OSTIS-2016
(Open Semantic Technologies for Intelligent Systems)

УДК 004.891

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ ДЕРЕВА КЛАССОВ ОНТОЛОГИИ И ИХ ЭКЗЕМПЛЯРОВ НА ОСНОВЕ ТЕКСТА ПО ПРЕДМЕТНОЙ ОБЛАСТИ

Андреев И.А., Бексаева Е.А., Желепов А.С., Клейн В.В., Серков И.П.

Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация

ares-ilya@ya.ru
ekaty@clukva.ru
a.zhelepov@yandex.ru
vikklein93@gmail.com
bivo@bivo.ru

В данной статье рассматривается алгоритм формирования дерева классов онтологии и их экземпляров на основе текста предметной области. Описывается техническая реализация программной системы, входные и выходные данные, оценивается результат работы реализованного алгоритма. Предлагаемый подход основан на применении лингвистических шаблонов и тезаурусов по предметным областям.
Ключевые слова: онтология; автоматическое формирование онтологии; дерево классов; экземпляры классов.

Введение

Задача формирования онтологии предметной области связана с задачами информационного поиска [Мозжерина, 2011], извлечения терминологии [Андреев и др., 2014а], интеграции разнородных баз данных [Соловьев и др., 2006]. Использование онтологии в качестве инструмента при решении задачи предполагает привлечение эксперта по предметной области, его обучению работе с онтологиями с последующим процессом ручного формирования самой онтологии.

Недостатками данного процесса является увеличение материальных и временных затрат на решение задачи, возможные неточности в результате работы эксперта. Таким образом, автоматизация процесса формирования онтологии, имеющая целью избавление от упомянутых недостатков, является актуальной задачей.

В данной статье приводится алгоритм автоматизации процесса формирования онтологии предметной области на основе текстов по соответствующей предметной области. В основу предлагаемого алгоритма положены преимущественно лингвистические методы.

В качестве базы знаний используется тезаурус по соответствующей предметной области. Особенностью алгоритма является зависимость результата от содержания, как тезауруса, так и текста, подаваемого на вход. Тезаурусы по предметным областям предварительно составляются экспертом-лингвистом, что делает их содержание зависимым от субъективного фактора, уровня подготовки эксперта, а также актуальности использованных при составлении источников. Для получения качественного результата подаваемый на вход алгоритму текст должен содержать термины, релевантные рассматриваемой предметной области, иначе в результате онтология не будет в полной мере представлять текст.

Предлагаемый алгоритм решает задачу построения дерева классов, а также выделяет экземпляры классов, но это не является полным решением задачи автоматического формирования онтологии. Классы в онтологии вместе с экземплярами являются основой для последующего извлечения связей, атрибутов и их значений. В данной статье рассматривается только начальный этап автоматического формирования онтологии из текстов по определенной предметной области.

Таким образом, целью данной статьи является описание разработанного алгоритма формирования дерева классов и их экземпляров и некоторые особенности его технической реализации.

1. Онтология, классы и их экземпляры

1.1. Понятие и модель онтологии

Большой спектр решаемых с использованием онтологии задач породил множество определений понятия «онтология», большинство которых рассматриваются в [Клещев и др., 2000].

В данной статье онтология понимается как формальное представление предметной области, представимое, согласно [Гаврилова и др., 2000] в виде тройки:

$$O = \langle T, R, F \rangle, \quad (1)$$

где T – конечное множество терминов рассматриваемой предметной области; R – конечное множество отношений между понятиями рассматриваемой предметной области; F – конечное множество функции аксиоматизации, заданных на множестве T и R. При этом присутствует обязательное требование непустоты множества T.

1.2. Классы и их экземпляры

Классы (или понятия) онтологии вместе с их экземплярами (или индивидами) являются основой онтологии любого типа, применяются и строятся одинаково практически в любом языке описания онтологий.

Классы онтологии представляют собой коллекции объектов, сформированные по определенному принципу. Классы могут включать в себя как другие классы, так и экземпляры. Возможно содержание одним классов как классов-потомков, так и собственных экземпляров. Таким образом, классы представляют собой иерархию понятий по отношению вложения.

Экземпляры классов в онтологии являются компонентами нижнего уровня. Выражают как физические, так и абстрактные объекты. Формально онтология может не включать в себя ни одного экземпляра класса.

Кроме классов и их экземпляров онтология содержит отношения и атрибуты. Извлечению этих компонентов из текста предшествует формирование дерева классов и их экземпляров.

