Московский физико-технический институт
Факультет общей и прикладной физики
Кафедра системной интеграции и менеджмента

Лабзин Григорий Александрович

Различные уровни семантики и задачи об автоматическом выделении семантики

Магистерская диссертация

Руководитель:

к.ф.н. Рыков В.В.

Рецензент:

к.ф.-м.н. Скорик В.А.

Долгопрудный
30 мая 2002 года

Аннотация

В этой работе рассматриваются разные уровни организации знания: от слов и предложений (семантика языка) до баз знаний (семантика знаний) и поставлена общая задача об автоматическом выделении семантики разного уровня. Проведен краткий обзор методов решения этой задачи, предложен информационно-геометрический подход к решению задачи о кластеризации. Здесь продолжены идеи, заложенные проектом Semantic Web:
Необходимо научить компьютер понимать человеческий язык, а так же разбираться в накопленном человечеством знании, чтобы помочь ему работать с возрастающими информационными потоками. Это значит, что необходимо научит компьютер семантике разных уровней и назначений. Эта задача решается с помощью языков разметки для "разных семантик" и инструментов для работы с знаниями и публикации новых знаний в сети Internet.

В данной работе уделено внимание вопросу о том, какие связи и какие структуры следует размечать при формализации знаний. Построение согласованной иерархии семантик является важнейшим этапом на пути формализации знаний. Автор указывает на перспективные направления и идеи связанные с созданием универсального языка для знаний, в частности Российский проект Knowledge Markup Language, разработки Semantic Web и W3.org.

Содержание

1 Введение

В данной работе под **семантикой** мы будем понимать отношение между языковыми выражениями (словами, сочетаниями, текстами), а также совокупность всех таких отношений с различными ролями и атрибутами.

Эти отношения состоят в том, что языковые выражения (слова, словосочетания, предложения, тексты) обозначают то, что есть в мире, — предметы, качества (или свойства), действия, способы совершения действий, последовательности действий, отношения, алгоритмы.

Термин «семантика» образован от греческого корня, связанного с идеей «обозначения» (ср. semantikos «обозначающий»). Именно такой смысл мы и будем вкладывать в корень слова семантика: обозначение свойств, ролей и связей. Причём, мы будем рассматривать семантику разных уровней: семантика слов, семантика предложений, семантика единиц знаний, семантика базы знаний.

Первые две “семантики” мы объединим в **семантику языка**, а две другие в **семантику знаний**. Под семантикой языка мы понимаем, в частности, связи между словами: “синонимичность”, “состоит из”, “является частным случаем”, и роли слов в предложении: “объект действующий”, “суть действия“, “описание действующего объекта”, “описание действия”, “объект действия”, а также роли предложений и связи между предложениями.

Семантика знаний — это связи между единицами знаний: “основывается на” (background links), “связано по смыслу” (related links), “является продолжением”, “предшествует” (sequence links), и роли единицы знаний: “аннотация”, “теорема”, “определение понятия”, “доказательство утверждения”, и т.д.

В действительности ролей и связей гораздо больше, они могут быть другие, в зависимости от уровня семантики и её назначения.

Компьютерная лингвистика — направление в прикладной лингвистике, ориентированное на использование компьютерных инструментов — программ, компьютерных технологий организации и обработки данных — для моделирования языка в тех или иных условиях, ситуациях, проблемных сферах и т.д.

Одно из направлений компьютерной лингвистики занимается созданием машинных языков, позволяющих выделять семантику различных уровней и назначений.

В данной работе проанализирована проблема формализации семантики различных уровней и задача об автоматическом выделении (или “маркапировании”) этой семантики. Рассмотрена иерархия языков и различные языковые метапереходы.

Markup — выделить, отметить, то есть чётко прописать роли и свойства объектов, а также роли и свойства связей.

Задача об автоматическом маркапировании семантики является, по сути, задачей о том, как научить компьютер естественному языку, то есть научить видеть семантику слов, предложений, текстов и единиц знаний. Эта задача непосредственно связана с вопросом о том, каким образом человек, зная слова и грамматические правила какого-либо **естественного языка**, оказывается способным передавать с их помощью разнообразную информацию и понимать, какую информацию о мире заключает в себе различные высказывания (текст, набор текстов). Можно сказать, что это задача о том, как человек выделяет смысл из текста, как определяет о чем данный текст и как классифицирует

тексты по тематикам. Этими вопросами занимается наука лингвистическая семантика.

Прежде, чем ставить задачу об автоматической разметке семантики, необходимо создать базовую схему языков, связанных и иерархически выстроенных. Этой первостепенной задаче посвящена большая часть работы — формализации семантики языка и семантике знания.

В первой части рассматривается семантика знания. Там также обозначены основные проблемы индустрии знаний и предложено несколько идей по преодолению этих проблем. Одно из решения — инструмент выделения семантики знаний язык знаний KML, который позволяет создавать базы знаний, которые понятны для компьютера (Computer Readable). Компьютер может эффективно помогать человеку при работе с такими базами знаний, поскольку он сам "видит" структуру знаний, связи между знаниями и роли различных элементов знаний.

Формализации естественного языка и созданию языков разметки, понятных компьютеру, посвящен также проект Semantic Web. Во второй части сделан небольшой его обзор, представлена одна из возможных "иерархий семантик" с некоторыми дополнениями и поправками, связанными с ориентацией на научные знания.

В третьей — поставлена задача об автоматическом выделении семантики, рассмотрены различные подходы, и приведен результат небольшого исследования, посвященного применению информационно-геометрических идей в задачах компьютерной лингвистики.

Часть I
Первый уровень. Семантика Знаний

Базовой идеей проекта Semantic Web является создания языков понимаемых как машинами, так и людьми. То есть не просто создавать большие архивы документов, а делать их структурированными и связанными друг с другом.

