Знания-Онтологии-Теории (ЗОНТ-11)

Об одном методе автоматического построения гипернимов с помощью внешней поисковой системы

Афонин С.А.[1], Бахтин А.В.[1]

[1] *НИИ механики МГУ, Мичуринский пр., д. 1, г. Москва, 119192, Россия.*

serg@msu.ru, amdxw51w@gmail.com

Аннотация. *Поисковая система позволяет получить множество сниппетов (коротких фрагментов документов), содержащих заданные ключевые слова. В данной статье предложен метод автоматического построения гипонимических пар (общее-частное) с помощью сниппетов, используя несколько известных приёмов. Используются лексические шаблоны, чтобы построить гиперонимы-кандидаты для заданного термина и фильтрация, основанная как на лексических шаблонах, так и на частотном анализе. Приведены первые экспериментальные данные, свидетельствующие о возможности реализации метода на практике.*

Ключевые слова: выделение гиперонимов, автоматическое построение онтологий, лексические шаблоны

1 Введение

Онтология — многозначный термин. Понятие онтологии в философиии восходит ко временам Древней Греции, и обозначает систему построения или способ описания мира [16]. Одни и те же сущности и ситуации можно описать как в терминах вещей, так и в терминах фактов (императивно формирующих мир) или идей, которым причастны объекты бытия. В современной лингвистике и информатике онтология представляет собой формализованное знание о некоторой предметной области, выраженное в виде схемы (объекты и связи между ними). Можно выделить несколько видов онтологий [20], однако в данной статье будет рассматриваться только терминологические онтологии — базы информации о *термах* естественного языка и связи между ними (гипонимия, мерономия, синонимия и другие).

Терминологические онтологии, такие как WordNet, содержат информацию о гипонимах-гиперонимах (общее-частное), синонимах и других семантических отношениях между термами. Такая информация может быть использована во многих областях, связанных с обработкой естественных языков, таких как уточнение запроса [9, 3], резюмирование текста [4], категоризация [11] , ответы на вопросы [12] и многие другие. К сожалению, высококачественные онтологии в большинстве созданы вручную и охватывают ограниченный набор тематических областей. Более того, сложно поддерживать онтологию актуальной в быстро развивающихся областях, когда часто появляются новые объекты и связи. Таким образом, построение онтологий становится «узким местом» для практической реализации многих проектов на этом направлении. Один из подходов к его устранению - автоматическое или полуавтоматическое построение онтологий.

В последние двадцать лет проблеме построения онтологий было посвящено большое число работ. Хотя информация о способах описания отношений между термами присутствует во многих источниках (базы данных, тексты с разметкой, словари), наиболее перспективным представляется подход, основанный на извлечении иерархии терминов из неструктурированных текстов. Хорошие перспективы данной задачи обусловлены тем,

что иерархия терминов лежит в основе каждой онтологии, а неструктурированные тексты - наиболее популярный формат текстовых данных.

Автоматическое построение онтологии - непростая задача. Хорошо известно, что набор понятий и терминов зависит от языка и культуры. Например, во многих северных языках существует много термов для обозначения различного состояния снега, аналогов которых в других языках нет. Даже в одном языке некоторые термы имеют различные значения в зависимости от местности. Так, в Великобритании слова *city* означает важный населенный пункт, отличающейся от *town* размером, плотностью населения или статусом. В тоже время в американском английском практически все поселения могут быть обозначены термом *city*, и существуют сотни страниц с текстом: «Canyon city is a small village».

Большинство работ, направленных на построение иерархии терминов, можно разбить на два класса. Первый состоит из работ, основанных на предложенной Harris *распределительной гипотезе* [7], которая утверждает, что семантически близкие термы встречаются в похожих контекстах. Например, можно предполагать, что контексты слов *car* и *automobile* содержат много общих слов. Хотя семантическая близость может быть измерена посредством статистики, зачастую сложно объяснить причину близости, отношения связывающие слова. Примером статистического подхода к построению онтологий может служить работа [14], использующая следующую гипотезу: множество документов, в которых два терма встречаются одновременно, должно быть подмножеством множества документов, в котором встречается их гипероним.