2. Формирование дерева классов и экземпляров

Разработанный алгоритм автоматического формирования онтологии включает в себя этапы первичной обработки текста, извлечения терминов, извлечения классов и извлечения экземпляров.

2.1. Первичная обработка текста

На вход реализованной информационной системе подается текст на естественном языке. Для определения части речи слова в тексте и правильного составления термина по лингвистическому шаблону используются инструменты Mystem [Андреев и др., 2013] и LingvoNET [Торгашов, 2015].

Mystem — морфологический анализатор русского языка с поддержкой снятия морфологической неоднозначности. Программа работает на основе словаря и способна формировать морфологические гипотезы о незнакомых словах.

LingvoNET - библиотека для склонения и спряжения слов русского языка для .NET Framework. Библиотека содержит встроенные словари глаголов, существительных, прилагательных и местоимений русского языка на основе словаря Зализняка.

Текст, который был размечен программой Mystem, обрабатывается информационной системой с помощью библиотеки LingoNET и заносится в базу данных.

2.2. Извлечение терминов

Для построения понятийного аппарата из текстов предметной области используется поиск и выделение субстантивных именных словосочетаний, выражаемых схемой: согласуемое слово + существительное. В этой модели существительное является главным словом, а согласуемое слово — зависимым и может выражаться как прилагательным, так и существительным. Словосочетания могут включать в свой состав также предлоги и сочинительные союзы.

Количество слов в именных словосочетаниях колеблется от двух до семи и в среднем составляет три слова. В ходе работы были выделены шаблоны именных словосочетаний, используемых для выделения терминов предметной области.

Для возможности выделения терминов из текстов предметной области были разработаны лингвистические шаблоны, с помощью которых удается выделить основные термины. В русском языке синтаксическая структура терминов предметной области более чем в 90 процентов случаев соответствует следующим пяти шаблонам:

- одиночные существительные;
- существительное + существительное в родительном падеже;
- прилагательное + существительное;
- прилагательное + прилагательное + существительное;
- существительное + прилагательное + существительное в родительном падеже.

Вместе с тем существуют сложные словосочетания, используемые для обозначения понятий и терминов, состоящих из трех и более значимых слов.

Выражение понятий и терминов словосочетаниями в пять и более слов, с использованием союзов и предлогов встречается

редко, особенно такими словосочетаниями, в которых части речи не чередуются (например, прилагательное + прилагательное + прилагательное + существительное + существительное в родительном падеже).

2.3. Автоматическое формирование дерева классов из текстов по предметной области

На начальном этапе построения онтологии необходимо построить дерево классов. Для разработки инструмента, осуществляющего формирование дерева классов, был разработан алгоритм автоматического построения дерева классов на основе данных из тезауруса, а также применялась методика слабоиерархической онтологии [Загорулько и др., 2006]. Данные алгоритмы представляют собой последовательность шагов по обработке характеристик, их структурированию, а также выделению классов на их основе.

Рассмотрим данные алгоритмы более детально. На первом этапе анализа текстов выделяются термины по лингвистическим шаблонам рассмотренных в параграфе 2.2. После выделения списка терминов они проходят отбор по стоп-словам. Стоп-слова были выделены экспертом лингвистом и в основном включают в себя: местоимения, междометия, частицы, вводные слова и т.п.

На втором этапе происходит непосредственно построение дерева классов. Формирование классов происходит при сопоставление терминов, выделенных из текста, с тезаурусом по предметной области предварительно составленный экспертом-лингвистом, который представляет следующие данные:

- Лемма термина;
- Отношение термина с другими терминами в тезаурусе (в качестве основного отношения рассматривается гипероним-гипоним).

Сам процесс построения тезауруса состоит из нескольких взаимосвязанных этапов:

Первый этап - формирование словаря. Словарь – первоначальные множества ключевых слов. При этом рассматривается представительный массив наиболее информативных для данной предметной области документов.

Выбираются слова, употребляемые в этих источниках, при всем этом устанавливается частота употребления слов, и учитываются все формы, которые могут иметь слова.

Второй этап – формирование множества ключевых слов. Из словаря формируется множество ключевых слов. При отборе ключевых слов учитывается информативность слова, которая определяется исходя из частоты встречаемости слова, роли слова в этой предметной области. Процесс выбора ключевых слов достаточно сложно формализовать.

Например, такой критерий, как частота встречаемости не может быть абсолютным. Если слово встречается в текстах очень часто, это может означать, что оно выражает чрезмерно широкое понятие, либо недостаточно четко определено, т.е. неинформативно.