2 Проблемы современных Баз Знаний и их решения

Индустрия знаний "Путешествия вглубь науки", о которых писал Станислав Лем, сейчас стали неотъемлемой частью научной деятельности всякого ученого. Открытия в области естественных наук, техники, архивы результатов исследований предстают перед ним необъятным информационным потоком. Так получается, что современный ученый вынужден тратить большую часть своего времени на "путешествия вглубь науки", то есть на изучение и анализ чужих работ. Изобретение велосипедов — это ещё не самая большая беда. Изобрести велосипед заново иногда проще, чем найти описание его конструкции в архиве. Есть другая опасность, а именно то, что ученые, занимающиеся вроде бы одной

проблемой, вдруг обнаруживают, что они не понимают друг друга, потому что выросли в разных научных школах.

Первый шаг к решению проблемы очевиден — публикация знаний в Интернет, создание открытых Интернет конференций, открытых международных архивов статей и т.п. Бумажная индустрия знаний не справляется со своей задачей, а электронная индустрия знаний потенциально содержит больше возможностей для хранения знаний, обеспечения доступа к ним и поиска информации.

Процесс "электронизации знаний" идет и набирает ход. Научно-образовательная деятельность, так или иначе, стала занимать свою нишу в WWW. В качестве примера приведу три успешных сайта: xxx.lanl.gov (российское зеркало xxx.itep.ru), www.citeseer.com, и новая научная российская сеть www.nature.ru.

Но шаг этот не решил проблему, а скорее обнажил её, выявил её размеры. Ученый просто теряется, когда обнаруживает, что ежедневно выходит с десяток публикаций о фундаментальных работах в той узкой области науки, которой он занимаются. Создается впечатление, что в некоторых областях науки уже достигнута ситуация "мегабитовой бомбы" или "информационного барьера", о которой писал С. Лем. Наука не может перейти этот барьер, не может справиться с обрушивающейся на неё лавиной информации.

Конечно, решения есть. Одно из них предлагает российский проект KML (Knowledge Markup Language, http://kml.mipt.ru),

Наметился целый ряд тенденций в области публикации и получения знаний в глобальной сети Интернет. На данный момент существует несколько проектов (KA [21], SHOE [24], RDF [19]), имеющих сходные цели и использующие одну модель представления знаний — онтологии. Их разработка связана с получением знаний и причесыванием уже существующего содержания страниц, опубликованных в Интернет. Однако, эти системы не получили достаточного распространения, поскольку дополнительное атрибутирование документов (на чем, в принципе, все эти системы и основаны), которое не приводит к существенному росту информативности, порождает еще большую переинформированность.

В этой работе предлагается несколько базовых идей создания "правильных баз знаний". Частично эти идеи были заложены проектом KML (Knowledge Markup Language, http://kml.mipt.ru) [31], который занимается вопросами формализации научных знаний. А именно

1. Как правильно писать Знания?

2. Как правильно организовать Базу Знаний в Интернет?

KML — язык Знаний (Knowledge Markup Language). Проект KML взял на себя задачу разработку спецификации этого языка, а также различного инструментария для работы со знаниями. Инструментарий — это программы

1. для создания знания и его публикации в WWW — Knowledge Builder;

2. для навигации по Базе Знаний — Knowledge Viewer;

3. для запросов к Базе Знаний — Knowledge Query;

4. для перевода знаний из KML в другие форматы (HTML, TEX, rtf)

Основные идеи проекта по решению проблемы "мегабитовой бомбы" таковы:

2.1 Стандартизация Языка Знаний

Это означает создание базовой спецификации Универсального Языка Знаний, а скорее даже создание общей технологии формализации элементов знаний в различных областях науки, плюс технологии построения самой Базы Знаний.

В качестве основы была взята технология XML (www.xml.org, www.w3.org) создания языков, которая уже хорошо зарекомендовала себя. С её помощью созданы языки для математических и химических формул, логики высказываний, доказательств, языки обмена информацией и др.

KML тоже является XML языком, и, кроме того, он содержит в себе возможность включения объектов, написанных на других XML языках. Эта расширяемость KML позволяет ему эволюционировать, что является неотъемлемым свойством языка знаний, а стандартизация есть обязательное условие построения единой глобальной базы знаний.

2.2 Грануляризация знаний

То есть хранение знаний в отдельных, небольших, относительно самостоятельных единицах, структурированных и снабженных метаинформацией.

Для этого в KML существует специальный термин — UNIT. Что из себя должен представлять UNIT? Каковы основные свойства информационной единицы? Следует запомнить главное: отказаться от использования контекста и эллипсиса ради гибкости. Определение терминов, доказательство теорем, описание объектов и алгоритмов — все должно быть кратко и читаться в отрыве от общего контекста. Разрешается только использование ссылок на используемые объекты, термины, обязательные знания, прошлые версии и пр. Эти требования должны привести к отсутствию повторений и желания писать все заново. Оптимальный объем UNIT'а — абзац текста, который лексически связывает воедино используемые термины, объекты и мысль, выраженную в виде алгоритма или семантических связей. Любая книга (проект), публикуемый в KML — это набор UNIT'ов, связанных связями последовательности изложения и другими зависимостями. Причем изложение должно быть построено (литературно) таким образом, чтобы текст последовательности UNIT'ов можно было читать с любого места, с учетом того, что в дальнейшем, при составлении ответов на запросы пользователей, текст может быть разорван в местах разрыва UNIT'ов.

Конечно, любые изначальные рамки и рафинирование 'губят' мысль. Но, возможно, иначе нельзя. Публикация повторений вызовет путаницу при поиске информации.

2.3 Программируемость знаний

Речь идёт об идее возведения знаний в ранг программного продукта.

Публикация знаний в формате KML очень напоминает программирование: "программировать — значит понимать".

