Московский государственный университет имени М. В. Ломоносова

На правах рукописи

Турдаков Денис Юрьевич

Методы и программные средства разрешения лексической многозначности терминов на основе сетей документов

05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель

д. т. н., проф.

Кузнецов Сергей Дмитриевич

Москва – 2010

Содержание

Введение

Актуальность темы

Разрешение лексической многозначности является одной из центральных задач обработки текстов. Задача заключается в установлении значений слов или составных терминов в соответствии с контекстом, в котором они использовались. Разрешение лексической многозначности используется для повышения точности методов классификации и кластеризации текстов, увеличения качества машинного перевода, информационного поиска и других приложений.

Для решения задачи необходимо определить возможные значения слов и отношения между этими значениями и контекстом, в котором использовались слова. На данный момент основным источником значений являются словари и энциклопедии. Для установления связей между значениями лингвистами создаются тезаурусы, семантические сети и другие специализированные структуры. Однако создание таких ресурсов требует огромных трудозатрат.

В начале 21-го века исследователи в области обработки естественного языка заинтересовались возможностью использования сетей документов, таких как Веб и Википедия, связанных гиперссылками, созданных огромным числом независимых пользователей, и обладающих высокой степенью актуальности.

Открытая энциклопедия Википедия является беспрецедентным ресурсом. Она позволяет автоматически составить словарь терминов, достаточный для описания любых текстовых документов, сопоставить термины со значениями, описанными в статьях Википедии, и на основе ссылочной структуры вывести отношения между этими значениями. Словарь Википедии позволяет автоматически находить в документах как отдельные слова, так и составные термины. На основе разрешения лексической многозначности выделенных

терминов, возможно определить основные тематические линии, нахождение которых необходимо для большого числа практических приложений.

Цель диссертационной работы

Целью диссертационной работы является разработка методов и программных средств разрешения лексической многозначности терминов на основе структурной и текстовой информации сетей документов. Разрабатываемые методы должны обладать следующими свойствами: они должны быть полностью автоматическими; соотношение точности и полноты должно быть равно или превышать аналогичный показатель методов, представленных в современной литературе; время работы алгоритмов должно линейно зависеть от количества обрабатываемых терминов; методы не должны быть привязаны к синтаксису конкретных языков.

Для достижения этой цели были поставлены следующие задачи:

1. разработать метод для автоматического определения отношений между значениями терминов Википедии;

2. разработать методы разрешения лексической многозначности терминов, на основе структурной и текстовой информации Википедии.

Научная новизна

Научной новизной обладают следующие результаты работы:

1. предложен подход к разрешению лексической многозначности терминов на основе сети документов Википедии.

2. разработан метод разрешения лексической многозначности, основанный на Марковской модели высокого порядка, где параметры модели оценивались на основе структурной и текстовой информации Википедии;

3. предложено обобщение Марковской модели на случай множества независимых Марковских процессов и разработан алгоритм вычисления наи-

более вероятной последовательности состояний, удовлетворяющей ограничениям модели;

4. разработан метод разрешения лексической многозначности и выделения лексических цепей, основанный на обобщенной Марковской модели.

Практическая значимость Разработанные методы разрешения лексической многозначности, основанные на Википедии, могут применяться для повышения точности реальных практических приложений, предназначенных для обработки и анализа текстовых данных.

На основе предложенных методов разработан прототип системы разрешения лексической многозначности. Этот прототип был использован в качестве основы для создания в Институте системного программирования РАН системы анализа текстов «Texterra».

Апробация работы и Публикации.

По материалам диссертации опубликовано восемь работ [1–8]. Основные положения докладывались на следующих конференциях и семинарах:

- на четвертом и пятом весеннем коллоквиуме молодых исследователей в области баз данных и информационных систем (SYRCoDIS) (2007 и 2008 гг.);
- на сто двадцать пятом и сто тридцать шестом заседаниях Московской Секции ACM SIGMOD (2008 и 2009 гг.);
- на тридцать четвертой международной конференции по очень большим базам данных (VLDB) (2008 г.);
- на международном симпозиуме по извлечению знаний из социального Веба (KASW) (2008 г.);

- на одиннадцатой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (2009 г.);

- на двадцать третей международной конференции по проблемам языка, информации и вычислений (PACLIC) (2009 г.).

Структура и объем диссертации

Работа состоит из введения, трех глав, заключения и списка литературы. Общий объем диссертации составляет 138 страниц. Список литературы содержит 119 наименований.

Глава 1

Разрешение лексической многозначности

Лексическая многозначность — это фундаментальное свойство естественных языков: каждое слово может иметь более одного значения. Так, например, каждое из 121 наиболее часто встречающихся в английском языке имен существительных имеет в среднем 7.8 значений, описанных в тезаурусе[1] английского языка WordNet [9].

Разрешение лексической многозначности — это установление значения слова в некотором контексте [10]. Для человека процесс устранения многозначности во многом является подсознательным и не представляет каких-либо трудностей. Несмотря на это, как вычислительная проблема он представляет собой сложнейшую задачу, относящуюся к «ИИ-полным» — задачам, чья сложность эквивалентна главной проблеме искусственного интеллекта — созданию ИИ в «сильном смысле» [11].

Важность задачи разрешения лексической многозначности сложно переоценить. В электронной библиотеке ACL (The Association for Computational Linguistics) содержится более 700 статей по данной теме [10]. Очевидно, что решение данной задачи является необходимым условием для полного понимания естественного языка. Также оно может быть полезным некоторым приложениям, чьей целью не является понимание естественного языка [11]:

- **Машинный перевод**: понимание смысла слова является неотъемлемой частью правильного перевода слов, значение которых зависит от контекста. Например, в зависимости от контекста английское слова *bar*

[1] ***Тезаурус*** —- это словарь, в котором слова и словосочетания с близкими значениями сгруппированы в единицы, называемые понятиями, концептами или дескрипторами, и в котором явно указываются семантические отношения между этими понятиями (концептами, дескрипторами)

может быть переведено на русский как *брусок, перекладина, отмель, полоса, музыкальный такт, стандарт, препятствие* и т. д.

- **Информационный поиск**: В процессе поиска специфичных ключевых слов, желательно оставлять только документы, в которых эти слова встречаются в нужном смысле. Например, при поиске комментариев к судебным решениям, желательно пропустить документы, в который слово *закон* ассоциируется с королевской властью [12]. Сейчас системы информационного поиска не используют специальные алгоритмы для разрешения лексической многозначности и основываются на предположении, что пользователь введет достаточно дополнительной информации о контексте, чтобы получить релевантные результаты.

- **Контент-анализ**: основным подходом в контент-анализе является анализ распределения категорий слов в текстовых коллекциях, то есть слов относящихся к заданной концепции, теме, и т. п. Очевидно, что установление смысла слова в каждом конкретном случае необходимо для построения верных распределений категорий [13].

- **Обработка речи**: разрешение многозначности необходимо для правильного воспроизведения слов при синтезе текстов, а также для сегментации слов и дифференциации омофонов при распознавании речи [14].

- **Обработка текстов**: разрешение многозначности используется для повышения точности методов классификации и кластеризации текстов [4], устранения сложных орфографических ошибок [15], анализа текстов [16, 17] и т. д.

В данной главе определяются основные понятия, необходимые для дальнейшего описания методов и алгоритмов. Описывается процесс лингвистиче-

ской обработки текстов, и очерчиваются границы решаемой задачи. Также приводится обзор литературы по данной тематике.

1.1. Используемая терминология

Терминология области, изучаемой в данной работе, тесно связана с терминологией классической лингвистики, возникшей намного раньше первых компьютеров. Еще Аристотель пытался систематизировать и объяснить феномены естественного языка [18]. Как следствие терминология классической лингвистики содержит множество нюансов известных только специалистам в области исследования языков. Однако так как исследование языка не является нашей целью, мы введем только определения и рассуждения, необходимые для понимания работы методов и алгоритмов, приведенных в следующих главах.

1.1.1. Терминология классической лингвистики

Во всех развитых языках присутствуют как ***однозначные***, так и ***многозначные*** слова. Способность слов выступать лишь в одном значении называется ***однозначностью*** или ***моносемией***. Примеры таких слов: *«бинокль»*, *«троллейбус»*, *«suitcase»*, *«noun»*. Однако большинство слов имеют не одно, а несколько значений. Они называются ***многозначными*** или ***полисемантическими***. Способность лексических единиц иметь несколько значений называется ***многозначностью*** или ***полисемией***. Примерами таких слов могут служить *«дом»* (жилище, строение, домашнее хозяйство, семья), *платформа, platform* (железнодорожная, политическая, компьютерная, континентальная).

Слово приобретает многозначность в процессе исторического развития языка. Объем словаря любого языка ограничен, поэтому развитие лексики

происходит не только благодаря созданию новых слов, но и в результате увеличения числа значений у ранее известных, отмирания одних значений и возникновения новых.

В то же время было бы неверно считать, что развитие значений слов вызывается только внеязыковыми факторами. Многозначность обусловлена и чисто лингвистически: слова способны употребляться в переносных значениях. Названия могут переноситься с одного предмета на другой, если у этих предметов есть общие признаки. Ведь в лексическом значении слов отражаются не все дифференциальные признаки называемого предмета, а лишь те, которые обратили на себя внимание в момент номинации. Таким образом, у многих предметов есть общие связи, которые могут послужить основанием для ассоциативного сближения этих предметов и переноса названия с одного из них на другой [19].

Слова, которые звучат одинаково, но имеют совершенно разные значения, принято выделять в отдельную группу. Такие слова называются лексическими ***омонимами***, а звуковое и грамматическое совпадение разных языковых единиц, которые семантически не связаны друг с другом, называется ***омонимией***. Например, *ключ* — «родник» (студеный ключ) или «металлический стержень особой формы для отпирания и запирания замка» (стальной ключ); *лук* — «растение» (зеленый лук) или «оружие для метания стрел» (тугой лук). В отличие от многозначных слов лексические омонимы не обладают предметно-семантической связью, т. е. у них нет общих семантических признаков, по которым можно было бы судить о полисемантизме одного слова.

Наряду с омонимией обычно рассматривают смежные с ней явления, относящиеся к грамматическому, фонетическому и графическому уровням языка. Среди созвучных форм выделяют ***омоформы*** — слова, совпадающие лишь в какой-нибудь одной грамматической форме (*«три»* — числительное в именительном падеже (три друга) или глагол в повелительном наклонении един-

ственного числа 2-го лица). Слова, которые звучат одинаково, но пишутся по-разному, называются ***омофонами*** (луг и лук, молод и молот). Слова, которые пишутся одинаково, но произносятся по-разному, называются ***омографами*** (крӯжки - кружкѝ).

Современной наукой выработаны критерии разграничения омонимии и многозначности, помогающие развести значения одного и того же слова и омонимы, которые возникли в результате полного разрыва полисемии. Однако нередки разночтения в определении границ омонимии и многозначности [19]. Для детального изучения этого вопроса следует обратиться к специальной литературе.

В данной работе мы будем использовать термины *«многозначность»* и *«полисемия»* как синонимы для обозначения наиболее широкого понятия, включающего в себя все приведенные феномены, не вдаваясь в детали определения границ между различными классами.

1.1.2. Терминология компьютерной лингвистики

В специальной литературе делается различие между терминами ***«компьютерная лингвистика»*** («computational linguistics») и ***«обработка естественного языка»*** («natural language processing»). Компьютерная лингвистика ставит своей целью использование компьютеров для развития и продвижения лингвистической теории. В свою очередь, обработка естественного языка является прикладной областью, развивающей и использующей технологии для обработки речи и текста. Большинство современных методов для разрешения лексической многозначности относятся последнему направлению, при этом, некоторые теоретические разработки в частности и решение задачи разрешения лексической многозначности в целом относится к области ответственности компьютерной лингвистики. В дальнейшем мы будем исполь-

зовать термин «компьютерная лингвистика» в наиболее широком смысле, включающем в себя оба понятия.

Исследователи выделяют несколько типов многозначности естественного языка, и для работы с каждым из этих типов существуют собственные методы:

- ***Морфологическая (грамматическая) многозначность.*** Значения слова могут относиться к разным частям речи, например, англ. «look» может быть существительным «взгляд» или глаголом «смотреть». Этот тип многозначности более свойственен английскому языку, чем русскому. Морфологическая многозначность представляет собой основной объект изучения задачи определения частей речи слов (part of speech tagging). Однако современные системы способны эффективно решать эту задачу, используя методы машинного обучения, такие как метод опорных векторов и метод максимальной энтропии, и показывая точность более 97% [20, 21].

- ***Синтаксическая многозначность.*** Синтаксическая неоднозначность возникает в тех случаях, когда предложение может быть описано более чем одной синтаксической структурой. Примерами такой неоднозначности являются предложения *«flying planes can be dangerous»* («летящие самолеты могут быть опасны» или «летать на самолетах может быть опасно»); *мужу изменять нельзя* (кто потенциально может изменить — муж или жена). В данной работе не рассматривается уровень синтаксического анализа и, соответственно, не возникает проблема синтаксической многозначности.

- ***Лексическая многозначность.*** Значения слов могут относиться к одной части речи и различаются по смыслу, например «platform» —

железнодорожная или компьютерная платформа. Некоторые исследователи различают понятия ***смысл*** и ***значение*** [22]. Однако в данной работе мы будем использовать эти слова, как синонимы.

Разрешение лексической многозначности — наиболее употребляемый в русскоязычной литературе перевод английского термина ***«word sense disambiguation»***. В качестве синонимов к слову «многозначность», также часто употребляют слово *«неоднозначность»*. Синонимами к слову «разрешение» являются слова *«устранение»* и *«снятие»*. Кроме того, в русскоязычной литературе встречается заимствованное слово ***«дизамбигуация»*** для обозначения данной задачи. В дальнейшем, эти термины будут использоваться как синонимы. Этот тип многозначности представляет особый интерес в данной работе и будет подробно рассмотрен ниже.

- ***Семантическая многозначность.*** Под семантической многозначностью понимается возможность использования слова в переносном значении (*«лиса»* как обозначение хитрого человека). При наличии специальных словарей, большая часть методов разрешения лексической многозначности может применяться и для разрешения семантической многозначности, поэтому в литературе часто используют термин ***«лексико-семантическая многозначность»*** или просто «лексическая многозначность» для обозначения обоих явлений.

- ***Прагматическая неоднозначность***. Еще один тип неоднозначности возникает в результате употребления местоимений или специальных существительных вроде one, another (еще один). Так в предложении *«Она уронила карандаш на стол и сломала его»* невозможно однозначно определить, что именно было сломано — карандаш или стол, но при этом обоим случаям соответствует одинаковая синтаксическая

структура. Этот тип неоднозначности является одним из объектов исследования при разработке методов ***установления кореферентности*** («coreference resolution»), занимающихся поиском отношений между компонентами высказывания, которые обозначают один и тот же объект внеязыковой действительности. Эти методы тесно переплетены с синтаксическим анализом и не будут подробно рассматриваться в данной работе.

Основным объектом исследования в данной работе является лексическая многозначность. Далее, в этой главе, будут описаны методы разрешения лексической многозначности и основные проблемы, связанные с данной темой.

1.2. Основные проблемы разрешения лексической многозначности

Для решения задачи разрешения лексической многозначности необходимо:

1. для каждого слова, относящегося к тексту, определить какие оно может иметь значения;

2. на основании контекста, в котором встретилось слово, выбрать наиболее подходящее значение.

Таким образом, для формализации задачи требуется дать определение того, что является значением слова и что такое контекст. Отсутствие строгого определения этих понятий порождает дополнительные сложности для разработки и сравнения методов разрешения лексической многозначности.

В этом разделе будут рассмотрены существующие подходы к определению значений, контекста, а также методы сравнения различных алгоритмов разрешения лексической многозначности.

1.2.1. Значение

Формализация задачи устранения многозначности проблематична, так как не существует признанного всеми способа определить, где заканчивается одно значение слова и начинается другое. Большинство современных работ опираются на предопределенные значения: списки слов, найденные в словарях, переводы на иностранные языки и т. п. Однако различные словари могут содержать различное количество значений для одних и тех же слов. Другим подходом к определению значений является анализ способов употребления слов в текстовом корпусе[2] и описание значений на основе этого анализа.

Известный русский лингвист В. В. Виноградов отмечал сложность определения значений слова в современной лингвистике [23]:

> Термин «лексическое» или, как в последнее время стали говорить, «смысловое значение слова» не может считаться вполне определенным. Под лексическим значением слова обычно разумеют его предметно-вещественное содержание, оформленное по законам грамматики данного языка и являющееся элементом общей семантической системы словаря этого языка. Общественно закрепленное содержание слова может быть однородным, единым, но может представлять собою внутренне связанную систему разнонаправленных отражений разных «кусочков действительности», между которыми в системе данного языка устанавливается смысловая связь. Разграничение и объединение этих разнородных предметно-смысловых отношений в структуре слова сопряжено с очень большими трудностями. Эти трудности дают себя знать в типичных для толковых словарей непрестанных смешений значений и употреблений

[2] ***Лингвистический корпус*** — набор текстов, собранных в соответствии с определенными принципами, размеченных по определенному стандарту и обеспеченных специализированной поисковой системой.

слова, в расплывчатости границ между значениями и оттенками значений слова, в постоянных разногласиях или разноречиях по вопросу о количестве значений слова и о правильности их определения.

Одна из главных проблем для разрешения многозначности состоит в определении степени ***гранулярности*** значений. Использование слишком тонко различающихся значений создает практические трудности для автоматического разрешения многозначности: требуется делать слишком сложный выбор, даже для экспертов в лексикографии; возникает эффект комбинаторного взрыва при обсчете всех возможных комбинаций значений слов предложения или текста; существенно увеличивается сложность обучения методов, основанных на обучении с учителем. С другой стороны слишком грубое разбиение не подходит для многих задач обработки естественного языка.

На основании вышесказанного можно заметить, что для различных задач необходима различная гранулярность. Так для расстановки ударений [15] в испанском языке необходимо различать только омографы, в то время как для таких задач, как машинный перевод необходимо намного более тонкое различие между значениями, иногда более тонкое, чем можно найти в одноязычном словаре. Например, русскому слову *«город»* (населенный пункт) в английском языке соответствует два перевода: *«town»* и *«city»* — зависящих от размера конкретного населенного пункта. При этом не существует строгой связи между задачей и необходимой гранулярностью. Например, слово *«мышь»*, хотя и имеет два различных смысла (животное и устройство), переводится на английский в обоих случаях одинаково (*«mouse»*). С другой стороны для информационного поиска различие между этими значениями очень важно, в то время как сложно представить случай, когда будет необходимо различить английские значения слова *«город»*.

1.2.2. Контекст

Все работы по разрешению многозначности основываются на информации, которую предоставляет контекст многозначного слова, для выбора значения этого слова. Обычно контекст используется одним из двух способов:

- контекст представляется как окно из слов вокруг целевого слова, сгруппированное без учета расстояния, грамматических отношений и т. д.
- контекст рассматривается в терминах некоторого отношения с целевым словом, включающим в себя расстояние до цели, синтаксические связи, орфографические свойства, семантические категории и т. д.

Первый подход считается менее эффективным, но более дешевым, хотя для некоторых приложений он может показать впечатляющие результаты.

Информация о микро-контексте (несколько слов в ближайшем окружении целевого слова), тематическом контексте (несколько предложений вокруг целевого слова), и контексте, определяемым областью знаний, для которой решается задача снятия многозначности, играют важную роль при выборе значения. При этом отношения между этими типами контекстов, их роли и важность до сих пор плохо изучены. Далее каждый тип контекста будет рассмотрен подробнее.

Большинство работ по снятию многозначности используют локальный или «микро» контекст, как основной информационный источник для определения правильного значения. Локальным контекстом обычно считается небольшое окно из слов в окружении целевого слова, обычно не превышающее размера одного предложения. Возникает вопрос, какой размер микро-контекста контекста оптимален.

Первую попытку ответа этот вопрос можно встретить в работе Каплана [24], написанной в 1950 году. Он экспериментально показал, что двух слов

вокруг многозначного слова достаточно для определения его значения в большинстве случаев. Однако в современных работах длина микроконтекста может варьироваться и зависит от приложения.

Наиболее полное современное исследование этого вопроса провел Дэвид Яровски в 1993–94 годах [15, 25]. Он сделал наблюдение, что длина микроконтекста может варьироваться в зависимости от типа многозначности. Он предположил, что для разрешения локальной многозначности достаточно 3–4 слова контекста, в то время как для семантической многозначности необходимо большее окно, состоящее из 20-50 слов. Таким образом, исследователи до сих пор не пришли к единому мнению по поводу оптимальной длины микроконтекста.

Дополнительно для разрешения лексической многозначности в некоторых работах используется информация о словосочетаниях и синтаксических отношениях. Так Яровски [25] установил, что для одинаковых сочетаний из двух слов, вероятность употребления соответствующих слов в одинаковых значениях колеблется в пределах 90–99%. Это наблюдение послужило для использования во многих современных работах эвристики «одно значение для словосочетания» (one sense per collocation), то есть соответствующие слова в одинаковых словосочетаниях должны иметь один и тот же смысл.

Ранние работы по устранению лексической многозначности, часто основывались на полном синтаксическом разборе предложений. Однако впоследствии полный синтаксический разбор стал заменяться частичным, использующимся только для выделения именных, предложных и глагольных групп. Яровски [25] определил, что для слов, относящихся к различным частям речи, наибольший вклад в разрешение лексической многозначности дают различные источники. Так для устранения многозначности глаголов наиболее важны их объекты, чем субъекты, для существительных наибольшее значение имеют смежные с ними прилагательные и существительные, а для при-

лагательных практически всю информацию о их значениях можно получить, основываясь на существительных, которые они модифицируют. На данный момент, для устранения многозначности, в основном, используется информация о частях речи слов, в комбинации с другими методами, не относящимися к синтаксическому разбору.

Исследования по применению тематического контекста появились несколько позже микро-контекста и несколько лет активно обсуждались в области информационного поиска [12]. Современные работы в основном комбинируют тематический и микро- контексты. Вильям Гэйл и др. [26] улучшили точность своего метода с 86% до 90%, расширив контекст из 12 слов в окружении целевого до 100 слов. Кроме того, они показали, что важность слов контекста падает при удалении от целевого слова. В похожей работе [27] они показали, что в одинаковых тематических контекстах значения соответствующих многозначных слов совпадают (one sense per discourse). Яровски [25] показал, что в то время как большой контекст может быть использован для определения значений имен существительных, при разрешении лексической многозначности глаголов и прилагательных важность слов контекста быстро уменьшается с увеличением расстояния до целевого слова. Это указывает на то, что для успешного определения значений всех слов текста требуется использовать как микро- так и тематический контекст, и что для устранения многозначности различных типов слов необходимы разные методы.

Методы, использующие тематический контекст могут быть улучшены разделением анализируемого текста на подтемы. Очевидным решением является разбиение текста на параграфы, однако это довольно грубое деление, так как одна тема обычно развиваться в течение нескольких параграфов. Автоматическая сегментация текста на такие подтемы несомненно поможет улучшить методы разрешения лексической многозначности. Было замечено, что повторение слов в соседних сегментах или предложениях является сильным индика-

тором подтемы. Один из последних методов, использующих это наблюдение можно найти в статье Ричмонда и др. [28].

Контекст, определяемым областью или доменом, может дать существенный прирост точности методов устранения многозначности в применении к некоторым специфичным приложениям. Так если в некоторой организации будет заранее определен словарь терминов, употребляемых в официальных документах, то выбирать правильное значение будет намного легче. Однако очевидно, что в общем случае этот тип контекста необходимо комбинировать с другими типами для достижения хороших результатов.

Ярким примером применения знаний о глобальном контексте может считаться программа SHRDLU, созданная Терри Виноградом [29]. Пользователь общался с программой SHRDLU с помощью обычных выражений английского языка. По его приказу SHRDLU перемещала простые объекты в упрощенном «мире блоков»: кубики, конусы, шары и так далее. Мир SHRDLU был настолько прост, что полный набор объектов и локаций мог быть описан, не более чем пятьюдесятью различными словами — существительными, такими как «блок» или «конус», глаголами, такими как «помести на» или «перемести в», и прилагательными, такими как «большой» или «синий». Всевозможные комбинации этих базовых блоков языка были элементарны, и программа успешно понимала то, что имел в виду пользователь.

1.2.3. Методы оценки

Еще одной важной проблемой является оценка методов устранения многозначности и их сравнение. Так как разрешение многозначности является промежуточной задачей, существуют два способа оценки: in vitro — насколько хорошо работают методы сами по себе, и in vivo — как разрешение многозначности улучшает работу системы в целом. На данный момент, практически не

существует попыток оценки встроенных систем (in vivo), при этом, оценке отдельностоящих алгоритмов посвящено довольно большое количество работ.

Для оценки in vitro обычно используют два коэффициента: точность и полноту. ***Точность*** — это число слов, размеченных правильно, по отношению к числу всех слов, обработанных системой. ***Полнота*** — число слов, размеченных правильно, по отношению к числу слов в тестовом множестве. Также часто вводят F-меру, значением которой является среднее гармоническое между полнотой и точностью.

Пример: в тестовом множестве 100 слов, система работала с 75 словами, правильно определила значение 50 слов. Тогда точность вычисляется как $50/75 = 0.66$, полнота — $50/100 = 0.50$.

Для сравнения методов разрешения многозначности были разработаны специальные тестовые коллекции. Общей чертой всех коллекций является то, что в них используются заранее определенные наборы значений многозначных слов, которые берутся из некоторого словаря. Этот словарь во многом определяет результаты тестирования.

Наиболее популярным электронным словарем является WordNet [9]. WordNet — это большая электронная база данных, организованная в семантическую сеть, состоящую из различных отношений между словами, включающих синонимию, антонимию, обобщение и детализацию. WordNet является наиболее распространенной и часто используемой базой данных для исследований в области обработки естественного языка. Кроме того, существуют аналоги WordNet для некоторых языков, отличных от английского.

Помимо тестирования алгоритмов, размеченные корпусы также используются для тренировки алгоритмов, основанных на обучении. Заметим, что обучение и тестирование на похожих данных является одной из причин хороших показателей этих методов.

Первой попыткой создания централизованной коллекции для тестирова-

ния методов разрешения многозначности является Semcor (семантический конкорданс) [30]. Он был основан на подмножестве корпуса Brown [31] и значениях из словаря WordNet 1.6. К сожалению, Semcor был слишком мал для создания надежных алгоритмов снятия многозначности. Аналогами этого корпуса, были line-hard-serve [32] и interest [33], в которых собрано большое количество употреблений нескольких заранее выбранных слов.

Для тестирования расширяемости в 1996 году был создан корпус DSO [34]. Он включал в себя 192,800 экземпляров наиболее часто употребляемых слов английского языка, которые были вручную размечены значениями из WordNet. Этот корпус был на два порядка больше, чем предыдущие коллекции.

