**Редактор языковых баз Wordnet с использованием гиперграфовой базы данных**

Лопатин Илья, студент магистратуры

Московский Государственный Технический Университет имени Н.Э.Баумана

*В данной статье рассматриваются лингвистическая модель Wordnet, гипеграфовые базы данных и технологии реализации интерфейса для редактора подобной языковой модели.*

***Ключевые слова****: база данных (database), редактор баз данных (database editor), гиперграф (hypergraph), языки (languages).*

Повсюду проникающая информатизация не обошла стороной коммуникативную часть жизни людей. Сегодня во всемирной паутине можно найти множество сервисов, используемых для перевода предложений с одного языка на другой. Одним из самых знаменитых сервисов является Google translate.

Для создания подобных ресурсов необходимы мощные инструменты систематизации, обработки и хранения данных.

В 1985 г. в Принстонском университете (США) была разработана первая модель ментального лексикона человека. Данный ресурс получил название Wordnet (WN), что в дословном переводе означает «Словесная сеть».

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.

Не многие знают о существовании подобных баз данных. Ранее в NOSQL среде была распространена всего одна система такого плана — Neo4j [1]. Сегодня одной из популярных баз данных такого типа является HyperGraphBD.

HypergraphBD представляет собой хранилище данных общего назначения, которое находится в свободном доступе и базируется на математическом представлении ориентированных гиперграфов [2].

**Лингвистическая модель Wordnet.**

Как уже было сказано, модель Wordnet впервые была представлена миру в 1985 году. Позднее были созданы аналоги для других языков. В частности, в 1999 был создан российский словарь RussNet, включающий в себя основные особенности WN.

Словари, построенные по модели WN, включают в себя, элементы как

справочной системы так, и инструмента для осуществления исследований в области лингвистики.

Ядром WN, является синсет. Синсет представляет собой совокупность синонимов, связанных определенными отношениями.

WN определяет различные виды отношений между синсетами: «is- a», «category», «has-parts», «is-part-of» и тд. Наиболее простым является отношение «is-a», которое связывает гипоним (более специфичный синсет) с гиперонимом (более общий синсет) [3]. К примеру, слово «наука» является гиперонимом к слову «физика», и наоборот, слово «физика» является гипонимом к слову «наука».

При построении модели данных по типу WN, первоочередной задачей является построение ориентированного графа, отображающего синсет. Каждая вершина графа *v* представляет собой целое число, отображающее синсет, и каждое ребро v → w, означает, что w это гипероним v. Ориентированный граф в WN является ациклическим, и имеет одну родительскую вершину, для которой все остальные вершины графа считаются дочерними. На рисунке 1 приведен пример графа в WN.



Рис.1. Ориентированный граф Wordnet

Из рисунка видно, что слово «bird» обозначает родительскую вершину. Ближние вершины (fowl, doll, hoot и тд.) являются гипонимами, т.к как отображают частные виды птиц, а вершина «vertebrate» связанная отношением «is-a» является гиперонимом. Также на рисунке отображены другие виды отношений.

Понимание вышеупомянутых особенностей модели WN является базовым и необходимым условием для работы с подобной моделью данных.

**Графовая база данных HyperGraphDB**

Как было сказано ранее, синсеты в WN реализуются в виде графов. Более того многие слова могут иметь множество значений, синонимов, гипонимов и т.д., что усложняет выбор базы данных, способной обеспечить корректную и удобную работу с подобной моделью данных. Одним из лучших вариантов для реализации хранения данных словаря WN можно смело назвать графовую базу данных HyperGraphDB.

В целом, HyperGraphDB представляет собой ровно то, о чем и говорит ее название - база данных для хранения гиперграфов. HypergraphDB - это не просто база данных, это целый инструментарий для работы с данными неоднородной структуры и различными слоями сложности. Так, к примеру, в HyperGraphDB можно реализовывать управление данными как в виде отношений, так и в виде объектно-ориентированной модели.

HyperGraphDB не накладывает никаких ограничений и предлагает большую обобщенность, в сравнении с другими графовыми базами данных. Для большей ясности стоит разобраться, что представляет собой понятие «гиперграф».