"Писать знания = программировать знания" является лишь частичной аналогией, потому что большая часть знаний программиста остается все-таки в его голове и не попадает в код программы, так как программа — это МАШИННЫЕ ЗНАНИЯ. Действительно, программист формализует знания не для остальных людей, а для машины и, следовательно, остальное приходится дописывать в

комментариях для того, чтобы код был понятен остальным. Возьмём теперь экспертные системы. В их создании участвуют три постоянных роли: специалист предметной области, инженер по знаниям и программист. Первый из них — это носитель знаний, ЧЕЛОВЕЧЕСКИХ ЗНАНИЙ. Второй осуществляет формализацию знаний для занесения их в программу, при этом процесс выглядит как "допрос" эксперта предметной области, "вытягивание" из него знаний. Третий же должен запрограммировать результат. Технология XML предполагает совмещение в лице "писателя знаний" сразу трёх ролей. Конечно, это тяжело, но здесь в помощь приходят разрабатываемые KML инструменты для Визуального Программирования Знаний.

Таким образом, ещё одна цель проекта KML — это создание простого и понятного языка формализации знаний, который бы позволил специалистам и экспертам в предметной области не прибегать к услугам инженера по знаниям и программиста, а самостоятельно "программировать" свои знания (для этого существует специальный термин — автоформализация).

Язык KML представляет собой лишь первый шаг в направлении Визуального Программирования Знаний. На его основе в настоящее время проектируются системы построения семантических сетей, связанных текстов и визуализации алгоритмов.

Следующая важная параллель программирования для машин и формализации знаний для людей — это повторное использование. При разработках программных систем особое внимание уделяется возможности повторного использования и доработки. Повторное использование может относится практически к любым аспектам программирования (структуры данных, модели поведения, протоколы и т.п.) и заметно удешевляет и ускоряет процесс разработки. Почему бы не использовать эту технологию создания программ для построения структуры человеческих знаний? Конечно, чтобы знания, как программный компонент, можно было использовать повторно, их форма должна удовлетворять определенным условиям "стыковки". Эти условия и диктует KML, позволяя тем самым создавать хорошо оформленные строительные блоки знаний (UNIT'ы)

Итак, 1) KML претендует на решение проблемы чистоты содержания, то есть отсутствие "воды" в научном информационном потоке, и 2) KML предоставляет стандарт для объединения накопленной человечеством информации в единой распределённой Базе Знаний.

Каждый человек, публикующий документы в формате KML, должен всегда помнить три повелительных глагола:

1. Избегать повторения.
2. Использовать имеющееся.
3. Улучшать существующее.

А KML, в свою очередь, будет заботится о следующем:

1. Выразительные средства формализации всех типов знаний.
2. Поддержка версионности.
3. Единая среда разработки и хранения знаний.
4. Удобство навигации.

3 Технологии знаний и программные технологии

Чистота содержания, отсутствие “воды” в научном информационном потоке, не может быть достигнута просто переходом на новый язык представления знаний в компьютере. Необходим новый стиль работы с знаниями. KML накладывает ряд серьёзных требований к публикуемым знаниям. Указанные три базовые идеи (стандартизация, грануляризация и программируемость) усложняют процесс написания знания. “Писатели знания” должны придерживаться определенной формы, и много потрудиться прежде, чем опубликовать свои научные разработки. В связи с этим нужны удобные технологии публикации знаний.

3.1 Требования к знаниям и технологии публикации

Писатели знаний должны избегать повторения, использовать имеющееся, улучшать существующее знания. А это значит разобраться в том, что уже разработано, написано и изучено, дать на эти работы ссылки. Ещё нужно позаботиться о недостаточно подкованном читателе, дав ему ссылки на базовый материал (background), а также о заинтересованном читателе, который хочет знать больше, и предоставить ему ссылки на работы для дальнейшего чтения. Кроме того, тематика работы, используемые термины, обозначения, ссылки на другие работы должны быть общепринятыми и стандартизованными, чтобы база знаний адекватно воспринимала эти единицы знаний, правильно индексировала и помещала в “правильное место” в общей структуре знания.

Заботу о стандартах и правильности формы единицы знаний, а также задачу о помещении её в правильный контекст может взять на себя компьютер, но конечно, лишь частично.

Инструментарии включает в себя авторубрикаторы, автореферaторы (программы выделяющие ключевые слова и структуру), и программы автоматически выявляющие базу (background). Все они так или иначе связаны с обучением компьютера естественному языку. Об этом пойдет речь в следующей части “Семантика Языка”.

3.2 Повторное использование

Рассмотрим подробнее идею повторного использования на примере программных продуктов [4]. Она заключается в том, чтобы отказаться от написания алгоритмов или деклараций, если они уже существуют и максимально использовать готовый код, что, в принципе объясняется природной ленью программистов. Действительно, зачем каждый раз писать свою операционную систему для решения конкретной задачи, когда уже есть несколько готовых, отлаженных и достаточно универсальных платформ. Зачем изобретать свою систему управления базами данных, если есть множество вполне подходящих и систем. Так и происходит. Существует огромное количество компонентов, которые, соединяя воедино, можно получить прекрасный “фундамент” для решения своих задач. Все компоненты, пригодные для повторного использования должным образом документируются и реализовываются в виде библиотек или приложений с API (Application Programming Interface). Однако, не все так радужно. То, что мы описали, — это *повторное использование в большом*. Попробуем спуститься на

Рис. 1: Древовидный процесс разработки. Существует основная ветвь, или 'ствол', который нумеруется парой чисел, и боковые ветви, для нумерации версий которых добавляется еще пара чисел.

нижний уровень и посмотреть, что происходит с *повторным использованием в малом*.