Следующей ступенью в эволюции методов оценки стал корпус Senseval. На момент написания этой работы существовало три версии этого корпуса и проведены три конференции по сравнению методов разрешения лексической многозначности на этих коллекциях. Летом 2010 года планируется провести четвертую конференцию, где будет учтен опыт предыдущих лет.

Senseval-1 [35] состоит из размеченных профессиональными лексикографами предложений из корпуса Hector [36]. Критерием служила правильность определения смысла 34 слов, значения которых брались из словаря Hector. Для различных слов степень согласия о значении слова между экспертами была 80%–95.5%. Таким образом, степень согласия определяет ***верхнюю границу*** точности автоматических методов. Лучший результат, показанный автоматическими методами на этом корпусе для слов с различной гранулярностью, был 77.1%–81.4%. В результате создания корпуса Senseval-1 и тестирования на нем методов разрешения лексической многозначности было показано, что для различных типов слов, необходима различная степень гранулярности. Это повлияло на создание следующего корпуса Senseval-2, где явным образом разделялись задачи определения смысла всех слов в тексте (all-word task) и снятия многозначности только некоторого типа слов(lexical sample

task).

В качестве основы Senseval-2 для определения значений использовался WordNet, и были составлены тесты для 10 языков. Для задачи определения смысла всех слов было размечено 5000 слов из трех статей Penn Treebank II [37], описывающих различные области.

Для разрешения лексической многозначности, учитывающей тип слова, из WordNet 1.7 было выделено всего 73 существительных, прилагательных и глагола, и вручную размечено от 70 до 300 вхождений каждого, в зависимости от числа возможных значений. В качестве основного корпуса для разметки использовался Penn Treebank II, а для малоупотребимых значений слов дополнительно использовался British National Corpus. Для имен существительных степень согласия экспертов была 64%-85.5%, при среднем количестве значений на слово 4.9.

В качестве основы для корпуса Senseval-3 [38] были взяты результаты проекта OMWE [39], где пользователям WEB в виде игры было предложено разметить слова в некотором контексте значениями из WordNet 1.7. Так как в этом проекте участвовали обычные пользователи, степень согласия между ними была ниже (67%) по сравнению с экспертами, разметившими Senseval-2 (85.5%). Лучшие результаты на этом корпусе показали методы, основанные на машинном обучении, такие как метод опорных векторов (SVM).

Необходимо заметить, что общая производительность систем в целом сильно зависит от качества разметки тренировочного корпуса, согласия экспертов, а также качества и гранулярности словаря.

По результатам последней конференции Senseval, исследователи пришли к мнению, что традиционный метод оценки отдельностоящих алгоритмов разрешения лексической многозначности (in vitro) достиг своей вершины и не может привести к новым фундаментальным результатам [10]. Однако этот метод оценки продолжает широко использоваться, из-за простой и понятной

постановки задачи.

Основываясь на успехе проекта Senseval, можно утверждать, что оценка алгоритмов разрешения лексической многозначности in vitro хорошо изученная задача: существует три различных корпуса состоящих из различных слов, с различным распределением значений и частот использования. Однако все еще остается несколько открытых проблем.

Наиболее важный фактор в оценке — это выбор словаря значений, что сильно зависит от конечного приложения, частью которого является алгоритм устранения многозначности. Также остается вопрос, подходит ли степень гранулярности этого словаря для конечного приложения.

Кроме того, лексикографы давно признали, что в естественном языке значение слова может зависеть от некоторого недостаточно определенного контекста и соответствовать более чем одному специфическому смыслу. Также некоторые смыслы могут быть плохо описаны в словарях. В таких случаях и эксперты, и система должны иметь возможность выбора некоторого альтернативного значения или более широкого значения, которое описывает все нюансы более специфического.

1.3. Обзор работ

Задача разрешения лексической многозначности (word sense disambiguation) возникла в 50-х годах прошлого века в качестве подзадачи машинного перевода. С тех пор исследователи предложили огромное количество методов решения этой задачи, однако она остается более чем актуальной и по сей день.

Условно можно выделить три этапа развития методов устранения лексической многозначности. С 50-х по 80-е года были разработаны основные подходы, однако из-за отсутствия хороших машинных словарей и баз знаний,

в этот период были созданы только «игрушечные» системы, покрывающие лишь крошечную часть языка.

Следующий этап, пик которого пришелся на 90-е годы, был обусловлен созданными вручную крупномасштабными базами знаний, такими как WordNet [9] и CyC [40] и сбалансированными корпусами документов (Brown [31], Penn TreeBank [41]). Алгоритмы, разработанные в этот период, использовали структуру баз знаний или обучались на общепризнанных корпусах. Исследователи получили хорошие результаты, однако сложность ручного создания и поддержки в актуальном состоянии больших структур ограничила область применения этих алгоритмов.

В начале 21-го века исследователей в области обработки естественного языка заинтересовала возможность использования **сетей документов**, таких как WWW и Wikipedia, связанных гиперссылками и созданных огромным числом независимых пользователей. Большим преимуществом таких сетей является то, что пользователи Интернета поддерживают их всегда в актуальном состоянии, а их документы описывают детально все области человеческой жизнедеятельности. Структура таких сетей отличается от созданных экспертами в 90-х гг. баз знаний, что влечет за собой необходимость разработки новых моделей и алгоритмов.

Подробно сети документов рассматриваются в следующей главе, поэтому далее будет приведен обзор алгоритмов относящихся только к первым двум этапам. Алгоритмы снятия многозначности, использующие сети документов будут рассмотрены в конце второй главы в разделе 2.4.

1.3.1. Работы 50-х — 80-х годов

Изначально в задаче машинного перевода выделялись две подзадачи — анализа и синтеза, а основная проблема заключалась в том, чтобы, взяв вход-

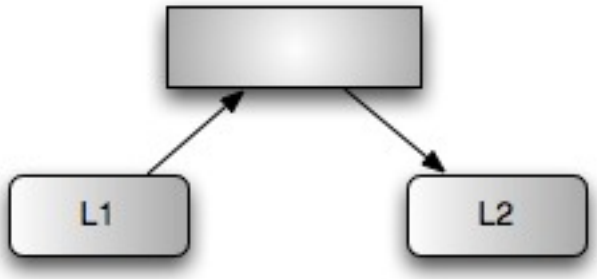

Рис. 1.1. Модель машинного перевода

ную цепочку на естественном языке $L1$, произвести ее разбор в нечто, и затем это нечто синтезировать в язык $L2$ (рис. 1.1) [22].

Первоначальный подход к машинному переводу испытывал сильное влияние синтаксической теории Хомского [42]. Однако пока содержимое пустого прямоугольника на рис. 1.1 рассматривалось как синтаксическое дерево и смысл полностью оставался в стороне добиться сколько-нибудь значимых успехов не удалось.

С тех пор, как стало очевидно, что синтаксической теории недостаточно для успешного машинного перевода, многие исследователи обратили свое внимание на проблему представления знаний [22, 29, 43]. Так в конце 50-х возникло понятие семантической сети[3] [44, 45]. На основе первой семантической сети и тезауруса Роджетса была создана первая машинно-ориентированной база знаний [44] и применена к проблеме устранения лексической многозначности при попытке машинного перевода «Георгики» Вергилия с латинского языка на английский [45].

Следующий этап в развитии методов разрешения лексической многозначности был связан с бурным развитием методов искусственного интеллекта. В 60-х годах создавались системы, пытающиеся полностью смоделировать

[3]Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (ребра) задают отношения между ними.

понимание естественного языка человеком.

Квилиан [46] в своих работах использовала семантическую сеть, состоящую из слов (токенов) и концепций (типов). Связи были помечены семантическими отношениями (IS-A). Далее программа моделировала последовательную активацию узлов в соответствии со ссылками, и неоднозначность устранялась выбором наиболее близкого узла-концепции. Эти работы стали основой для создания подхода, основанного на внешних источниках знаний.

Также исследователи активно использовали понятие ***фреймов*** [43]. Хаес [47] использовал семантическую сеть, состоящую из узлов-существительных и связей-глаголов, на которую с помощью фреймов наложены отношения IS-A и PART-OF. Наиболее подходящее значение выбиралось способом, аналогичным подходу Квилиан. Хаес достиг хороших результатов в разрешении лексической многозначности омографов, но для других типов полисемии результаты были намного менее успешными.

В начале 70-х годов группа исследователей из Стэндфордского университета во главе с Р. Шенком предложила теорию концептуальной зависимости для моделирования процесса обработки естественного языка человеком [22]. Основной идеей работы было использование семантической составляющей при разборе входных цепочек. Авторы полностью не отказывались от синтаксического разбора, однако сводили его к минимуму, мотивируя это тем, что обычному человеку не обязательно знать синтаксическую структуру предложения, чтобы понять смысл. Более того, одним из базовых предположений являлось то, что смысловое представление не зависит от языка (концептуальный уровень), в том числе и от его синтаксических правил. Таким образом, Р. Шенк показал, что для понимания языка необходимо оперировать понятием смысла. На примере нескольких предложений Шенк показывал, что его теория работоспособна, однако при этом признавал, что это лишь начальный этап на пути к созданию полноценной системы, понимающей любые предло-

жения естественного языка.

Работы в психолингвистике 60—70-х годов установили, что в процессе разрешения лексической многозначности, осуществляемом человеком, введение новой концепции влияет на определение смыслов уже введенных концепций. Основываясь на этой идее, Коллинс разработал [48] метод ***распространяющейся активации*** (spreading activation), в котором узлы-концепции в семантической сети активируются, когда встречается новое слово, и эта активация распространяется к соединенным узлам, ослабевая по мере удаления (spreading activation). Таким образом, в качестве значения выбирается наиболее активный узел.

В 50–80-х годах прошлого века было разработано множество идей, которые используются в современных методах, однако из-за нехватки вычислительных мощностей и данных, пригодных для машинной обработки, проверку многих идей пришлось отложить на десятки лет.

1.3.2. Методы, основанные внешних источниках знаний

Из алгоритмов, появившихся в последнее время, можно выделить на два доминирующих класса. Первый класс составляют алгоритмы разрешения лексической многозначности на основе внешних источников знаний (knowledge-based methods). В основном, они предлагались и тестировались для тезауруса английского языка WordNet. Большинство из этих алгоритмов основано на анализе взаимного расположения значений многозначного слова и слов контекста в структуре тезауруса. С появлением больших корпусов данных, активное развитие также получили статистические методы, основанные на машинном обучении. Лучшие результаты на корпусе Senseval-3(59.1%) были достигнуты системами, основанными на обучении по размеченным корпусам и использующими комбинации нескольких классификаторов [38].

Однако методы, основанные внешних источниках знаний обладают рядом преимуществ, тем самым, привлекая внимание исследователей. Они могут быть легко адаптированы к документам, полученным из любых источников, в отличие от методов, основанных на обучении, которые применяются, только к словам, доступным в размеченном корпусе. Другим важным преимуществом этих методов, является то, что они не зависят от доступности размеченных корпусов и могут быть легко применены к любым другим языкам.

Алгоритмы, относящиеся к этой категории методов можно разделить на четыре основные группы:

1. **Алгоритм Леска** [49] и его вариации: выбор наиболее правдоподобного значения в заданном контексте осуществляется посредством измерения степени пересечения словарных определений значений целевого слова и слов контекста.

2. **Меры семантической близости, вычисленные на основе семантических сетей.** В эту категорию входят методы вычисления семантической плотности/расстояния между концепциями. В зависимости от размера контекста, с которым они работают, эти методы, в свою очередь, подразделяются на две категории:

   - Методы, применяемые к локальному контексту, где семантические меры используются для устранения лексической многозначности слов связанных через а) синтаксические отношения; б) их месторасположение.
   - Методы, применяемые к глобальному контексту, где лексические цепи[4] (lexical chains) получаются на основе мер семантической близости.

[4] ***Лексическая цепь*** — это последовательность слов в обрабатываемом текст, связанных по значению.

3. Автоматически или полуавтоматически выведенные **правила ограничения** значений слов, на основе отношений с другими словами контекста.

4. **Методы, основанные на эвристиках**, состоящие из простых правил позволяющих с большой вероятностью определить значения для некоторых категорий слов. Включают в себя:

    - Наиболее частое значение (Most frequent sense)
    - Одно значение в одинаковых словосочетаниях (One sense per collocation)
    - Одно значение в одинаковых тематических контекстах (One sense per discourse)

Чтобы текст был согласован, слова, из которых он состоит, должны быть связаны по смыслу. Это естественное свойство языков является наиболее сильным ограничением для автоматического снятия лексической многозначности. В то время как этот тип семантического ограничения поддерживает целостность текста в целом, границы его применения обычно лимитированы небольшим числом слов, найденных в непосредственной близости от целевого, или других слов, связанных синтаксическими зависимостями с целевым. Однако существуют и методы, зависящие от глобального контекста и пытающиеся найти последовательности связанных по смыслу слов, проходящие через весь текст. Лексические цепи являются хорошим примером таких семантических отношений.

Для методов, основанных на внешних источниках, в качестве нижней границы часто используют результат работы алгоритма Леска. ***Алгоритм Леска*** – это классический алгоритм автоматического снятия многозначности, введенный М. Леском в 1986 году [49]. Алгоритм основан на предположении,

что многозначное слово и его окружение относятся к одной теме. Простая реализация алгоритма Леска состоит из трех шагов: 1) выбрать многозначные слова и их контексты; 2) взять определение всех найденных слов в некотором словаре; 3) в качестве значения многозначного термина выбрать то, которое максимизирует количество общих слов в словарном определении данного значения и определений терминов контекста. Существует несколько вариаций алгоритма Леска, отличающихся способом выбора значения [10].

Классическим примером применения алгоритма Леска является определение значений слов выражения «PINE CONE». Слово «PINE» имеет два смысла:

1. kinds of evergreen tree with needle-shaped leaves (сосна)
2. waste away through sorrow or illness (чахнуть, томиться)

слово CONE — три:

1. solid body which narrows to a point (конус)
2. something of this shape whether solid or hollow (предмет в форме конуса)
3. fruit of certain evergreen trees (шишка)

Таким образом вычисляя пересечение слов в словарных определениях получаем $Pine\#1 \cap Cone\#3 = 2$.

Кроме алгоритма Леска для оценки нижней границы точности часто используется метод, который всем словам присваивает их наиболее частое значение. Так как для создания этого алгоритма требуется собрать статистику употребления значений, он чаще всего используется для сравнения точности методов, основанных на обучении по размеченным корпусам.

Слова, разделяющие общий контекст, обычно имеют похожий смысл, поэтому, подходящее значение может быть найдено по наименьшему семанти-

ческому расстоянию до контекста. На этом наблюдении основаны методы, использующие семантическую близость. Большинство мер семантической близости вычисляется на основе иерархии WordNet. Далее приводятся несколько таких мер, часто используемых в исследованиях.

Определение 1. *Семантической близостью называется отображение* $f : X \times X \to \mathcal{R}$*, ставящее в соответствие паре слов, терминов или их значений действительное число и обладающее следующими свойствами:*

1. $0 \leq f(x, y) \leq 1$,
2. $f(x, y) = 1 \Leftrightarrow x = y$.

В одной из первых работ [50], использующих семантическую близость двух концепций C_1 и C_2, эта мера определяется через путь минимальной длины, проложенный в иерархии синсетов WordNet:

$$sim(C_1, C_2) = -log\left(\frac{Path(C_1, C_2)}{2D}\right) \quad , \tag{1.1}$$

где D — общая глубина таксономии.

Хирст [51] интегрировал в меру близости направление связей, формирующих путь. В дополнение к длине, путь не должен «слишком часто менять направление». В уравнении 1.2 C и k — константы, а d — это число изменений направления.

$$sim(C_1, C_2) = C - Path(C_1, C_2) - kd \tag{1.2}$$

Резник [52] определил ***информационное содержание*** (information content), определяющий специфичность концепции. Эта мера определяется через вероятность появления концепции в большом корпусе.

$$IC(C) = -log(P(C)) \tag{1.3}$$

$P(C)$ — вероятность встретить слово, относящееся к классу C. Таким образом, значение $P(C)$ больше для концепций расположенных выше в иерархии и достигает своего максимума на вершине иерархии.

Основываясь на понятии информационного содержания, Резник определил меру семантической близости между двумя концепциями через измерение информационного содержания ближайшего общего предка (lowest common subsumer или LCS), то есть первого общего узла в семантической сети, встретившегося при подъеме из двух данных концепций к вершине иерархии.

$$sim(C_1, C_2) = IC(LCS(C_1, C_2)) \tag{1.4}$$

Альтернатива такому определению, использующая различие в информационном содержании двух концепций, предложена в работе [53].

$$sim(C_1, C_2) = 2 \times IC(LSC(C_1, C_2)) - (IC(C_1) + IC(C_2)) \tag{1.5}$$

Лин [54] предложил комбинировать значения информационного содержимого по-другому:

$$sim(C_1, C_2) = \frac{2 \times IC(LSC(C_1, C_2))}{IC(C_1) + IC(C_2)} \tag{1.6}$$

Все предыдущие меры применимы только к концепциям, явно соединенным через ребра семантической сети. Михалсиа и Молдован [55] предложили формулу для вычисления близости между независимыми иерархиями, включая иерархии для слов, относящихся к различным частям речи (ур. 1.7). В уравнении 1.7 $|CD_{12}|$ — это число общих слов в словарных определениях слов, встретившихся в иерархиях C_1 и C_2, $desc(C_2)$ — число концепций в иерархии C_2, W_k — вес, присвоенный каждой концепции, определяющий глубину концепции в иерархии.

$$sim(C_1, C_2) = \frac{\sum_{k=1}^{|CD_{12}|} W_k}{log(desc(C_2))} \tag{1.7}$$

Метод ***концептуальной плотности*** (conceptual density) использует иерархию WordNet для вычисления наиболее подходящего значения слова, основываясь на контексте в котором это слово встретилось [56]. Значения обрабатываемого слова и слов контекста отмечаются в иерархии WordNet, а затем производится поиск поддерева, имеющего наибольшую плотность отметок. В качестве ответа выбирается значение, содержащееся в наиболее плотном поддереве.

Плотность для концепции c, поддерево которой содержит m отметок вычисляется по формуле

$$CD(c, m) = \frac{\sum_{i=0}^{m-1} nhyp^{i^{0.20}}}{desc_c}, \tag{1.8}$$

где $nhyp$ — среднее количество потомков в каждом узле, $desc_c$ - количество потомков узла c. Параметр 0.20 используется для сглаживания степени i и подобран на основании экспериментов. Точность, показанная этим методом на различных подмножествах корпуса SemCor с использованием различной длины контекста, колеблется от 40% до 70.1%.

Основным недостатком этого метода является невозможность выбора правильного результата, когда два или более возможных значения попали в наиболее плотное поддерево. Таким образом, увеличение гранулярности значений (раздел 1.2.1) автоматически понизит точность метода. Кроме того, возможен случай, когда выбранное на i-том шаге значение слова w_i, не будет принадлежать наиболее плотным поддеревьям, полученным на последующих шагах, а лучший результат на этих шагах достигается с использованием другого значения слова w_i. Таким образом, появляется некоторая несогласованность в выборе значений.

Дополнительным ограничением для методов, основанных на семантической близости, могут служить синтаксические зависимости. В работе [57] представлен метод, основанный на синтаксических зависимостях между словами и на очень простой мере близости, которая определяет близкие слова, как слова принадлежащие одному синсету WordNet. Эксперименты на основе 100 синтаксических зависимостях, полученных из корпуса Semcor, показали точность 80.3%, измеренную на 15 файлах Semcor. На том же тестовом множестве выбор наиболее частого значения давал точность 75.2%

Нахождение лексических цепей [58] — последовательностей семантически связанных слов — считается очень полезной задачей для обработки естественного языка, особенно для задач реферирования и категоризации текстов, а также для устранения лексической многозначности слов. Лексические цепи не зависят от грамматических структур и могут существовать на больших промежутках текста. В большинстве работ, алгоритм поиска лексических цепей состоит из трех этапов:

1. Выбрать из текста слова-кандидаты, для которых можно посчитать семантическую близость. В большинстве случаев, это слова, относящиеся к одной части речи.

2. Для каждого слова-кандидата и для каждого значения этого слова найти цепь, в которую входит слово в конкретном значении, на основании семантической близости между значением слова-кандидата и концепций, которые уже присутствуют в лексической цепи.

3. Если цепь найдена, добавить слово в цепь; в противном случае создать новую цепь.

В завершении алгоритма, часто делают фильтрацию цепей по некоторому порогу.

В работе [59] предложен алгоритм разрешения лексической многозначности с помощью минимального разбиения на цепи: авторы выбирали в качестве ответов пары значений слов с дистанцией 0 или 1 в иерархии WordNet. Лучшая точность достигалась, если слова цепи были расположены рядом в тексте. Для реализации этой идеи авторы разбивали текст на фрагменты по 15 строк каждый. В последней главе данной работы предлагается метод (раздел 3.4), обобщающий и формализующий данный подход для цепей, состоящих из любого количества слов, где локальность достигается через степень расширенной модели Маркова.

В одной из последних работ Михалсиа [60] предложила алгоритм разрешения лексической многозначности основанный на графе связанных терминов. Граф создавался автоматически с использованием мер близости, описанных выше, а затем осуществлялся выбор наиболее близкого значения с помощью алгоритмов ранжирования вершин, таких как PageRank [61]. Применение этого алгоритма к задаче определения значений всех слов корпуса Senseval-2 дало точность 55.2%.

Устранение лексической многозначности через соединение значений в лексические цепи также описано в работе [62]. Каждая цепь выявляется независимо с использованием скрытой марковской модели. Хотя этот алгоритм и является обобщением предыдущих, основанных на лексических цепях, он показывает точность меньше, чем нижний порог, присваивающий всем словам их наиболее употребляемое значение. Основываясь на этом результате, авторы утверждают, что существует «внутренний конфликт между лексическими цепями и разрешением лексической многозначности». Авторы показали, что из-за разреженности значений в WordNet, лексические цепи не позволяют достичь значимых результатов. Из чего они сделали слишком общее заключение: «не использовать лексические цепи для устранения многозначности».

В данной работе показано, что это утверждение некорректно и предложен метод (раздел 3.4) показывающий точность, значительно превосходящую нижнюю границу. Так же, как и в работе [62], предполагается, что текст состоит из нескольких лексических цепей, но лежащая в основе модель отличается и предполагается более плотная связь между значениями.

Методы, основанные на ограничивающих значения правилах, собирают информацию о возможных отношениях между категориями слов. Например, «EAT-FOOD» и «DRINK-LIQUID» являются такими семантическими ограничениями, которые могут быть использованы для фильтрации значений, не относящихся к контексту.

Хотя, эти методы интуитивно понятны, их сложно применить к задаче устранения многозначности. Основной причиной является необходимость создания больших размеченных значениями корпусов. В работе [63] приведен детальный анализ этого подхода.

Для русского языка также разрабатываются тезаурусы и онтологии [64, 65]. Методы, относящиеся к данному классу и представленные в отечественной литературе, используют тезаурус русского языка РуТез [64]. В работе [66] описана автоматическая процедура выбора значений на основе структуры общественно-политического тезауруса, то есть использующая глобальный контекст, относящийся к конкретной предметной области. Работа [67] содержит описание метода, использующего семантическую близость, вычисленную на базе тезауруса русского языка РуТез.

1.3.3. Методы, основанные на обучении по размеченным корпусам

Задача устранения неоднозначности может быть легко представлена как задача классификации: каждое входное слово $w_i \in T$ текста T необходимо отнести к одному из классов $m_i^j \in M_i$, где M_i — множество значений слова

w_i. Методы, основанные на обучении по размеченным корпусам, занимаются выводом гипотезы h, аппроксимирующей отображение f слов текста во множество их значений.

Успех подходов, основанных на обучении по размеченным корпусам, зависит от доступности больших аннотированных коллекций. Быстрый прогресс в области автоматического определения частей речи и синтаксического разбора был достигнут, в частности, благодаря большим размеченным вручную корпусам, таким как Penn Treebank [41]. Модели, полученные из аннотированных корпусов методами машинного обучения, показывают хорошую производительность во многих задачах обработки естественного языка, а их точность часто превосходит точность алгоритмов, основанных на созданных экспертами правилах.

По сравнению с задачами определения частей речи и синтаксического разбора, задача устранения лексической многозначности связана с дополнительными трудностями. Так как каждое слово ассоциировано с его уникальным значением, для полного обучения алгоритмов потребуется огромное количество примеров. Эта проблема усложняется тем, что в естественных текстах значения слов распределены не по равномерному закону, а по закону Зипфа (Zipf law). Будем ссылаться на эту проблему как на ***проблему разреженности языка***.

Проблему разреженности языка пытаются обойти с помощью выбора признаков, использующихся в обучении алгоритмов. Для обучения алгоритмов разрешения лексической многозначности, обычно использующиеся признаки могут быть скомбинированы в следующие группы:

1. **Локальные признаки.** Признаки, соответствующие локальному контексту, включающие n-граммы частей речи, леммы, словоформы и их расположение относительно целевого слова.

2. **Признаки темы.** Эти признаки представляют собой больший контекст: более широкие окна слов, другие предложения, параграфы или документы.

3. Также для улучшения моделей часто используются **синтаксические зависимости**.

4. **Семантическая близость.** В последнее время, все чаще стали появляться алгоритмы, решающие проблему нехватки данных для обучения использованием семантической близости в качестве признака. Семантическая близость привлекательна тем, что существует возможность посчитать эту величину для всех пар слов в словаре.

Методы, основанные на обучении по размеченным корпусам, можно разделить на два класса: скрытые модели и открытые модели. Открытые модели могут быть разделены на классы в соответствии с предположением о независимости признаков. Логарифмически линейная модель (Log linear models) [68] предполагает, что все признаки условно независимы. Метод максимальной энтропии (Maximum Entropy) [69, 70] и обучение на основе экземпляра (Instance-based Learning) не делают никаких предположений о зависимости признаков. Разложимые модели (Decomposable models) [71] делают заключение о зависимостях на основе тренировочного корпуса. Рассмотрим эти модели подробнее.

Логарифмически линейная модель (LLM). Условная вероятность каждого значения s_i вычисляется с помощью правила Байеса:

$$P(s_i|c_1,\ldots,c_k) = \frac{P(c_1,\ldots,c_n)P(s_i)}{P(c_1,\ldots,c_n)} \ ,$$

где c_j — j-й признак. Так как знаменатель дроби одинаков для всех значений слова, он просто игнорируется. Далее делается предположение об условной

независимости признаков, тогда

$$P(c_1, \ldots, c_n) = \prod_{j=1}^{k} P(c_j|s_i) \ .$$

Таким образом для каждого обучающего примера должно выполняться

$$s = \arg\max_{s_i} \left(logP(s_i) + \sum_{j=1}^{k} logP(c_j|s_i) \right) \ .$$

Вычисляя частоту каждого признака, можно оценить величину $logP(c_j|s_i)$.

