**Использование гиперграфовой базы данных HypergraphDB для работы со словарями WordNet**

Лопатин Илья, студент магистратуры

Московский Государственный Технический Университет имени Н.Э.Баумана

*В данной статье рассматриваются ключевые аспекты работы с лингвистической моделью WordNet внутри графовой базы данных HypergraphDB*

***Ключевые слова****: база данных (database), HypergraphDB, гиперграф (hypergraph), лингвистическая модель WordNet.*

Реляционные базы данных уже долгое время занимают доминирующее положение в современном информационном мире. Реляционные базы просты в использовании, имеют ясный, стандартизированный вид и во многих случаях обеспечивают хорошую скорость работы. Однако при большом количестве связей между таблицами, приходится применять все больше уровней вложенности, используя конструкции SELECT…JOIN…, а также, в классическом варианте отношения многие-ко-многим, создавать промежуточную таблицу. При большом объеме данных и уровне вложенности производительность запроса резко падает.

Именно поэтому, все чаще стали применяться так называемые графовые БД. Основной отличительной особенностью графовых баз данных является их уникальная модель данных, которая позволяет хранить объекты и связи в качестве узлов и связующих их ребер. Это позволяет получить хорошую скорость обработки запроса при емких запросах.

Однако, для выполнения простых запросов графовые базы данных не подходят, так как проигрывают реляционным. Именно поэтому, перед тем как сделать выбор в пользу таких БД, нужно ясно представлять архитектуру разрабатываемого проекта.

Большинство людей с легкостью могут привести пример применения графовых баз данных. Первое, что приходит на ум – социальные сети. И это действительно один из самых распространенных случаев применения, поскольку связи между людьми представляют собой граф.

Однако есть и другой яркий пример применения графовых бд – обработка естественного языка. В данной области активно применяется лингвистическая модель WordNet (WN), разработанная Принстонским университетом в 1985 г.

Не вдаваясь в подробности можно сказать, что WN это множество английских слов, связанных различными видами отношений, такими как синонимия, гипонимия, меронимия и т.д. Уже из этого краткого определения, видно, что применение реляционной базы данных в рамках работы с WN, будет неэффективным.

В случае с WN следует отдать предпочтение графовой БД. Однако таких баз данных не мало (FlockDB, Giraph, Neo4j и др ). Какую из них стоит выбрать?

Поскольку модель WN представляет собой не просто набор слов и их значений, а многочисленные иерархические списки слов и связей, то для ее эффективного применения необходима база данных, в основе которой лежит математическое понятие графа и его разновидностей, в частности – гиперграф (мультиграф). В 2010 году компанией Kobrix Software была разработана подобная база данных, которая получила соответствующее название HypergraphDB.

В данной статье будет рассказано о базовых возможностях применения HypergraphBD для работы с WN.

**Подготовка рабочего места**

Поскольку HypergraphDB написана только для Java, требуется установка Java выше 5 версии. HypergraphDB предоставляется в виде jar файлов, которые можно найти в maven репозитории. Следовательно, все, что нужно для подключения HypergraphDB к вашему проекту, это прописать соответствующие пути в вашем .pom файле (рис 1.).



Рисунок 1 - подключение HypergraphDB

Для работы с WN требуется скачать .zip архив с сайта <http://wordnet.princeton.edu/> и распаковать его. Все необходимые файлы лежат в архиве.

Также необходимо выгрузить проект Бориса Иорданова (Borislav Iordanov), представляющий собой набор классов составляющих Hypergraph wordnet (HGWN) API (рис.2). В системе контроля версий Github, данный разработчик известен под псевдонимом Bolerio.

На этом подготовка рабочего места заканчивается. Далее следует загрузить весь набор слов и связей из WN в HypergraphDB.

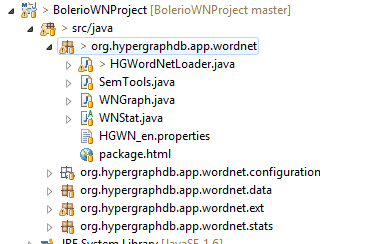


Рисунок 2 - Пакеты HGWN API

**Загрузка Wordnet в HypegraphDB**

Для выгрузки WN в гиперграф вам понадобиться класс HGWordNetLoader. В основе кода это класса лежит библиотека JWNL (Java WordNet Library) - - <http://sourceforge.net/projects/jwordnet> [1]. Все необходимые файлы конфигураций необходимые для этой библиотеки включены в выгруженный проект BolerioWNProject (рис. 2).

