Ю. В. Малинина

АВТОМАТИЧЕСКОЕ ВЫЯВЛЕНИЕ ТАКСОНОМИИ В ОБЛАСТИ ПРЕОБРАЗОВАНИЙ ПРОГРАММ НА ОСНОВЕ АНАЛИЗА СЕМАНТИЧЕСКИХ СВЯЗЕЙ В ПУБЛИКАЦИЯХ

ВВЕДЕНИЕ

В конечном счете научное знание воплощается в тексты и познается через тексты, поэтому можно допустить, что абстрактная модель научного знания является производной от множества реальных текстов и может рассматриваться как обобщающее представление этих текстов на уровне семантических моделей.

То есть задача построения классификации преобразований программ на данном уровне может быть интерпретирована как построение семантических сетей публикаций в области преобразования программ, содержащих информацию о научных терминах и их связях между собой с последующим обобщением их в единую семантическую сеть предметной области.

Следует заметить, что с 90-х годов множество проектов, посвященных автоматическому извлечению терминов, было выполнено, однако попрежнему остаются открытыми следующие вопросы [7]:

(1) идентификация сложных терминов, особенно вопрос начала и окончания терминологической фразы произвольной длины;
(2) распознавание сложных терминов, то есть решение составляет набор слов;
(3) соответствие терминологической единицы словарю предметной области.

На основе сравнения существующих систем авторы [7] заключают, что для того, чтобы улучшить извлечение терминов и сделать результат более релевантным, то есть уменьшить информационный шум и потери, следующие условия должны быть выполнены.

Во-первых, должны быть проведены лингвистически ориентированные исследования семантических связей терминов и условий ограничения терминологических единиц в пределах данной специальной области и в данном текстовом типе.

Во-вторых, программные системы должны научиться сочетать статистические и лингвистические методы и поддерживать более одной стратегии. Также должна быть полезной разработка общей шкалы тестирования и оценки/сравнения качества извлекаемых терминов.

1. ТИПЫ СЕМАНТИЧЕСКИХ СВЯЗЕЙ В ТЕКСТЕ

Сжатие информации при переходе от лексического к семантическому описанию документов приводит к ее обобщению, что эквивалентно получению некоторого знания. Ведь возможность более сжатого описания данных есть следствие скрытых в этих данных закономерностей. Сжатие информации по сути и сводится к выявлению этих закономерностей, выражающих наши знания о структуре данных.

Семантические сети с самого начала активно использовались для построения систем обработки естественного языка. В семантическую сеть включаются наиболее часто встречающиеся слова текста, которые несут основную смысловую нагрузку. Для каждого понятия формируется набор ассоциативных (смысловых) связей, т.е. список других понятий, в сочетании с которыми оно встречалось в предложениях текста. При этом считается, что чем чаще встречаются вместе два термина в предложениях текста, тем выше вероятность того, что они связаны по смыслу.

При этом нельзя сбрасывать со счета особенности стиля научного текста. Общеизвестно, что для научного стиля характерно использование следующих языковых средств:

- на уровне лексики:
  - насыщенность терминами данной предметной области;
  - использование слов с абстрактным значением: закон, число, предел, свойство; отглагольных существительных со значением действия: переработка, приземление, использование;
  - употребление слов в прямых значениях, отсутствие образности (метафор, междометий, восклицательных частиц);
  - частое использование лексических средств, указывающих на связь и последовательность мыслей: сначала, прежде всего, во-первых, следовательно, наоборот, потому что, поэтому;
- на уровне морфологии:
  - редкое использование личных местоимений я и ты и глаголов в форме 1 и 2 лица единственного числа;

- специальные приемы авторизации: авторское «мы», неопределенно-личные (считают, что...) и безличные конструкции (известно, что...; представляется необходимым...);
- использование причастий, деепричастий и оборотов с ними;

- на синтаксическом уровне:
  - употребление сложных предложений с использованием союзов, указывающих на связь явлений;
  - неупотребление восклицательных предложений, незначительное употребление вопросительных предложений;
  - частые цитаты, ссылки;
  - использование в качестве компонентов текста формул, графиков, схем.