Недостатками этого метода являются недостаточно обоснованное предположение об условной независимости, а также необходимость применения специальных алгоритмов сглаживания для оценки вероятности комбинаций признак-значение, не присутствующих в тренировочном множестве.

Разложимая вероятностная модель (DPM). Решение о зависимости признаков принимается на основе тренировочных данных. Обычно в разложимой модели некоторые признаки зависимы, а некоторые нет, что может быть представлено в виде графа зависимостей [72]. Система Grling-Sdm [71], основанная на разложимой модели показала средние результаты на корпусе Senseval, так как для заключения о зависимости или независимости признаков необходимо большее количество тренировочных данных.

Обучение на основе экземпляра (IBL). Эта модель классифицирует новые примеры путем экстраполяции предыдущих, наиболее похожих на данный. Простая мера близости приведена в [73]:

$$\Delta(X, Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i) \ ,$$

где

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{max_i - min_i} & \text{для численных переменных} \\ 1 & \text{если } x_i = y_i \\ 0 & \text{если } x_i \neq y_i \end{cases}$$

Существует еще несколько способов измерения близости между экземплярами.

Так как обучение на основе экземпляра позволяет использовать одновременно численные и символьные признаки, оно позволяет интегрировать различные признаки в одну модель. Стивенсон и Уилкс [74] создали систему устранения лексической многозначности на основе этого метода и получили очень высокие результаты: 90.37%.

Метод максимальной энтропии. В терминах задачи устранения многозначности, метод максимизирует энтропию условной вероятности $P_\lambda(y|x)$ значения y при условии фактов x, где множество фактов выводится из тренировочного множества. Каждый факт представляется как двоичный признак, выраженный через индикаторную функцию:

$$y = \begin{cases} 1 & \text{если значение } y \text{ при условии } x \\ 0 & \text{в остальных случаях} \end{cases}$$

Далее находим

$$P_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right) \quad ,$$

где $Z_\lambda(x)$ — коэффициент нормализации, определяющийся с помощью ограничения $\sum_y P_\lambda(y|x) = 1$ для всех x.

Таким образом, значение слова в каждом примере должно быть.

$$y = \arg\max_y \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

Параметр λ вычисляется на основе тренировочного множества с помощью алгоритма Improved Iterative Scaling [69].

Данг и Палмер [75] применили метод максимальной энтропии к задаче устранения лексической многозначности и получили очень хорошие результаты. Однако метод максимальной энтропии обычно использует большое количество признаков, и поэтому имеет высокую вычислительную сложность.

EM-алгоритм. Метод решает задачу максимизации, содержащую скрытую информацию, итеративным способом. Относительно задачи разрешения лексической многозначности под скрытой информацией имеются ввиду контекстные признаки, напрямую не ассоциированные со значениями слов. Например, если взять два параллельных текста на разных языках, и ассоциировать каждое слово с его значением на одном из языков, тогда это значение будет скрытой переменной.

Обозначим тройку (W_{L_1}, W_{L_2}, S) как X, где $W_{L_{1,2}}$ — слова различных языках, S — их значение. Тогда задача сводится к максимизации вероятности $P(X|Y, \Theta)$.

$$X(W_{L_1}, W_{L_2}, S) = \arg\max_T P(X|Y, \Theta) \ ,$$

где

$$\Theta = p(T), p(W_{L_1}|T), p(W_{L_2}|T)$$

EM-алгоритм состоит из двух этапов, E-шаг (expectation) и M-шаг (maximisation), и оценивает параметр Θ из тренировочных данных. Достижение глобального максимума зависит от начальных значений, поэтому следует осторожно инициализировать начальные параметры. Часто для этого используют статистику, полученную из словарей.

EM-алгоритм может быть обучен на двуязычном корпусе и не требует ручной разметки. При этом он сохраняет высокую производительность. В работе [76] точность и полнота, показанные предложенной реализацией, достигают 67.2% и 65.1% соответственно.

Для обучения алгоритмов разрешения лексической многозначности русского языка используется Национальный корпус русского языка (НКРЯ) [77]. Работы, представленные в отечественной литературе и использующие НКРЯ уделяют особое внимание семантической многозначности [78]. Автоматическое разрешение семантической неоднозначности в Национальном корпусе

русского языка производится с помощью глубинных фильтров — на основе глобальных правил сочетаемости семантических классов, например, «названия одежды не могут употребляться в роли субъекта при глаголах эмоции» [79]. В работе [80] представлен метод использующий поверхностные фильтры, работающие на базе устойчивых коллокаций, включающих данные (1) о лемме, (2) о частеречных, (3) словоклассифицирующих и (4) словоизменительных признаках составляющих, (5) об их исходной семантической разметке, а также (6) о некоторых грамматических и лексико-семантических характеристиках ближайшего контекста (например, "родительный падеж"для оборота типа кого-чего-л.). Метод представленный в работе [81] комбинирует автоматическое построение БД высокочастотных устойчивых коллокаций с их последующим (полу)ручным аннотированием.

Также внимание уделяется многозначности глаголов [82] и прилагательных [83]. Так в работе [82] описывается метод разрешения семантической неоднозначности глаголов с использованием информации об их моделях управления, извлеченной из толковых словарей, а также из специализированного словаря глагольного управления. В работе [83] для разрешения многозначности имен прилагательных используются поверхностные фильтры.

1.3.4. Методы, основанные на обучении по неразмеченным корпусам

Существует еще один тип алгоритмов снятия лексической многозначности — методы основанные на обучении по неразмеченным корпусам (см. [10]). Основной предпосылкой для их разработки послужила трудность создания размеченных корпусов, семантических сетей и других необходимых ресурсов.

Существует два альтернативных подхода к созданию алгоритмов этого класса:

1. **Дистрибутивный подход**, разделяющий значения слов на основе предположения, что слова, встречающиеся в одинаковом контексте, имеют одинаковые значения.

2. **Подход эквивалентного перевода**, который использует перевод слов на некоторый язык, зависящий от значения слова на исходном языке. Эти зависящие от значения переводы, могут быть использованы как словарь значений для исходного языка.

Оба этих подхода не зависят от внешних источников знаний, так как не требуют ресурсов, таких как аннотированные одноязычные корпусы или выровненные по словам параллельные тексты.

Дистрибутивные методы не присваивают значение слову, и только позволяют сделать различие между значениями путем установления кластеров с одинаковым контекстом, где каждый кластер показывает, что слово в нем было использовано в конкретном частном значении.

Методы, основанные на эквивалентном переводе, используют тот факт, что различные значения слов на одном языке могут быть переведены в абсолютно разные слова на другом языке. Этот подход имеет два привлекательных свойства. Первое, такие методы автоматически выводят словарь значений с гранулярностью, подходящей для машинного перевода. Второе, на основании работы таких методов может быть создан размеченный корпус, который может быть использован, как тренировочный для традиционных методов, основанных на обучении с учителем.

Хотя, методы, основанные на неразмеченных корпусах, очень привлекательны с точки зрения трудозатрат на их создание и поддержку, лучшие из них показывают слишком малую точность — меньше чем выбор наиболее вероятного значения.

1.4. Выводы к первой главе

В данной главе рассмотрено понятие многозначности или полисемии, описаны основные типы многозначности и приводится детальное описание задачи устранения лексической многозначности.

Задача автоматического устранения лексической многозначности существует более 60 лет, однако до сих пор остается множество открытых вопросов. Основные проблемы перечислены ниже:

- До сих пор не существует строгого определения понятия «значение слова». В большинстве методов значения берутся из словарей. Однако словари могут различаться по количеству значений и степени гранулярности. Необходимая степень гранулярности зависит от задачи.

- Исследователи выделяют несколько типов контекстов.

  - микро-контекст (несколько слов в ближайшем окружении целевого слова),
  - тематический контекст (несколько предложений вокруг целевого слова), и
  - контекст, определяемый областью знаний, для которой решается задача снятия многозначности,

  Наилучшие результаты дают методы, использующие комбинацию этих типов. Несмотря на это, не существует работ, оценивающих вклад каждого типа контекста в решение задачи разрешения лексической многозначности.

- Третьей проблемой является поиск способа оценки производительности алгоритмов. Для оценки алгоритмов создано несколько корпусов, наи-

более известные из которых Senseval-1,2,3. Однако возможность применения этих корпусов зависит от словаря, на котором основан метод.

Из множества всех существующих алгоритмов можно выделить два доминирующих класса.

- алгоритмы разрешения лексической многозначности на основе внешних источников знаний (knowledge-based methods) могут быть легко адаптированы к документам, полученным из любых источников и не привязаны к конкретному языку.

- методы, основанные на машинном обучении показывают лучшие результаты из всех алгоритмов, представленных в современной литературе, однако требуют обучения на документах похожих на обрабатываемые в дальнейшем. Это связано с проблемой разреженности языка.

Большинство алгоритмов основано на лексико-семантических ресурсах, созданных экспертами. Однако эти ресурсы содержат далеко не все слова и их значения, употребляемые в языке, а их создание требует огромных трудозатрат. Одним способом решения этой проблемы является автоматический вывод значений на основе анализа неразмеченных текстовых коллекций. Однако на данный момент эти методы обладают слишком малой точностью. Альтернативным решением является использование сетей документов, созданных огромным числом независимых пользователей и описывающих большинство понятий реального мира.

В следующей главе будет дано определение сетей документов и показано, как их можно использовать для устранения лексической многозначности.

Глава 2

Вычисление семантической близости в сетях документов

В данной главе вводится понятие сети документов, рассматриваются отличия сетей документов, от баз знаний, созданных экспертами и описываются методы вычисления семантической близости в сетях документов. Наиболее используемой в качестве ресурса для обработки естественного языка сетью является Википедия, так как она относительно легко поддается машинной обработке. Поэтому в данной работе особое внимание уделяется этой открытой энциклопедии, вычислению семантической близости между значениями ее терминов и применение вычисленной семантической близости к задаче устранения лексической многозначности.

2.1. Сети документов

Понятие ***сети документов*** (document network) было впервые предложено Филиппо Менцером в 2004 году в работе [84]. ***Сеть документов*** — это случайный граф, вершинами которого являются текстовые документы, а ребрами — гипертекстовые ссылки между ними. Примерами таких сетей служат Веб, Википедия, сети цитирования в научной литературе, сети соавторства и т. д.

Сети документов появляются очень быстро и по размерам на порядки превышают любые базы знаний, созданные экспертами для применения в компьютерной лингвистике. В последнее время исследователи в области обработки естественного языка обратили внимание на такие сети документов, как Web и Wikipedia. Было предложено несколько подходов по использова-

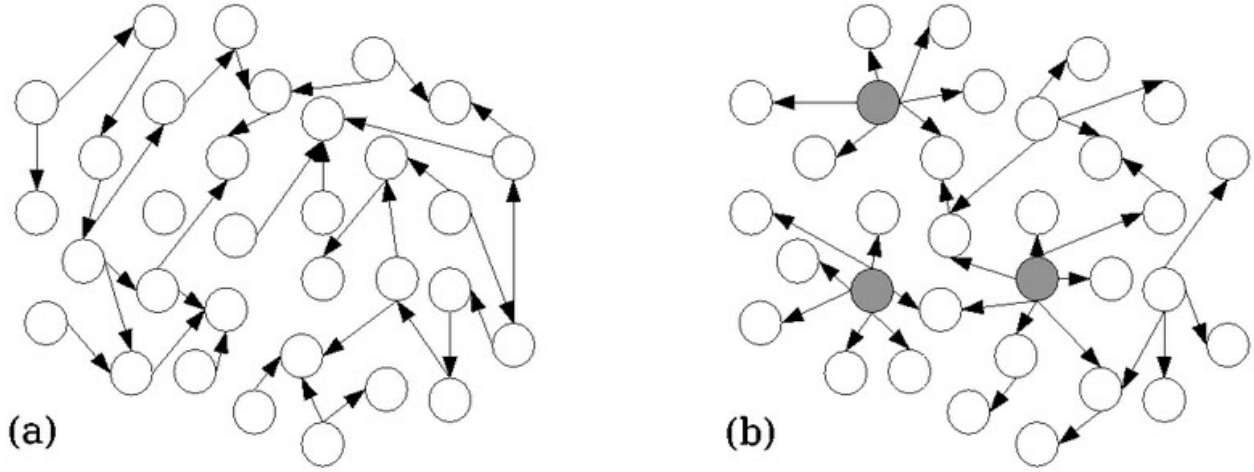

Рис. 2.1. (a) Случайный граф и (b) Безмасштабный граф

нию Web, как корпуса для методов компьютерной лингвистики [85]. Однако в данное время большей популярностью пользуется Википедия, так как она обладает более четкой структурой и проще поддается машинной обработке.

Сети документов являются частным случаем ***безмасштабных графов*** (scale-free graph). Безмасштабные графы — это класс графов, с распределением степеней узлов, отвечающим степенному закону, то есть доля $P(k)$ вершин в сети, имеющих k связей с другими узлами описывается законом $P(k) \sim k^{-\gamma}$, где γ — константа, для большинства реальных сетей находящаяся в интервале $2 < \gamma < 3$. История безмасштабных графов начинается с работы Барабаси и Альберта [86]. Они обнаружили, что множество случайных графов в реальном мире не удовлетворяют классической теории случайных графов (рис. 2.1b), предложенной Эрдосом и Реньи [87], так как свойства безмасштабных графов сильно отличаются от свойств классических случайных графов. Рассмотрим эти различия подробнее.

Распределение степеней вершин безмасштабных графов подчинено степенному закону, что означает присутствие узлов с очень большой степенью по сравнению со средним значением степеней (***хабов***) и длинного «хвоста» из вершин с малой степенью.

Важной характеристикой безмасштабных графов является коэффициент кластеризации (вероятность того, что два соседа одной вершины, также соединены между собой). Коэффициент кластеризации распределен по степенному закону и уменьшается с ростом степени вершин. Это означает, что вершины с малой степенью принадлежат очень плотным подграфам, а связь между этими кластерами осуществляется через хабы.

Очевидно, что для такой топологии характерно определяющее значение хабов. Так, безмасштабные графы устойчивы к удалению случайных узлов: для уничтожения наибольшей связной компоненты путем удаления случайных узлов, требуется удалить 20-30% всех узлов [88]. Однако граф уязвим для целенаправленной атаке на хабы, то есть граф быстро распадается после удаления нескольких процентов узлов с максимальной степенью. Это свойство имеет множество практических применений. В случае компьютерных сетей оно используется при атаках на сети и, с другой стороны, для обеспечения их безопасности.

Как следствие, характерной особенностью является эффект суммирования шумов (funneling effect), когда большая часть кратчайших путей от заданной вершины проходит через одного наиболее значимого соседа этой вершины [89, 90]. Такая топология позволяет производить быстрый эвристический поиск [91]. Также, на этом свойстве основана эвристика, предложенная для вычисления семантической близости и описанная ниже (раздел 2.3.1).

Относительно диаметра безмасштабных графов интересное исследование провели Коэн и Хавлин [92]. Они показали, что специально построенный граф с распределением степеней вершин вида $P(k) \sim k^{-\gamma}$, $2 < \gamma < 3$, где узлы высокой степени в первую очередь соединялись с другими узлами высокой степени, имеет диаметр $O(\log \log N)$, где N — общее число вершин. Затем, опираясь на вероятностные рассуждения, авторы показали, что такой диаметр характерен для большинства случайных графов со степенным рас-

пределением степеней вершин. Таким образом, для безмасштабных графов диаметр можно принимать за константу, независящую от числа вершин.

В названии «безмасштабные» заключено еще одно свойство — самопохожесть этих графов. То есть свойства не зависят от размера сети. Например, граф Википедии, являясь подграфом безмасштабного графа Веба, также является безмасштабным графом [93].

Основным подходом, к исследованию свойств этого класса графов является моделирование процесса их роста. Наиболее распространенной гипотезой, описывающей процесс образования таких сетей, является ***предпочтительное соединение*** (preferential attachment): в растущем графе вероятность появления новой связи у вершины пропорциональна ее текущей степени. Однако она не описывает свойства всех графов, встречающихся в реальном мире. Поэтому существует еще несколько альтернативных подходов, для описания процесса образования безмасшабных графов.

Филиппо Менцер показал, что процесс образования графов, где узлами служат текстовые документу лучше описывается гипотезой, что «документы с похожим содержимым имеют тенденцию ссылаться друг на друга». Таким образом, появляется основание для использования ссылочной структуры сетей документов, для определения коэффициента семантической близости между этими документами.

2.2. Семантическая близость в сетях документов

Методы вычисления семантической близости между документами можно разделить на два широких класса:

- методы, использующие контент или текстовое содержимое документа и представляющие каждый документ в виде множества элементов или вектора весов слов; и

- методы, основанные на ссылочной структуре сети, вычисляющие отношения между объектами в терминах ссылок между документами.

В работе [84] было показано, что вероятность возникновения ссылки между документами коррелирует с близостью документов, вычисленной с помощью мер из первого класса.

Методы, использующие текстовое содержимое документа, хорошо изучены и используются в информационном поиске. Наиболее распространенной является векторная модель в комбинации с весовой схемой TF-IDF, где каждый документ в коллекции представляется в виде векторов, состоящих из весов терминов, а близость между документами определяется как косинус между этими векторами. Веса терминов вычисляются на основе частоты их употребления в каждом документе в отдельности и в коллекции в целом. Хороший обзор и сравнение этих методов приведен в статье Зобеля и Моффата [94]. Пример применения этих моделей в сетях документов можно найти в [95].

Одной из наиболее перспективных моделей для вычисления близости слов в коллекциях неразмеченных документов является латентно-семантический анализ (Latent semantic analysis) [96]. Для каждого слова вычисляется частота встречаемости в каждом из документов. Полученная матрица нормализуется и к ней применяется сингулярное разложение, чтобы получить приближенную матрицу меньшей размерности. В получившейся матрице слова представляют собой векторы; близость между словами вычисляется как угол между соответствующими векторами. Латентно-семантический анализ основан на математической технике разложения матрицы на сомножители, требующей большое количество ресурсов. Это служит главной причиной того, что этот метод до сих пор не применяется к сетям документов.

В данной работе основное внимание уделяется методам, основанным на ссылочной структуре сети. Последние исследования по оценке различных мер семантической близости [97, 98] показали, что меры, основанные на ссылочной структуре, позволяют достичь лучшей корреляции с оценками, произведенными экспертами, по сравнению с текстовыми методами. В качестве примера стоит упомянуть успех поисковой системы Google, которого она достигла благодаря способности ранжировать результаты поисковых запросов в соответствии с ожиданиями пользователей. А в основе этого метода ранжирования лежал алгоритм PageRank, основанный на ссылочной структуре Веба [61].

Методы, основанные на ссылочной структуре, можно в свою очередь разделить на два подкласса: локальные и глобальные методы. Локальные методы вычисляют близость между любой парой вершин, основываясь на локальной информации и не затрагивая большинство других вершин. Глобальные методы одновременно вычисляют близость между всеми вершинами графа.

2.2.1. Локальные методы

Самыми простыми из локальных методов могут считаться алгоритмы, использующие длину минимального пути между вершинами. Однако для топологии безмасштабных графов эти методы не позволяют получить приемлемых результатов. Из-за малого диаметра графов почти для всех вершин максимальная длина минимального пути не превышает некоторой небольшой константы ($c \cdot \log \log N$), поэтому становится невозможным отличить близость одной пары вершин от другой. Более того, из-за эффекта суммирования шумов промежуточные вершины в кратчайших путях часто совпадают.

Широко используемыми методами являются методы, основанные на близости между множествами ближайших соседей каждого из узлов 2.2. Самой

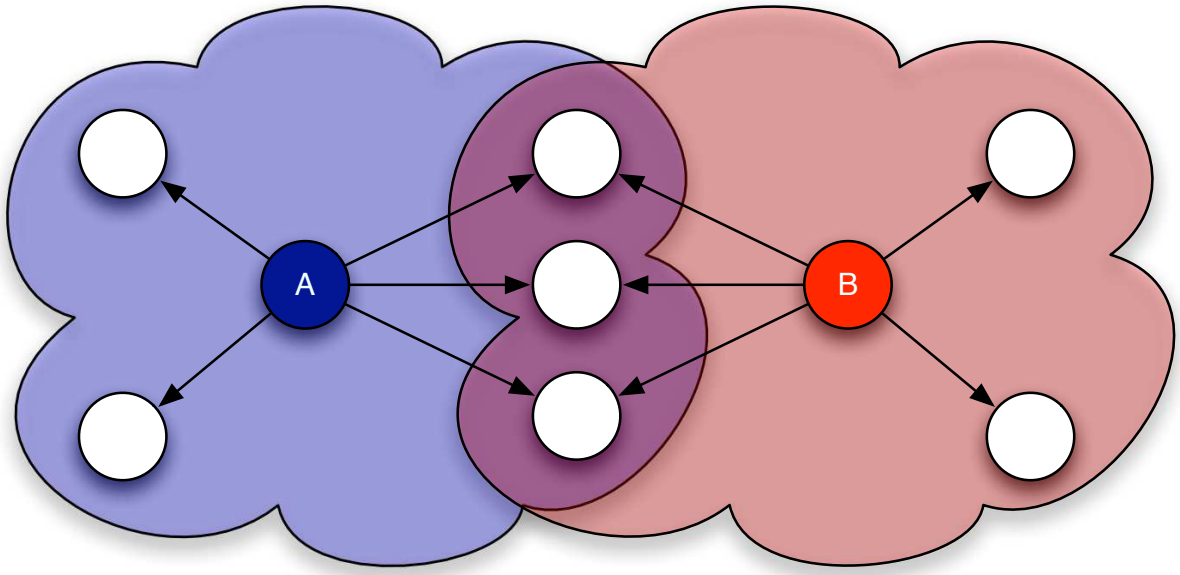

Рис. 2.2. Множества ближайших соседей узлов

простой из этих мер является косинус. Обозначим как $N(a)$ множество ближайших соседей узла a — узлов графа, напрямую связанных ребром с узлом a. Тогда мера близости определяется как

$$sim_{cos}(a,b) = \frac{|N(a) \cap N(b)|}{\sqrt{|N(a)|^2 + |N(b)|^2}}$$

Эта мера использовалась для вычисления близости между значениями терминов Википедии в работе [99].

В работе [100] используется коэффициент Жаккара:

$$sim_{Jaccard}(a,b) = \frac{|N(a) \cap N(b)|}{|N(a) \cup N(b)|}$$

Аналогичным образом определяется коэффициент Дайса:

$$sim_{Dice}(a,b) = \frac{2\,|N(a) \cap N(b)|}{|N(a)| + |N(b)|}$$

В одной из последних работ [101] используется мера Google Distance:

$$sim_{GD}(a,b) = \frac{\log(\max(|N(a)|, |N(b)|)) - \log(|N(a) \cap N(b)|)}{\log(|W|) - \log(\min(|N(a)|, |N(b)|))},$$

где W — число вершин графа.

Для направленных графов наиболее часто используются меры ***со-цитирования*** (co-citation) и ***библиографического сочетания*** (bibliographic coupling) [102]. Как следует из названия, эти меры применяются для графов цитирования, однако они могут быть использованы для нахождения близости между узлами в любых направленных графах. В случае первой меры, близость между двумя узлами p и q основана на количестве статей, одновременно ссылающихся на p и q. Для библиографического сочетания, мера близости базируется на количестве узлов, на которые одновременно ссылаются p и q.

2.2.2. Глобальные методы

Наиболее известные методы второго подкласса основаны на модели случайного блуждания (random walk). Существует несколько работ, пытающихся адаптировать методы ранжирования, основанные на случайном блуждании, такие как PageRank, для вычисления семантической близости.

В работе [103] предложена мера близости, основанная на распространении значения PageRank. Если обозначить как $PG(u, v)$ величину PageRank, которую узел u передает узлу v и которая вычисляется как

$$PG(u,v) = \begin{cases} \sum_{p \in PATH(u,v)} \frac{d \cdot PR(u)}{\prod_{w \in p, w \neq v | O(w)|}} & \text{, если } u \neq v, \\ PR(u) & \text{, если } u = v, \end{cases}$$

то близость между двумя вершинами вычисляется как

$$PS(u,v) = \sum_{i=1}^{n} \frac{\min(PG(v_i,u), PG(v_i,v))^2}{\max PG(v_i,u), PG(v_i,v)} \ ,$$

где $PR(u)$ — значение PageRank в узле u, $O(w)$ — ребра исходящие из узла w, $PATH(u, v)$ — множество всевозможных путей из вершины u в вершину v. Сложность вычисления этой меры определяется как $O(d^{2r})$, где d — средняя степень узлов, а r — константа, определяющая радиус распространения.

Также близость между вершинами графа может вычисляться на основе персонализированного PageRank. Авторы работы [104] предложили использовать локальную близость между узлами [95] для создания векторов переходов для каждого узла, а после этого использовать персонализированный PageRank для вычисления стационарных распределений в соответствующих узлах, с использованием соответствующих векторов переходов. Семантическая близость в этой работе вычисляется как косинус между векторами распределений узлов. Авторы добились несколько лучших результатов, чем были показаны локальной мерой близости [95].

Наиболее теоретически обоснованной является мера SimRank [105], использующая наработки предыдущих методов. SimRank основан на простом, интуитивно понятном предположении: *Два объекта похожи, если на них ссылаются похожие объекты.* Заметим, что это предположение рекурсивно. В качестве базы рекурсии предполагается, что любой объект максимально похож сам на себя.

Для графа $G(V, E)$, состоящего из множества узлов V и множества ребер E, для каждого узла v, обозначим как $I(v)$ множество вершин графа, которые ссылаются на узел v. Тогда SimRank определяется как:

$$s(a, a) = 1, s(a, b) = \frac{C}{|I(a)|\,|I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \ ,$$

где C — коэффициент затухания, $0 < C < 1$.

SimRank вычисляется итеративно. Для получения приемлемых результатов в оригинальной статье [105] предлагалось сделать не менее 5 итераций при значении коэффициента затухания $C = 0.8$. Однако в статье [3] было показано, что для большинства приложений этого будет недостаточно, и предложена оценка погрешности

$$s(a, b) - R_j(a, b) \leq C^{k+1} \ ,$$

где $R_k(a,b)$ — приближенное значение меры, полученной на k-й итерации.

Хотя рекурсивные методы теоретически позволяют достичь хороших результатов, их применение для вычисления близости узлов в реальных сетях связано с большими трудностями из-за:

1. сложности вычисления алгоритмов;

2. трудностей при организации хранения полученной информации и доступа к ней.