В математике, гиперграф - это обобщенное понятие графа, в котором ребра могут соединять любое количество вершин. Формально говоря, гиперграф *H* - это пара *H* = (*X*,*E*), где *X* - это набор элементов, называемых узлами или вершинами, и *E* - это набор не пустых подмножеств *X*, называемых гиперребрами или ребрами. Следовательно, *E* - это подмножество *P*(*X*) \{∅}, где *P*(*X*) – супермножество [2].

На рисунке 2 изображен пример математической модели гиперграфа, где и .

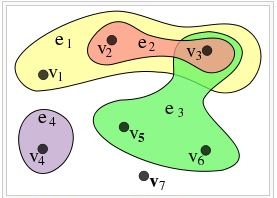


Рис. 2. Пример гиперграфа

Основное преимущество HyperGraphDB лежит в хорошо продуманной структуре, рефлексивной модели данных, в динамической схеме расширения и открытой архитектуре хранения, позволяющей производить специфичные доменные оптимизации. Также следует отметить, что гибкость структуры служит предпосылкой для увеличения производительности во время эксплуатации.

В то время как гиперграф может быть представлен в виде однородного графа, обратное также возможно, хоть и не тривиальным способом. Многие графы будут представлять собой повторяющиеся шаблоны, вытекающие из ограничений, которые связаны с классической моделью графа. В HyperGraphDB подобные шаблоны могут быть абстрагированы в виде гиперграфов, что приведет к уменьшению количества узлов и операций в базе данных. К примеру, потоковые графы, в которых ребра отображают многоканальные соединения входов-выходов, могут храниться намного более компактно при использовании гиперграфовой модели данных.

Применение графовых баз данных обоснованно при применении емких запросов, когда необходимо получить большое количество информации. В таких случаях графовые базы данных значительно превосходят реляционные. Однако применение графовых баз данных для простых запросов не целесообразно.

Приведем простой пример, демонстрирующий разницу между реляционными и графовыми базами данных.

Предположим, что в нашей реляционной базе данных используется связь многие –ко-многим. К примеру, нам необходимо хранить информацию о сотрудниках и отделах, в которых они работают. Традиционно в таком случае необходимо создавать третью (связующую) таблицу, хранящую пары ключей обеих таблиц (рис. 3).

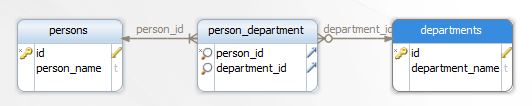


Рис. 3. Пример связующей таблицы

В таком случае наши запросы к такой базе данных будут содержать в себе вложенные команды select или join, и с каждой такой командой время выполнения запроса будет расти. Более наглядная иллюстрация подобной базы данных изображена на рис 4.

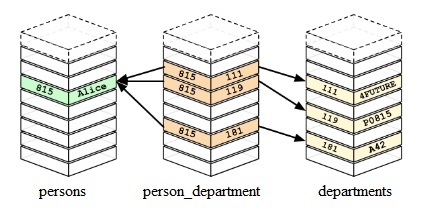


Рис. 4. Реализация в виде реляционной базы данных

На рисунке 5 изображена аналогичная база данных, но уже в виде графовой базы данных. Сразу видно, что теперь уже больше нет связующих таблиц, и данные из обеих таблиц соединены напрямую через ребра, что приведет к увеличению скорости обработки.

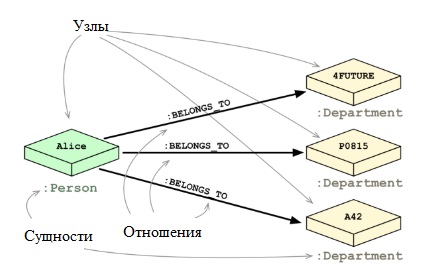


Рис. 5. Реализация в виде графовой базы данных

**Редакторы Wordnet**

WN имеет огромное значение в большинстве задач связанных с обработкой языковых структур. За все время существования WN было не так много систем, предоставляющих возможность редактировать WN. Основным недостатком разработанных систем являются строгие лицензионные ограничения, а так же тот факт, что большинство систем разработаны под какой-либо конкретный язык.