Оказывается, повторного использования в малом, на уровне отдельных алгоритмов и структур, практически нет [13]. Каждый новый проект стартует с написания своих библиотек работы со строками, списками, деревьями и т.д. [17]. Отчего это происходит? В первую очередь от того, что эти компоненты довольно легко реализуемы. То есть относительная цена или трудоемкость их написания довольно мала. Во-вторых, сейчас, в основном, программистами используется порядка десятка языков программирования и "крупным" проектам легче сделать привязки (интерфейс) к каждому языку, чем для каждого языка реализовывать свою 'базовую' библиотеку.

Более того, использование готовых малых компонентов в любом случае требует изучения дополнительной спецификации, что в большинстве случаев не оправдано.

Но, не смотря на это, попытки стандартизировать различные библиотеки, хотя бы для одного языка (и не только [18]) существуют и приносят свои плоды (CPAN, CTAN). Во всяком случае, выигрыш в повторном использовании появляется только в случае достаточно высокого уровня языка (когда все 'мелкие' конструкции уже реализованы), либо в полном абстрагировании от исходного кода [18].

3.3 Системы контроля версий совместной разработки

Особой популярностью среди разработчиков программного обеспечения пользуются так называемые *системы контроля версий* [14, 15]. Они позволяют вести совместный процесс разработки, при этом позволяют хранить всю историю разработки (все состояния проекта). Также, немаловажной особенностью этих систем является ветвление процесса разработки. Рассмотрим все более детально. В каждом процессе существует несколько коммитеров (от слова commit), которые имеют право накладывать сделанные изменения в проекте. Для этого сначала коммитер 'берет' опорную версию проекта, делает в ней изменения и накладывает их. Обычно, если число коммитеров больше одного, то они работают по очереди, чтобы в случае одновременного изменения одной и той же версии проекта, их изменения не конфликтовали друг с другом.

Существует другой способ избежать этого. Коммитеры ответвляют новую ветвь от той, в которой находилась опорная версия и сохраняют свои изменения в новой ветви, чтобы потом, влить их в общий поток. Таким образом, процесс

Рис. 2: Наложение Автором нового Проекта (версии) на существующую структуру в общей Базе.

развития проекта представляет собой дерево версий.

Итак, стандартизация, создание расширяемых продуктов (продуктов приспособленных для доработки), системы контроля версий, технология “апгрейдов” широко и с пользой используется при создании программного обеспечения и имеет смысл перенести эти технологии на базы знаний.

Часть II
Второй уровень. Семантика Языка

Прежде, чем приступать к задаче обучения компьютера человеческому языку, следует создать формализованную систему языков для различных нужд (задач). То есть дополнительную разметку естественного языка, которую понимает компьютер (Machine-Understandable information), и которая фиксирует роль и атрибуты каждого слова в предложениях, роли и атрибуты отдельных предложений и кусков текста, а также фиксирует структуру и отношения между этими элементами.

Такая система языков разрабатывается в частности проектом Semantic Web. Она ориентирована на различные Web приложения, протоколы обмена и контроля знаний в Internet. Мы приведем здесь краткую, классификацию уровней языков Semantic Web несколько адаптированную для Научных Знаний и дополненную.

4 Иерархия языков Semantic Web

4.1 Язык утверждений

Первое, чему следует научить компьютер — это языку высказываний. С помощью этого языка можно вводить объекты, указывать их свойства, вводить взаимоотношения между объектами. Например,

<объект> имеет <свойство>

<объект> <глагол действия> <объект действия>

<объект> <глагол действия> <объект действия> <место действия>

Предложение состоит из связанных понятий различного типа. Кроме того этот уровень позволяет проводить “суждения о суждениях”, то есть снабжать суждения метаинформацией и связывать их отношениями.

Язык предложений позволяет нам описывать статическую семантическую сеть, и многие базы знаний, по сути дела, ограничиваются этим первым уровнем.

Язык, позволяющий вводить объекты, свойства, взаимоотношения между объектами, снабжать их метаинформацией, принято называть **онтологией**.

4.2 Язык схем. Проверка корректности объектов

На языке схем можно описывать грамматики языков. Можно также сказать что язык схем это язык для создания других языков. В частности существуют DTD схемы и SGML схемы различных языков (форматов), например HTML, которые позволяют осуществлять проверку, действительно ли данный документ является корректным документом предполагаемого формата.

“Язык Языков” — первый метапереход.

4.3 Язык перевода

Следующее, чему мы хотим научить компьютер — это перевод текстов с одного языка (онтологии) на другой. Различных онтологий огромное количество и невозможно создать единый алгоритм перевода. Но иногда, между категориями двух онтологий можно провести соответствие. Язык этого уровня должен предоставить средства для описания этих соответствий и алгоритмов конвертации.

“Язык Описания Алгоритмов Перевода” или “Язык межъязыковых соответствий” — это второй метапереход.

4.4 Язык логики

Эта естественная надстройка над языком предложений, позволяющая составлять составные логические выражения из имеющихся (предикаты И, ИЛИ, НЕ, ... и кванторы ДЛЯ ЛЮБОГО, СУЩЕСТВУЕТ ..) Во-первых, алгебра высказываний логики черезвычайно важная составляющая, когда речь идет о формализации естественно научных знаний, математических теорий, но не только. Этот уровень позволяет создавать составные свойства, составные соотношения, и кроме того, является неотъемлимой частью языка алгоритмов.

Появляются категории “причина”, “следствие”, множества, удовлетворяющие набору абстрактных свойств. Этот уровень **позволяет развивать язык**, определяя новые понятия с помощью логических выражений. И наконец, мы можем **записывать доказательства, объяснения и алгоритмы**. Доказательство, написанное на машинном языке, можно абсолютно надежно проверить на его надежность, если создать соответствующую семантическую сеть аксиоматических основ теории и научить компьютер логике.