В статье [105] показано, что вычисление одной итерации SimRank требует $O(N^4)$ операций. В работе [3] предложены оптимизации, уменьшающие сложность вычисления до $O(\min(NL, \frac{N^3}{\log_2 N}))$, где L — число ребер в графе. Несмотря на это, вычислить SimRank, например, для графа Википедии, содержащего 3,000,000 узлов, на современном настольном компьютере не предоставляется возможным.

Если для связного графа с N вершинами и диаметром d количество итераций K удовлетворяет уравнению $2 < d \leq K$, тогда, обозначив $D = \lceil d \rceil$, можно получить оценку для числа ненулевых значений S итеративной функции $R_K(*,*)$, полученных после K итераций[3]:

$$\zeta \geq \frac{6((D+1)(N-1) - dN)^2}{D(D+1)(2D+1)} \ .$$

Таким образом для связного графа с диаметром $d < 5$ число ненулевых значений близости после пяти итераций будет $\zeta \geq \frac{(N-6)^2}{55}$. Для графа с 3,000,000 вершин и диаметром $d = \log\log N < 5$ SimRank после 5 пяти итераций выдаст $163 * 10^9$ ненулевых значений похожести. Даже при условии использования 8 байт на каждое значение, для их хранения потребуется более терабайта. Для такого объема информации, организация эффективного доступа становится отдельной трудоемкой задачей.

2.3. Википедия

Википедия — это открытая энциклопедия, создающаяся пользователями Веба. Сейчас Википедия содержит более 3 млн. статей, не считая специальные страницы. Граф Википедии (вершины графа — это страницы, а ребра — гипертекстовые ссылки между страницами) имеет свойства сети документов [93]. Кроме того, Википедия содержит огромное количество дополнительной информации, которая используется для различных исследований. Рассмотрим структуру открытой энциклопедии более подробно.

Каждая статья Википедии имеет заголовок, состоящий из одного или нескольких слов, и тела, описывающего значение ***термина***-заголовка (***концепцию***). Все статьи Википедии связаны гипертекстовыми ссылками и образуют граф, обладающий свойствами сети документов [93]. При этом, в соответствии с правилами редактирования статей[1], авторы могут ссылаться на только на статьи, релевантные текущей.

Структурно ссылки состоят из двух основных частей: концепции энциклопедии, на которую указывает ссылка, и термина, который видит пользователь. Например, при использовании ссылки [[Platform (computing)|Platform]], пользователь видит многозначный термин «Platform», а ссылка указывает на конкретное значение «Platform (computing)». Значимость ссылок зависит от места статьи, в котором они стоят. Например ссылки в разделе «Смотри также» или «See also», указывают на концепции, наиболее релевантные данной.

Кроме того, существуют специальные страницы для определения синонимов (страницы переадресации) и списков значений многозначных слов. Таким образом, можно автоматически создать словарь терминов и определить их возможные значения. При этом, вычислив семантическую близость между статьями Википедии, можно оценить отношения между значениями терми-

[1] $http://en.wikipedia.org/wiki/Wikipedia:Manual\_of\_Style\#Links$

	Википедия Октябрь '08	Википедия Март '09	WordNet
Количество концепций	2 500 000	2 800 000	150 000
Многозначных репрезентаций	260 000	290 000	26 000
Среднее количество значений	5.4	5.47	2.95

Таблица 2.1. Statistics for Wikipedia and WordNet

нов. Заметим, что среднее количество значений терминов намного превосходит аналогичное количество в тезаурусе WordNet (табл. 2.1).

Каждая статья Википедии принадлежит одной или нескольким категориям. Сами категории также могут принадлежать более общим категориям. Граф категорий, хотя и обладает иерархией, не образует таксономию, так как может содержать циклы. Более того, он обладает свойствами [106] безмасштабных графов, то есть имеет высокий коэффициент кластеризации, а степени узлов подчиняются степенному закону.

Кроме того, Википедия содержит еще много дополнительных структур (шаблоны, инфобоксы, списки и т.д.), в дальнейшем они будут использоваться для определения типа ссылок и подсчета семантической близости между статьями Википедии.

Исследование журнала Nature показало, что точность Википедии сравнима с точностью самой большой энциклопедии, созданной экспертами — Британики [107]. Проверка 42 случайных статей экспертами выявила по 4 серьезные ошибки в обеих энциклопедиях; в среднем статья Википедии содержала четыре неточности, а Британики — три. При этом, размер Википедии

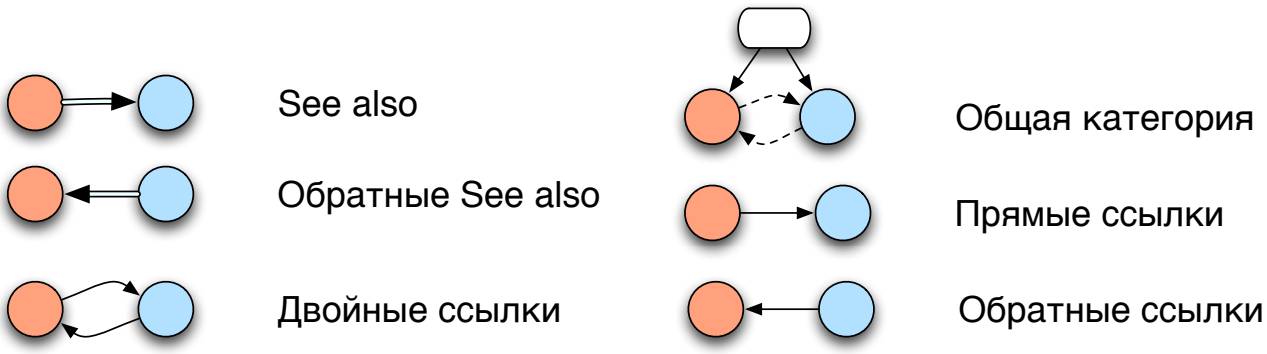

Рис. 2.3. Типы ссылок

намного превосходит размер Британики.

2.3.1. Вычисление семантической близости между статьями Википедии

Для вычисления семантической близости между концепциями Википедии использовались меры, основанные на нормализованном количестве ближайших соседей, так как они позволяют одновременно получить хорошие результаты и приемлемую скорость вычисления. Мы исследовали структуру Википедии и заметили, что некоторые типы ссылок чрезвычайно релевантны по отношению к семантической близости, в то время как другие могут привести к неверным результатам. Поэтому, в дополнение к основной мере, вводится схема весов, основанная на типах ссылок (рис. 2.3):

- Ссылки «See also». В большинстве статей Википедии есть секция «Смотреть также», которая содержит список похожих статей. Этими ссылками авторы страницы предлагают читателям ознакомится с концепциями, близкими к данной; таким образом, эти ссылки неявно означают, что страницы, на которые они указывают, семантически очень близки к данной странице, поэтому им присваивается наибольший вес. Если на статью ссылается другая статья из секции «see also», то такая входящая ссылка содержит не менее важную информацию и поэтому тоже

должна иметь большой вес при вычислении близости.

- Двойные ссылки. Статьи, которые ссылаются друг на друга прямыми ссылками, в большинстве случаев довольно близки по смыслу, поэтому их вес идет следующим в нашей схеме весов.

- Ссылки между статьями с общей категорией. Википедия имеет богатую структуру категорий, а статьи, находящиеся в одной категории, семантически близки. Однако некоторые категории охватывают очень широкие области и состоят из статей, мало связанных друг с другом. Так, в категорию «Living people» входят все статьи о людях, живущих в настоящее время. Поэтому в качестве следующего наиболее значимого типа связей выделяется связь между статьями, которые одновременно имеют ссылку с одной на другую и находятся в одной категории.

- Свойствами категорий также обладают страницы содержащие списки статей «List of ...» (например, List of Hindu gurus), некоторые порталы и шаблоны («Template:Capitals in Europe» и т.п.).

- Википедия обладает огромным количеством статей, описывающих события, произошедшие в ту или иную дату. Однако для большинства приложений, ссылочная структура, предоставляемая этим типом статей, является бесполезной. Например, статя «(Люди) родившиеся в 1905 году» может быть полезна только для создания узконаправленных приложений. Поэтому, в нашем случае, связи через даты при вычислении семантической близости не используются.

- Инфобоксы — это специальные шаблоны, зависящие от типа статьи. Например, для стран используется шаблон «государство», в котором описываются географические, политические, экономические и демогра-

See Also	5	Двойные ссылки	2
Обратные See Also	2	Общая категория	1.5
Обычные ссылки	1	Обратные обычные ссылки	0.5
Даты	0	Ссылки в инфобоксах	1

Таблица 2.2. Вес различных типов ссылок

фические особенности каждого государства. Обычно инфобоксы располагаются наверху страницы и содержат чрезвычайно релевантную информацию. Однако часто в них появляются нерелевантные ссылки, например при описании ВВП государств значения даются в американских долларах и ставится ссылка на статью про американский доллар, что для большинства государств является мало релевантной связью. Можно сделать вывод, что из инфобоксов можно извлечь много полезной информации, однако для этого потребуется их обрабатывать специальным образом. В данный момент, специальная обработка не производится, поэтому ссылки в инфобоксах приравниваются к обычным ссылкам.

В наших экспериментах использовалась схема весов, приведенная в таблице 2.3. Эти веса подобраны вручную экспертами из Hewlett-Packard Labs, во время совместного проекта по созданию системы интеллектуального анализа текстов.

Наконец, для нормализации весов, вес каждой ссылки делится на сумму весов всех ссылок узла, соответствующего статье, в которой содержится ссылка.

В качестве мер близости использовались косинус, коэффициенты Дайса и Жаккара и Google Distance (секция 2.2.1). Для реализации теоретико-множественных операций используется теория нечетких множеств [108], где вес ссылки характеризует степень принадлежности ссылки к нечеткому множе-

ству.

В общем виде операция пересечения нечетких множеств определяется следующим образом

$$\mu_{A\cap B}(x) = T(\mu_A(x), \mu_B(x)) \ ,$$

где функция T — это так называемая T-норма. Операция объединения нечетких множеств определяется как

$$\mu_{A\cup B}(x) = S(\mu_A(x), \mu_B(x)) \ ,$$

где функция S — S-норма (T-конорма). Ниже приведены способы реализации T-нормы и S-нормы, использующиеся в данной работе:

1.

$$\begin{aligned} \mu_{A\cap B}(x) &= \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \\ \mu_{A\cup B}(x) &= \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \end{aligned} \tag{2.1}$$

2.

$$\begin{aligned} \mu_{A\cap B}(x) &= \mu_A(x)\mu_B(x) \\ \mu_{A\cup B}(x) &= \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \end{aligned} \tag{2.2}$$

3.

$$\begin{aligned} \mu_{A\cap B}(x) &= \max\{0, \mu_A(x) + \mu_B(x) - 1\} \\ \mu_{A\cup B}(x) &= \min\{1, \mu_A(x) + \mu_B(x)\} \end{aligned} \tag{2.3}$$

See Also	5	Двойные ссылки	2
Обратные See Also	2	Общая категория	1.5
Обычные ссылки	1	Обратные обычные ссылки	0.5
Даты	0	Ссылки в инфобоксах	1

Таблица 2.3. Вес различных типов ссылок

4.

$$\mu_{A\cap B}(x) = \begin{cases} \mu_A(x), & \mu_B(x) = 1 \\ \mu_B(x), & \mu_A(x) = 1 \\ 0, & \mu_A(x) < 1, \mu_B(x) < 1. \end{cases} \tag{2.4}$$

$$\mu_{A\cup B}(x) = \begin{cases} \mu_A(x), & \mu_B(x) = 0 \\ \mu_B(x), & \mu_A(x) = 0 \\ 1, & \mu_A(x) > 0, \mu_B(x) > 0. \end{cases}$$

Кроме этих мер была введена еще одна, определяемая как

$$sim(A,B) = \frac{\sum\limits_{N\in n(A)\cap n(B)} [w(A,N) + w(B,N)]}{\sum\limits_{N\in n(A)} w(A,N) + \sum\limits_{N\in n(B)} w(B,N)}\ , \tag{2.5}$$

Так как основной задачей работы является разработка алгоритмов устранения неоднозначности, сравнение мер семантической близости производилось косвенно, через алгоритм разрешения лексической многозначности, описанный в разделе 3.2. Результаты работы этого алгоритма с использованием различных мер приведены в разделе 3.2.3 в таблицах 3.2, 3.3 и 3.4. Наибольшую точность показала мера, задаваемая формулой 2.5.

2.3.2. Обработка Википедии

Существует несколько проектов, занимающихся обработкой Википедии с целью извлечения структурированной информации, пригодной для дальней-

шего использования.

Первым таким проектом был Yago [109]. Задачей этого проекта было построение онтологии, содержащей информацию о людях, организациях и городах и факты об этих сущностях. Информация извлекалась из Википедии и структурировалась с помощью WordNet.

В данный момент наиболее известным проектом по извлечению структурированной информации из Википедии является DBpedia [110]. Информация в DBpedia представляется в виде RDF-троек (Resource Description Framework). На ноябрь 2008 года DBpedia содержала около 274 миллиона таких RDF-троек, содержащих информацию о 213,000 людей, 328,000 мест, 57,000 музыкальных альбомах, 36,000 фильмов, 20,000 компаний и т. д. Также в DBpedia содержится информация о статьях Википедии и ссылках между ними на 30 языках.

Самым новым проектом является Wikipedia-Miner [111]. Wikipedia-Miner использует структуру и статьи Википедии для построения систем анализа текста. Для экстракции информации был разработан набор скриптов, написанных на языке Perl. На основе извлеченной с помощью Wikipedia-Miner была создана система для обогащения текстов ссылками на статьи Википедии [101].

В рамках данной работы использовался собственный анализатор Википедии. Это обусловлено тем, что на момент начала данной работы, описанные проекты только начали развиваться, и не позволяли получить всю необходимую информацию. Кроме того, ни один из описанных проектов не позволяет получить тип ссылки, а это является обязательным для вычисления меры семантической близости, описанной выше.

На данный момент Википедия содержит статьи, описывающие более 3 миллионов концепций. Названия статей используются для создания словарей, которые применяются для поиска однозначных и многозначных терминов в

текстах. После того, как все термины в тексте будут найдены, для устранения многозначности используется мера близости.

Первым этапом является очистка Википедии.

1. Ссылки через страницы перенаправления преобразовываются в прямые ссылки на целевые статьи. Например, на рисунке 2.4 ссылка со статьи «Bb (sport)» (id 3) на редирект «F» (id 7) исчезает и появляется ссылка на статью «A» (id 1).

2. Фильтруются циклические ссылки со специальных страниц (id 21 → id 22, id 17 → id 18)

3. Убираются висячие ссылки (id 5, id 17 → id 22)

Для формирования словаря однозначных терминов берутся названия всех статей, описывающих соответствующие концепции и названия всех страниц переадресации на эти статьи. Основным источником многозначных терминов является категория «Disambiguation pages». Статьи, входящие в эту категорию, содержат списки возможных значений и ссылки на страницы, описывающие эти значения. Однако в Википедии не существует четких правил для создания таких страниц, поэтому часто они содержат много лишних ссылок, напрямую не связанных с многозначной концепцией. Поэтому при обработке этих страниц выделяются только те значения, которые содержат в своем названии словоформы многозначной концепции, или для которых она является акронимом. Эта эвристика очень жесткая и отсеивает много хороших значений (мы добавляем их позднее при анализе ссылок). Например, для термина «NATO» будет найдено значение «North Atlantic Treaty Organisation» но пропущено значение «Mora (plant)» – растение, которое часто называют «нато».

Система обработки текстов заранее не имеет информации, какие тексты придется анализировать, следовательно, не должна быть чувствительна к ре-

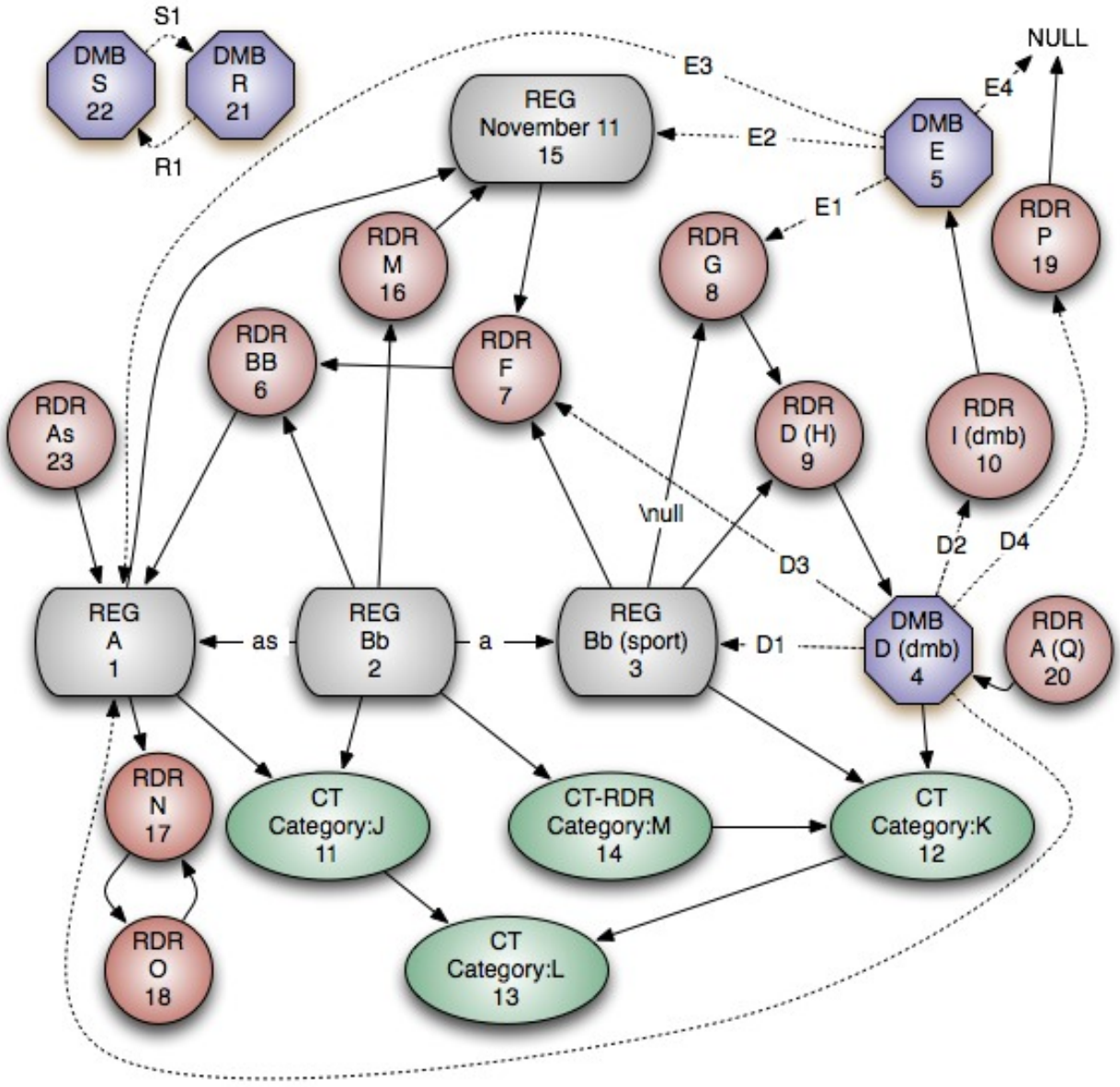

Рис. 2.4. Модельный граф Википедии. Каждая вершина содержит тип, имя и идентификационный номер.

гистру слов. Поэтому все слова в словаре приводятся к верхнему регистру. Википедия, напротив, чувствительна к регистру, и один набор слов, написанные в разном регистре, может указывать на различные концепции («Algol» — звезда в созвездии Персея и «ALGOL» — название языка программирования). Для решения этой проблемы, такие наборы слов добавляются в словарь многозначных терминов, и значение выбирается на этапе анализа текста.

Кроме того, большое количество терминов содержит в названии уточняющие концепции, например «Platform (computing)». Анализатор убирает текст в скобках и в случае коллизий применяется подход, аналогичный тому, который используется при приведении к одному регистру.

Источником значений многозначных терминов могут быть как словари, так и способы употребления терминов в текстах. В нашем случае дополнительным источником значений являются ссылки между страницами. Любая ссылка содержит две части: текст который видит пользователь, и концепцию Википедии, на которую в действительности ведет ссылка. Анализатор просматривает употребления всех многозначных терминов из созданного словаря в качестве текста ссылок и добавляет в список значений этих терминов концепции, на которые указывали данные ссылки. Чтобы удалить шум, в конечную базу знаний включаются только те термины, которые встретились в ссылках не менее 30 раз.

На этом заканчивается формирование словаря многозначных терминов. В итоге словарь однозначных терминов содержит более 5 500 000 терминов, соответствующих 2 800 000 концепциям, словарь многозначных терминов – 290 000 элементов, а среднее количество значений равно 5,4.

2.4. Обзор работ, использующих Википедию для устранения лексической многозначности

В работах, использующих Википедию для снятия многозначности, продолжают использоваться парадигмы, заданные в 90-х годах. Основными проблемами, решаемыми в этих работах, стали извлечение информации из Википедии, разработка алгоритмов вычисления семантической близости [100, 112] и адаптация существующих алгоритмов к новым моделям.

В работе [113] Википедия использовалась как аннотированный корпус для обучения Наивного Байесовского классификатора. Многозначные термины и их значения выделялись из ссылок ([[статья-значение|многозначный термин]]). Для каждого многозначного термина, представленного в Википедии, строился вектор признаков, составленный из:

- части речи многозначного термина;
- локального контекста из трех слов слева и справа от многозначного термина с их частями речи;
- глобального контекста из не более пяти самых частых специфичных для значения ключевых слов, встречающихся во всевозможных контекстах термина.

Авторы вручную отобразили часть терминов Википедии в термины WordNet, чтобы провести эксперименты на общепринятых эталонных тестах корпуса SenseEval. Эксперименты показали, что Наивный Байесовский классификатор, построенный по Википедии, показывает лучшие результаты, чем алгоритм Леска и baseline algorithm. Эксперименты также подтвердили предположение, что с ростом Википедии улучшается точность снятия многозначности.

Кроме того, было показано, что распределение терминов в статьях Википедии сильно отличается от тренировочного множества корпуса SenseEval.

Исследование, описанное в [113], получило развитие в системе автоматического обогащения документов ссылками на статьи Википедии. В статье [114], посвященной этой системе, описываются методы для автоматического извлечения ключевых слов и снятия лексической многозначности на основе Википедии. Для выделения ключевых фраз использовался словарь, состоящий из заголовков Википедии, расширенный морфологическими формами, которые встречаются во внутренних ссылках Википедии на соответствующие статьи не менее пяти раз. Выделение ключевых фраз происходит в два этапа:

1. поиск кандидатов;

2. ранжирование ключевых фраз.

На первом этапе выделяются всевозможные N-граммы, присутствующие в словаре, после этого на втором этапе выделенным терминам присваиваются веса на основе одного из трех методов: tf-idf, χ^2, и информативность (вероятность использования термина в качестве текста ссылки):

$$P(keyword|W) \approx \frac{|D(keyword)|}{|D(term)|} \ ,$$

где $D(keyword)$ — количество статей Википедии, в которых термин отмечен как ключевой (является частью ссылки), $D(term)$ — количество документов, в которых встречается этот термин. На основании результатов проведенных экспериментов утверждается, что последний метод показывает наилучшие результаты.

Во второй части работы приводится алгоритм снятия многозначности в выделенных ключевых фразах. Авторы использовали комбинацию алгоритма Леска [49] и статистического алгоритма, обученного на Википедии. В алгоритме Леска в качестве словарного определения термина бралась соответствую-

щая статья Википедии, а в качестве контекста — абзац, в котором встретился термин. В качестве статистического алгоритма использовался классификатор [113], описанный в выше. Наконец, предполагая ортогональность этих методов, авторы использовали расхождения в результатах как признак потенциальной ошибки и игнорировали такие результаты. Оценка получившейся системы производилась на тестовом наборе из 85 случайно выбранных статей Википедии, специально размеченных вручную. Эксперименты показали, что статистический метод показывает бóльшую точность (92.91%) и полноту (83.1%), чем алгоритм Леска (80.1% и 71.86% соответственно), а комбинирование алгоритмов увеличивает точность (94.33%), но снижает полноту(70.51%).

Работа [115] является развитием алгоритма Леска с учетом дополнительной информации, которую можно извлечь из Википедии. Для создания словаря и поиска возможных значений терминов использовались заголовки статей, страницы переадресации, страницы значений (disambiguation page) и тексты ссылок. Для каждого термина словаря собирался вектор признаков, состоящий из тэга категории, контекста (слова или термина, встречающиеся вместе с данным термином) и класса термина (Человек, Место, Организация, Остальное). В качестве значения многозначного термина выбирался кандидат, максимально похожий на контекст, где похожесть вычислялась как косинус между векторами признаков.

В работе [116] также рассматривается использование векторной модели для снятия многозначности имен собственных, однако в отличие от [115], практически не уделяется внимания нахождению различных признаков, а в дополнение к векторной модели предлагается использовать иерархию категорий Википедии для обучения линейного классификатора, основанного на методе опорных векторов (Support Vector Machine, SVM).

В работе [117] для снятия многозначности используется семантическая близость терминов [112]. Расстояние между терминами вычисляется на осно-

ве графа ссылок Википедии. Затем для каждого возможного значения термина выбирается наиболее близкое. Авторы показали, что их алгоритм работает лучше, чем алгоритм Леска. Для оценки качества алгоритма использовалось тестовое множество, созданное на основе аннотаций ссылок из статей Википедии.

Для снятия многозначности авторы [101] использовали подход, основанный на машинном обучении в комбинации с семантической похожестью, описанной в [112]. В качестве тренировочного множества использовалось 500 случайных статей Википедии. Положительными примерами устранения многозначности служили термины, на которые указывали ссылки, а остальные возможные значения, полученные из ссылок Википедии, как и в [117], служили отрицательными примерами. В качестве признаков использовались вероятность значения многозначного термина, полученная просмотром всех ссылок Википедии, и расстояние до терминов контекста. Для вычисления расстояния до контекста использовалась та же мера, что и в [117]. Но, кроме того, терминам контекста придавался вес, вычисленный как среднее между информативностью [114] данного термина и близостью термина к центральной нити документа, которая была посчитана как средняя близость между текущим термином и остальными терминами контекста. Еще одним признаком послужило качество контекста, определенное как сумма весов терминов, посчитанных на предыдущем шаге. Основываясь на данных признаках, авторы провели сравнительное тестирование нескольких алгоритмов машинного обучения (Naive Bayes, SVM, C4.5 с вариациями) на тестовом множестве из 100 случайных статей Википедии и показали, что данный подход дает лучшие результаты (97.1%), чем описанный в [117]. Однако эти результаты были получены на корпусе из статей Википедии, поэтому потенциально могут быть необъективными.