Перечисленные выше языковые средства в свою очередь могут быть использованы как индикаторы семантических отношений.

Рассмотрим несколько основных семантических отношений, которые можно выделить как ключевые для построения классификации терминов предметной области.

1.1. Вариантность и синонимия

К синонимии относятся отношения, основанные на полном или частичном совпадении значений. Слова, связываемые отношением синонимии, называются синонимами [8].

Существует по крайней мере 2 разных подхода к определению синонимии:

- семантическая близость слов: в данном случае в качестве критерия рассматривается наличие у слов некоего общего значения [11].
- взаимозаменяемость в контексте: два слова считаются синонимами, если существует высказывание (или ряд высказываний), в котором замена этих слов друг на друга не влияет на истинность высказывания ([12, 13, 14]).

Таким образом синонимия − это отношения тождественных (или близких) семем, формально выраженные разными лексемами, например: Elimination / Removal.

Elimination − act of getting rid of; omission, act of leaving out; process of solving simultaneous equations by removing the variables (Algebra).

1) удаление; исключение; выбрасывание; 2) отсев; выбывание; 3) устранение; уничтожение, ликвидация; 4) физиол. очищение; выделение; экскреция, удаление из организма; 5) мат. исключение (неизвестного).

Removal − act of taking off, act of shedding; act of taking away; elimination; ejection, dismissal.

1) перемещение; переезд; вывоз; 2) смещение (с должности); 3) устранение, удаление; ликвидация; исключение; 4) горн. выемка перемещение.

Синонимия тесно связана с проблемой распознавания употребленных в научных текстах терминов и понятий.

Согласно [4] в терминоведении до некоторого времени преобладал подход, согласно которому в специальных текстах одно понятие должно иметь только один способ выражения. Соответственно, синонимия и полисемия терминов признавались нежелательными явлениями, и научно-технические терминологии подвергались логической и лингвистической регламентации, направленной на устранение этих явлений. Однако стремительный рост числа новых специальных терминов, сопровождавший развитие науки и техники, и неизбежное функционирование в речи большого количества синонимичных вариантов, а также исторические особенности употребления терминов различными научными школами и сообществами требуют учета вариативности специальных терминов.

Особенно хорошо это заметно в области преобразования программ.

Например, в одних работах используется термин dead code elimination (устранение неиспользуемого кода) [15, 16,17,18], а в других dead code removal (удаление неиспользуемого кода) [20, 21, 22].

1.2. Гипонимия

В лингвистической литературе принято слова, обозначающие родовые понятия, называть гиперонимами, а слова, обозначающие видовые понятия – гипонимами [8].

Гипонимия — это отношения родовидового включения, формально выраженные разными лексемами: *дом/строение*. Гипероним обозначает общее родовое понятие или совокупность, целое по отношению к составляющим его элементам, частям. Гипоним обозначает видовое понятие или название элемента, части какого-нибудь множества, целого.

Гиперонимы и гипонимы образуют гипонимические ряды, в которых гипонимы занимают подчиненное положение по отношению к гиперонимам. В гипонимический ряд входят один гипероним, занимающий ведущее место и обозначающий общее понятие, и один или несколько гипонимов, занимающие подчиненное положение по отношению к нему [8].

Например, согласно [10] мы можем выделить следующие гипонимические связи между терминами в области преобразования программ.

- Partial Evaluation (частичные вычисления) включает:
  - Constant Propagation (втягивание констант);
  - Constant Folding (свертка констант);
  - Copy Propagation (втягивание копий выражений);
  - Statement Substitution (подстановка выражений);
  - Reassociation (перегруппировка);
  - Algebraic Simplification (алгеброическое упрощение);
  - Function Cloning (клонирование функции);
  - I/O Format Compilation (компиляция формата ввода-вывода).

- Redundancy Elimination (устранение избыточности) включает:
  - Unreachable Code Elimination (устранение недостижимого кода);
  - Useless Code Elimination (устранение бесполезного кода);
  - Dead Variable Elimination (устранение мертвых переменных);
  - Common Subexpression Elimination (устранение общих подвыражений).