Именно поэтому разработка общедоступного редактора WN с поддержкой нескольких языков является перспективной.

Для разработки редактора, который сможет привлечь к себе интерес пользователей, необходимо учесть недостатки уже разработанных систем и реализовать доступный и богатый интерфейс.

К наиболее известным редакторам WN относятся: VisDic (2004), GernEdit (2010), KUI (2008).

Анализ перечисленных систем, помогает выделить некоторые недостатки каждой из них.

* **VisDic**. Полезность данной системы ограничена тем, что обработка данных ведется при помощи xml парсинга, что является менее эффективным в сравнении с непосредственной обработкой текстовых файлов. Позднее на смену VisDic пришла DEBVisDic, которая, однако, тоже не является идеальной. Данная система полностью зависит от браузера Mozilla Firefox. Кроме того интерфейс DEBVisDic не является максимально интуитивным для пользователя;
* **GernEdiT**.Предназначен для работы только лишь с одним языком (немецким) и имеет лицензионные ограничения (запрет на распространение и коммерческое использование);
* **KUI**.Крайне ограниченный функционал. Система позволяет работать только с леммами.

Как видно из списка, к основным недостаткам относятся лицензионные ограничения, неинтуитивный интерфейс и бедность функционала. Но если лицензионные ограничения и функционал зависят от предпочтений конкретного разработчика, то выбор технологии для проектирования интерфейса может быть более или менее общим.

**Технологии проектирования интерфейса**

Поскольку одной из наиболее предпочтительных баз данных для разработки редактора WN является HypergraphDB, к которой удобно обращаться с помощью языка программирования Java, то и разработка интерфейса на Java будет весьма логичным решением. Однако крайне желательно придерживаться методологии разделения логической и графической части. Наиболее известными технологиями создания интерфейса на языке Java являются:

* Swing;
* JavaFX;
* Google Widget ToolKit (GWT).

Для начала стоит сравнить Swing и JavaFX. Две этих библиотеки очень похожи, как с точки зрения элементной базы, так и с точки зрения синтаксиса. Однако есть и весомые различия. JavaFX, в отличие от Swing, можно применять не только для разработки настольных приложений, но и для разработки под iOS и Android, а также для создания веб-сервисов. Кроме того, графические решения, реализованные при помощи Swing, зависимы от платформы, на которой запускается приложение. Это означает, что стандартные элементы (окна, кнопки и т.д.) на разных платформах будут выглядеть по-разному.

Кроме того, в JavaFX прослеживается четкое разделение бизнес-логики и интерфейса. Для написания логики используется Java 1.5 и выше, а для создания интерфейса применяется JavaFX Script, называемый FXML.

GWT также как и JavaFX отлично подходит для разработки RIA (англ. Rich Internet Application). При их сравнении, следует отметить, что GWT более приемлем для разработки веб-сервисов. Однако используемый в этой технологии компилятор весьма медленный, что приводит к большим задержкам при разработке. Компиляция небольшого проекта на компьютере средней мощности может занять несколько минут.

**Заключение**

При разработке редактора для лингвистической модели Wordnet необходимо крайне ясно представлять ее внутреннюю реализацию. Выбор базы данных, в которой будет храниться весь набор отношений должен предполагать возможность представления данных в виде графа. Для реализации интерфейса следует выбирать технологии разработки RIA.

Литература:

1. HyperGraphBD – база данных на основе графов [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/82171/> , свободный (25.08.2016).
2. HyperGraph [Электронный ресурс]. – Режим доступа:  [https://en.wikipedia.org/wiki/Hypergraph](https://ru.wikipedia.org/wiki/Гиперграф%20) , свободный (25.08.2016).
3. Wordnet [Электронный ресурс]. – Режим доступа: <http://wordnet.princeton.edu/> , свободный (27.08.2016).