Второй класс работ, к которому относится и данная работа, использует *лексические шаблоны* [8], которые сигнализируют о возможных семантических отношениях между словами. Рассмотрим пример гипонимии. Терм A является гипонимом терма B (терм B является гиперонимом терма A), если A - подтип или экземпляр B. Таким образом, терм *Шекспир* - это гипоним для терма *автор* (а слово *автор* - гипероним для слова *Шекспир*), *письменный стол* - гипоним для *мебели*, и так далее. Для поиска кандидатов на гипонимы-гиперонимы, можно поискать в корпусе выражения вида: *«B is a A»*, *«B is a kind of A»*, or *«B, C and other As»*. Аналогичные шаблоны были найдены и для отношения мерономии (часть-целое) [1, 6]. Данный подход основан на предположении, что в достаточно большом корпусе текстов содержится «определение» слова B, написанное по одному из таких шаблонов. Набор лексических шаблонов может быть построен как вручную, так и автоматически с помощью машинного обучения. В отличие от статистического подхода, лексические шаблоны позволяют выделять гиперонимы даже при единичном вхождении соответствующей фразы в корпус текстов. Применение полученных данных к задаче измерения близости позволяет получить результаты лучшие, чем при статистическом подходе [2]. Это достаточно неожиданный вывод, так как совсем неочевидно, посредством каких шаблонов можно выразить близость между такими словами как *гвоздь* и *молоток*, *группа* и *гомоморфизм*.

В классическом, лингвистическом определении гипонимии, гиперонимом называет слово, множество поименованных объектов которым является подмножеством поименованных объектов гипонима. В работе [8] и других работах, связанных с лексическими шаблонами, принято более широкое понятии гипонимии: пара (T,H) рассматривается как гипоним-гипероним, если утверждение «T is a H» имеет смысл. Стоит отметить, что в связи с разнообразием синтаксических форм и бóльшей выразительностью пунктуации применение в русском языке шаблонов, аналогичных английским, не представляется перспективным. Так, в русском отсутствует однозначный аналог «индикатора» гиперонимии «is a».

Применение лексических шаблонов сталкивается с двумя затруднениями: небольшим набором шаблонов, которые можно использовать для обычных текстов (что, впрочем, одновременно является и плюсом), и шум на выходе. Существует сотни тысяч страниц, содержащих шаблон «robbery is a», и только сто страниц, содержащих «oroshi is a». В последнем случае, несмотря на то, что в выдаче поисковой системы находиться страница с правильным определением, автоматически извлечь её не представляется возможным из-за «нестандартного» построение предложения: «Oroshi is a word with a very specific meaning: wind blowing down from the mountain». Проанализировав такое предложение, система сделает ошибочный вывод, что *oroshi* является гипонимом *word*. Для более редких термов шаблонов не будет найдено вовсе. Комбинированный, лексико-статистический метод был недавно предложен в [5].

Цель исследования, результаты которого представлены в настоящей работе, — построение атомарной единицы иерархии понятий (пары гипоним-гипероним) для английского

языка. То есть для заданного гипонима с использованием лексических шаблонов и поисковой системы (в качестве единственного источника информации) необходимо построить его гиперонимы.

2 Лексические шаблоны и предшествующие работы

Лексические шаблоны впервые были введены в [8] как средство для автоматического извлечения гипонимов. Было замечено, что если два терма A и B, то скорее всего они встречаются в конструкции вида *A is a (kind of) B*. Таким образом, из таких конструкций можно получить список возможных гипонимов. В отличии от статистических методов, лексические шаблоны позволяют выявлять гипонимы (и гиперонимы) заданного терма по единственному вхождению в корпус анализируемых текстов. Привлекательным свойством таких методов является возможность их реализации с помощью широко используемых интернет-поисковиков, что позволяет запускать алгоритм на огромном корпусе текстов. С другой стороны, возможности таких поисковиков ограничены их языком запросов, обработка которых, как правило, не обеспечивает производительность, необходимую для решения сложных задач, связанных с обработкой естественных языков. В случае, если документы доступны целиком, можно использовать дополнительную информацию. Например, в [15] используется структура HTML-документов. Алгоритм основан на предположении, что термы, встречающиеся в HTML в неупорядоченных списках (тег *<ul>*), могут быть гипонимами одного слова, которое, вероятнее всего, находится в тексте сразу перед списком. В [18] представлен метод автоматического построения ключевых слов (атрибутов) объектов, комбинирующий лексические шаблоны с анализом HTML.