2. ПОДХОД К РЕАЛИЗАЦИИ

2.1. Предварительная обработка

Как уже предлагалось в [1], текст будем рассматривать, как множество соответствующих основ слов, входящих в него. Последовательность обработки текста будет следующая:

1. Построение неупорядоченного списка слов текста с учетом пропуска так называемых стоп-слов.
2. Выделение списка основ слов (stemming).

3. Подсчет частоты вхождения и частоты появления рядом слов текста.
4. Составление списков близких слов, на основе частоты появления рядом.
5. Определение меры близости для любых двух слов.

2.2. Извлечение терминов

Для автоматического извлечения многословных терминов предполагается использовать хорошо известный подход основанный на N-gram модели, которая является разновидностью вероятностной модели для определения следующего члена последовательности.

N-gram размера 1 называется «unigram»; размера 2 − «bigram» (или, менее обычно, «биграмма»); размера 3 является «триграммой»; и размера 4 или более просто «n-gram».

Согласно [5], алгоритм прямого подсчета количества пар (freq), который является модификацией метода автоматического выделения двусловий на основе bigrams, может быть использован для составления списка терминов-кандидатов в задачах полуавтоматического формирования терминологических ресурсов. Он основан на простейшем методе упорядочиния двусловий по убыванию их встречаемости в тексте

Например, для фрагмента *Strength reduction is a compiler optimization where a costly operation is replaced with an equivalent, but less expensive operation. Operator strength reduction involves using mathematical identities to replace slow math operations with faster operations,* получаем, что только словосочетание «strength reduction» встречается больше одного раза и может служить кандидатом термина.

И далее для терминов любой длинны указанный метод модифицируется наращиванием словосочетаний, если более короткие словосочетания достаточно часто встречаются в составе более длинных.

Для повышение качества алгоритмов на множестве текстов выбранной узкой специальности за счет удаления устойчивых выражений общей лексики желательно использовать дополнительный «контрастный» корпус текстов более широких областей науки. Следует также отметить, что этот способ выделения терминов эффективен только при использовании достаточно большого корпуса общенаучных текстов [6].

2.3. Выявление синонимов и вариантов терминов

Стратегия распознавания синонимов (вариантов) терминов в общем случае опирается на разбиение всего набора терминов и кандидатов в термины, полученных на предыдущем шаге, на группы синонимичных вариантов (одна группа соответствует одному понятию) на основе похожести терминов с учетом словарей синонимов общей лексики и словарей сокращений

Например,

- синонимия ключевого слова в словосочетании Dead code elimination, Dead code removal может быть определена на основе словаря;
- использование сокращений Input/Output Format Compilation, I/O Format Compilation.

2.4. Выявление гипонимов/гиперонимов

Шаблонный метод извлечения гипонимов и гиперонимов (для английского языка) описан в [9], он основан на извлечении гипонимов с помощью заранее заданных лексико-синтаксические шаблонов. Фактически утверждается что для всех семантических конструкций выделенных по шаблону

NP0 <шаблон> {NP1, NP2, …, (and|or)} NPn.

Существует связь

for all Npi, 1<= i <=n, гипоним (Npi, NP0),

где <шаблон> = “such that” (такой как), “or other” (или др.), including (включая), especially (особенно)

Например

Although related to caching, memorization refers to a specific case of this optimization, distinguishing it from forms of caching such as buffering or page replacement.

Гипоним (buffering, forms of caching); Гипоним (page replacement, forms of caching)

ЗАКЛЮЧЕНИЕ

Описанные подходы могут быть полезны для проведения терминологического анализа текста в системах литературно-научного редактирования, а также для автоматического реферирования и аннотирования документов в информационно-поисковых системах

Автоматическое извлечение гипонимических рядов терминов из текста с учетом синонимии открывает дополнительные возможности построением автоматической классификаций предметной области.

