CIKM '13. — New York, NY, USA: ACM, 2013. — Pp. 1649–1654.

[155] *Zhou S., Li K., Liu Y.* Text categorization based on topic model // *International Journal of Computational Intelligence Systems.* — 2009. — Vol. 2, no. 4. — Pp. 398–409.

[156] *Zuo Y., Zhao J., Xu K.* Word network topic model: A simple but general solution for short and imbalanced texts // *Knowledge and Information Systems.* — 2016. — Vol. 48, no. 2. — Pp. 379–398.

БОЛЬШАКОВА Елена Игоревна
ВОРОНЦОВ Константин Вячеславович
ЕФРЕМОВА Наталья Эрнестовна
КЛЫШИНСКИЙ Эдуард Станиславович
ЛУКАШЕВИЧ Наталья Валентиновна
САПИН Александр Сергеевич

Автоматическая обработка текстов на естественном языке и анализ данных

Подписано в печать 20.07.17.
Формат 60x84/16. Бумага типографская № 2. Печать - цифровая.
Усл.печ. л. 16,8 Уч.-изд. л. 13,5. Тираж 60 экз. Заказ № .
Типография НИУ ВШЭ
101000, г. Москва, ул. Мясницкая, д. 20.