Основные проблемные вопросы и задачи, связанные с применением лексических шаблонов, — это фильтрация результатов, ограниченный, построенный вручную набор шаблонов, работа с многозначными словами и выделение редких значений слов.

Многих исследователей смущает малочисленность и искусственное построение шаблонов. В качестве решения этой задачи может рассматриваться метод их автоматического построения, предложенный в [17]. Используя фиксированный набор составленных вручную гипонимических пар из WordNet, из корпуса текстов извлекаются контексты, в которых эти пары встречаются. Далее, по контекстам строятся лексические шаблоны (у авторов получилось около 70000 шаблонов). С помощью статистических методов каждому шаблону приписывается вес — бóльший вес соответствует бóльшей корреляции между событиями вхождения пары слов в данный шаблон и существования гипонимической связи между ними. Следует отметить, что среди шаблонов с наибольшим весом были и все шаблоны, предложенные [8]. Методы, использующие данный алгоритм, позволяют получить существенное увеличение точности по сравнению с теми, в которых применяются «наивные» алгоритмы, использующие один шаблон. Однако его использование требует существенно бóльшего времени для вычислений. С другой стороны, метод, предложенный в [13] использует ограниченный набор шаблонов, но предоставляет сравнимые результаты.

Можно выделить две основные причины ошибки выделения гипонимов. Первый тип ошибок связан с *неправильной* структурой предложения. Этот факт означает, что синтаксическое значение шаблона не совпадает с предполагаемым. Так, предложение «... urban birds in cities such as pigeons ...», подходит под шаблон «Hs such as T», где T - это «city», а Y - «pigeon». Получаем, что *pigeon* (голубь) - гипоним для *city*. Это связанно с локальной природой шаблонов, когда синтаксический шаблон заменяется лексическим. Следует заметить, что проверка предложений на синтаксическую корректность при работе с поисковой системой не представляется возможной в связи с тем, что возвращаемые сниппеты редко содержат предложения целиком. Второй тип ошибок связан определением формы того, что мы хотим найти. В рамках общеупотребительной лексики более длинные понятия являются общими для более коротких. (Стоит заметить, что это не верно для таких областей, как, например, химия или медицина). С другой стороны, при «укорачивании» некоторые гиперонимы теряют свой смысл (например, *branch* как гипероним для *algebra*). В рамках данной работы мы будем рассматривать в качестве гиперонимов как слова, так и словосочетания, используя понятие «фраза» для обоих вариантов. При выделении кандидатов-гиперонимов наибольший интерес представляют фразы являющиеся именными группами. Такое ограничение порождает две задачи. Во-первых, необходимо уметь определять части речи в предложении. Во-вторых, в случае длинной именной группы зачастую неочевидно, что следует рассматривать как фразу-гипероним. Рассмотрим пример: *Typhoon is a twin*

engine canard delta wing multirole aircraft. Из этого предложения можно выделить сразу ряд корректных фраз, а именно — *aircraft*, *multirole aircraft*, *twin engine canard delta aircraft*, *twin engine canard delta wing multirole aircraft* и другие, а также некоторые некорректные *twin canard aircraft*, *wing aircraft*. Как и в первом случае, система не в состоянии проверить корректность из контекста. Для решения задачи фильтрации можно использовать внешние источники (например, в [21] для этих целей используется Wikipedia), или дополнительные запросы к поисковой системе ([13], [10]). Поскольку одной из основных задач данного исследования является разработки алгоритма с минимальным использования внешних данных, используется второй из рассматриваемых вариантов.

3 Наивный алгоритм выделения гиперонимов

В статье [8] был предложен метод автоматического построения гипонимических пар, основанный на лексических шаблонах. С помощью шаблона «Hs such as T» по известному гипериониму H находились его гипонимы. Данный метод позволял добиться приемлемых результатов, однако обладал двумя существенными ограничениями. Во-первых, поиск вниз (от гиперонима к гипонимам) представляет меньший интерес, чем поиск вверх (от гипонима к гиперонимам), так как поиск вверх позволяет дополнительно ответить на вопрос «что это?». Во-вторых, поиск производился по энциклопедии (Academic American Encyclopedia, 1980). Как уже отмечалось, использование специально подготовленных текстов позволяет уменьшить количество ошибок (в связи с более стандартизированной лексикой и отсутствием фактологических ошибок), но существенно уменьшает терминологический охват.