2.5. Выводы ко второй главе

Сети документов структурно отличаются от семантических сетей и тезаурусов, созданных экспертами. Это накладывает отпечаток на способы их использования, в частности, на методы вычисления семантической близости. Наиболее эффективными считаются локальные методы, основанные на пересечении множеств ближайших соседей узлов сети, так как позволяют получить хорошие оценки при малой вычислительной сложности. В данной главе приводится метод вычисления семантической близости между концепциями Википедии.

Википедия является современной электронной энциклопедией, поддающейся машинной обработке и позволяющей создать высококачественную базу знаний, необходимую для построения современных систем интеллектуального анализа текстов. Ссылки Википедии обладают различной релевантностью по отношению к семантической близости. В данной главе предложен способ взвешивания графа Википедии и вычисления семантической близости концепций во взвешенном безмасштабном графе с использованием теории нечетких множеств.

Оценка качества мер близости может производиться с помощью корреляции с оценками данными людьми. Однако, так как вычисления семантической близости является лишь вспомогательной задачей, в рамках данной работы выбрана мера, дающая наилучшие результаты в контексте применения к задаче устранения лексической многозначности.

Основными проблемами, с которыми сталкиваются исследователи, использующие Википедию как ресурс для построения интеллектуальных систем, являются:

- **Необходимость предварительной обработки данных энциклопедии, с использованием различных эвристик.** Это связано с тем,

что Википедия создается огромным количеством людей, и, хотя правила создания и редактирования страниц четко определены, гарантии, что абсолютно все страницы будут обладать одинаковой структурой, не существует.

- **Извлечение необходимой информации.** Википедия превышает по объему все существующие энциклопедии и содержит огромное количество информации. Однако для извлечения полезных знаний необходимо создавать сложные системы обработки данных Википедии.

- **Определение значений терминов, не имеющих соответствующих концепций в Википедии.** Решение этой проблемы необходимо для создания хороших методов устранения лексической многозначности. В современной литературе предложена единственная эвристика, называемая «информативностью термина». Один из описанных в следующей главе методов использует эту эвристику для повышения полноты.

Глава 3

Снятие лексической многозначности

Ниже описываются три метода снятия лексической многозначности, основанные на семантической близости концепций Википедии. Первый метод основан на выборе наиболее близкого к однозначному контексту значения многозначного слова. Этот метод является самым простым с вычислительной точки зрения, однако не позволяет достичь точности последующих методов.

Результаты первого метода зависят от однозначного контекста, которого может быть недостаточно для принятия правильного решения. Этот существенный недостаток, исправляется в последующих методах, где задача устранения лексической многозначности сводится к задаче максимизации.

В секции 3.3 показано как можно решать задачу разрешения лексической многозначности с помощью модели Маркова. Основной трудностью применения марковских моделей к обработке естественного языка является оценка параметров моделей. Для преодоления этой трудности предложено использовать семантическую близость концепций Википедии для вычисления параметров модели перехода.

Метод основанный на марковской модели показывает результаты лучше, чем первый метод. Однако марковская модель не позволяет полностью описать структуру дискурса, состоящего из нескольких тем. В разделе 3.4 предложено расширение марковской модели на случай множества независимых марковских процессов для моделирования структуры текста и применение этого расширения к задаче устранения лексической многозначности.

3.1. Общий процесс обработки

Перед тем как перейти к устранению лексической многозначности, необходимо разбить текст на предложения, предложения – на лексемы, определить части речи слов, устранив, таким образом, морфосинтаксическую многозначность слов, и поставить в соответствие последовательностям лексем термины Википедии из созданного словаря.

Для разбиения теста и определения частей речи используется пакет OpenNLP. Алгоритмы представленные в этом пакете основаны на методе максимальной энтропии, и позволяют достичь лучших результатов среди всех открыто доступных инструментов.

Для большинства задач не нужно определять смысл каждого отдельного слова, намного полезнее является определение смысла терминов, состоящих из одного или более слова. Например, для правильной классификации необходимо определить значение всего термина «информационный поиск», а не каждого слова в отдельности. В общем случае задача поиска словосочетаний и связанных фраз не является тривиальной. Однако в нашем случае, можно ограничиться только поиском терминов представленных в Википедии. При этом, для выделения терминов в тексте производится только поиск именных фраз, так как в Википедии практически не представлены другие типы фраз.

Также необходимо находить термины, употребленные во множественном числе. Система ищет в словаре все возможные словоформы, основываясь на правилах их образования в естественном языке, и, если найдено более одного термина, получившаяся многозначность снимается на следующем этапе.

Для английского языка существует четыре правила изменения окончания слов для образования множественного числа ($-s \rightarrow -es$; $-y \rightarrow -ies$; $-f, -fe \rightarrow -ves$; в остальных случаях добавляется буква s). Исключения из этих правил представлены в виде редиректов Википедии, поэтому присут-

ствуют в словаре и могут быть легко найдены. Основываясь на этих правилах, система убирает окончания слов и ищет получившиеся термины в словаре. Возможные коллизии, когда разным словоформам соответствуют разные концепции, решаются на этапе устранения лексической многозначности. Этот метод дает результаты лучше, чем стемминг[1], и при этом вычислительно более эффективен, чем алгоритмы приведения слова к нормальной форме.

Кроме того, для предварительной обработки текста используются две эвристики:

- в анализируемом тексте названия (фильмов, песен и т.д.) должны начинаться с большой буквы;
- если рядом с именем человека стоит имя собственное, то с большой вероятностью это фамилия; необходимо соединить их в одну последовательность и в таком виде искать в словаре.

Эти эвристики позволяют на раннем этапе исключить ошибки выделения терминов.

Эта предварительная обработка является общей частью для всех методов устранения лексической многозначности, представленных в данной работе. В ее результате получаем список терминов и списки возможных значений для каждого термина. Все коллизии возникшие на данном этапе решаются выбором наиболее подходящего значения в дальнейшем.

[1] ***Стемминг*** — это процесс нахождения основы слова для заданного исходного слова.

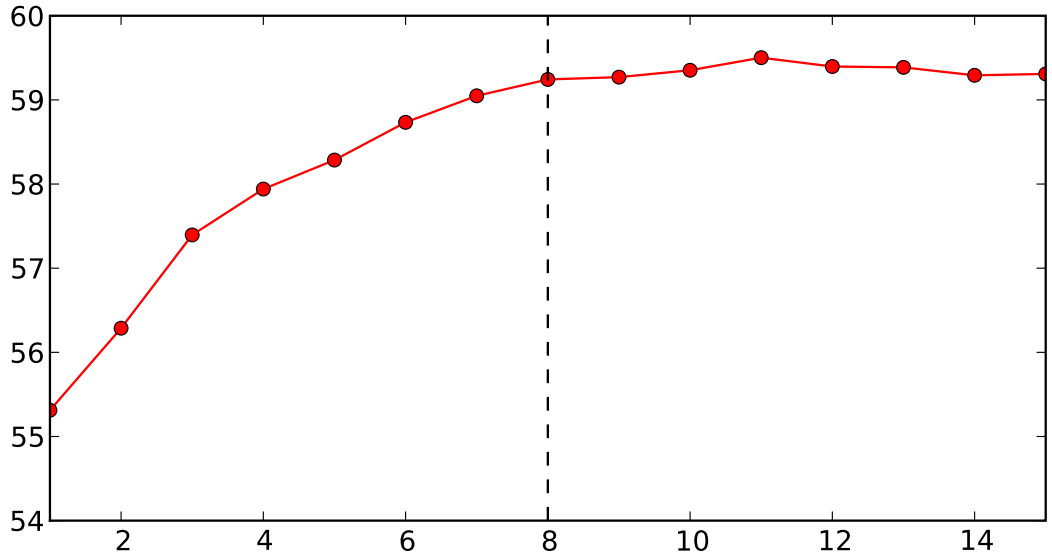

Рис. 3.1. Зависимость точности метода от размера контекста

3.2. Метод, использующий однозначный контекст

3.2.1. Описание метода

Наиболее простым методом снятия многозначности является выбор значения, семантически наиболее близкого к однозначному контексту, в котором встретился многозначный термин. Для формального определения такого метода необходимо определить, что является контекстом термина и критерий близости значения к контексту.

В наших экспериментах в качестве контекста брались однозначные термины текущего и соседних предложений так, чтобы их количество было не меньше заранее определенного значения. Среди исследователей нет общего мнения, каким размером должен обладать контекст, чтобы методы устранения многозначности показывали наилучшие результаты. Мы провели эксперименты и выяснили, что контекст длиной от четырех терминов позволяет данному методу показывать хорошие результаты (рис. 3.1).

Для определения критерия близости значения к контексту воспользуемся

семантической близостью (раздел 2.3.1). Заметим, что семантическая близость является функцией от двух параметров, где одним из параметров служит значение многозначного слова. Контекст, обычно, состоит из нескольких однозначных терминов, поэтому необходимо отдельно определить понятие близости между конкретным значением и контекстом.

Наиболее очевидным способом является определение такой близости через линейную комбинацию близостей значения многозначного термина m и значений каждого термина c_i в контексте $c(c_1, \ldots, c_n)$.

$$sim(m, c(c_1, \ldots, c_n)) = \sum_{i=1}^{n} \alpha_i \cdot sim(m, c_i)$$

Коэффициенты α_i позволяют ранжировать термины контекста по важности в каждом конкретном случае, например, отражать степень удаленности в тексте. В наших экспериментах эти коэффициенты уравниваются $\alpha_i = 1$.

Другой подход состоит в представлении контекста, как единой концепции. То есть, контекст моделируется как временный узел в графе, который соединен с соседями всех концепций, представленных в контексте:

$$N(c_1, \ldots, c_n) = \bigcup_{i=1}^{n} N(c_i) \ .$$

Эксперименты показали, что второй подход позволяет достичь лучшей точности метода.

Еще одной важной задачей, при использовании Википедии является определение терминов, для которых нет правильной концепции в Википедии. Обычно такими терминами служат часто употребляемые выражения и их части. Например, слово «lot» часто употребляется в выражении «a lot of (time, money)», однако в Википедии нет статьи об этом слове в значении «множество» или «много».

В работах [101, 114] было замечено, что на концепции, которые имеют наиболее полное описание в Википедии, чаще ставят внутренние ссылки. На

основе этого наблюдения был введен показатель информативности как отношение количества употреблений термина в качестве ссылки к общему количеству употреблений термина

$$I(t) = \frac{|C(t_{inlinks})|}{|C(t)|} \ .$$

Например, термин «Of course» имеет в Википедии только одно значение – название песни американской группы. Однако это термин часто употребляется в качестве наречия, поэтому его информативность будет низкая. Та же ситуация будет и с термином «lot». Основываясь на этом коэффициенте, вводился порог для отсеивания плохо описанных терминов.

Мы также использовали информативность для улучшения точности нашего метода (рис. 3.2). Система не обрабатывает термины с информативностью ниже некоторого порога. Это уменьшает полноту результата, однако позволяет повысить точность.

3.2.2. Эксперименты

В наиболее известных тестовых коллекциях используются заранее определенные наборы значений многозначных слов, которые берутся из словаря WordNet [9], что накладывает ограничение на возможность их использования этих коллекций. Так, методы использующие словарь Википедии, нельзя напрямую сравнить с методами, использующими словарь WordNet, так как количество значений слов в Википедии намного превосходит аналогичное число в WordNet (Табл. 2.1).

В работе [113] авторы смогли отобразить используемые значения на словарь WordNet, однако в дальнейшем [114] отказались от такой процедуры. Это связано с тем, что Википедия растет и изменяется очень быстро, а объем ее словаря на порядок превосходит словарь WordNet. Кроме того, если значения слов находятся путем анализа употребления слов в текстах, то для

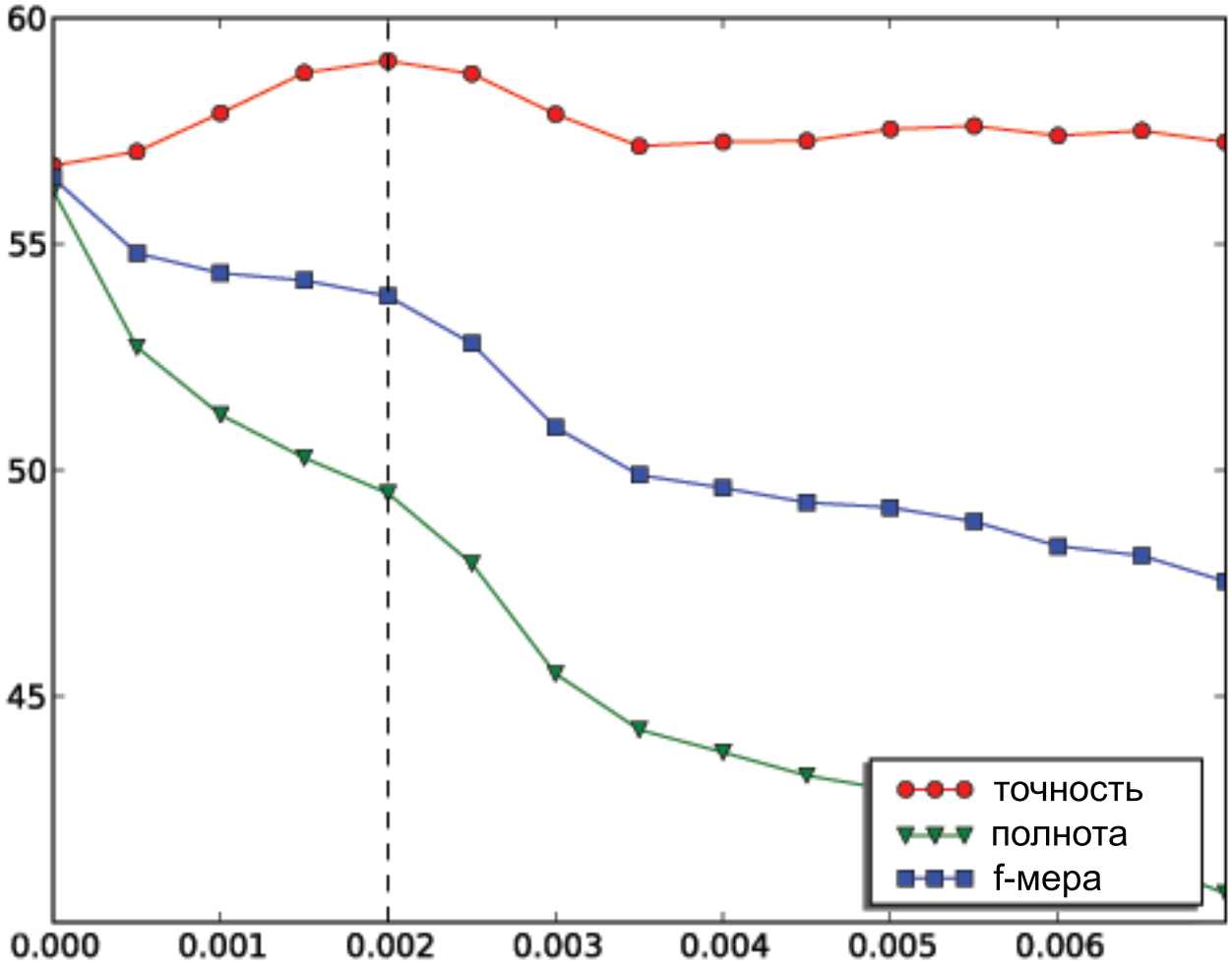

Рис. 3.2. Выбор порога информативности

оценки методов можно использовать только тестовые коллекции, созданные специально для оценки этого метода.

В методах, основанных на Википедии, в качестве тестового корпуса часто используются сами статьи Википедии, причем обрабатываются только термины, представленные в виде ссылок, а значениями этих терминов служат концепции Википедии, на которые указывают ссылки. Несложно заметить, что для таких тестовых корпусов методы, основанные на машинном обучении и обученные на Википедии, дают наилучшие результаты из-за схожести распределений обучающего и тестового множеств [101].

Для оценки нашего метода мы создали тестовое множество, выделив 500 случайных статей Википедии. Использовались только статьи, описывающие однозначные концепции, так как они наиболее близки к неструктурированным текстам. Кроме этого, для составления более полной картины, мы вручную разметили тестовые коллекции из 131 документа, состоящие из новостных сообщений, взятых из различных источников, и нескольких научных статей. Также, мы использовали тестовую коллекцию, предоставленную авторами работы [101] и созданную с помощью сервиса Amazon Mechanical Turk. Характеристики данных коллекций представлены в таблице 3.1.

Среднее число значений многозначных терминов в обеих коллекциях сильно превышает аналогичное число во всем языке (см. табл. 2.1). Это происходит потому, что у часто употребляемых терминов значений больше. Кроме того, процент многозначных терминов в коллекции, размеченной вручную, намного превышает аналогичное число в коллекции, автоматически созданной из статей Википедии, что, несомненно, влияет на результаты алгоритмов устранения многозначности (табл. 3.5, 3.7 и 3.6). Также следует заметить, что с изменением Википедии, также приходится изменять и тесты, так как появляются новые термины, и увеличивается количество значений. И если создать тестовую коллекцию по Википедии можно автоматически, то тесты,

	статьи Википедии	Milne and Witten [101]	Новости и научные статьи
#документов	500	50	131
#терминов	50947	727	8236
#многозначных терминов	39332	479	6952
#среднее кол-во значений	35.34	29.94	22.34

Таблица 3.1. Статистика для тестовых коллекций

	Норма 2.1	Норма 2.2	Норма 2.3	Норма 2.4
cos	58.955536	53.77517	64.11493	39.639263
Dice	59.18624	53.94295	**64.345634**	39.723156
Jaccard	37.520973	26.489094	45.658558	39.177853
Google	33.372465	25.960512	33.137674	37.641409

Таблица 3.2. Результаты работы алгоритма с использованием различных мер близости и норм на коллекции новостей и научных статей (октябрь 2008)

созданные вручную, придется заново вручную переразметить.

3.2.3. Выбор параметров и результаты

При выборе меры семантической близости, T-нормы и S-нормы использовались коллекции, описанные в предыдущем разделе, однако в коллекции новостных и научных статей использовались только те термины, значения которых представлены в Википедии в виде соответствующих концепций. Это позволяет избежать погрешности, которая возникает при фильтрации низкоинформативных терминов (см. ниже).

Выбор порога информативности осуществлялся для коллекции новостей

	Норма 2.1	Норма 2.2	Норма 2.3	Норма 2.4
cos	68.31683	59.40594	**78.81188**	55.247524
Dice	68.31683	59.40594	**78.81188**	55.247524
Jaccard	44.752476	36.435642	50.09901	52.475246
Google	37.903225	35.080647	37.903225	42.54032

Таблица 3.3. Результаты работы алгоритма с использованием различных мер близости и норм на коллекции Milne and Witten [101] (октябрь 2008)

	Норма 2.1	Норма 2.2	Норма 2.3	Норма 2.4
cos	78.72156	74.70648	84.968834	74.03972
Dice	78.808525	74.67749	**84.997826**	74.03972
Jaccard	62.79171	54.70358	68.02435	72.735176
Google	56.47904	53.166225	56.37643	58.45793

Таблица 3.4. Результаты работы алгоритма с использованием различных мер близости и норм на коллекции статей Википедии (октябрь 2008)

и научных статей. Увеличение порога уменьшает количество обрабатываемых терминов, влияя, тем самым, на полноту и точность метода. Лучшая точность была достигнута при пороге равном 0.002 (рис. 3.2). При дальнейшем увеличении порога точность метода не растет, так как отсеиваются и однозначные термины из контекста. В работах [101, 114] использовался порог 0.001, но он подбирался для коллекции из статей Википедии. Отсюда можно сделать вывод, что значение порога следует выбирать, исходя из специфики документов.

Выбор размера контекста также осуществлялся на основе экспериментов для коллекции новостей и научных статей (рис. 3.1). Уже при длине контекста от четырех терминов алгоритм показывает приемлемые результаты. Исходя из экспериментов, можно заметить, что с увеличением контекста возрастает точность и полнота метода. Это можно объяснить спецификой коллекции, где, в основном, присутствуют документы с ярко выраженной основной темой (новости). Для многотемных документов контекст должен соответствовать конкретной теме.

Следует заметить, что время работы алгоритма линейно увеличивается в зависимости от длины контекста (так как количество значений многозначных слов фиксировано). Исходя их этих соображений, в дальнейших экспериментах использовался контекст длиной от 8 терминов. Результаты работы методов представлены в таблицах 3.5, 3.7 и 3.6. На тестовом множестве, составленном из документов Википедии, алгоритм показывает лучшие результаты. Это связанно с характеристиками коллекций (табл. 3.1).

3.2.4. Выводы

Метод, представленный в предыдущем разделе, несмотря на свою простоту, показывает хорошие результаты. Основным преимуществом перед дру-

	Точность	Полнота	F-мера
Самое частое значение	43.00	33.22	37.48
Без учета весов ссылок	54.66	45.81	49.84
С учетом весов ссылок	**59.19**	49.60	53.97
С нулевым порогом информативности	56.77	**56.42**	**56.60**
Только термины со значениями, описанными в Википедии	*64,34*	*64,34*	*64,34*

Таблица 3.5. Результаты работы алгоритма на коллекции новостей и научных статей (октябрь 2008)

	Точность	Полнота	F-мера
Самое частое значение	47.86	47.25	47.55
Без учета весов ссылок	71.61	68.86	70.21
С учетом весов ссылок	**74.28**	**71.42**	**72.82**
Только термины со значениями, описанными в Википедии	*78,81*	*78,81*	*78,81*

Таблица 3.6. Результаты работы алгоритма на коллекции Milne and Witten [101] (октябрь 2008)

	Точность	Полнота	F-мера
Самое частое значение	76,92	71,28	73,99
Без учета весов ссылок	83.05	80.98	82.01
С учетом весов ссылок	**85.93**	**83.62**	**84.76**
Только термины со значениями, описанными в Википедии	*87,12*	*87,12*	*87,12*

Таблица 3.7. Результаты работы алгоритма на коллекции статей Википедии (октябрь 2008)

	Точность	Полнота	F-мера
Новости и научные статьи	40.25	32,40	35,70
Milne and Witten	74,28	71,42	72,82
Статьи Википедии	80,55	78,16	79,34

Таблица 3.8. Лучшие результаты, показанные алгоритмом на аналогичных коллекциях и снимке Википедии, сделанном в апреле 2009

гими методами является простота реализации и интерпретации результатов. При этом метод полностью автоматический и не требует специально размеченных коллекций документов. Кроме того, метод может быть легко адаптирован к любым естественным языкам, для которых существует раздел Википедии.

Основным недостатком метода является то, что он неявно используют предположение, что в тексте существуют однозначные термины, на основании значений которых впоследствии определяются значения многозначных терминов. Однако было замечено, что с ростом Википедии словарь многозначных терминов увеличивается, причем дополнительные значения появляются у наиболее употребляемых терминов. Это приводит к тому, что в неспецифических сообщениях, таких как новостные статьи, все термины имеют более одного значения, либо встречающиеся однозначные термины мало связаны с основной темой документа. Это ухудшает точность алгоритмов, основанных на однозначном контексте, и ведет к необходимости разработки метода, лишенного такого недостатка. Заметим, что этим недостатком обладают все методы устранения лексической многозначности, использующие Википедию, представленные в разделе 2.4.

3.3. Метод на основе специализированной марковской модели

Одним из перспективных, но мало изученных направлений в области устранения лексической многозначности является использование методов оптимизации. Современные системы устраняют многозначность слов по очереди, обрабатывая каждое слово изолированно. Очевидно, что значения слов зависимы, и фиксирование смысла одного слова может повлиять на контекст другого. Зависимость между значениями слов может быть смоделирована, представлением задачи разрешения лексической многозначности в виде задачи оптимизации (в отличие от классификационной модели) [10].

Далее будет приведено описание такого метода, основанного на скрытой модели Маркова. Вероятности модели перехода мы оцениваем с помощью семантической близости концепций, способ подсчета которой описан в разделе 2.3.1. Вероятности модели наблюдений и априорная вероятность значений оценивается с помощью эмпирического распределения значений и терминов во внутренних ссылках Википедии.

3.3.1. Описание метода

Обозначим через T множество терминов в тексте, а через M — множество соответствующих значений. Для входной последовательности терминов $\tau = t_1, \ldots, t_n$, где $t_i \in T$, задача максимизации состоит в поиске наиболее вероятной последовательности значений $\mu = m_1, \ldots, m_n$, где $m_i \in M$, соответствующей входным терминам и согласованной с ограничениями модели:

$$\hat{\mu} = \arg\max_{\mu} P(\mu \mid \tau) = \arg\max_{\mu} \left(\frac{P(\mu)P(\tau \mid \mu)}{P(\tau)} \right) \quad . \tag{3.1}$$

Так как вероятность $P(\tau)$ постоянна для фиксированной входной последовательности, задача редуцируется к максимизации числителя правой части

равенства (3.1). Для решения этого уравнения делается марковское предположение, что значение i-го термина зависит только от конечного числа значений предыдущих терминов:

$$\hat{\mu} = \arg\max_{\mu} \left(\prod_{i=1}^{n} P(m_i \mid m_{i-h:i-1}) \cdot P(t_i \mid m_i) \right) \quad , \tag{3.2}$$

где h — порядок модели.

Множители уравнения (3.2) определяют скрытую марковскую модель h-го порядка, где наблюдения соответствуют входным терминам, состояния соответствуют значениям терминов, $P(m_i \mid m_{i-h:i-1})$ — модель перехода между состояниями и $P(t_i \mid m_i)$ — модель наблюдения, описывающая вероятность появления термина t_i в каждом состоянии m_i.

Несмотря на то, что рассматриваемую задачу нетрудно формализовать с помощью скрытой марковской модели, дальнейшее использование этого формализма связано с проблемой разреженности языка. Так, чтобы построить модель перехода для марковской модели первого порядка, необходимо оценить вероятность каждой пары состояний, что для задачи устранения лексической многозначности является вероятностью того, что два термина в конкретных значениях встретились вместе. Если для задачи определения частей речи слов, параметры марковской модели можно оценить на основе сравнительно небольшого размеченного корпуса, то для задачи устранения лексической многозначности проблема оценки параметров сильно усложняется. Это связано с количеством значений. Так, Википедия содержит более трех миллионов концепций, кроме того, задача осложняется тем, что частота употребления терминов в тексте распределена не по равномерному закону, а по закону Зипфа (Zipf law). Учитывая эти факты, несложно заметить, что для обучения марковской модели потребуется размеченный корпус огромного размера. Ниже мы предложим способ оценки модели перехода для поставленной задачи с помощью семантической близости концепций Википедии, вычисленной на

основе графа ссылок.