Если ключевое слово встречается очень редко, это может означать, что оно выражает новое понятие и таким образом является информативным.

Третий этап – формирование классов эквивалентности. Выделение дескрипторов. Автоматические информационно-поисковые тезаурусы являются составным элементом автоматического индексирования документов и запросов.

В словарной статье автоматического тезауруса, как правило, зафиксированы отношения условной эквивалентности, отношения подчинения и ассоциативные отношения.

Данные по тезаурусу хранятся в базе данных и представлены в следующем виде:

Id	Lemma	Topic
1	механика	Физика
2	сила	Физика
3	сложение сил	Физика
4	параллелограмм сил	Физика

Рисунок 1 – Тезаурус

Сопоставление терминов происходит по полному совпадению данных с леммой термина из тезауруса. При полном совпадении выделенный термин применяет на себя свойства тезауруса.

Данные по свойствам тезауруса хранятся в таблице данных ThesaurusProperty и представлены в следующем виде:

Id	Property	Subject	LemmaId
10	hyperonym	сила	7
11	hyperonym	сила	8
12	hyperonym	сила	9
13	hyperonym	сила	10

Рисунок 2 – Свойства тезауруса

Данные по выделенным терминам хранятся в таблице данных FilteredWord и представлены в следующем виде:

Id	Lemm	Template
20	согласование характеристики дизеля	SSS
21	история создания тепловоза	SSS
22	история создания передачи	SSS
23	сгорание жидкого топлива	SAS

Рисунок 3 – Выделенные термины

На следующем этапе происходит выделение классов по следующему алгоритму:

1. У выделенной Lemma из таблицы ThesaurusLemma рассматривается свойство hyperonym из таблицы ThesaurusProperty;
2. Subject из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а Lemma подклассом;

3. Ищем выделенный Subject из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma. Если Subject и Lemma полностью совпадают, то мы возвращаемся к шагу 1 и начинаем выделять надклассы у Subject.

4. Если у выделенного Subject с Lemma совпадений не найдено. То на этом шаге выделение классов останавливается.

Например, у нас есть следующие данные в таблице ThesaurusLemma:

Таблица 1 – Пример данных в базе данных ThesaurusLemma

Id	Lemma	Topic
1	блок цилиндров	машины
2	бензиновый двигатель	машины

Они соотносятся со следующими данными в таблице 2.

1. Был выделен термин “блок цилиндров”;
2. В таком случае “бензиновый двигатель” из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а “блок цилиндров” подклассом;
3. Ищем выделенный класс “бензиновый двигатель” из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma на полное совпадение. При анализе было выделено совпадение. “Бензиновый двигатель” имеет Id равное 2 в таблице ThesaurusLemma. Повторяем шаги 1, 2 и 3 еще раз:
   1. Был выделен термин “бензиновый двигатель”;
   2. В таком случае “двигатель” из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а “бензиновый двигатель” подклассом;
   3. Ищем выделенный класс “двигатель” из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma на полное совпадение. При анализе совпадений не найдено;
   4. Выделение классов остановлено. Получена следующая структура:

Класс: Двигатель
Подкласс1: Бензиновый двигатель
Подкласс2: Блок цилиндров

Таблица 2 – Пример данных в базе данных ThesaurusProperty

Id	Property	Subject	LemmaId
1	hyperonym	бензиновый двигатель	2
2	hyperonym	двигатель	3

На третьем этапе рассматриваются термины, которые отсутствуют в тезаурусе. К ним применяется методика слабоиерархической онтологии. Выделенные термины объединяются в слабоиерархическую онтологию.

Основанием для объединения типа родовидового было вхождение термина из среднечастотной зоны в словосочетание в качестве главного компонента, например: воздействие => антропологическое воздействие; изотоп => тяжелый изотоп. Термины, которые входят в выделенные словосочетания в качестве зависимых, позволяют объединять словосочетания в ассоциативные группы, например: почва <= эрозия почвы / плодородие почвы / дефляция почвы.

2.4. Автоматическое формирование экземпляров

Выделение экземпляров можно разделить на два этапа. На первом этапе происходит определение последнего элемента в цепочке дерева классов по следующему правилу: если Lemma из таблицы ThesaurusLemma не имеет совпадений в столбце Subject в таблице ThesaurusProperty, то термин из столбца Lemma является последним в цепочке классов.

На втором этапе происходит непосредственно определение экземпляров. Для возможности выделения экземпляров из текста были разработаны следующие лингвистические шаблоны, с помощью которых удается выделить экземпляры: существительное + цифра, существительное + буква + цифра, существительное + буква, существительное + существительное + буква, существительное + существительное + цифра.