В данной работе в качестве корпуса выступают все тексты из Интернет и для поиска по ним используется система YahooBOSS API, позволяющая выполнять запросы к поисковой системе Yahoo и получать результаты (фрагменты текста, *сниппеты*) в удобном для машинной обработки виде (XML или JSON). С помощью кавычек и символа звёздочки «*» можно искать точное вхождение фразы, в котором на месте звёздочки стоит ровно одно слово. Например, запросу «moscow is a * city» удовлетворяет сниппет «moscow is a big city», но не «moscow is a city» или «moscow is a really big city».

Наивный алгоритм поиска гиперонимов заданного терма T состоит в выполнении запроса «T is a *» (например, «chimpanzee is a *») и выборе слов, которые стоят на месте звездочки в результатах выполнения запроса. Заметим, что описанный способ выделения гиперонимов совпадает с предложенным в [8]) за исключением того, что мы используем «is a» вместо «such as». Для ранжирования полученных результатов можно предложить два варианта. Во-первых, каждому слову можно сопоставить число — количество найденных сниппетов, в которых оно входило. Во-вторых, для каждого слова H можно вычислить количество хитов для фразы «chimpanzee is a H» (здесь и далее под количеством хитов понимается количество найденных страниц в поисковой системе для данного запроса, а все запросы выполняются «с кавычками»). Однако для слова *chimpanzee* наиболее частотными оказываются слова great, very, hug и tour, ни одно из которых не является корректным гиперонимом. Предполагалось, что вместо H будут стоять понятия, однако в действительности получаем произвольные слова.

Таблица 1: Гиперонимы для слова *chimpanzee*.

слово	**вхождений в запрос**
ape	59
animal	34
hug	22
bust	18
de	16
model	15
species	12
primate	11
synthesis	9
year	8

Таблица 2: Гиперонимы для слова *orthodoxy*.

слово	**вхождений в запрос**
series	26
book	20
movement	17
way	13
part	10
wave	9
religion	9
term	7
matter	7
tradition	7

Анализ показывает, что самое частотное вхождение (great) почти всегда порождается фразой вида: «. . . Dwarf or Gracile Chimpanzee is a great ape . . . ». Естественной эвристикой в данном случае представляется выбор последнего существительного в первой именной группе, что может быть сделано с использованием статистических методов определения частей речи (так называемые POS-тэггеры). В результате её применения (таблица 1) наиболее частотными кандидатами для слова *chimpanzee* оказываются «ape», «monkey» и «hug» (крепкое объятие, захват). Однако частоты этих слов настолько близки, что никакого естественного решающего правила, которое бы приняло первые два слова, но отвергло бы «hug», построить нельзя. Более того, самое частотное слово не всегда является корректным гиперонимом. Так, для слова *orthodoxy* (православие) первый правильный гипероним *religion* занимает 7-ю строку (таблица 2). Таким образом, алгоритм из [8] может использоваться для построения множества кандидатов, однако для выделения гиперонимов необходимо проведение дополнительной фильтрации.

4 Фильтрация и уточнение результатов

Повышение эффективности наивного алгоритма связано с разработкой метода фильтрации кандидатов, который позволит исключить или уменьшить количетво «мусора», а также с более точным определении фразы, являющейся в рамках заданного шаблона гипонимом или гиперонимом. Для создания фильтрующего критерия предлагается ввести набор числовых характеристик, признаков, позволяющих оценить гипонимичность заданной пары слов. Уточнение гипонимов и гиперонимов осуществляется на основании частотного анализа.

4.1 Правые и левые шаблоны

Первая группа критериев состоит из количества вхождений пары (T,H) в шаблоны специального вида. В [13] было предложено поделить лексические шаблоны на *правые* и *левые* (те, в которых гипероним стоит справа и слева, соответственно). Пример правого шаблона — стандартный «T is a (kind of) H», примеры левого — «Hs such as T» или «Hs including T». В работе было отмечено, что для каждый гипонимической пары (T, H) требование вхождения хотя бы в один правый шаблон и хотя бы в один левый шаблон, позволяет существенно увеличить точность при сохранении полноты. Необходимо отметить, что из метода построения кандидатов не следует, что каждый из них вместе с исходным словом входит в какой-либо правый шаблон. Это обстоятельство связано со спецификой выделения и определения термов. Например, в случае, если в качестве термов рассматриваются именные группы, могут быть выделены некорректные кандидаты, для которых запрос «T is a H» может вернуть пустой результат (например, для запроса «typhoon is a twin canard aircraft»). По этой причине на данном шаге нельзя ограничиться только левыми шаблонами.