Для оценки модели наблюдения воспользуемся ссылками Википедии. На основании способа построения словарей (раздел 2.3.2) можно заметить, что термины, соответствующие синонимам концепции, могут появиться только из заголовка статьи, описывающей концепцию, названий редиректов на концепцию и терминов, совпадающих с текстом ссылок на концепцию. Исходя из этого, определим условную вероятность термина t_i^j, соответствующего значению m_i через эмпирическую вероятность $\hat{P}(t_i^j|m_i)$:

$$P(t_i^j|m_i) = \hat{P}(t_i^j|m_i) = \frac{C(t_i^j, m_i)}{C(m_i)} \ , \tag{3.3}$$

где $C(t_i^j, m_i)$ — количество ссылок на концепцию m_i в которых термин которых совпадал с t_i^j, включая редиректы и название концепции, как специальный тип ссылок, а $C(m_i)$ — общее количество ссылок на концепцию.

Чтобы оценить модель перехода сделаем предположение, что

Эвристика 1. *Вероятность значения m_i, при условии предыдущего контекста $m_{i-h}, \ldots, m_{i-1}$ пропорциональна линейной комбинации (а) близости значения к контексту и (б) априорной вероятности этого значения.*

$$P(m_i|m_{i-h}, \ldots m_{i-1}) = \hat{P}(m_i|m_{i-h}, \ldots m_{i-1}) = \\ = \alpha \cdot (sim(m_i; m_{i-h}, \ldots m_{i-1}) + \beta \cdot P(m_i)) \tag{3.4}$$

Для модели первого порядка близость значения к контексту, соответствующему предыдущему значению, вычисляется через семантическую близость, описанную в разделе 2.3.1. Чтобы оценить близость значения к контексту из нескольких терминов, так же как и в предыдущем методе воспользуемся представлением контекста в виде обобщенной концепции, объединяющей все входящие в нее значения:

$$N(m_1, \ldots, m_n) = \bigcup_{i=1}^{n} N(m_i) \ .$$

Тогда близость вычисляется так же, как и для двух обычных концепций.

Априорную вероятность значения будем оценивать на основе ссылок, способом аналогичным тому, который мы использовали при оценке модели наблюдения.

$$P(m_i) = \hat{P}(m_i) = \frac{C(m_i)}{\sum_i C(m_i)} \tag{3.5}$$

Коэффициент нормализации α в уравнении 5 не влияет на решение задачи максимизации, поэтому его можно не учитывать. Коэффициент β на основании экспериментов принят равным 1.

После определения всех параметров модели задача максимизации решается с помощью алгоритма Витерби. Этот алгоритм использует замечание, что наиболее вероятный путь до каждого следующего состояния зависит только от наиболее вероятного пути через h предыдущих состояний. Таким образом, количество сравнений на каждом шаге экспоненциально зависит от h и равно

$$\prod_{i=n-h}^{n-1} |m_i| \ .$$

Чтобы сократить время работы алгоритма, мы выдвинули наивное предположение:

Эвристика 2. *для задачи устранения лексической многозначности, наиболее вероятный путь до состояния m_i зависит только от h последних значений наиболее вероятного пути до состояния m_{i-1} .*

В этом случае каждое состояние должно хранить дополнительную информацию не более чем о h предыдущих терминах, и, таким образом, модель сведется к специализированной марковской модели первого порядка. Наиболее вероятная последовательность состояний для такой модели находится так же, как и для обычной марковской модели первого порядка, за исключением вычисления вероятности перехода между состояниями.

Порядок	Модель Маркова	ММ с эвристикой
0	53.12	53.12
1	54.00	54.00
2	54.50	54.49
3	**54,76**	54.72

Таблица 3.9. Результаты работы алгоритма на коллекции новостей и научных статей

Порядок	Модель Маркова	ММ с эвристикой
0	83.10	83.10
1	83.30	83.30
2	83.69	83.69
3	83.69	**83.88**

Таблица 3.10. Результаты работы алгоритма на коллекции Milne and Witten [101]

Конечно, это предположение в общем случае неверно, однако в рамках данной задачи, оно позволяет уменьшить порядок модели и, при этом, учесть контекст из нескольких терминов, тем самым не сильно ухудшить точность метода (табл. 3.9, 3.11 и 3.10).

3.3.2. Эксперименты

Эксперименты проводились на тестовых коллекциях, описанных в разделе 3.2.2. Результаты представлены в таблицах 3.9, 3.11 и 3.10.

Все эксперименты проводились на снимке Википедии, полученном в октябре 2008 г. Алгоритм применялся ко всем найденным терминам текста, поэтому точность и полнота совпадают.

Сравнительно низкие результаты, полученные на первом корпусе, связаны с тем, что мы считали заведомо неверным ответ алгоритма, данный для

Порядок	Модель Маркова	ММ с эвристикой
0	90,10	90,10
1	90,13	90,13
2	91,51	91.48
3	**91,62**	91,52

Таблица 3.11. Результаты работы алгоритма на коллекции статей Википедии

терминов, не имеющих правильного значения среди концепций Википедии. Такими терминами, в основном, являются имена людей и слова, входящие в устойчивые выражения, например, слово “lot” в выражении “a lot of time. . .”. Если не учитывать такие термины, точность алгоритма достигает 76,84%.

3.3.3. Выводы

Предложенный метод устранения лексической многозначности терминов естественного языка, основанный на марковской модели, параметры которой вычислены с помощью данных Википедии, показывает высокую точность, превышающую точность предыдущего алгоритма. Проблема разреженности языка решается предположением, что апостериорная вероятность значения термина при условии предыдущего контекста пропорциональна линейной комбинации семантической близости соответствующих концепций Википедии и априорной вероятности значения. Для ускорения алгоритма предложена эвристика, которая, в рамках поставленной задачи, дает выигрыш по времени выполнения, при этом незначительно ухудшая точность результата.

Анализ ошибок алгоритма позволил выявить существенный недостаток: метод неявно предполагает, что все термины в тексте имеют общий смысл. Однако часто в тексте кроме основной семантической линии существует несколько параллельных, таких как место и время основных событий. Основываясь

на этом замечании, мы пришли к выводу, что применение данного алгоритма необходимо комбинировать с методом, выделяющим последовательности семантически связанных терминов (lexical chains).

3.4. Метод на основе марковской модели, обобщенной на случай нескольких независимых цепей

3.4.1. Мотивация и примеры

В предыдущем разделе показано, как оценить параметры марковских моделей высокого порядка с помощью меры семантической близости. Но даже при возможности получить такие оценки, марковская модель не является достаточно хорошим способом описания последовательности значений терминов, встретившихся в тексте на естественном языке. Рассмотрим несколько примеров. На основании следующих примеров можно заметить, что практически все тексты на естественном языке описывают несколько тем или несколько аспектов одной темы, что не может быть должным образом отражено с помощью марковской модели.

Пример 1. Предположим, что система автоматического снятия лексической многозначности обрабатывает текст о спортивной медицине: новой способ лечения одной из профессиональных болезней с помощью лекарственного препарата, который ранее не использовался для этих целей. Пусть термины, встретившиеся системе, будут: «*football*», «*The drug*», «*sports medicine*», и т. д.

Так как конкретный лекарственный препарат (называемый для иллюстрации «*The drug*») никогда не использовался для лечения данной болезни, наиболее вероятно, что имя этого препарата не связанно со спортивными концепциями, то есть $P(\mathit{The\ drug} \mid \mathit{football}) = 0$. В случае использования классической скрытой марковской модели, система должна использовать некоторые

эвристики или методы сглаживания, для того чтобы найти правильную последовательность значений. С другой стороны, если модель поддерживает несколько цепей, тогда возникнут две различные цепи, и вероятность этого события будет равна сумме вероятностей двух независимых событий, что соответствующие термины имеют свои наиболее вероятные значения. Более того, эти цепи могут объединиться в одну во время последующей обработки входного текста, например при обработке термина «*sports medicine*», значение которого пересекается с обеими темами.

Пример 2. Рассмотрим пример, который был найден во время анализа ошибок при использовании классической марковской модели. Этот пример представляет собой фрагмент новостной статьи о профессиональных футболистах и их машинах. В предложении

> *Cristiano Ronaldo* hit the *headlines* when he crashed his *Ferrari* (Кристиано Рональдо попал в заголовки газет, когда разбил свою Феррари).

представлено три термина из словаря Википедии: «*Cristiano Ronaldo*», «*headline*» и «*Ferrari*». Термин *Ferrari* — многозначный и имеет, по крайней мере, два значения: (а) Спортивная машина известного итальянского производителя и (б) Маттео Феррари — Итальянский футболист. Основываясь на контексте, представленном термином *Cristiano Ronaldo* и сильно связанным с темой футбола, большинство алгоритмов разрешения лексической многозначности, включая основанные на классической марковской модели, имеют тенденцию к выбору в качестве значения термина «*Ferrari*» футболиста, а не спортивную машину, даже если значение «спортивная машина» чаще встречается в текстах. С другой стороны, модель, поддерживающая несколько цепей, в данном случае найдет более вероятным начать новую цепь, содержащую наиболее часто употребляемое значение; другие, связанные с машинами, значения терминов будут добавляться к этой цепи, если встретятся при дальнейшей

обработке текста.

Таким образом, наша цель состоит в разработке модели, предоставляющей естественный способ описания смысла текста как множества независимых марковских цепей. В последующих разделах описывается обобщенная модель в целом, а затем обсуждается один из вариантов оценки ее параметров, с целью применения к задачи устранения лексической многозначности.

3.4.2. Обобщение марковской модели

Также как и в классической марковской модели, исследуются стационарные процессы, моделируемые марковскими цепями некоторого порядка m. Основное различие между классической марковской моделью и моделью представленной в данном разделе заключается в том, что здесь текущее значение переменной состояния может стать частью одной из существующих цепей, а может сформировать новую цепь, независимую от других цепей.

В дальнейшем, будем использовать обычные математические символы для обозначения ненаблюдаемых переменных состояния: состояние системы на k-м шаге обозначается как $S_k \in \mathbb{S}$, где $\mathbb{S}$ — множество всевозможных состояний. Марковские цепи будем обозначать каллиграфическими символами, например $\mathcal{L}$, $\mathcal{N}$. Используем черту сверху для соединения компонентов, составляющих одну цепь, так $\overline{\mathcal{L}S_k}$ обозначает цепь, состоящую из последовательности состояний $\mathcal{L}$ и заканчивающуюся последним состоянием S_k. Кроме того, запись $S_k \in \mathcal{L}$ означает, что состояние S_k принадлежит цепи $\mathcal{L}$. Наконец, запись $\widehat{S_i S_j}$ используется для обозначения пары состояний S_i, S_j, принадлежащих одной цепи.

Далее предложена формализация обобщенной модели. Сначала, опишем случай когда все предыдущие состояния системы составляют одну цепь. После, этот частный случай расширим до общего случая для множества цепей.

Все предыдущие состояния принадлежат одной цепи

Случай, когда все предыдущие состояния S_1, S_2, ..., S_{k-1} соединены в одну цепь $\mathcal{L}$ отличается от классической модели Маркова способом обработки текущего состояния. А именно, для нового состояния S_k возможны два случая: (а) S_k присоединяется к $\mathcal{L}$, как новое состояние ($\overline{\mathcal{L}S_k}$), или (б) S_k не принадлежит $\mathcal{L}$ и формирует новую цепь $\mathcal{N}$, ($\mathcal{L}, \mathcal{N}$). Вероятности этих событий вычисляются как:

$$P(\overline{\mathcal{L}S_k}) = P(\mathcal{L}) \cdot P(S_k \in \mathcal{L}) \cdot P(S_k \mid \mathcal{L}) \ , \tag{3.6}$$

$$P(\mathcal{L}, \mathcal{N}) = P(\mathcal{L}) \cdot P(S_k \notin \mathcal{L}) \cdot P(S_k) \ . \tag{3.7}$$

Правые части каждого из уравнений состоят в точности из трех сомножителей; в дальнейшем будем называть их *первый*, *второй* и *третий* соответственно. Каждое из уравнений (3.6) и (3.7) определяет полную вероятность одного из возможных событий; таким образом, оба уравнения включают в себя вероятность цепи $\mathcal{L}$, как первый из сомножителей. Второй сомножитель определяет вероятность принадлежности текущего состояния S_k существующей цепи $\mathcal{L}$. Если S_k присоединяется к цепи $\mathcal{L}$, тогда третий сомножитель в уравнении (3.6) выражает вероятность появления S_k в этой цепи. Если S_k формирует новую цепь, тогда третий сомножитель в уравнении (3.7) не зависит от состояний цепи $\mathcal{L}$.

Наша цель найти наиболее вероятную последовательность состояний и их разделение на отдельные цепи; эта задача включает в себя вычисление вероятностей $P(\overline{\mathcal{L}S_k})$ и $P(\mathcal{L}, \mathcal{N})$. Для дальнейшего исследования уравнений (3.6) и (3.7) необходимо вывести формулы для вычисления каждого из трех описанных сомножителей.

Первые сомножители уравнений (3.6) и (3.7) вычисляются рекурсивно для каждого состояния цепи.

Для нахождения вторых сомножителей сделаем предположение аналогичное марковскому для классической модели:

Предположение 1. *Вероятность того, что текущее состояние S_k принадлежит цепи $\mathcal{L}$, зависит только от конечного числа предыдущих состояний S_{t_1}, S_{t_2}, ..., S_{t_h} цепи $\mathcal{L}$, где $t_i < k, \forall i = \overline{1,h}$.*

В дальнейшем будем называть состояния S_{t_1}, S_{t_2}, ..., S_{t_h}, описанные в сделанном предположении ***активными***; цепи, содержащие активные состояния, будем называть ***активными цепями***. Заметим, что число активных цепей всегда меньше или равно числа активных состояний. Не ограничивая общности рассуждений, будем рассматривать только конечную ***историю*** активных состояний, то есть конечную последовательность предыдущих состояний S_{k-1}, S_{k-2}, ..., S_{k-h}. Таким образом, в дополнение к порядку m классической марковской модели для связанных цепей, вводится ***порядок обобщенной модели***, в дальнейшем обозначаемый как h.

На основании Предположения 1, вторые сомножители уравнений (3.6) и (3.7) выражаются через отношение между состоянием S_k и множеством активных состояний цепи $\mathcal{L}$. Для дальнейшего упрощения вычисления вторых сомножителей, выразим их через попарные отношения между состояниями. Для этого введем еще одно предположение:

Предположение 2. *Для различных состояний S_i, S_j и S_k, событие «S_i и S_k принадлежат одной цепи» является независимым от события «S_j и S_k принадлежат одной цепи». То есть для $i \neq j$, $i \neq k$ и $j \neq k$:*

$$P(\widehat{S_iS_k} \text{ and } \widehat{S_jS_k}) = P(\widehat{S_iS_k}) \cdot P(\widehat{S_jS_k}) \ .$$

Обозначим множество активных состояний в цепи $\mathcal{L}$ как $\Omega = \{S_{k-1}, \ldots, S_{k-h}\}$. Тогда, на основании сделанных предположений 1 и 2, вторые сомножители

уравнений (3.6) и (3.7) вычисляются как вероятности комплиментарных событий:

$$\begin{aligned} P(S_k \notin \mathcal{L}) &= \prod_{S_i \in \Omega} [1 - P(\widehat{S_i S_k})] \ , \\ P(S_k \in \mathcal{L}) &= 1 - \prod_{S_i \in \Omega} [1 - P(\widehat{S_i S_k})] \ . \end{aligned}$$

Третьи сомножители в уравнениях (3.6) и (3.7) могут быть оценены путем машинного обучения на размеченном корпусе или способом аналогичным представленному в разделе 3.3 для классической марковской модели. Конкретный способ оценки модели перехода $P(S_k \mid S_{k-1}, \ldots, S_{k-m})$ и вероятности $P(\widehat{S_i S_k})$ зависит от конкретного приложения, в котором применяется данная модель. В разделе 3.4.4 показан способ вычисления этих вероятностей с помощью Википедии в контексте применения данной модели к задаче устранения лексической многозначности.

Случай множества цепей

Рассмотрим общий случай, когда существует несколько цепей. Обозначим множество всех цепей как $\Lambda = \{\mathcal{L}_1, \ldots, \mathcal{L}_q\}$. Новое состояние может принадлежать одной или более цепей из Λ или не принадлежать ни одной. Если состояние S_k принадлежит более чем одной цепи, это означает что S_k объединяет эти цепи в одну. Формально, для случайного подмножества цепей $\lambda \subset \Lambda$, $\lambda = \{\mathcal{L}_{i_1}, \mathcal{L}_{i_2}, \ldots, \mathcal{L}_{i_r}\}$, вероятность события, что S_k принадлежит в точности этому подмножеству записывается через комбинацию уравнений (3.6) и (3.7):

$$P(\overline{\lambda S_k}, \Lambda \setminus \lambda) = P(\Lambda) \cdot P(S_k \in \lambda, S_k \notin \Lambda \setminus \lambda) \cdot P(S_k \mid \lambda) \ . \qquad (3.8)$$

Здесь, запись $S_k \in \lambda$ означает событие, что S_k принадлежит каждой цепи в λ; запись $S_k \notin \Lambda \setminus \lambda$ обозначает событие, что S_k не принадлежит ни одной цепи в $\Lambda \setminus \lambda$.

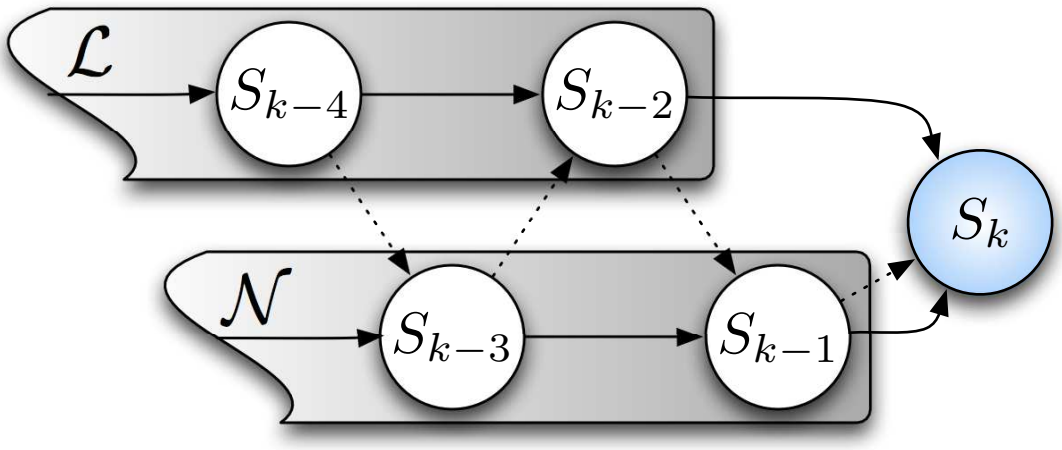

Рис. 3.3. Новое состояние S_k принадлежит цепям $\mathcal{L}$ и $\mathcal{N}$

Так же, как и в случае одной цепи, первый сомножитель в уравнении (3.8) вычисляется рекурсивно по k. Второй сомножитель уравнения (3.8) вычисляется с использованием независимости отдельных цепей:

$$P(S_k \in \lambda, S_k \notin \Lambda \setminus \lambda) = \prod_{L_i \in \lambda} P(S_k \in \mathcal{L}_i) \times \prod_{L_j \in (\Lambda/\lambda)} P(S_k \notin \mathcal{L}_j) \ .$$

Если λ состоит в точности из одной цепи, то третий сомножитель уравнения (3.8) вычисляется таким же образом, как и в предыдущем разделе. Вследствие того, что общий случай, когда λ содержит несколько цепей, означает, что все эти цепи соединяются вместе в одну цепь, вероятность состояния S_k при условии принадлежности цепям λ определяется как вероятность того, что S_k принадлежит объединенной цепи:

$$P(S_k \mid \mathcal{L}_{i_1}, \ldots, \mathcal{L}_{i_r}) = P(S_k \mid \mathcal{L}) \ , \quad \mathcal{L} = \bigcup_{j=1}^{r} \mathcal{L}_{i_j} \ .$$

Например, если порядок марковской модели для цепей $m = 2$, и состояние S_k принадлежит двум цепям $\mathcal{L}$ и $\mathcal{N}$ (Рис. 3.3), тогда

$$P(S_k|\mathcal{L}\mathcal{N}) = P(S_k|S_{k-1}S_{k-2}) \ .$$

Если S_k принадлежит только цепи $\mathcal{L}$, тогда

$$P(S_k|\mathcal{L}\mathcal{N}) = P(S_k|\mathcal{L}) = P(S_k|S_{k-2}S_{k-4}) \ ,$$

и т. д. Заметим, что перестроение цепей не требуется, достаточно вычислить вероятность нового состояния при условии последних m состояний, принадлежащих цепям в λ.

Для классической марковской модели, наиболее вероятная последовательность состояний при условии заданной последовательности наблюдений может быть найдена с помощью алгоритма Витерби. Алгоритм Витерби основан на рекурсивной связи между наиболее вероятным путем до каждого состояния S_{k+1} и наиболее вероятным путем до каждого предыдущего состояния S_k. В следующем разделе описывается аналогичный алгоритм для обобщенной модели, представленной выше.

3.4.3. Алгоритм для нахождения наиболее вероятной последовательности состояний

Алгоритм (Algorithm 1) поиска наиболее вероятной последовательности состояний для заданной последовательности наблюдений похож на свой аналог для классической марковской модели, за исключением функции $computePath$. Функция $computePath$ принимает на вход наиболее вероятные пути (последовательности состояний) до предыдущих состояний и новое состояние, соответствующее текущему наблюдению, и вычисляет наиболее вероятный путь до этого нового состояния; определение функции дается ниже. Пути через предыдущие состояния запоминаются в ассоциативный контейнер $prevPath$ (строки 2, 8, 10 алгоритма). Функция, $combination(s^j_{i-h:i-1})$ вызываемая в строке 6 алгоритма, выдает множество всех комбинаций состояний, соответствующих наблюдениям $i-h,\ldots,i-1$. Заметим, что в случае равновероятных путей алгоритм предпочитает путь, содержащий наименьшее количество цепей.

Вычислительная сложность алгоритма сильно зависит от вычислительной

Algorithm 1 Поиск наиболее вероятной последовательности состояний

Input: h, sequenceOfObservation

Output: mostProbableSequenceOfStates

{Initialization}

for $s_1^j \in$ {states corresponding to first observation} **do**

  prevPath[s_1^j]=s_1^j

end for

{Induction}

for $i = 2$ to n **do**

  for $s_i^j \in$ {states corresponding to ith observation} **do**

    for all $u \in$ $combination(s_{i-h:i-1}^j)$ **do**

      if $i <= k$ **then**

        prevPath[$u \cup s_i^j$] = computePath(prevPath[u],s_i^j)

      else

        prevPath[$u/\{s_{i-h}\} \cup s_i^j$] =$\arg\max_{s_{i-h}}$ ($computePath$(prevPath[u], s_i^j)

      end if

    end for

  end for

end for

{Termination}

return $\arg\max_{u=\{u_{n:n-h}\}}$ prevPath(u)

сложности функции *computePath*. Определим эту функции для двух случаев. Так как вероятность нового состояния S_k присоединенного к нескольким цепям потенциально изменяется во втором и третьем сомножителях уравнения (3.8) (для различных значений λ), полезно рассмотреть эффект, производимый на модель каждым из сомножителей, индивидуально. Для достижения этой цели, рассмотрим отдельно специальный случай для моделирования цепей с помощью марковской модели нулевого порядка (т. е. случай $m = 0$), так как в этом случае $P(S_k \mid \lambda) = P(S_k)$, и, таким образом, третий сомножитель в уравнении (3.8) является постоянным относительно λ. В дальнейшем, будем ссылаться на этот специальный случай обобщенной модели, как на ***слабую*** модель, и рассмотрим сначала ее. Общий случай, когда $m > 0$, будет называться ***полной*** моделью и будет рассмотрен далее.

Слабая модель

Для слабой модели $m = 0$ правая часть уравнения (3.8) для различных значений λ отличается только во втором сомножителе. Такая единообразность уравнения (3.8) для слабой модели позволяет уменьшить вычислительную сложность алгоритма поиска наиболее вероятной последовательности состояний.

Далее представлена теорема, позволяющая уменьшить область поиска наиболее вероятного пути; после ее доказательства приводится функция *computePath*.

Доказательство теоремы производится по индукции по количеству цепей, используя рекурсивную структуру алгоритма, для перехода от наиболее вероятного пути до некоторого промежуточного состояния к искомому наиболее вероятному пути, содержащему все состояния. Для нахождения базиса индукции, сначала рассмотрим случай, когда Λ состоит только из двух цепей:

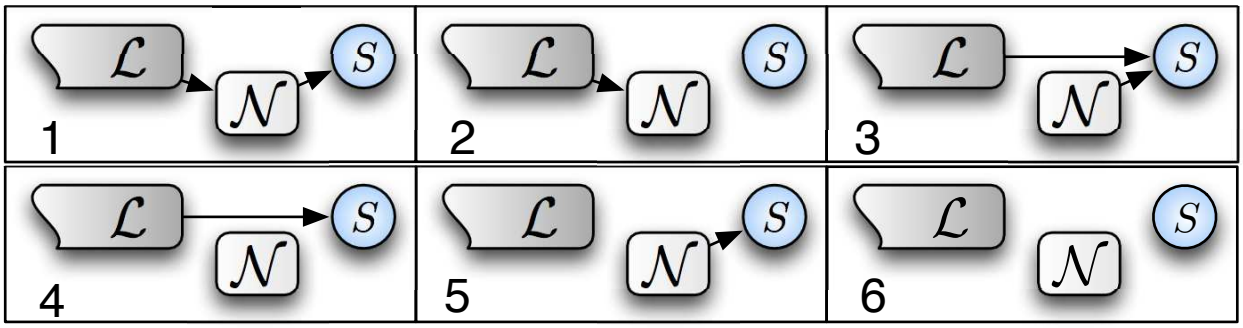

Рис. 3.4. Возможные соединения между цепями $\mathcal{L}$ и $\mathcal{N}$ и новым состоянием S. В случаях 1–2, $\mathcal{L}$ и $\mathcal{N}$ формируют одну цепь. В случаи 3–6 $\mathcal{L}$ и $\mathcal{N}$ представлены как разные цепи.

$\Lambda = \{\mathcal{L}, \mathcal{N}\}$. На рисунке 3.4 перечислены все возможные соединения между $\mathcal{L}$ и $\mathcal{N}$ и текущим состоянием S. В первых двух случаях на рисунке изображен случай, когда цепь $\mathcal{N}$ является продолжением цепи $\mathcal{L}$, в оставшихся случаях $\mathcal{N}$ является отдельной цепью.