СПИСОК ЛИТЕРАТУРЫ

1. Малинина Ю.В. Семантическая сеть как формальный метод описания и обработки текстов по преобразованиям программ // Методы и инструменты конструирования и оптимизации программ. – Новосибирск, 2005. – С. 137–144.
2. Большакова Е.И., Васильева Н.Э., Морозов С.С. Лексико-синтаксические шаблоны для автоматического анализа научно-технических текстов // Тр. 10-й национальной конф. по искусственному интеллекту с международным участием – КИИ'2006. – М.: Физматлит, 2006. – Т. 3. – С. 832–840.
3. Лавренова О.А. Моделирование семантической структуры текстов научно-технического содержания в связи с автоматизацией информационных процессов. Дис. канд. филол. наук. – М.: Московский гос. университет, 1978. – 280 с.
4. Большакова Е.И., Васильева Н.Э.. Терминологическая вариантность и ее учет при автоматической обработке текстов // Тр. XI Национальной конф. по ИИ с международным участием КИИ-2008, г. Дубна. 29 сентября–3 октября 2008 г. – http://www.raai.org/cai-08/files/cai-08_paper_208.doc
5. Браславский П.И., Соколов Е.А. Сравнение пяти методов извлечения терминов произвольной длины // Тр. конф. ДИАЛОГ, 2008. – С. 67–74.
6. Митрофанова О.А., Мухин А.С., Паничева П.В. Автоматическая классификация лексики в русскоязычных текстах // Тр. междунар. конф. «Диалог'2007» (Бекасово, 30 мая–3 июня 2007 г.). – С. 413–422.
7. Castellvi M., Bagot R., Palatresi J. Automatic term detection: A review of current systems // Recent Advances in Computational Terminology. – Amsterdam: John Benjamins, 2001. – P. 53–87.
8. Новиков Л.А. Семантика русского языка. – М.: Высшая школа, 1982
9. Hearst M. Automatic Acquisition of Hyponyms from Large Text Corpora // Proc. of COLING 92, Nantes. – 1992. – Vol. 2. – P. 539–545.
10. Bacon D.F., Graham S.L., Sharp O.J. Compiler transformations for high-performance computing // ACM Computing Surveys. – 1994. – Vol. 26, № 4. – P. 345–420.

11. Словарь синонимов русского языка: В 2 т. / АН СССР, Институт русского языка; Под ред. А. П. Евгеньевой. Л.. – Наука, Ленинградское отделение, 1970.
12. Lyons J. Semantics. (2 vol.) – London; New York, 1977.
13. Miller G. et al. Five Papers on WordNet. – Princeton University, 1990. – (CSL-Report / Vol. 43).
14. Новый объяснительный словарь синонимов русского языка. / Под рук. Ю. Д. Апресяна. – М: «Языки русской культуры». Вып. 1, 1997.
15. Knoop J., Ruthing J., Steffen B. Partial dead code elimination // ACM SIGPLAN Notices. – 1994. – Vol. 29, N 6. – P. 147–158.
16. Hongwei Xi. Dead code elimination through dependent types // The First Internat. Workshop on Practical Aspects of Declarative Languages (PADL '99). – Lect. Notes Comput. Sci. – 1999. – Vol. 1551. – P. 228–242.
17. Gupta R. Partial dead code elimination using slicing transformations // Proc. of the ACM SIGPLAN '97 Conf. on Programming Language Design and Implementation (PLDI). – SIGPLAN Notices. – 1997. – Vol. 32(5). – P. 159–170.
18. Ancourt C. etc. How to add a new phase in PIPS: The case of dead code elimination // Sixth Workshop «Compilers for Parallel Computers» (CPC'96), Aachen, Konferenzen des Forschungszentrums Jülich. – P. 19–30.
19. Gold N., Harman M. An empirical study of static program slice size // ACM Transactions on Software Engineering and Methodology (TOSEM). – Vol. 16 , Iss. 2. – http://www.cs.loyola.edu/~binkley/papers/tosem-slice-size.ps
20. Chang P., Scott A. and etc. Using pro le information to assist classic code optimizations // Software Practice and Experience. – 1991. – Vol. 21(12). – P. 1301–1321.
21. Madou M., Van Put L., De Bosschere K.. Understanding Obfuscated Cod e// Proc. of the 14th IEEE Internat. Conf. on Program Comprehension (ICPC). – IEEE Computer Society, 2006. – P. 268–274.