На начальном этапе термины по выше выделенным лингвистическим шаблонам из таблицы FilteredWord сопоставляются с последним элементом в цепочке классов. Сопоставление происходит по полному совпадению существительных.

Например, есть два правила существительное + существительное + буква и прилагательное + существительное, если существительное из правила прилагательное + существительное совпадает с существительным из правила существительное + существительное + буква, то термин существительное + существительное + буква является экземпляром в цепочке классов прилагательное + существительное.

3. Техническая реализация проекта

Для данного проекта был выбран язык C# (Microsoft Visual Studio 2015), на платформе .NET 4.5.2 и СУБД Microsoft SQL Server 2012.

При разработке инструмента была использована технология Entity Framework. В Entity Framework модель данных реляционной базы данных сопоставляется объектной модели, выраженной в языке программирования разработчика.

При запуске приложения Entity Framework преобразует запросы LINQ из объектной модели в SQL и отправляет их в базу данных для выполнения. Когда база данных возвращает результаты, Entity

Framework преобразует их обратно в объекты, с которыми можно работать на собственном языке программирования.

Основное предназначение разработанной базы данных заключается в хранении сущностей выбранной пользователем онтологии и предоставление данных для методов, осуществляющих работу алгоритма.

После написания программы, система должна выполнять следующие функции с БД:

- Отображать содержимое базы данных;
- Выводить информацию из базы данных на экран;
- Удалять информацию из базы данных;
- Предоставлять данные для методов, осуществляющих подсчет критериев.

Таким образом, программа использует БД, которая хранится в двух файлах: с расширением mdf и ldf, которые используются запускаемым в операционной среде экземпляром СУБД.

Взаимодействие с системой осуществляется через web-интерфейс, расположенный по адресу http://auto-ontology.ru. Доступ пользователя к сайту осуществляется при помощи любого современного браузера.

На стороне сервера работают службы IIS, обеспечивающие выполнение скомпилированных исходных файлов системы. Для автоматической первичной обработки текста на сервере должен содержаться исполняемый файл Mystem, который выполняется при соответствующем запросе пользователя. Кроме того, на сервере должна быть размещена СУБД MS SQL Server 2012.

Данная конфигурация системы имеет 2 основных преимущества: независимость от платформы и облачная обработка данных. Таким образом, система доступна почти с любого современного устройства и не использует его ресурсов для собственной работы.

Из недостатков следует отметить зависимость от подключения к интернету.

4. Оценка качества построенной онтологии

Проблема оценки качества онтологии является одной из актуальных проблем современного онтологического инжиниринга. Распространенным критерием оценки качества онтологии является оценка работы приложения, использующего онтологию.

В настоящее время за методику оценки качества онтологии можно рассматривать системы семантического поиска, которые используют онтологии.

Применяя онтологию, поисковые системы могут улучшить качество поиска за счет динамического расширения запросов пользователя. Для таких систем необходимо обрабатывать большое количество текстовой информации и переработка данных текстов в онтологию при помощи экспертов. Но экспертная оценка трудозатратна.

Построение качественной онтологии без помощи эксперта усложняется отсутствием некоторого эталона онтологии. В связи с этим, для проверки качества построенной онтологии имеет смысл проводить начальную оценку работы алгоритма, только в одной тематике (например, машиностроение).

Именно по этой теме был построен тезаурус терминов, что позволит оценить работу метода на первых этапах.

Заключение

В статье рассматривается подход к автоматизации процесса построения онтологии по текстам предметной области, на основании лингвистических шаблонов, тезаурусов и методики слабоиерархической онтологии.

В результате работы был создан инструмент автоматизированного создания онтологии предметной области. Полученная в результате онтология представляет собой структурированную систему базы знаний на основе текста предметной области.

Использование данного инструмента позволит значительно сократить затраты времени на составление и редактирование онтологии и будет полезен во многих областях знаний, таких как компьютерная и корпусная лингвистика, лексикография, библиотечное дело, семантический поиск и многие другие.

Разработанный инструмент находится в свободном доступе и расположен по адресу: http://auto-ontology.ru

Библиографический список

[Мозжерина, 2011] Мозжерина Е. С. Автоматическое построение онтологии по коллекции текстовых документов //Электронные библиотеки: Перспективные Методы и Технологии, Электронные коллекции–RCDL. – 2011. – С. 293-298.

[Соловьев и др., 2006] Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Учебное пособие. – Казань, Москва. – 2006.