Кроме правых и левых шаблонов предлагается используется несколько шаблонов, позволяющих оценить распространённость гипонима и гиперонима. Они призваны помочь ответить на следующие вопросы: десять хитов для шаблона «T is a H» — это много или мало? является ли T элементарным понятием? является ли H высшим, наиболее общим понятием? Такими шаблонми являются «T», «T is a *», «H is a *» и «* is a T».

4.2 Построение братьев для данного гипонима

В работе [10] было введено понятие *семантических классов*, — множества слов, имеющих общего гиперонима. Примерами семантических классов служат названия стран или игр с мячом. В терминах иерархии понятий элементы семантических понятий — братья (т.е. элементы поддерева с корнем в гипероними). Для поиска братьев T относительно H в [10] предлагается использовать шаблон «Hs such as T and *». Слова, стоящие на месте звёздочки и будут искомыми братьями. Преимуществом данного алгоритма является малое количество мусора за счёт использования двух слов — H и T. Однако данный алгоритм обладает и недостатками. Во-первых, такой шаблон накладывает слишком жёсткие требования на расположение слов. Вместо «and» могут использоваться запятые, а исходный элемент семантического класса T не обязательно встречается сразу после «such as». Во-вторых, использование одного шаблона ограничивает область поиска.

Таблица 3: Братья «orthodoxy» относительно «religion».

baptism	catholicism	catholocism
gnosticism	heterodoxy	islam
lamaism	maruni christians	nestorianism
pentecostals	protestantism	sacraments
sunni islam	vatican	vestments
words and symbols		

Предлагаемая далее модификация решает эти проблемы. Определим *уровень братьев* следующим образом. Братьями T относительно H уровня 0 называется множество, содержащее единственный элемент — T. Братьями T относительно H уровня $n + 1$ называется множество, для каждого терма в котором существует брат n-го уровня, встречающийся с ним в одном шаблоне, и не существует братьев меньшего уровня, удовлетворяющих этому условию. Для поиска братьев использовался запросы вида «T and other Hs», «T * and other Hs», ..., и «Hs such as T», «Hs such as * T», ..., которые возвращают связные фрагменты текстов, содержащие H и T.

Задача извлечения братьев представляет отдельный интерес (например, в статье [10] поиск семантических классов рассматривается в качестве основной задачи). В рамках данной работы в большей степени важна чувствительность данного метода к ошибкам определения гипернимов. Рассмотрим, например, самый частотный гипероним-кандидат для «orthodoxy» — «series». Для этой пары алгоритм не находит ни одного брата. С другой стороны, для пары «orthodoxy» — «religion» получаем 16 братьев уже на первом уровне (приведены в таблице 3). Таким образом, количество братьев на каждом уровне можно считать содержательным признаком пары.

5 Расширение гиперонимов и гипонимов

В предыдущем разделе предложены признаки, которые призваны помочь в построении критерия «хорошести» пары. Вместе с тем, не менее интересен и другой вопрос: откуда берутся неправильные варианты? Если не рассматривать ошибки POS-тэггера и фактологические ошибки, то можно выделить две причины: либо в исходном фрагменте определялось что-то не то, что мы искали, либо не так, как мы нашли. В пером случае слева от «is a» стояло более специализированное (и, наверное, более длинное) понятие. Во втором — мы выделили понятие справа слишком грубо. Именно методы расширения, уточнения фраз, разработанные для идентификации и исправления таких ошибок, описаны далее в настоящем разделе.

В качестве примера расширения гиперонимов рассмотрим пару (ubuntu, distribution), полученную при выделении кандидатов, например, из фрагмента «I have to admit folks, Ubuntu is the best distribution in all these years». Является ли она гипонимической? С одной стороны, представляется очевидным, что пара (ubuntu, linux distribution) является гипонимеской, и, как следствие, исходную пару стоит считать такой. С другой стороны, основное значение слова «distribution» — распространение, распределение. Среди нескольких первых страниц по запросу «distribution» нет ни одного результата связанного с дистрибутивами операционных систем.