4.5 Проверка доказательств

Проверка верности доказательств или корректности утверждений это задача состоящая из двух этапов: 1) проверка грамматики 2) проверка логики.

Первый этап осуществляется с помощью схемы (заданной на языке схем) описывающей грамматику доказательств. Соответственно, мы могли бы ввести язык логик, и проверить, удовлетворяет ли данное доказательство определённой логике.

Конечно, в математике принята одна единственная правильная логика, а другие логики рассматриваются лишь как специфические отклонения.

Но в принципе, язык логик можно воспринимать как дополнительное средство описания корректности (правильности) объекта. Очень часто бывает ситуации когда, важно чтобы объект не только был правильно записан с точки зрения грамматики, но и удовлетворял некоторой внутренней сложное логике формата, которую уже нельзя или неестественно описывать как набор грамматических правил.

Языки логик нашёл бы применение в юриспруденции, потому что законы — это ни что иное, как определения и алгоритмы, а любой судебный процесс состоит в выявлении логичности доводов адвоката и прокурора.

4.6 Язык описания эволюции объектов

Этот язык является надстройкой над языком предложений и предоставляет возможность описывать изменения. То есть описывать алгоритм для динамики семантической сети.

Не смотря на то, что логический уровень уже позволяет нам писать любые алгоритмы, алгоритмы эволюции языка – отдельный разговор.

"Язык Эволюции Языка" — подробнее об этом метапереходе можно прочитать в [23, Refal, метакомпиляция].

5 Поисковые системы будущего

Поисковые системы Internet находят множество ответов на получаемые запросы и "покрывают" огромную часть Web, однако при этом они возвращают много несоответствующих ответов. В большинстве поисковых систем практически не существует понятия "соответствия документа запросу". Их язык запросов слишком элементарен, чтобы говорить о соответствии. Они способны ограничивать множество полученных результатов и выдавать наиболее вероятные правильные ответы. Такие системы не могут осуществлять точный поиск в массивах связанных данных и конструировать при этом верные ответы.

"Комбинаторный взрыв" числа вариантов при обработке сложных запросов и их обработка представляет собой весьма трудную задачу. Но, возможно, в будущем, когда хранимая в Internet информация будет надлежащим образом размечена, появится возможность создавать "мыслящие" Поисковые Машины (ПМ), которые смогут "понимать содержание" просматриваемых документов и смогут адекватно вычислять меру соответствия документа данному запросу.

Такие ПМ будут способны доходить до каталогов, содержащих наиболее полный перечень мест употребления того или иного термина, а затем, применяя логические методы, удалять из этого списка все нерелевантные ссылки, оставляя лишь те, которые соответствуют запросу и которые можно использовать для решения поставленной задачи. Таким образом, даже при сохранении проблемы "комбинаторного взрыва", использование простого метода пошаговых заключений/выводов (например, двухшаговых) позволит решать задачи ПМ на несколько другом качественом уровне. При этом задачи построения умозаключений, логических связок и их обоснования по-прежнему представляют собой отдельную проблему и направление исследований. Есть определенный коммерческий интерес к развитию интеллектуальных ПМ и алгоритмов, предназначенных для эффективного решения отдельных задач Баз Знаний.

Этот процесс может включать в себя создание "хранилищ" промежуточных результатов, во многом аналогичных каталогам сегодняшних поисковых машин, а также выделение скрытых кластеров, или параметров порядка Знаний, автоматическая индексация единиц знаниям по этим параметрам порядка и другие задачи.

Часть III
Задачи об автоматическом выделении семантики

6 Задачи компьютерной лингвистики

Человеческий язык сам по себе уже содержит всю необходимую информацию о связях и структуре. Другое дело, что трудно научить компьютеры понимать этот язык, выделять **смысл** документа.

Есть целый ряд важных задач компьютерной лингвистки:

- **Резюмирование документа** или (автореферат).
- **Рубрикация**, то есть отнесение данного документа к определенной рубрике
- **Кластеризация**, то есть разделение множества документов на на рубрики, — классы близких друг другу по смыслу документов
- **Перевод документа** с одного языка на другой
- **Ответ на запрос**, то есть проверка документа, написанного на естественном языке, на удовлетворение сложному запросу. Например “Верно ли что автор документа женщина?”

Есть много других. Все они довольно сложные и сильно взаимосвязаны.

В большинстве случаев компьютер неспособен решить их надлежащим способом, так хорошо, как это сделал бы человек. Это связано с черезвычайно сложным устройством человеческой памяти и функционированием мозга. Ни то, ни другое пока не нашло эффективной формализации.

Именно поэтому компьютерным системам нужна помощь — нужно разрабатывать универсальные языки (machine and human understandable), иерархию которых мы описали во второй части. Знания нужно помещать в компьютер уже с выделенной семантикой, “замаркапированной” с помощью этих языков.

Создание таких языков, конечно, не убирает самой проблемы формализации человеческого языка и мышления. Это чрезвычайно интересная область компьютерной лингвистики, связанная с созданием искусственного интеллекта и эвристических лингвистических алгоритмов.

6.1 Мера семантической близости

Многие существующие алгоритмы решающие задачи компьютерной лингвистики так или иначе опираются на **меру семантической близости**.

Существует два различных метода вычисления семантической близости документов (единиц знаний), первый основан на “внутренней семантике”, например на семантике естественного языка, второй — на “внешней семантике”, то есть связях между единицами знаний в глобальной базе знаний (семантика Базы Знаний).

Поясним это подробнее. Имея две лингвистические единицы, мы можем разбить их на более мелкие и вычислять их семантическую близость на основе сходности их состава, то есть семантической близости их отдельных составляющих. То есть близость двух текстов определяется сходностью слов (частот слов), которые в них встречаются.