[Андреев и др., 2014] Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г. Семантическая метрика терминологичности на основе онтологии предметной области // Автоматизация процессов управления. – 2014. – № 4 (38). – С. 76 – 84.

[Гаврилова и др., 2000] Гаврилова Т. А., Хорошевский В. Ф. базы знаний интеллектуальных систем: учеб. для вузов. — СПб.: Питер, 2000. — 384 с.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В.Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[Клещев и др., 2000] Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия «онтология».

// Научно-техническая информация, серия 2 «информационные процессы и системы». — М.: ВИНИТИ, 2001. — №2. — С. 20–27.

[**Андреев и др., 2013**] Андреев И. А., Башаев В. А., Клейн В. В. Разработка программного средства для извлечения терминологии из текста на основании морфологических признаков, определяемых программой Mystem //Интегрированные модели и мягкие вычисления в искусственном интеллекте». – М.: Физматлит. – 2013. – С. 1227-1236.

[**Торгашов, 2015**] NuGet Gallery | LingvoNET - Библиотека для склонения и спряжения слов русского языка 1.1.0 [Электронный ресурс] // NuGet Gallery [Интернет-портал]. URL: https://www.nuget.org/packages/LingvoNET/ (Дата обращения: 29.11.2015)

[**Загорулько и др., 2006**] Загорулько Ю. А. и др. Подход к построению предметной онтологии для портала знаний по компьютерной лингвистике //Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог». – 2006. – С. 148-151.

[Ярушкина и др., 2014] Ярушкина Н.Г., Мошкин В.С. Применение онтологического подхода к анализу состояния локальной вычислительной сети. Радиотехника. – 2014. – № 7. – С. 120-124.

AUTOMATIC CREATION OF CLASS HIERARCHY AND SEARCH INSTANCES FROM TEXT FOR ONTOLOGY

Andreev I.A., Beksaeva E.A., Zhelepov A.S, Klein V.V., Serkov I.P.

Ulyanovsk State Technical University, Ulyanovsk, Russia

ares-ilya@ya.ru

ekaty@clukva.ru

a.zhelepov@yandex.ru

vikklein93@gmail.com

bivo@bivo.ru

The article deals with the algorithm of automatic creation of class hierarchy and search instances from text of subject domain for ontology.

Introduction

Over the past decade, ontologies and knowledge bases have gained popularity due to their high potential benefits in a number of applications including data/knowledge organization and search applications. Though ontology integration is beneficial, it is very well known that ontology creation is an expensive process.

The modeling of non-trivial domain ontologies is difficult, and is time and resource intensive. The knowledge acquisition bottleneck problems in ontology creation and maintenance have resulted in expensive procedures for maintaining and expanding the ontology library available to support the growing and evolving needs of analysts in various domains.

Ontology generation is the automatic creation of ontologies, including extracting the corresponding domain's terms and the relationships between those concepts from a corpus of natural language text. As building ontologies manually is extremely labor-intensive and time consuming, there is great motivation to automate the process.

Main Part

This article provides the algorithm of automatic creation of ontology from text. The proposed algorithm is put mainly linguistic methods. The algorithm is based mainly linguistic processors such as phrase chunking.

We described a method to automatically extract key terms, and relationships between the key terms.

For automatic building class hierarchy and searching instances used specialized thesaurus and concept hierarchy derivation. Specialized thesaurus are made expert and submit the following information: lemma of term and the relationship between terms in the thesaurus.

In the concept hierarchy derivation step, the system tries to arrange the extracted concepts in a taxonomic structure. This is mostly achieved by unsupervised hierarchical clustering methods.

Conclusion

Knowledge intensive applications require extensive domain-specific knowledge in addition to general-purpose knowledge bases.

However, domain-specific ontology creation and maintenance is an expensive process and hence is referred to as the knowledge acquisition bottleneck. Thus, you need an expert to be involved either in the process of creating the ontology itself and to evaluate the effectiveness of the developed algorithm.

In this paper, we presented a generalized and improved procedure to automatically extract information from text resources and rapidly create ontologies while keeping the manual intervention to a minimum.

We also defined evaluation metrics to assess the quality of the ontologies created using our methodology. The results show that a decent amount of knowledge can be accurately extracted while keeping the manual intervention in the process to a minimum.

Another conclusion we came into is that you need to use the developed algorithm in several real tasks before evaluating its effectiveness.

Using the developed algorithms, a tool that uses C# and Microsoft SQL Server 2012 has been created for automatic creation of ontology from text. This tool is available at auto-ontology.ru. Using this tool you can make an ontology of your own, based on a text you choose.