Простейший код выгрузки словаря Wordnet в HypergraphDB представлен на рисунке 3.

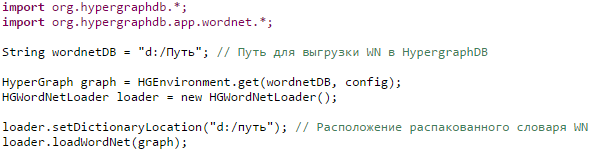


Рисунок 3 - Выгрузка WordNet в HypergraphDB

Стоит заметить, что скорость выгрузки сильно зависит от объема оперативной памяти доступной компьютеру. При 4 Гб оперативной памяти выгрузка занимает примерно 35 минут, а при 6 Гб – 17 минут. Тестирование проводилось на операционной системе Windows 7, с процессором Intel (R) Core (TM) i5 – 2410М 2.3 ГГц. Объем выгруженных данных составляет 831 Мб.

**Работа с WN внутри HypergraphDB**

Приложение HGWN это представление информации из WordNet внутри HypergraphDB. Оно включает в себя следующие функции:

1. запрос к базе на получение определенного слова;
2. запрос на нахождение определенного семантической или лексической связи для данного слова;
3. морфологический анализ – например, нахождение основной формы заданного слова или его предка и тд.

Одним из нюансов, с которыми может столкнуться начинающий разработчик при разработке HGWN приложения, это то, что HGWN API не предоставляет все методы необходимые для работы с Wordnet. К примеру, API не содержит методов, для поиска слов, связанных с определенными отношениями. Т.е если вы захотите найти синоним, гипоним и тд., какого-либо слова, например «dog», то вы не найдете соответствующего метода в HGWN API. По заявлению Бориса Иорданова, причина этому – большое количество различных видов отношений в WordNet, что требует большого количества кода и «утяжеляет» API. Подобные операции можно без особого труда можно выполнить посредством методов из HypergraphDB.

По большому счету, после того как вы загрузили WordNet в HypergraphDB, все что вам нужно это функционал HypergraphDB.

Самые базовые примеры работы с HGWN рассмотрим на примере небольшого приложения (от автора статьи), представляющего собой пользовательский интерфейс для поиска слов из WordNet. Интерфейс реализован с помощью технологии JavaFX (рис.4).

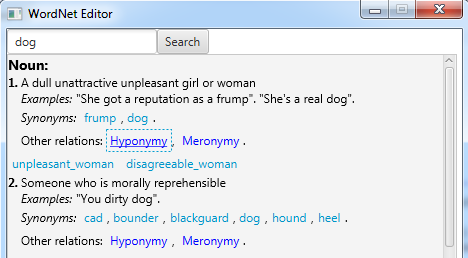


Рисунок 4 - Пользовательский интерфейс для словаря WordNet

Как видно из рисунка, функционал, реализованный в данном приложении, позволяет находить значение слова в разных частях речи (существительное(на рисунке), глагол и др), его синонимы, меронимы и гипонимы. На рисунке 5 показан код, позволяющий получить значение слова.

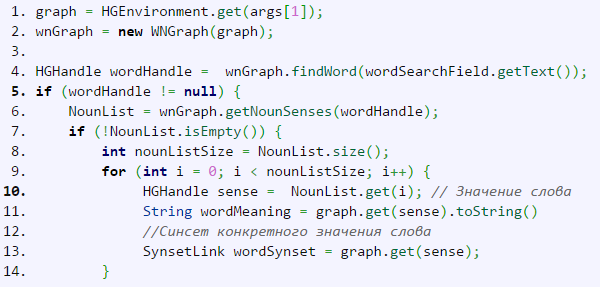


Рисунок 5 - Получение значения слова

В первой строке инициализируется гиперграф. Стоит заметить, что существует два способа инициализации. Первый – тот, что указан на рисунке 5, а второй это вызов конструктора Hypergraph(“Путь к HypergraphDB”). Первый метод является предпочтительней. В 4-й строке происходит получение указателя на искомое слово внутри HypergraphDВ.