Можно пойти другим путём. Найти положение данных двух лингвистических единиц в общей семантической сети соответствующего уровня. И рассмотреть насколько сильно они разнесены в этой сети. То есть близость текстов определять по их внешним связям, сходностью рубрик и тем. "Насколько долго нужно двигаться по ссылкам в Интернете, чтобы добраться от одного документа до другого".

При вычислении семантической близости можно использовать явные правила, а так же накопить различную статистику. Например, статистику об частотах слов (сочетаний, языковых конструкций) в различных рубриках, накапливать и анализировать статистику запросов и блужданий по сети документов (единиц знаний) пользователей. В последнем случае близость будет имеет вид "пользователи, которые наблюдали данную единицу интересовались также ... ".

Информационно-геометрический подход позволяет вычислять семантическую близость на основе набора частотных характеристик. В следующей части мы подробно рассмотрим этот подход а также основанный на нём алгоритм решения задачи о кластеризации.

Следующая, очень плодотворна идея заключена в комбинировании семантики разных уровней. Семантическую близость между двумя единицами знаний можно вычислять, анализируя тексты с использованием семантической базы языка и рассматривая их общее положение в базе знаний, и из анализа семантических связей на разных уровнях находить меру семантической близости.

6.2 Информационно-геометрические идеи

Информационно-геометрические идеи в семантических задачах — это, грубо говоря, построение метрики на объектах семантической сети на основе статистических данных.

Роль объектов, как мы уже писали, могут играть различные лингвистические единицы: слова, предложения, тексты, рубрики, а статистические данные — это частоты их появления и корреляции между ними.

Пусть мы имеем n вероятностных характеристик лингвистической единицы — n чисел от 0 до 1, сумма которых равна 1. Это могут быть вероятности (частоты появления) лингвистических единиц следующего уровня (более мелких), из которых составлены лингвистические единицы рассматриваемого уровня.

Множество вероятностных распределений $\{p_1, p_2, \ldots, p_n\}$ на n точках образуют многообразие размерности $n-1$, поскольку $p_i \geqslant 0$ и $p_1 + p_2 + \ldots + p_n = 1$. Это симплекс S_n.

Введём величины $z_i = 2\sqrt{p_i}$, тогда $\sum z_i^2 = 4$ и таким образом $(n-1)$-мерный симплекс запараметризован частью n-сферы. Пусть $z(t)$ гладкая кривая. Тогда квадрат длины касательного вектора есть

$$\langle \partial_t z, \partial_t z \rangle = \sum_i (\partial_t z_i)^2 = \sum_i p_i(t)(\partial_t \ln p_i(t))^2,$$

Рис. 3: Геодезические на симплексе S_3

что совпадает с метрикой Фишера [25, 26, 27].

Этот результат можно получить непосредственно, решив уравнение геодезической.

Для $n = 2$ имеем

$$\langle \partial_t p, \partial_t p \rangle_{fisher} = \frac{(\partial_t p_1(t))^2}{p_1} + \frac{(\partial_t p_2(t))^2}{p_2} = (\partial_t p_1(t))^2 \left(\frac{1}{p_1} + \frac{1}{1-p_1} \right) = \frac{(\partial_t p_1(t))^2}{p_1(1-p_1)}.$$

То есть расстояние между распределениями $(p, 1-p)$ и $(q, 1-q)$ равно

$$\rho(p,q) = \int_p^q \frac{dx}{\sqrt{x(1-x)}},$$

откуда $\rho(p,q) = |2\arccos(\sqrt{p}) - 2\arccos(\sqrt{q})|$.

В общем случае

$$\rho(p,q) = 2\arccos \sum_{i=1}^{n} \sqrt{p_i q_i}.$$

На рисунке 3 показаны геодезические на треугольнике S_3.

Определенная таким образом метрика на многообразии распределении обладает рядом важных свойств, связанных с задачей оценки параметров по наблюдению некоторой случайной величины.

А именно, она адекватно отражает меру близости на распределениях, так как является монотонной относительно стохастических отображений [29, Ченцов]. Подробнее об этом можно прочитать в трудах Дениса Петца (Petz) и Амари (Amari S.-I.), где описано почему эта метрика является привилегированной в задачах об оценках параметров случайной величины.

В задаче о рубрикации, случайной величиной является множество характеристик внутренней семантики элемента, а параметры которые нам нужно определить — меры соответствия данного элемента различным рубрикам.

А задачу о кластеризации можно поставить, как выделение рубрик, то есть разделении набора элементов на группы близких в смысле информационной метрики групп, и вычислении статистических характеристик этих групп (среднего значения, дисперсии).

7 Задача о кластеризации

Задача о кластеризации — это задача о разделении множества документов на рубрики, то есть о разбиении на группы близких **по смыслу** документов.

Любопытно, что можно написать программу, решающую эту задачу не закладывая в компьютер семантическую базу языка, то есть научить замечать близость по смыслу, не зная языка. Не более, чем статистический анализ позволяет ввести понятия **синонимы** и **омонемы**.

Статистических подход к задаче о кластеризации позволяет переформулировать её на языке задачи о разделении смесей.

7.1 Задача о разделении смесей

1. Постановка задачи.

Если плотность вероятности некоторой случайной величины представима в виде линейной комбинации плотностей вероятности, то в этом случае говорят, что функция плотности описывается конечной смесью распределений. Плотность распределения вероятностей смеси есть $F(x) = \sum_{j=1}^{k} p_j f_j(x, \theta_j)$, где θ_i — набор параметров, p_i — доля компонента в смеси, а f_i — компоненты смеси, которые в свою очередь представляют собой функции плотности распределения, зависящие от вектора параметров θ_i . Далее мы везде будем полагать $f_i \equiv f$, хотя исследуемые ниже алгоритмы могут быть использованы и в случае разнотипных компонентов в смеси.