Пусть цепь $\mathcal{N}$ состоит из состояний $S_{i_1}, \ldots, S_{i_p}$. Для краткости изложения последующих вычислений обозначим $\pi = P(S_{i_1} \in \mathcal{L})$, $p_1 = P(S \in \mathcal{L})$, $p_2 = P(S \in \mathcal{N})$. Тогда вероятность каждого случая, представленного на рисунке 3.4 записывается как:

$$P(Case_i) = P(\mathcal{L}) \cdot P(\mathcal{N}) \cdot P(S) \cdot P(Links_i) \ , \tag{3.9}$$

где $i = \overline{1..6}$, и последний множитель определяет вероятности связей и имеет следующее значение в каждом из случаев соответственно:

$$\begin{aligned}
P(Links_1) &= \pi(1 - (1 - p_1)(1 - p_2)) \ , \\
P(Links_2) &= \pi(1 - p_1)(1 - p_2) \ , \\
P(Links_3) &= (1 - \pi)p_1 p_2 \ , \\
P(Links_4) &= (1 - \pi)p_1(1 - p_2) \ , \\
P(Links_5) &= (1 - \pi)(1 - p_1)p_2 \ , \\
P(Links_6) &= (1 - \pi)(1 - p_1)(1 - p_2) \ .
\end{aligned}$$

Докажем, что если на некотором шаге алгоритма для фиксированных зна-

чений переменных состояний вероятность объединения этих состояний в одну цепь больше вероятности создания двух цепей, то, если переменные состояния в итоговом наиболее вероятном пути будут иметь те же значения, то они также будут принадлежать одной цепи.

Утверждение 1. *Если цепь $\mathcal{N}$ является частью наиболее вероятного пути, и первое состояние S_{i_1} цепи $\mathcal{N}$ принадлежит цепи $\mathcal{L}$ с вероятностью более $\frac{1}{2}$, тогда $\mathcal{L}$ также является частью наиболее вероятного пути, а $\mathcal{N}$ является продолжением цепи $\mathcal{L}$.*

Другими словами, утверждается что для любых $i = \overline{1..6}$:

$$P(Case_1) \geq P(Case_2) \Rightarrow P(Case_1) \geq P(Case_i) \ ,$$
$$P(Case_2) \geq P(Case_1) \Rightarrow P(Case_2) \geq P(Case_i) \ .$$

Доказательство. Условие $P(S_{i_1}) > \frac{1}{2}$ означает, что $\pi > 1 - \pi$, то есть $\mathcal{L}$ и $\mathcal{N}$ образуют одну цепь в наиболее вероятном пути. Введем обозначение $\xi = P(\mathcal{L})P(\mathcal{N})P(S)$. Тогда,

$$\begin{aligned}\mathbf{P(Case_1)} &= \xi\pi(1 - (1 - p_1)(1 - p_2)) = \\ &= \xi\pi(p_1 + p_2 - p_1p_2) \geq [p_1 \geq 0, p_2 \geq 0] \geq \\ &\geq \xi\pi(p_1 + p_2) \geq [0 \leq p_1 \leq 1, 0 \leq p_2 \leq 1] \geq \\ &\geq \xi\pi(p_1^2 + p_2^2) \geq [\text{Неравенство Коши}] \geq \\ &\geq 2\xi\pi p_1p_2 > 2\xi(1 - \pi)p_1p_2 > \mathbf{P(Case_3)}.\end{aligned}$$

$$\begin{aligned}\mathbf{P(Case_1)} &= \xi\pi(p_1 + p_2 - p_1p_2) \geq [p_2 \geq 0] \geq \\ &\geq \xi\pi(p_1 - p_1p_2) = \xi\pi p_1(1 - p_2) \geq \mathbf{P(Case_4)}.\end{aligned}$$

$$\begin{aligned}\mathbf{P(Case_1)} &= \xi\pi(p_1 + p_2 - p_1p_2) \geq [p_1 \geq 0] \geq \\ &\geq \xi\pi(p_2 - p_1p_2) = \xi\pi(1 - p_1)p_2 \geq \mathbf{P(Case_5)}.\end{aligned}$$

Пусть теперь $P(Case_1) \geq P(Case_2)$, тогда

$$\mathbf{P(Case_1)} \geq \xi\pi(1-p_1)(1-p_2) \geq \mathbf{P(Case_6)}.$$

Теперь рассмотрим случай $P(Case_2) \geq P(Case_1)$. Выше было показано, что вероятность первого случая больше или равна вероятностям 3–5 случаев, безотносительно к начальным условиям, поэтому:

$$\mathbf{P(Case_2)} \geq \mathbf{P(Case_1)} \geq \mathbf{P(Case_i)}, \forall i = 3, 4, 5$$

Для завершения доказательства заметим что если $\pi \geq 1 - \pi$, то

$$\mathbf{P(Case_2)} \geq \mathbf{P(Case_6)} \ .$$

□

Следующее утверждение устанавливает, что новое состояние может повлиять на соединение между уже существующими цепями, но только между активными.

Утверждение 2. *Для любых $\mathcal{L}$ и $\mathcal{N}$, являющихся отдельными активными цепями (случаи 3–6 на рисунке 3.4), существует такое состояние S, что наиболее вероятный путь до S содержит $\mathcal{L}$ и $\mathcal{N}$, в качестве частей одной цепи, и S принадлежит этой объединенной цепи (случай 1 на рисунке 3.4).*

То есть, необходимо проверить что существуют p_1, p_2 и π, такие что для всех $j = \overline{3..6}$:

$$P(Case_1) > P(Case_j) \ ,$$

и, если $\pi < 1 - \pi$, то для всех $i = \overline{1..6}$:

$$P(Case_2) \leq P(Case_i) \ .$$

Доказательство. Решая систему уравнений:

$$\begin{cases} \pi < 1 - \pi, \\ P(Case_1) > P(Case_j), \forall j = \overline{3..6} \end{cases}$$

можно найти π, p_1 и p_2. В частности этой системе удовлетворяют $p_1 = p_2 = \frac{1}{3}$, $\pi = 0,49$, что доказывает первую часть утверждения.

Вторая часть утверждения следует из следующего неравенства, учитывая что $\pi < 1 - \pi$:

$$\max_{i=\overline{3..6}} P(Case_i) \geq P(Case_6) > [\pi < 1 - \pi] > P(Case_2).$$

□

Теорема 1 обобщает утверждения 1 и 2 на случай, когда Λ состоит более чем из двух цепей. То есть доказывается два утверждения:

1. если на некотором шаге алгоритма для фиксированных значений переменных состояний вероятность объединения этих состояний в одну цепь больше вероятности создания нескольких цепей, то, если переменные состояния в итоговом наиболее вероятном пути будут иметь те же значения, то они также будут принадлежать одной цепи;

2. на любом шаге алгоритма, для любого количества различных активных цепей, существуют такие значения текущего состояния S и состояний, входящих в эти цепи, что вероятность случая, где все n цепей различны, а часть из них, возможно, объединяется состоянием S, меньше чем вероятность случая, где все эти состояния образуют одну цепь.

Используя рекурсивную структуру алгоритма 1, сформулируем теорему для любого промежуточного состояния S.

Теорема 1.

1. $\forall S \in \mathbb{S}$, $\forall \mathcal{L} \in \Lambda$, $\mathcal{L} = \overline{\mathcal{L}_1, \ldots, \mathcal{L}_n}$, $\forall n \in \mathbb{N}$, $\forall k = \overline{1, n}$*:*

$$P(\overline{\mathcal{L}S}) \geq P(\mathcal{L}, S) \Rightarrow P(\overline{\mathcal{L}S}) \geq \prod_{i=1}^{k-1} P(\mathcal{L}_i) \times P(\overline{\mathcal{L}_k S}) \times \prod_{i=k+1}^{n} P(\mathcal{L}_i) \ ,$$

$$P(\mathcal{L}, S) \geq P(\overline{\mathcal{L}S}) \Rightarrow P(\mathcal{L}, S) \geq \prod_{i=1}^{k-1} P(\mathcal{L}_i) \times P(\overline{\mathcal{L}_k S}) \times \prod_{i=k+1}^{n} P(\mathcal{L}_i) \ .$$

2. $\forall \lambda \in \Lambda$, $\lambda = \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$, $\exists S \in \mathbb{S}$*:*

$$P(\overline{\mathcal{L}_1 \ldots \mathcal{L}_n S}) \geq P(\mathcal{L}_{i_1}, \ldots \mathcal{L}_{i_k}, \overline{\eta S}) \ ,$$

где $\eta = \{\mathcal{L}_{i_{k+1}}, \ldots, \mathcal{L}_{i_n}\}$.

Доказательство. Проверим утверждения теоремы с помощью индукции по n, с базисом индукции $n = 2$, проверенным выше в утверждениях 1 и 2. Докажем обобщение неравенств, приведенных в этих утверждениях.

1. Обобщение неравенства $P(Case_1) \geq P(Case_3)$ из Утверждения 1 для n цепей. В этом неравенстве вероятность одной связанной цепи, без учета нового состояния больше, чем вероятность n различных цепей. поэтому достаточно проверить, что вероятность присоединения нового состояния к этой связанной цепи больше вероятности присоединения к нескольким различным цепям. Формально, необходимо проверить, что $1 - \prod_{i=1}^{n}(1 - p_i) \geq \prod_{i=1}^{n} p_i$:

$$1 - \prod_{i=1}^{n}(1 - p_i) = 1 - (1 - p_n) \prod_{i=1}^{n-1}(1 - p_i) =$$

$$= 1 - \prod_{i=1}^{n-1}(1 - p_i) + p_n \prod_{i=1}^{n-1}(1 - p_i) \geq [\text{по индукции}] \geq$$

$$\geq \prod_{i=1}^{n-1} p_i + p_n \prod_{i=1}^{n-1}(1 - p_i) > \prod_{i=1}^{n-1} p_i + p_n \geq \prod_{i=1}^{n} p_i \ .$$

2. Обобщение неравенства $P(Case_1) \geq P(Case_4)$ для n цепей: текущее состояние объединяет k цепей, не включая n-ую цепь. Необходимо проверить, что $1 - \prod_{i=1}^{n}(1-p_i) \geq \prod_{i=1}^{k} p_i \prod_{j=k+1}^{n}(1-p_j)$:

$$
\begin{aligned}
&1 - \prod_{i=1}^{n}(1-p_i) = 1 - (1-p_n)\prod_{i=1}^{n-1}(1-p_i) = \\
=&1 + (1-p_n)(1 - \prod_{i=1}^{n-1}(1-p_i) + 1) \geq [\text{по индукции}] \geq \\
\geq&1 + (1-p_n)(\prod_{i=1}^{k} p_i \prod_{j=k+1}^{n-1}(1-p_j) + 1) = \\
=&2 - p_n + \prod_{i=1}^{k} p_i \prod_{j=k+1}^{n}(1-p_j) > \prod_{i=1}^{k} p_i \prod_{j=k+1}^{n}(1-p_j) \ .
\end{aligned}
$$

3. Обобщение неравенства $P(Case_1) \geq P(Case_5)$ для n цепей: текущее состояние соединяет k цепей, включая n-ую цепь, $k \leq n$. Формально необходимо доказать что: $1 - \prod_{i=1}^{n}(1-p_i) \geq \prod_{i=n-k+1}^{n} p_i \prod_{j=1}^{n-k}(1-p_j)$:

$$
\begin{aligned}
&1 - \prod_{i=1}^{n}(1-p_i) = 1 - (1-p_n)\prod_{i=1}^{n-1}(1-p_i) = \\
=&1 - \prod_{i=1}^{n-1}(1-p_i) + p_n \prod_{j=1}^{n-1}(1-p_j) \geq [\text{по индукции}] \geq \\
\geq&\prod_{j=1}^{n-k}(1-p_j) \prod_{i=n-k+1}^{n-1} p_i + p_n \prod_{j=1}^{n-1}(1-p_j) > \\
>&\prod_{j=1}^{n-k}(1-p_j) \prod_{i=n-k+1}^{n-1} p_i + p_n \prod_{j=1}^{n-k}(1-p_j) = \\
=&(\prod_{i=n-k+1}^{n-1} p_i + p_n)\prod_{j=1}^{n-k}(1-p_j) > \prod_{i=n-k+1}^{n} p_i \prod_{j=i}^{n-k}(1-p_j) \ .
\end{aligned}
$$

Обобщение остальных случаев проверяется таким же способом, как в Утверждениях 1 и 2. □

Теорема утверждает, что во время итеративного поиска наиболее вероятной последовательности состояний, на каждом шаге нет необходимости рассматривать отдельные части уже соединенных цепей, и показывает, что различные цепи могут быть преобразованы в одну на последующих шагах, однако это соединение может произойти, только если эти цепи остаются активными. Таким образом, эта теорема позволяет сократить пространство поиска наиболее вероятного пути.

Теперь можно записать функцию *computePath* для слабой модели. Определение этой функции представлено в Алгоритме 2.

Algorithm 2 Функция *computePath* для слабой модели

Input: previousPath, newState

Output: newPath

1: $chainsSet = getChainsCombinations(previousPath)$
2: **for all** $joinedChain \in chainsSet$ **do**
3: Compute $P(newState \in joinedChain)$
4: **end for**
5: return $\arg\max_{processed\ paths} P(path)$

Локальная переменная *chainsSet* в Алгоритме 2 содержит множество всевозможных комбинаций активных цепей. Одним из способов создания такого множества является использование рекурсивной функции, основанной на следующем рассуждении. Представим активные цепи в виде узлов графа; соединенные узлы представляют собой соединенные цепи. Тогда *chainsSet* является множеством всевозможных неупорядоченных разбиений множества узлов, где каждая компонента связная. Пусть $S(n, k)$ — множество, состоящее из всех возможных разбиений n узлов графа на k связных компонент. Множество $S(n, k)$ может быть создано добавлением узла к одной из связных компонент для каждого элемента в $S(n-1, k)$ или добавлением новой

компоненты, состоящей из одного узла, к множеству $S(n-1,k-1)$. Таким образом,

$$|S(n,k)| = |S(n-1,k)| \cdot k + |S(n-1,k-1)| \,. \tag{3.10}$$

Последнее уравнение задает рекурсивную формулу для чисел Стирлинга второго рода. Используя уравнение (3.10), получим число элементов множества $chainSet$, вычисляемое следующим образом:

$$|chainsSet| = \sum_{k=0}^{n} S(n,k) = B_n \ ,$$

где B_n — числа Белла. Таким образом, было доказано утверждение

Утверждение 3. *Для каждого пути, обрабатываемого Алгоритмом 1, Алгоритм 2 обрабатывает B_n цепей, где n — число активных цепей в пути.*

Полная модель

Для полной модели, уравнение (3.9) принимает более сложную форму:

$$P(Case_i) = P(\mathcal{L}) \cdot P(\mathcal{N} \mid \mathcal{L}) \cdot P(S \mid \mathcal{LN}) \cdot P(Links_i) \ . \tag{3.11}$$

Утверждения 1 и 2 и Теорема 1 перестают выполняться для полной модели. В самом деле, даже если цепь $\mathcal{N}$ является продолжением цепи $\mathcal{L}$, нельзя утверждать о каких-либо априорных отношениях между ними в конечном, наиболее вероятном пути. В терминах рисунка 3.4 нельзя заранее выбрать никакой из случаев, из-за множителя $P(S \mid \mathcal{LN})$ в уравнении (3.11). Для каждого из случаев, этот множитель не позволяет заранее оценить соотношение с другими случаями (за исключением полностью совпадающих множителей). Например, вероятность $P(S \mid \mathcal{LN})$ в случае 1 может быть и больше и меньше, чем вероятность $P(S \mid \mathcal{L})$ в случае 4.

Для каждого пути, Алгоритму необходимо обработать все активные цепи, которые потенциально могут быть сформированы. Определение функции

computePath остается таким же, как и в Алгоритме 2, с единственным различием, что *chainsSet* теперь содержит множество комбинаций всех возможных цепей. Число таких комбинаций также определяется числом Белла B_n, но n теперь означает число всех активных состояний пути.

В то время как вычисления в слабой модели часто ограничиваются одной связной цепью, алгоритм для полной модели обрабатывает все возможные комбинации активных цепей для каждого пути. Это свойство полной модели увеличивает среднюю вычислительную сложность Алгоритма 1. Что касается качества результатов, предоставляемых каждой моделью, сравнительные оценки слабой и полной моделей представлены в разделе 3.4.5.

3.4.4. Применение модели к задаче устранения лексической многозначности

Чтобы применить описанную модель к задаче устранения лексической многозначности, необходимо оценить три параметра модели:

1. Вероятность $P(\widehat{m_i m_j})$ — два значения принадлежат одной цепи;
2. Модель перехода $P(m_i \mid m_{i-h:i-1})$;
3. Модель наблюдения $P(t_i \mid m_i)$.

Для этого воспользуемся словарем и ссылочной структурой Википедии.

Для оценки *вероятности, что два значения принадлежат одной цепи* вводится следующая эвристика:

Эвристика 3. *вероятность события, что два значения принадлежат одной цепи, является функцией от семантической близости:*

$$P(\widehat{m_1 m_2}) = \phi(sim(m_1, m_2)) \ . \tag{3.12}$$

Функция ϕ может быть получена из корпуса с размеченными лексическими цепями. Чтобы избежать ручной обработки, воспользуемся идеей описанной в работе [118], о существовании корреляции между разбиением на лексические цепи и кластеризацией графа концепций, описанной в работе. Представим каждый документ некоторой коллекции в виде взвешенного графа, где вершинами являются концепции, соответствующие терминам документа, а ребра между вершинами имеют вес равный семантической близости концепций. Заметим, что никакая дополнительная обработка текста не требуется и берутся все возможные концепции для каждого найденного в тексте термина.

Далее, к каждому полученному графу применяется алгоритм кластеризации [119]. Две концепции считаются принадлежащими одной лексической цепи, если соответствующие вершины графа принадлежат одному кластеру. Таким образом, получаем тренировочный корпус для оценки функции ϕ, где пары концепций, принадлежащих одному кластеру, являются положительными примерами, а отрицательные примеры формируются парами концепций, принадлежащих одному документу, но разным кластерам. Было получено множество из 137,324 положительных и 859,076 отрицательных примеров, на их основе из пространства ступенчатых функций с шагом 0.01 выбирается функция ϕ, представленная на рисунке 3.5.

Модель перехода оценивается таким же образом, как и для классической модели (раздел 3.3.1). Воспользуемся Эвристикой 1:

$$P(m_i \mid m_{i-1}) = \alpha \cdot (sim(m_i \mid m_{i-1}) + \beta \cdot P(m_i)) \quad . \tag{3.13}$$

Коэффициент α не влияет на результат задачи максимизации (Алгоритм 1), поэтому может не учитываться. Кроме того, если текущее состояние не зависит от предыдущего и является первым состоянием новой цепи, правая часть уравнения 3.13 должна редуцироваться к априорной вероятности, отсюда, естественным образом получаем коэффициент $\beta = 1$.

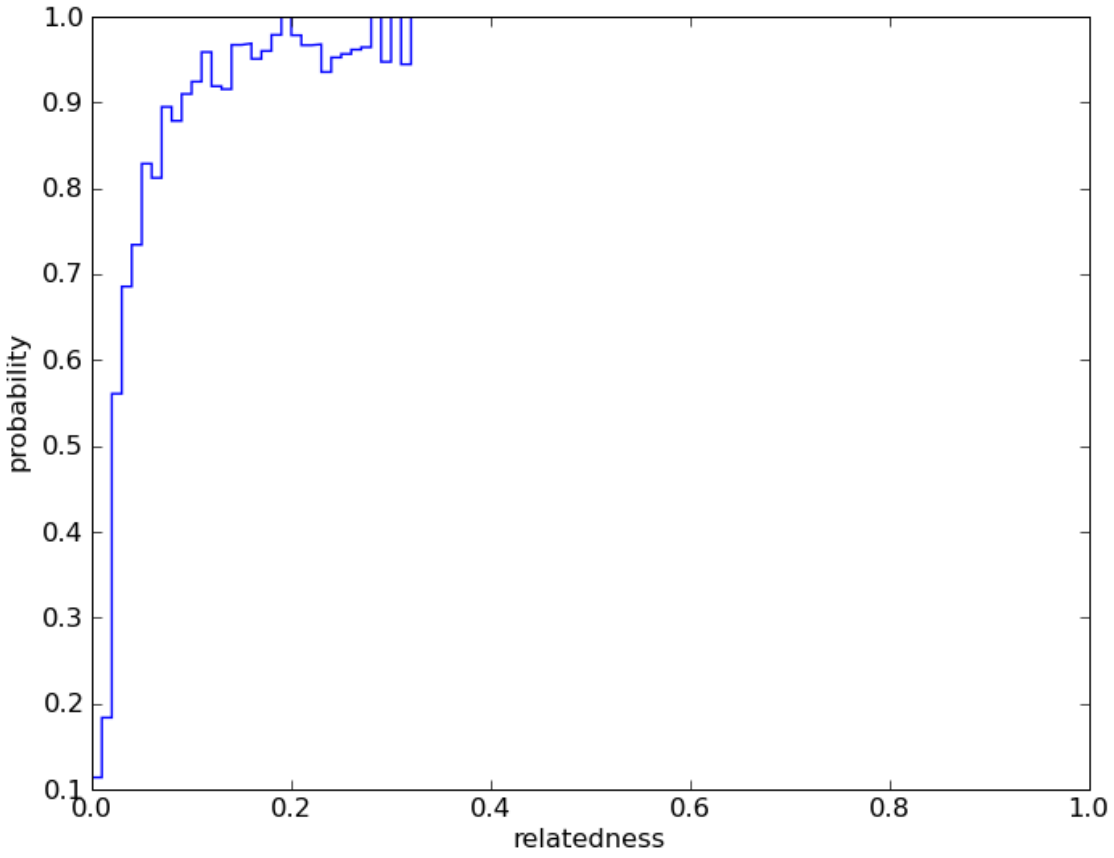

Рис. 3.5. Функция ϕ полученная на основе корпуса, содержащего автоматически найденные лексические цепи

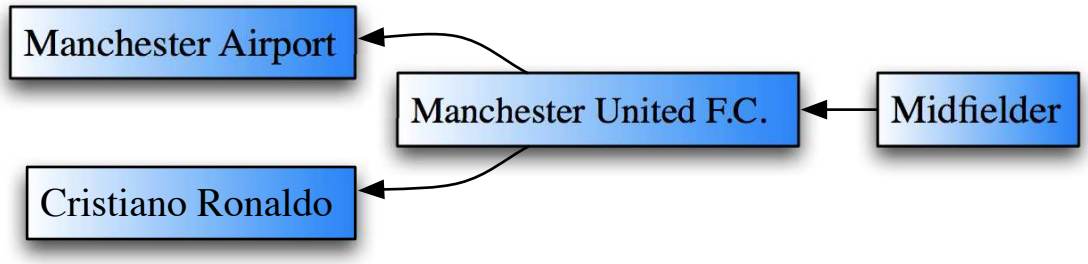

Рис. 3.6. Пример полученной лексической цепи

Модель наблюдения оценивается в точности таким же способом, как и в разделе 3.3.1.

Имея оценки параметров обобщенной марковской модели, можно применить ее к последовательности терминов, содержащихся во входном тексте, и получить наиболее вероятную последовательность значений, то есть устранить лексическую многозначность во входном тексте.

Для иллюстрации работы алгоритма приведем пример. Рассмотрим часть новостной статьи об известном футболисте, который разбил свою машину:

> *Cristiano Ronaldo* crashed his *Ferrari* early this *morning* in the *tunnel* under *Manchester Airport*. The *Manchester United winger* was his way to practice when he totaled the $400,000 *sports car*.

Термины из словаря Википедии выделены наклонным шрифтом. На основании работы метода был получен следующий результат:

> [Cristiano Ronaldo : Cristiano Ronaldo : 1] crashed his [Ferrari : Ferrari : 2] early this [morning : Morning : 3] in the [tunnel : Tunnel : 4] under [Manchester Airport : Manchester Airport : 1]. The [Manchester United : Manchester United F.C. : 1] [winger:Midfielder : 1] was his way to practice when he totaled the $400,000 [sports car : Sports car : 2].

В квадратных скобках представлены тройки

[выделенный термин : значение выделенного термина : номер цепи].

В данном фрагменте точность метода составила 100%. Было выделено четыре различные цепи, и, как и предполагалось в секции 3.4.1, термины, связанные с футболистом и спортивными машинами, попали в различные лексические цепи. Рассмотрим цепь №1 (рис. 3.6). В процессе обработки текста термины *Cristiano Ronaldo* и *Manchester Airport* попали в разные лексические цепи. Однако при последующей обработке они были объединены в одну цепь, значением *Manchester United F.C.*, которое связано со значениями обоих терминов.

В следующем разделе оценивается качество результатов алгоритма разрешения лексической многозначности, основанного на предложенной модели.

3.4.5. Эксперименты

Эксперименты проводились на коллекциях, описанных в разделе 3.2.2. Результаты представлены в таблице 3.12. Предложенный алгоритм применялся ко всем терминам, имеющим, по крайней мере, одну концепцию Википедии, таким образом, точность и полнота совпадают. Как следствие, для значений терминов, не имеющих соответствующих концепций, алгоритм не может выдать правильных результат (см. раздел 3.2.1). Для того, чтобы оценить качество алгоритма, вне зависимости от покрытия Википедии, такие термины были удалены и тестовых коллекции.

Показатели, которые позволяет достичь описанная модель (даже в случае использования слабой модели), значительно превышают точность и полноту, которые можно получить на основе классической марковской модели. Также, эксперименты показали, что слабая модель позволяет получить хорошие результаты, лишь немного хуже, чем те, которые демонстрирует полная модель.

Для сравнения в таблице 3.12 представлены результаты работы наиболее

	Wikipedia articles	Milne and Witten [101]	News and scient. papers
Метод, использующий однозначный контекст	85.12	78.81	64.34
Наиболее вероятное значение	90.10	83.10	67.61
HMM-1	90.13	83.30	67.61
HMM-2	91.51	83.69	67.72
$h = 2$, $m = 0$	94.36	90.00	75.25
$h = 2$, $m = 1$	94.68	90.18	75.60
$h = 2$, $m = 2$	93.87	89.80	74.53
$h = 3$, $m = 0$	94.67	89.41	76.04
$h = 3$, $m = 1$	94.72	90.38	75.96
$h = 3$, $m = 2$	**94.98**	90.38	**76.73**
Milne and Witten [101]	94.85	**91.78**	76.49

Таблица 3.12. Точность (%) различных методов устранения лексической многозначности

точного метода, представленного в современной литературе, описанного в статье [101]. Для корректного сравнения все методы использовали один словарь, созданный на основе снимка Википедии, сделанного в марте 2009 г. Сравнение проводилось на тестовых коллекциях, описанных выше.