Попробуем удлинить гипероним влево на одно слово. Выполним запрос «ubuntu is a * distribution» и выберем 500 результатов. В найденных фрагментах выделим множество слов, стоящих на месте звёздочки. Каждому слову можно сопоставить пару — количество найденных сниппетов, в которых это слово входило, и отношение этого числа к количеству всех сниппетов. Очевидно, что если какое-то расширение содержится в большинстве результатов, то его следует принимать в качестве «лучшего» гиперонима. Наиболее популярными расширениями оказываются *linux* (381), *great* (33) и *popular* (16). Согласно предыдущему рассуждению вместо «distribution» следует принять «linux distribution». Дальнейшее расширение влево не даёт явного лидера. Аналогично можно расширять фразы вправо, а также комбинировать направления.

Однако расширения не являются панацеей от ошибок выделения кандидатов. Более того, они привносят новые задачи. Во-первых, поиск функции штрафа, функции остановки является нетривиальной задачей. Более того, выбор единственного кандидата не все-

гда является наилучшим решением. Например, расширение «type of» вправо в контексте «chimpanzee is a type of» приводит к двум равноправным вариантам: «chimpanzee is a type of ape» и «chimpanzee is a type of monkey». Процент вхождения каждого в выборку — 40%. Таким образом, использование глобальной «планки» неэффективно. В реализованном алгоритме используются комбинированный подход. С помощью «низкой планки» отсекаются результаты, которые не могут рассматриваться как расширения. Далее, среди оставшихся с помощью кластеризация выбирается «головные» распределения.

Во-вторых, так как расширения работают со словами как со знаками, нет никаких гарантий синтаксической корректности результатов. Например, после расширения «creature» в контексте «elephant is a creature» имеем два варианта: «nobble creature of special» и «creature of majesty and great power and may». Оба этих расширения не являются синтаксически целостными. Для решения этой задачи можно применить таггер и отбросить все слова после последнего существительного. Получим: «nobble creature» и «creature of majesty and great power».

Как отмечено ранее, результат расширения даже в одну сторону неоднозначен. Должно ли расширение заменять исходное слово, а именно — является ли наличие расширения (множества расширений) признаком «качества» исходной пары? В текущей реализации все эти вопросы переложены на механизмы фильтрации в пространстве признаков: все найденные расширения рассматриваются как полноценный кандидаты в гиперонимы.

Аналогичным образом можно производить и расширения гипонимов. В некоторых случаях найденный гипероним соответствует некоторому побочному значению гипонима. Например, рассмотрим пару (scanner, program). Очевидно, что она является ошибочной. Если посмотреть на фрагменты, в которых встречается данная пара, то можно встретить выражения вида «scanner is a program designed for searching for httpssocks». Для выделения таких случаев предлагается использовать расширение гипонимов. Выполним запрос «* scanner is a program» и обработаем полученные фрагменты также, как и ранее. Получим *proxy scanner* (147), *a scanner* (56), *ftp scanner* (51), *port scanner* (29). Таким образом, наличие нетривиального расширения можно рассматривать как признак того, что гипероним относится к побочному значению гипонима, а наличие тривиального расширения — дополнительным признаком «качества» пары. Под тривиальными расширениями понимается исходное слово, а также артикль в сочетании с исходным словом.

Очевидно, что для более точного, длинного гиперонима существование расширенного гипонима более вероятно, чем гиперонима, представленного одним словом. С другой стороны, если гипероним расширяется «мягко» (то есть с низким порогом, при котором расширение принимается), то и наличие расширения гипонима для какого-либо другого варианта расширения гиперонима более вероятно. В предельном случае, если принимаются все найденные фрагменты (всё, что стоит после «is a») в качестве расширений, то для каждого из них будет существовать расширение гипонима — всё, что стоит до «is a». По этой причине используются только «жёсткие» расширения. Жёсткое левое расширение получается в результате рекурсивного расширения гиперонима влево, до тех пор, пока есть вариант, входящий в более чем в половину сниппетов в относительном исчислении и более чем в 30 — в абсолютном. Аналогичным образом получаем правое жёсткое расширение. Например, для пары (scanner, tool) такими расширениями будут «network scan tool» и «tool», соответственно.