Задача разделения смеси распределения состоит в нахождении по заданной выборке $X = (x_1, ..., x_n)$ наблюдений (случайных величин, функция распределения которых является смесью) оценок параметров p и θ . При этом вид функции предполагается известным.

Решение этой задачи не всегда бывает однозначным. В таком случае говорят, что смесь распределений неразличима. В качестве примера неразличимых смесей можно указать произвольное разбиение точек, равномерно распределенных на отрезке прямой, на два класса. В работах [320, 321, 327] сформулированы необходимые и достаточные условия различимости для смесей распределений. Из них, в частности, следует, что различимыми являются конечные смеси нормальных (в том числе и многомерных), экспоненциальных, пуассоновских распределений.

2. Метод максимального правдоподобия.

Пусть $F_\theta(x)$ — некоторое параметрическое семейство функций распределения, $X = (x_1, ..., x_n)$ — выборка наблюдений при некотором фиксированном значении параметров θ.

Введем функцию правдоподобия $L = \prod_{i=1}^{n} F_\theta(x_i)$. Метод максимального правдоподобия заключается в максимизации функции правдоподобия по параметрам распределения θ.

Оценкой максимального правдоподобия θ^* называется значение параметров θ, при котором достигается максимум функции правдоподобия. Получаемые таким образом оценки обладают двумя важными асимптотическими свойствами — несмещенностью (т.е. математическое ожидание оценки θ^* совпадает с истинным значением параметров θ) и состоятельностью (последовательность θ_n^* сходится по вероятности к истинным значениям параметров при $n \to \infty$).

Однако метод максимального правдоподобия напрямую не применим к задаче разделения смесей, поскольку получающиеся уравнения для нахождения θ^* решить на практике не удается. Поэтому был предложен алгоритм EM, который представляет собой итерационную процедуру нахождения оценок метода максимального правдоподобия.

3. Описание некоторых алгоритмов.

Рассмотрим сначала алгоритм EM (Estimation Maximization). Применив метод максимального правдоподобия к задаче разделения смесей, мы сведем ее к оптимизационной задачи вида

$$ln(L) = \sum_{i=1}^{n} ln \sum_{j=1}^{k} p_j f(x_i, \theta_j) \to \max_{F,\theta}$$

(1)

Если обозначить через g_{ij} так называемые апостериорные вероятности, то логарифмическую функцию правдоподобия можно переписать в виде

$$ln(L) = \sum_{i=1}^{n} \sum_{j=1}^{k} g_{ij} ln(p_j) + \sum_{i=1}^{n} \sum_{j=1}^{k} g_{ij} ln(f(x_i, \theta_j)) - \sum_{i=1}^{n} \sum_{j=1}^{k} g_{ij} ln(g_{ij})$$

, (2)

где

$$g_{ij} = \frac{p_j f(x_i, \theta_j)}{\sum_{j=1}^{k} p_j f(x_i, \theta_j)}$$

(3) Идея построения итерационного алгоритма вычисления оценок для параметров состоит в том, что, отправляясь от некоторого начального приближения для и , вычисляют по формуле (3) начальные приближения для апостериорных вероятностей (этап оценивания). Затем, возвращаясь к (2), при вычисленных значениях g_{ij} определяют значения p_j и θ_j из условия максимизации отдельно каждого из первых двух слагаемых в формуле (2), поскольку первое слагаемое зависит только от параметров p, а второе — только от параметров θ (этап максимизации). Максимизация первого слагаемого дает выражение для (здесь t — номер итерации):

$$p_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} g_{ij}^{(t)}$$

(4) Решение оптимизационной задачи (максимизация второго слагаемого) можно получить, зная конкретный вид функции.

Таким образом, мы можем вычислить параметры компонент смеси. Подставив их в (3), мы переходим к новому шагу итерации. Таким образом, используя алгоритм EM, мы получаем последовательность . (здесь условия, при которых эта последовательность сходится к оценкам максимального правдоподобия)

Алгоритм адаптивного вероятностного обучения (алгоритм SEM — Stochastique - Estimation - Maximization) использует схему EM алгоритма, дополняя ее этапом статистического моделирования. При этом SEM позволяет в рамках основной процедуры решить задачу об оценке числа компонент (классов) в смеси, чего не позволяет сделать EM.

7.2 Автоматическая кластеризация

Предположим, у нас есть множество документов $D = \{d_1 ... d_N\}$ на базе фиксированного словаря терминов(слов) $W\{w_1 ... w_M\}$. каждый документ можно рассматривать как распределение вероятностей на множестве последовательностей терминов (слов). Мы сфокусируем своё внимание на более частном случае, когда текстовый документ рассматривается как набор не связанных друг с другом слов, то есть информационный поток без "памяти".

Это предположение означает, что каждый документ может быть представлен как распределение вероятностей $P(w_j|d_i)$, (которые равны вероятностям того, что определённое слово j встретится в документе i), и, как следствие, данные могут быть представлены в виде матрицы $n(d_i, w_j)$, где n_{ij} показывает частоту повторения слова j в документе i.

Итак, документу d_i можно поставить в соответствие вектор $\mathbf{n^i} = \{n^{i1}, ..., n^{iM}\}$. Первое упрощение, которое мы сделаем, заключается в том, что мы будем характеризовать документ не вектором повторения слов $\mathbf{n^i}$, а вектором "направления" $\mathbf{x^i} = \frac{\mathbf{n^i}}{\sum_j \mathbf{n_j^i}}$. У нас есть множество документов, которые мы хотим разбить на K рубрик без участия человека, то есть найти "наилучшие" вектора $\mathbf{w^k}$ для заданного множества документов $\{\mathbf{d^n}\}$. Здесь мы вынуждены использовать второе предположение — о том, что рубрика характеризуется только частотами повторяемости слов.