Метод, представленный в данной работе, и основанный на обобщении марковской модели, не уступает по точности методу Дэвида Милна и Яна Уиттена [101]. Однако в дополнение к устранению лексической многозначности, предложенный метод разбивает текст на лексические цепи, что может быть полезно многим практическим приложениям, например, для автоматического аннотирования текстов.

3.4.6. Выводы

В разделе 3.4 представлено обобщение марковской модели на случай множества независимых марковских цепей и приложение предложенной модели к задаче устранения лексической многозначности. Сопутствующим результатом описанного алгоритма является сегментация концепций и соответствующих терминов в лексические цепи. Представленный алгоритм показывает значительно лучшие результаты, чем алгоритм разрешения лексической многозначности, основанный на классической марковской модели.

Одним из возможных способов улучшить результаты алгоритма является исследование более тонких стратегий выбора активных состояний. Предположение 1 сделанное в этой главе является более слабым, чем марковское предположение, так как в данном подходе возможно сохранять конечное число состояний активными для использование при обработке любых частей текста. Например, разумно сохранять активными концепции, ассоциированные с терминами из заголовка документа. Такая модификация сделает темы, описанные в заголовке документа, активными на протяжении всего текста и поз-

волит сформировать цепи, соответствующие этим основным темам.

Другим способом улучшения результатов разрешения лексической многозначности является исследование функции ϕ, устанавливающей вероятность того, что две концепции принадлежат одной цепи, на основе их семантической близости. Эта функция является главным параметром алгоритма, определяющим длину получившихся цепей. Таким образом, разные реализации функции ϕ позволят получить различные цепи с различной длиной. В этой работе предлагается один из способов вычисления данной функции; однако возможно предложить другие реализации, позволяющие получить лучшие результаты при снятии лексической многозначности. Такой анализ является хорошей темой для дальнейшей работы.

3.5. Выводы к третей главе

В данной главе описаны три алгоритма снятия лексической многозначности именных фраз, использующие семантическую близость концепций Википедии для выбора наиболее подходящего значения термина в каждом конкретном случае.

Преимуществом первого, наиболее простого, алгоритма является легкость понимания причин выбора конкретного значения и скорость обработки текстов. Главным недостатком, служит то, что при выборе значения он опирается на однозначный контекст, которого может и не существовать в конкретном тексте. Более того, с ростом Википедии, растет количество многозначных терминов, а следовательно увеличивается доля документов, содержащих исключительно такие термины.

Решение проблемы однозначного контекста состоит в использовании моделей, позволяющих решать задачу устранения лексической многозначности методами оптимизации. Второй из предложенных методов адаптирует мар-

ковскую модель для решения данной задачи. Основной проблемой при использовании марковской модели, является оценка ее параметров. В данной главе показано, как с помощью семантической близости и ссылок Википедии можно оценить модели наблюдения и перехода. Однако марковская модель позволяет описать только последовательности терминов, относящихся к одной теме. Для решения этой проблемы предложено обобщение марковской модели на случай множества независимых цепей.

Алгоритм, основанный на обобщенной марковской модели, показывает результаты, превосходящие все результаты, представленные в современной литературе, на основании этого, можно сделать вывод, что обобщенная марковская модель является хорошей моделью для разрешения лексической многозначности терминов текста.

Заключение

В ходе диссертационной работы получены следующие результаты:

1. Предложен подход к разрешению лексической многозначности терминов на основе сети документов Википедии.

2. Предложен метод измерения семантической близости узлов взвешенной сети документов.

3. В рамках предложенного подхода разработаны и формально обоснованы мето-ды разрешения лексической многозначности терминов на основе структурной и текстовой информации сетей документов с использованием: контекста из однозначных терминов; Марковской модели высокого порядка; обобщения Марковской модели.

4. Для экспериментального подтверждения эффективности предложенных методов разработан прототип системы разрешения лексической многозначности терминов Википедии и проведены эксперименты, доказывающие эффективность предложенных методов.

5. Разработанный прототип был использован в качестве основы для создания в Институте системного программирования РАН системы анализа текстов Texterra.

Литература

[1] *Denis Turdakov.* Recommender System Based on User-generated Content // Proceedings of the SYRCODIS 2007 Colloquium on Databases and Information Systems. — 2007.

[2] *Denis Turdakov, Pavel Velikhov.* Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation // Proceedings of the SYRCODIS 2008 Colloquium on Databases and Information Systems. — 2008.

[3] *Dmitry Lizorkin, Pavel Velikhov, Maxim Grinev, Denis Turdakov.* Accuracy estimate and optimization techniques for SimRank computation // *Proceedings of the 34rd International Conference on Very Large Data Bases.* — 2008. — Vol. 1, no. 1. — Pp. 422–433.

[4] *Maria Grineva, Maxim Grinev, Denis Turdakov et al.* Harnessing Wikipedia for Smart Tags Clustering // KASW: International Workshop on «Knowledge Acquisition from the Social Web». — 2008.

[5] *Д. Ю. Турдаков, С. Д. Кузнецов.* Автоматическое разрешение лексической многозначности терминов на основе сетей документов // *Программирование.* — 2010. — Vol. 36, no. 1. — Pp. 11–18.

[6] *Турдаков Денис.* Устранение лексической многозначности терминов Википедии на основе скрытой модели Маркова // XI Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». — 2009.

[7] *Dmitry Lizorkin, Pavel Velikhov, Maxim Grinev, Denis Turdakov.* Accuracy

estimate and optimization techniques for SimRank computation // *The VLDB Journal.* — 2009. http://dx.doi.org/10.1145/1453856.1453904.

[8] *Denis Turdakov, Dmitry Lizorkin.* HMM Expanded to Multiple Interleaved Chains as a Model for Word Sense Disambiguation // Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation. — Hong Kong: City University of Hong Kong, 2009. — December. — Pp. 549–558.

[9] *George A. Miller, Richard Beckwith, Christiane Fellbaum et al.* WordNet: An on-line lexical database // *International Journal of Lexicography.* — 1990. — Vol. 3. — Pp. 235–244.

[10] Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology), Ed. by E. Agirre, P. G. Edmonds. — 1 edition. — Springer, 2007. — November.

[11] *Nancy Ide, Jean Véronis.* Word Sense Disambiguation: The State of the Art // *Computational Linguistics.* — 1998. — Vol. 24. — Pp. 1–40.

[12] *Gerard. Salton.* Automatic Information Organization and Retrieval. — McGraw Hill Text, 1968.

[13] *Kenneth C. Litowski.* Desiderata for tagging with WordNet sysnsets or MCAA categories // In Proceedings of the ACL-SIGLEX Workshop "Tagging Text with Lexical Semantics: Why, What, and How?"pages 12–17. — Washington, DC, 1997. — April.

[14] *Stephanie Seneff.* TINA: a natural language system for spoken language applications // *Comput. Linguist.* — 1992. — Vol. 18, no. 1. — Pp. 61–86.

[15] *David Yarowsky.* Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French // Proceedings of the 32nd annual meeting on Association for Computational Linguistics. — Morristown, NJ, USA: Association for Computational Linguistics, 1994. — Pp. 88–95.

[16] *M. Grineva, M. Grinev, D. Lizorkin.* Effective Extraction of Thematically Grouped Key Terms From Text // AAAI-SSS-09: Social Semantic Web: Where Web 2.0 Meets Web 3.0. — 2009.

[17] *Maria Grineva, Maxim Grinev, Dmitry Lizorkin.* Extracting Key Terms From Noisy and Multi-theme Documents // 18th International World Wide Web Conference. — 2009. — April. — Pp. 661–661.

[18] *Аристотель.* Категории // Аристотель. Сочинения: в 4 т. Т.2–4 / ред. З.Н.Микеладзе. — М.: Мысль, 1978–1984.

[19] *Розенталь Д.Э., Голуб И.Б., Теленкова М.А.* Современный русский язык.

[20] *Jesus Gimenez, Lluis Marquez.* SVMTool: A general POS tagger generator based on Support Vector Machines. — 2004.

[21] *Robert Malouf.* A comparison of algorithms for maximum entropy parameter estimation // COLING-02: proceeding of the 6th conference on Natural language learning. — Morristown, NJ, USA: Association for Computational Linguistics, 2002.

[22] *Roger C. Schank.* Conceptual Information Processing. — Amsterdam: North Holland, 1975.

[23] *В. В. Виноградов.* Основные типы лексических значений слова // *"Вопросы языкознания".* — 1953.

[24] *Abraham Kaplan.* An experimental study of ambiguity and context // *Mechanical Translation.* — 1955. — Vol. 2, no. 2. — Pp. 39–46.

[25] *David Yarowsky.* One sense per collocation // HLT '93: Proceedings of the workshop on Human Language Technology. — Morristown, NJ, USA: Association for Computational Linguistics, 1993. — Pp. 266–271.

[26] *W. A. Gale, K. W. Church, D. Yarowsky.* A method for disambiguating word senses in a large corpus. // Computers and the Humanitzes. — Vol. 26. — 1993. — Pp. 415–439.

[27] *William A. Gale, Kenneth W. Church, David Yarowsky.* One sense per discourse // HLT '91: Proceedings of the workshop on Speech and Natural Language. — Morristown, NJ, USA: Association for Computational Linguistics, 1992. — Pp. 233–237.

[28] *Korin Richmond, Andrew Smith, Einat Amitay.* Detecting Subject Boundaries Within Text: A Language Independent Statistical Approach // In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, EMNLP-2. — Providence, RI: Brown University, 1997. — August. — Pp. 47–54.

[29] *Terry Winograd.* Procedures as a Representation for Data in a Computer Program for Understanding Natural Language: Tech. rep.: 1971.

[30] *George A. Miller, Claudia Leacock, Randee Tengi, Ross T. Bunker.* A semantic concordance // HLT '93: Proceedings of the workshop on Human Language Technology. — Morristown, NJ, USA: Association for Computational Linguistics, 1993. — Pp. 303–308.

[31] *Nelson W. Francis, Henry Kučera.* Frequency Analysis of English Us-

age: Lexicon and Grammar. — Boston: Houghton Mifflin, 1982. — April. — Vol. 18. — Pp. 64–70.

[32] *Claudia Leacock, Geoffrey Towell, Ellen Voorhees.* Corpus-based statistical sense resolution // HLT '93: Proceedings of the workshop on Human Language Technology. — Morristown, NJ, USA: Association for Computational Linguistics, 1993. — Pp. 260–265.

[33] *Rebecca Bruce, Janyce Wiebe.* Word-Sense Disambiguation Using Decomposable Models // Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics. — 1994. — Pp. 139–146.

[34] *Hwee Tou Ng, Hian Beng Lee.* Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach // Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics / Ed. by A. Joshi, M. Palmer. — San Francisco: Morgan Kaufmann Publishers, 1996. — Pp. 40–47.

[35] *Adam Kilgarriff.* SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs // In LREC. — 1998. — Pp. 581–588.

[36] *Sue Atkins.* Tools for computer-aided corpus lexicography: The Hector project. — 1993. — Vol. 41.

[37] *Martha Palmer, Christiane Fellbaum, Scott Cotton et al.* English tasks: All-words and verb lexical sample // Proceedings of Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems. — Toulouse, France: 2001. — P. 21–24.

[38] *Rada Mihalcea, Philip Edmonds* // Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. — Barcelona, Spain: 2004.

[39] *Timothy Chklovski, Rada Mihalcea.* Building a sense tagged corpus with open mind word expert // Proceedings of the ACL-02 workshop on Word sense disambiguation. — Morristown, NJ, USA: Association for Computational Linguistics, 2002. — Pp. 116–122.

[40] *R. V. Guha, Douglas B. Lenat.* CYC: a mid-term report // *Appl. Artif. Intell.* — 1991. — Vol. 5, no. 1. — Pp. 45–86.

[41] *Mitchell P. Marcus, Mary Ann Marcinkiewicz, Beatrice Santorini et al.* Building a Large Annotated Corpus of English: The Penn Treebank. — 2004.

[42] *Noam Chomsky.* Syntactic Structures. — Mouton, The Hague, 1957.

[43] *Минский М.* Фреймы для представления знаний. — М.: Мир, 1979.

[44] *Richard H. Richens.* Interlingual machine translation // Computer Journal. — Vol. 3. — 1958. — Pp. 144–147.

[45] *Margaret Masterman.* Semantic message detection for machine translation, using an interlingua // International Conference on Machine Translation of Languages and Applied Language Analysis. — London: Her Majesty's Stationery Office, 1962. — Pp. 437–475.

[46] *M. Ross Quillian.* The teachable language comprehender: a simulation program and theory of language // *Commun. ACM.* — 1969. — Vol. 12, no. 8. — Pp. 459–476.

[47] *Philip J Hayes.* A process to implement some word-sense disambiguation // *Working paper 23. Institut pour les Etudes Sémantiques et Cognitives. Université de Genéve.* — 1976.

[48] *Allan M. Collins, Elisabeth F. Loftus.* A spreading activation theory of

semantic processing // *Psychological Review.* — 1975. — Vol. 82, no. 6. — Pp. 407–428.

[49] *Michael Lesk.* Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone // SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation. — New York, NY, USA: ACM Press, 1986. — Pp. 24–26.

[50] *Claudia Leacock, George A. Miller, Martin Chodorow.* Using Corpus Statistics and WordNet Relations for Sense Identification. — 1998.

[51] *Graeme Hirst, David St-Onge.* Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. — 1997.

[52] *Philip Resnik Sun.* Using Information Content to Evaluate Semantic Similarity in a Taxonomy // In Proceedings of the 14th International Joint Conference on Artificial Intelligence. — 1995. — Pp. 448–453.

[53] *J. J. Jiang, D. W. Conrath.* Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy // International Conference Research on Computational Linguistics (ROCLING X). — 1997. — September.

[54] *Dekang Lin.* An Information-Theoretic Definition of Similarity // ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. — Pp. 296–304.

[55] *Rada Mihalcea, Dan I. Moldovan.* A method for word sense disambiguation of unrestricted text // Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. — Morristown, NJ, USA: Association for Computational Linguistics, 1999. — Pp. 152–158.

[56] *Eneko Agirre, German Rigau.* Word Sense Disambiguation using Conceptual Density // In Proceedings of the 16th International Conference on Computational Linguistics. — 1996. — Pp. 16–22.

[57] *Jiri Stetina, Sadao Kurohashi, Makoto Nagao.* General Word Sense Disambiguation Method Based on a Full Sentential Context // In Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop. — 1998.

[58] *Jane Morris, Graeme Hirst.* Lexical cohesion computed by thesaural relations as an indicator of the structure of text // *Comput. Linguist.* — 1991. — March. — Vol. 17, no. 1. — Pp. 21–48.

[59] *Rada Mihalcea, Dan I. Moldovan.* A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation // *International Journal on Artificial Intelligence Tools.* — 2001. — Vol. 10, no. 1-2. — Pp. 5–21.

[60] *Rada Mihalcea.* Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling // HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. — Morristown, NJ, USA: Association for Computational Linguistics, 2005. — Pp. 411–418.

[61] *Sergey Brin, Lawrence Page.* The Anatomy of a Large-Scale Hypertextual Web Search Engine // Computer Networks and ISDN Systems. — 1998. — Pp. 107–117.

[62] *R. Nelken, S.M. Shieber.* Lexical chaining and word-sense-disambiguation: Technical Report TR-06-07: School of Engineering and Applied Sciences, Harvard University, 2007.

[63] *Carsten Brockmann, Mirella Lapata.* Evaluating and combining approaches to selectional preference acquisition // EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics. — Morristown, NJ, USA: Association for Computational Linguistics, 2003. — Pp. 27–34.

[64] *Лукашевич Н.В., Добров Б.В.* Тезаурус русского языка для автоматической обработки больших текстовых коллекций // Компьютерная лингвистика и интеллектуальные технологии: Труды Международного семинара «Диалог 2002» / под ред. А.С. Нариньяни. — М.: Наука, 2002.

[65] *Добров Б.В., Лукашевич Н.В.* Онтологии для автоматической обработки текстов: описание понятий и лексических значений // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2006» (Бекасово, 31 мая - 4 июня 2006 г.) / под ред. Н.И. Лауфер, А. С. Нариньяни, В. П. Селегея. — М.: Изд-во РГГУ, 2006. — Pp. 138–142.

[66] *Добров Б.В., Лукашевич Н.В.* Разрешение лексической многозначности на основе тезауруса предметной области // Труды международной конференции «Диалог 2007». — 2007.

[67] *Н.В. Лукашевич, Д.С. Чуйко.* Автоматическое разрешение лексической многозначности на базе тезаурусных знаний // Сборник работ участников конкурса «Интернет-математика 2007». — 2007.

[68] *Martin Chodorow, Claudia Leacock, George A. Miller.* A topical local classifier for word sense identification // *Computers and the Humanities.* — 2000. — Vol. 34. — Pp. 115–120.

[69] *Adam L. Berger, Vincent J. Della Pietra, Stephen A. Della Pietra.* A maximum entropy approach to natural language processing // *Comput. Linguist.* — 1996. — Vol. 22, no. 1. — Pp. 39–71.

[70] *C. Fellbaum, M. Palmer.* Manual and Automatic Semantic Annotation with WordNet // Proceedings of NAACL 2001 Workshop. — 2001.

[71] *Tom O'Hara et al.* Selecting decomposable models for word sense disambiguation: the grling-sdm system // *Computers and the Humanities.* — 2000. — Vol. 34. — Pp. 159–164.

[72] *Rebecca F. Bruce, Janyce M. Wiebe.* Decomposable modeling in natural language processing // *Comput. Linguist.* — 1999. — Vol. 25, no. 2. — Pp. 195–207.

[73] *Walter Daelemans, Jakub Zavrel, Ko van der Sloot, Antal van den Bosch.* TiMBL: Tilburg Memory-Based Learner – version 4.0 – Reference Guide. — 2001.

[74] *Mark Stevenson, Yorick Wilks.* The interaction of knowledge sources in word sense disambiguation // *Comput. Linguist.* — 2001. — September. — Vol. 27, no. 3. — Pp. 321–349.

[75] *Hoa Trang Dang, Martha Palmer.* Combining Contextual Features for Word Sense Disambiguation // In Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions. — 2002. — Pp. 88–94.

[76] *Indrajit Bhattacharya, Lise Getoor, Yoshua Bengio.* Unsupervised sense disambiguation using bilingual probabilistic models // ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. —

Morristown, NJ, USA: Association for Computational Linguistics, 2004. — P. 287.

[77] *Плунгян В. А., Резникова Т. И., Сичинава Д. В.* Национальный корпус русского языка: общая характеристика // НТИ, сер. 2, 2005, № 3, 9-13.

[78] *Кобрицов Б.П.* Методы снятия семантической многозначности // Научно-техническая информация, сер.2, N 2. — 2004.

[79] *Кобрицов Б. П., Ляшевская О. Н.* Автоматическое разрешение семантической неоднозначности в Национальном корпусе русского языка // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог'2004». Москва, Наука, 2004.

[80] *Кобрицов Б. П., Ляшевская О. Н., Шеманаева О. Ю.* Поверхностные фильтры для разрешения семантической омонимии в текстовом корпусе // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог'2005» (Звенигород, 1-6 июня 2005 г.)/ Под ред. И. М. Кобозевой, А. С. Нариньяни, В. П. Селегея. — М.: Наука, 2005.

[81] *Кобрицов Б. П., Ляшевская О. Н., Шеманаева О. Ю.* Снятие лексико-семантической омонимии в новостных и газетно-журнальных текстах: поверхностные фильтры и статистическая оценка // Интернет-математика — 2005. Москва, 2005.

[82] *Кобрицов Б. П., Ляшевская О. Н., Толдова С. Ю.* Снятие семантической многозначности глаголов с использованием моделей управления, извлеченных из электронных толковых словарей. —

Электронная публикация. http://download.yandex.ru/IMAT2007/kobricov.pdf.

[83] *Шеманаева О. Ю., Кустова Г. И., Ляшевская О. Н., Рахилина Е. В.* Семантические фильтры для разрешения многозначности в Национальном корпусе русского языка: прилагательные // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2007». С. 582-587.

[84] *Filippo Menczer.* Evolution of document networks // *Proceedings of the National Academy of Sciences of the United States of America.* — 2004. — April. — Vol. 101, no. Suppl 1. — Pp. 5261–5265.

[85] *Adam Kilgarriff, Gregory Grefenstette.* Introduction to the Special Issue on the Web as Corpus // *Computational Linguistics.* — 2003. — Vol. 29. — Pp. 333–347.

[86] *A. L. Barabasi, R. Albert.* Emergence of scaling in random networks // *Science.* — 1999. — October. — Vol. 286, no. 5439. — Pp. 509–512.

[87] *P. Erdős, A. Rényi.* On random graphs. I // *Publ. Math. Debrecen.* — 1959. — Vol. 6. — Pp. 290–297.

[88] *Reka Albert, Hawoong Jeong, Albert-Laszlo Barabasi.* Error and attack tolerance of complex networks // *Nature.* — 2000. — July. — Vol. 406, no. 6794. — Pp. 378–382.

[89] *M. E. Newman.* Scientific collaboration networks. I. Network construction and fundamental results. // *Phys Rev E Stat Nonlin Soft Matter Phys.* — 2001. — July. — Vol. 64, no. 1 Pt 2.

[90] *M. E. J. Newman.* Clustering and preferential attachment in growing networks. // *Phys. Rev. E.* — 2001. — Vol. 64.

[91] *Lada A. Adamic, Rajan M. Lukose, Bernardo A. Huberman.* Local Search in Unstructured Networks // *CoRR.* — 2002. — Vol. cond-mat/0204181. — informal publication.

[92] *Reuven Cohen, Shlomo Havlin.* Scale-Free Networks Are Ultrasmall // *Physical Review Letters.* — 2003. — Feb. — Vol. 90, no. 5.

[93] *V. Zlatic, M. Bozicevic, H. Stefancic, M. Domazet.* Wikipedias: Collaborative web-based encyclopedias as complex networks // *Physical Review E.* — 2006. — Vol. 74. — P. 016115.

[94] *Justin Zobel, Alistair Moffat.* Exploring the similarity space // *SIGIR Forum.* — 1998. — Vol. 32, no. 1. — Pp. 18–34.

[95] *E. Gabrilovich, S. Markovitch.* Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis // *Proceedings of the 20th International Joint Conference on Artificial Intelligence.* — 2007. — Pp. 6–12.

[96] *Thomas K. Landauer, Peter W. Foltz, Darrell Laham.* An Introduction to Latent Semantic Analysis // *Discourse Processes.* — 1998. — no. 25. — Pp. 259–284.

[97] *Ana Gabriela Maguitman, Filippo Menczer, Fulya Erdinc et al.* Algorithmic Computation and Approximation of Semantic Similarity // *World Wide Web.* — 2006. — Vol. 9, no. 4. — Pp. 431–456.

[98] *W. N. Lee, N. Shah, K. Sundlass, M. Musen.* Comparison of ontology-based semantic-similarity measures. // *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium.* — 2008. — Pp. 384–388.

[99] *D. Milne*. Computing Semantic Relatedness using Wikipedia Link Structure // Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC). — Hamilton, New Zealand: 2007.

[100] *Michael Strube, Simone Paolo Ponzetto*. WikiRelate! Computing Semantic Relatedness Using Wikipedia. // 21. AAAI / 18. IAAI 2006. — AAAI Press, 2006. — july.

[101] *D. Milne, I.H. Witten*. Learning to link with Wikipedia // 17th ACM Conference on Information and knowledge management. — ACM, 2008. — Pp. 509–518.

[102] *M. Kessler*. Bibliographic coupling between scientific papers // *American Documentation*. — 1963. — Vol. 14. — Pp. 10–25.

[103] *Zhenjiang Lin, Irwin King, Michael R. Lyu*. PageSim: A Novel Link-Based Similarity Measure for the World Wide Web // WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. — Washington, DC, USA: IEEE Computer Society, 2006. — Pp. 687–693.

[104] *Eric Yeh, Daniel Ramage, Christopher D. Manning et al.* WikiWalk: Random walks on Wikipedia for Semantic Relatedness // Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4). — Suntec, Singapore: Association for Computational Linguistics, 2009. — August. — Pp. 41–49.

[105] *Glen Jeh, Jennifer Widom*. SimRank: a measure of structural-context similarity // KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. — ACM Press, 2002. — Pp. 538–543.

[106] *Torsten Zesch, Iryna Gurevych.* Analysis of the Wikipedia Category Graph for NLP Applications // Proceedings of the TextGraphs-2 Workshop (NAACL-HLT). — 2007.

[107] *Jim Giles.* Internet encyclopaedias go head to head // *Nature.* — 2005. — December. — Vol. 438. — Pp. 900–901.

[108] *Lotfi A. Zadeh.* Fuzzy Sets // *Information and Control.* — 1965. — Vol. 8, no. 3. — Pp. 338–353.

[109] *Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum.* Yago: A Large Ontology from Wikipedia and WordNet. — 2007.

[110] *Sören Auer, Christian Bizer, Georgi Kobilarov et al.* DBpedia: A Nucleus for a Web of Open Data. — 2008. — Pp. 722–735.

[111] *D. Milne, I.H. Witten.* An Open-Source Toolkit for Mining Wikipedia. — 2009.

[112] *D. Milne, I.H. Witten.* An effective, low-cost measure of semantic relatedness obtained from Wikipedia // AAAI 2008 Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WIKI-AI '08). — 2008.

[113] *Rada Mihalcea.* Using Wikipedia for Automatic Word Sense Disambiguation // North American Chapter of the Association for Computational Linguistics (NAACL 2007). — 2007.

[114] *Rada Mihalcea, Andras Csomai.* Wikify!: linking documents to encyclopedic knowledge // CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. — New York, NY, USA: ACM, 2007. — Pp. 233–242.

[115] *S. Cucerzan.* Large-Scale Named Entity Disambiguation Based on Wikipedia Data // *EMNLP 2007: Empirical Methods in Natural Language Processing, June 28-30, 2007, Prague, Czech Republic.* — 2007.

[116] *Razvan C. Bunescu, Marius Pasca.* Using Encyclopedic Knowledge for Named entity Disambiguation // EACL. — The Association for Computer Linguistics, 2006.

[117] *O. Medelyan, I. H. Witten, D. Milne.* Topic indexing with Wikipedia // 1st AAAI Workshop on Wikipedia and Artificial Intelligence. — 2008.

[118] *O. Medelyan.* Computing Lexical Chains with Graph Clustering // ACL. — 2007.

[119] *A. Clauset, M. E. J. Newman, C. Moore.* Finding community structure in very large networks // *Physical Review E.* — 2004. — Vol. 70. — P. 066111.