Отметим, что wordHandle это не само слово, а всего лишь указатель на него. На строке 6 происходит получение списка указателей на значения искомого в роли существительного. Метод getNounSenses возвращает список HGHandle. На строке 10 идет получение указателя на одно из значений слова и уже на строке 11 происходит извлечение значения из гиперграфа при помощи указателя.

На строке 13 инициализируется синсет, связанный с искомым словом. Этот синсет необходим для поиска других слов связанных с искомым какими-либо отношениями. Важно понять, что поиск взаимосвязанных слов происходит по лексическому значению, а не по самому слову. Т.е если вы ищете синонимы слова «dog», то вы не можете просто написать что-то вроде findSynonym(«dog»). В качестве параметра вы должны передать значение этого слова в виде указателя. В HypergraphDB он будет представлен типом данных HGHandle. На рисунке 6 представлен поиск синонимов на основе полученного синсета wordSynset (рис.5).

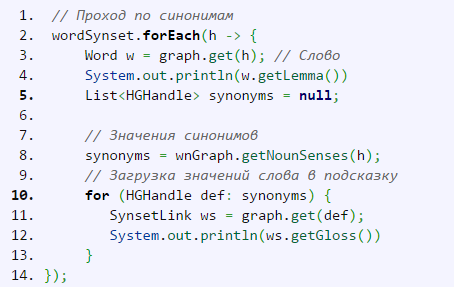


Рисунок 6 - Поиск синонимов

Получение синонимов происходит по следующему алгоритму: осуществляется проход по всем указателям внутри синсета, далее из графа извлекается каждое слово, связанное с указателем, после чего происходит поиск значений данного слова и поиск слов внутри графа, соответствующих данным значениям.

Проход по синсету является ключевым моментом для поиска слов связанных другими отношениями. Приведем пример поиска гипонимов. В HGWN этот тип отношения задается как KindOf. На рисунке 7 приведена реализация поиска.

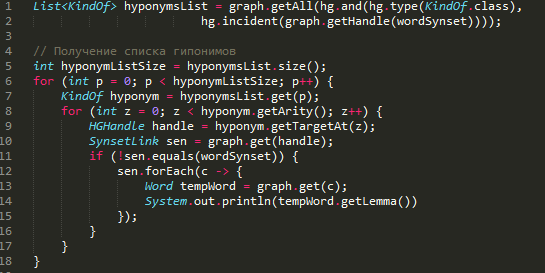


Рисунок 7 - Поиск гипонимов

На первой строке происходит получение списка гипонимов. Вызов graph.getAll(..) возвращает список указателей удовлетворяющих заданному условию [2]. Как видно из рисунка происходит поиск слов связанных отношением KindOf, и входящих в состав синсета wordSynset. HyponymsList представляет собой список указателей для всех значений искомого слова. Далее для каждого значения необходимо найти гипонимы (строка 7). Еще раз отметим, все связи ищутся для значения слова, а не для него самого. На строке 8 происходит проход по всем словам, связанным с конкретным значением искомого слова отношением KindOf. Строка 9 – получение указателя на гипоним. Строка 10 – получение синсета значений гипонима. Далее идет проход по списку значений. Условие на строке 11 необходимо для избежания вывода дубликатов на экран, поскольку значения гипонимов искомого слова ссылаются на само искомое слово и вместе находятся в одном синсете.

**Заключение**

Лингвистическая модель WordNet обладает большим количеством связей между словами, поэтому для работы с ней требуется графовая база данных. HypergraphDB позволяет работать с WN, преобразуя его в гиперграф. При работе с WN внутри HypergraphDB необходимо помнить, что вся работа производится со значениями слов, а не с самими словами. Для извлечения слов и их значений из гиперграфа используются указатели HGHandle.

Литература:

1. HyperGraphBD – WordNet Representation in HypergraphDB [Электронный ресурс]. –Режим доступа: <http://www.hypergraphdb.org/?page=WordNet&project=hypergraphdb>, свободный (25.04.2017).
2. Github – HypergraphDB GettingStarted [Электронный ресурс]. –Режим доступа: <https://github.com/hypergraphdb/hypergraphdb/wiki/GettingStarted> свободный (25.04.2017).