6 Пример работы алгоритмы

Введём обозначения для числа хитов, найденных для шаблонов, описанных в разделе 4:

- `isa_strict` — «T is a H»
- `isan_strict` — «T is an H»
- `isthe_strict` — «T is the H»
- `kindof_strict` — «T is a kind of H»
- `suchas_strict` — «Hs such as T»
- `including_strict` — «Hs including T»
- `especially_strict` — «Hs especially T»

- `word_hits` — «T»
- `isa` — «T is a *»

Пусть `brothers1` — количество братьев первого уровня пары; `hyponym`, `hypernym` — гипоним и гипероним соответственно; `hyponym_extension_all`, `hyponym_extension_all_from_left`, `hyponym_extension_all_from_right` — расширения гипонима в общем случае, при жёстком левом расширении гиперонима и при жёстком правом расширении гиперонима соответственно; `extension_depth` — 1, если гипероним является расширением, и 0 — иначе. Для краткости обозначим `ln(word_hits)` за `ln_w_hits`.

Тогда критерий для гиперонимов, которые не являются расширениями (т.е. для гиперонимов, состоящих из одного слова), формулируется следующим образом. Гипероним является корректным, если одновременно выполнены следующие условия:

- число братьев первого уровня больше 10;
- `isa_strict>ln_w_hits*1.5` или `isan_strict>ln_w_hits*1.5` или `isthe_strict>ln_w_hits*1.5` или `kindof_strict>ln_w_hits`;
- `suchas_strict>ln_w_hits` или `especially_strict>ln_w_hits` или `including_strict>ln_w_hits`;
- расширения гипонима в общем случае, при жёстком левом и жёстком правом расширениях гиперонима либо не существуют, либо содержат тривиальное расширение.

Критерий для гиперонимов, которые являются расширениями, формулируется следующим образом. Гипероним является корректным, если одновременно выполнены слудующие условия:

- существуют братья первого уровня;
- `isa_strict>0` или `isan_strict>0` или `isthe_strict>0` или `kindof_strict>0`;
- `suchas_strict>0` или `especially_strict>0` или `including_strict>0`;
- расширения гипонима в общем случае, при жёстком левом и жёстком правом расширениях гиперонима либо не существуют, либо содержат тривиальное расширение.

Структура критерия продиктована идеями, обсуждаемыми в разделе 4. Численные значения были подобраны с помощью обучающей выборки.

Для тестирования был выбран набор слов, для каждого из которых построены кандидаты в гиперонимы, что дало 5311 пар фраз. К ним был применён предложенный в данном разделе критерий гипонимичности. Некоторые примеры представлены в таблице 4. Корректные (на основе экспертной оценки) гиперонимы выделены жирным. Их количество в процентном отношение ко всем гиперонимам составляется 62%. Следует отметить, что отношение выбранных критерием пар ко всем парам — 2%.

Таблица 4: Пары слов, отобранные критерием, и их основные признаки.

hyponym	hypernym	brothers1	isa_strict	isan_strict	isthe_strict	kindof_strict	suchas_strict	including_strict	especially_strict
ak47	**gun**	27	30	0	15	0	35	46	2
ak47	**machine gun**	1	12	0	1	0	1	13	0
ak47	**weapon**	40	28	0	48	0	113	100	4
begonia	**flower**	36	33	0	11	0	25	22	1
begonia	**plant**	115	29	0	1	0	182	32	3
chimpanzee	**mammal**	20	11	0	41	0	6	21	0
chimpanzee	**primate**	28	21	0	9	1	84	111	10

Продолжение на следующей странице

Таблица 4: Пары слов, отобранные критерием, и их основные признаки.

hyponym	hypernym	brothers1	isa_strict	isan_strict	isthe_strict	kindof_strict	suchas_strict	including_strict	especially_strict
daiquiri	**cocktail**	16	60	0	3	1	17	5	0
daiquiri	**drink**	12	23	0	9	0	16	0	3
elephant	**animal**	110	26	281	103	5	1140	626	63
elephant	**creature**	31	44	0	0	0	29	19	0
elephant	film	14	114	0	12	0	76	50	7
elephant	**herbivore**	37	67	47	0	0	74	46	6
elephant	**large mammal**	42	32	0	0	0	112	105	5
elephant	**mammal**	74	121	0	9	5	332	256	10
guitar	c instrument	8	26	2	0	0	5	45	0
guitar	combination	21	243	0	23	0	21	14	0
guitar	course	55	118	2	19	0	52	51	2
guitar	**fretted string instrument**	3	26	0	0	0	17	39	0
guitar	hobby	37	135	0	2	1	56	60	0
guitar	part	31	192	0	92	0	87	98	6
guitar	skill	43	190	1	9	0	62	29	2
guitar	string	25	175	0	52	4	46	185	11
guitar	style	13	110	0	62	0	14	98	6
guitar	talent	14	79	0	4	0	50	35	4
guitar	thing	56	155	0	145	0	116	63	21
ipv6	**new protocol**	10	68	0	49	0	42	2	0
ipv6	new service	1	9	0	0	0	14	3	0
ipv6	**protocol**	80	113	0	81	0	440	179	13
ipv6	solution	16	73	0	131	0	55	33	2
ipv6	**standard**	40	41	0	43	0	133	86	1
ipv6	**technology**	80	79	0	23	0	843	166	0