Так вот, рассматривая каждый документ как случайную величину, мы можем свести нашу задачу к задаче о разделении смесей, причем $\{\theta_j\} \equiv \mathbf{w^j}$, а p_j число документов, принадлежащих рубрике j, отнесеённое к общему числу документов N.

8 Заключение

Итак, есть семантика разных уровней и необходимо создавать языки формализующие эту семантику. Одна из перспективных технологий создания таких языков — XML схема, а также специализированный расширяемый язык знаний KML.

Мы перечислили ряд идей, выдвинутых проектами KML и Semantic Web, которые позволяют добиться качественности знаний а также полноты и правильности семантики на разных уровнях. Это идеи программируемости знаний, вторичного использования знаний, стандартизации языка знаний.

Задачу о выделении семантики можно сформулировать как задачу о авторазметке ("автомаркопировании") текста и об автоматическом определении метаинформации. При решении этой задачи полезно использовать информацию о семантике верхнего уровня и семантике составных частей. Вполне адекватным оказывается статистический подход, использующий информационную метрику для вычисления меры семантической близости.

9 Благодарности

Прежде всего хочу поблагодарить своего научного руководителя Владимира Васильевича Рыкова, Станислава Владимировича Клименко, заведующего ка-

федрой системной интеграции и менеджмента, и декана факультета общей и прикладной физики Федора Федоровича Каменца, без которых написание и защита диплома были бы невозможны, а также Ворожцова Артема за полезные обсуждения чернового варианта работы. Особую благодарность хочу выразить своей Алёне, ради которой я всё это делаю.

Список литературы

[1] Скрэгг Г., 1983. *Семантические сети как модели памяти. Новое в зарубежной лингвистике.* Вып. **12**, М.: Радуга. С. 163-190

[2] Dorffner G., 1992. Taxonomies and Part-Whole Hierarchies in the Acquistion of Word Meaning – Connectionist Model. Austrian Research Institute of Artifical Intelligence. Submited for Publication.

[3] Kruiijff Geert-Jan M. June 1998. Basic Dependency-Based Logical Grammar. Technical Report (UFAL-TR-5).

[4] Брукс Ф.П., 1979. Как проектируются и создаются программные комплексы. Мифический человеко-месяц: очерки по системному программированию. М.: Наука.

[5] Буч Г., 1992. Объектно-ориентированное проектирование с примерами применения. М.: Конкорд.

[6] Мельчук И.А., 1974. Опыт теории лингвистических моделей "Смысл-Текст". Семантика, синтаксис. М.: Наука.

[7] Wilks Y., 1976. Parsing English II. Computational Semantics. Eds. Y. Wilks, E. Sharniak. N.Y.: North-Holland. P. 155-184.

[8] Resnik Ph., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Porceedings of IJCAI-95.

[9] Mladenic D., Grobelnik M., 1999. Predicting Content from Hyperlinks. Submitted for publication in IJS.

[10] Briggite Endres-Niggemeyer. Human-Style WWW Summarization.

[11] Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. Институт Проблем Управления РАН. http://www.ipu.ru/publ/epstn.htm.

[12] Аверкин А.Н., Батыршин И.З., Блишун А.Ф., Силов В.Б., Тарасов В.Б., 1986. Нечеткие множества в моделях управления и искосственного интеллекта. Под ред. Д.А. Поспелова. М.: Наука.

[13] Проблема повторного использования (IBM).

[14] Cederqvist P. Version Management with CVS. http://www.cvshome.org/docs/manual/cvs.html

[15] Tichy Walter F. 1993. RCS-A System for Version Control. http://www.cs.purdue.edu/homes/trinkle/RCS/rcs.ps

[16] Паронджанов В.Д. 1998. Как улучшить работу ума. М.: Радио и Связь.

[17] Горбунов-Посадов М.М., 2000. Расширяемые программы. М.: Полиптих.

[18] Роджерсон Д., 2000. Основы COM (2-е издание). Русская Редакция. Microsoft Press.

[19] Resource Definition Format. http://www.w3.org/rdf

[20] Chr. Koch, P. Petta et. al., 2000. On Information Integration In Large Sciencific Collaborations.

[21] Benjamins V.R., Fensel D., et. al., 1998. Community is Knowledge!. In KA^2, In: Proc. KAW'98, Banff, Canada.

[22] Турчин В., О КИБЕРНЕТИЧЕСКОЙ ЭПИСТЕМОЛОГИИ: Turchin V. On Cybernetic Epistemology. Systems Research, Vol.10, No.1, 1993, p. 3-28. http://www.dgap.mipt.ru/ artema/ai/turchin.pdf.

[23] Турчин В., REFAL, Идея метакомпиляции.

[24] Heflin J., Hendler J., Luke S., 1998. Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. In AAAI-98 Workshop on AI and Information Integration.

[25] Amari S.-I., **Differential-Geometrical Method in Statistics**, *Lecture Notes in Statistics* **28** (1985), Springer-Verlag, New York.

[26] Amari S.-I., **Information Geometry**, *Contemp. Math.*, **203** (1985).

[27] Streater R. F., Classical and Quantum Probability, math-ph/0002049, (1999).

[28] Petz D., Monotone metrics on matrices spaces, *Lin. Alg. Appl.*, **244** (1996), pp. 81-96 .

[29] Ченцов Н.Н. **Статистические решающие правила и оптимальные выводы**, М., "Наука", (1972), 520 с. и **Statistical Decision Rules and Optimal Inferences** *Translations of Mathematical Monographs*, American Mathematical Society, (1982).

[30] Винокуров Н.А., Ворожцов А.В., 2000. Формат Хранения Знаний KML. Спецификация и Использование. http://kml.mipt.ru

[31] Винокуров Н.А., 2002. Информативность структур данных, http://kml.mipt.ru/ndiplom.zip