7 Заключение

В настоящей статье предложен метод автоматического построения гипонимов/гиперонимов с помощью единственного источника данных — внешней поисковой интернет системы. Построенные пары гипоним-гипероним являются атомарными единицами иерархии понятий, которая может применяться не только как основа для построения онтологий, но и для расширения поисковых запросов, классификации текстов и других задач, связанных с обработкой естественных языков. Ключевое отличие данной работы заключается в том, что используются неподготовленные тексты, полученные с помощью интернет-поисковика, и построение ведётся снизу вверх (от гипонима к гипериниму). В работе исследована проблема выделения и уточнения кандидатов-гиперонимов; оптимизированы методы фильтрации, основанные на количестве хитов и количестве братьев [10]; предложен метод отделения побочных значений исходного терма; представлены результаты тестовых испытаний программных средств, реализующих предложенные алгоритмы.

В качестве возможного продолжения работы можно рассматривать задачи поиска редких значений, более тонкого оценивания качество братьев, а также перенесение описанных методов на русский язык.

8 Благодарности

Работа выполнена при поддержке гранта РФФИ № 09-07-00366-а.

Список литературы

[1] Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64, Morristown, NJ, USA, 1999. Association for Computational Linguistics.

[2] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 803–812, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[3] Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14, New York, NY, USA, 2007. ACM.

[4] Chenghua Dang, Xinjun Luo, and Haibin Zhang. Wordnet-based summarization of unstructured document. *W. Trans. on Comp.*, 7(9):1467–1472, 2008.

[5] Lucas Drumond and Rosario Girardi. Extracting ontology concept hierarchies from text using markov logic. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1354–1358, New York, NY, USA, 2010. ACM.

[6] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 1–8, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[7] Z. S. Harris. *Mathematical Structures of Language*. Wiley, New York, 1968.

[8] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545, 1992.

[9] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA, 2006. ACM.

[10] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the Association for Computational Linguistics*, 2008.

[11] Jianqiang Li, Yu Zhao, and Bo Liu. Fully automatic text categorization by exploiting wordnet. In *AIRS '09: Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, pages 1–12, Berlin, Heidelberg, 2009. Springer-Verlag.

[12] Vanessa Lopez, Victoria Uren, Enrico Motta, and Michele Pasin. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Web Semant.*, 5(2):72–105, 2007.

[13] A Ritter, S Soderland, and O Etzioni. What is this, anyway: Automatic hypernym discovery. In *In Proceedings of AAAI-09 Spring Symposium on Learning*, pages 88–93, 2009.

[14] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA, 1999. ACM.

[15] Keiji Shinzato and Kentaro Torisawa. Extracting hyponyms of prespecified hypernyms from itemizations and headings in web documents. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 938, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[16] Barry Smith. Ontology. In *The Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Blackwell, 2004.

[17] R. Snow, D. Jurafsky, and A.Y. Ng. Learning syntactic patterns for automatic hypernym discovery. *Advances in neural information processing systems*, 17:1297–1304, 2005.

[18] Kosuke Tokunaga and Kentaro Torisawa. Automatic discovery of attribute words from web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05), pages 106–118, Jeju Island, Korea*, pages 106–118, 2005.

[19] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 467–474, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[20] Gertjan van Heijst, A. Th. Schreiber, and Bob J. Wielinga. Using explicit ontologies in kbs development. *Int. J. Hum.-Comput. Stud.*, 46(2):183–292, 1997.

[21] I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Saeger, F. Bond, and A. Sumida. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 929–937. Association for Computational Linguistics, 2009.