Глава 2

1.1 Правила и требования к работе базы знаний

Прежде чем вести работу с базой знаний, нужно сформулировать ряд правил, которым нужно следовать при добавлении новых знаний, структурировании уже существующих знаний и последующей их обработке.

1) знания должны быть представлены в максимально простой и однородной модели данных с поддержкой всех возможных связей между объектами. С этой задачей хорошо справляется графовая модель данных;

2) база знаний должна предоставлять алгоритмы преобразования хранимых знаний (в виде графа) в другие модели данных, например, иерархию или агрегированные объекты;

3) семантика знаний должна быть абстрагирована от их морфологии — смысловая и символьная часть слова хранятся независимо друг от друга;

4) язык запросов к базе знаний не должен основываться на английском языке — формат языка запросов должен позволять совершать запросы на любом естественном языке, поддерживаемом базой знаний;

5) база знаний должна предоставлять знания в формате, пригодном для использования в проектах более высокого уровня — семантической поисковой сети, вопросно-ответной системы, системы перевода естественных языков, прототипа искусственного интеллекта и т. д.;

6) наполнение базы знаний должно происходить по принципу «конструктора»: наполнение начинается с самых простых и элементарных понятий, которые затем при помощи специальных ссылок используются для описания более сложных понятий, и далее по восходящей;

7)

Концепция информационной системы

Для хранения данных со множеством внутренних связей и ссылок лучше всего приспособлены графовые базы данных [1, с. 142]. В качестве базы данных для представления знаний выбрана мультимодельная (поддерживающая хранение данных в виде графов, документов и «ключ-значение») СУБД ArangoDB. Её преимуществом по сравнению с такими более популярными решениями, как Neo4j и OrientDB, является возможность записывать в качестве свойств документов числовые идентификаторы.

Главными особенностями рассматриваемой базы знаний являются использование «универсальных понятий» и раздельное хранение морфологии и семантики.

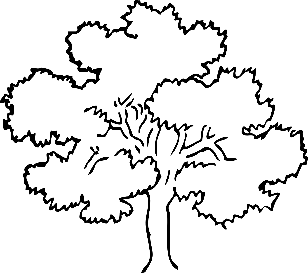
Одна из ключевых особенностей концепции – выделение своеобразных «универсальных понятий», которые одинаковы в представлении любого человека независимо от его родного языка (рис. 1).

рус. «дерево»

англ. «tree»

нем. «Baum»

франц. «arbre»



id: 43

Рисунок 1 - Пример соответствия слов естественных языков с универсальным понятием

Универсальным понятиям присваивается уникальный числовой идентификатор (например, 432683), и они отражают семантическую часть слова. Многие значения слов естественных языков абстрактны и расплывчаты, что позволяет соединить эквивалентные значения с помощью численных идентификаторов. Однако для слов с более специфическим значением семантика определяется с помощью специального выражения, составленного из ссылок на другие универсальные понятия.

Использование численных идентификаторов вместо строковых обусловлено тем, что строки могут иметь несколько значений, что может породить нежелательные смысловые коллизии и требует дополнительного уточнения.

Любое слово находит своё выражение на нескольких языковых уровнях: фонетико-фонологическом, морфемно-морфологическом, синтаксическом, семантическом и прагматическом [2, с. 78]. Фонетико-фонологический, морфемно-морфологический и семантический уровни слова (проще говоря, произношение, состав слова и его морфологические свойства) могут быть объединены в одном документе, описывающем слово, синтаксический и прагматический уровни принадлежат предложению и тексту соответственно.

Морфология и семантика разделены по разным базам данных. Каждый из естественных языков (русский, английский, немецкий и т. д.) хранится в отдельной коллекции документов и никаким образом не пересекается с другими языками. Необходимые для морфологического анализа части слов (морфемы — префиксы, корни, суффиксы, окончания) также хранятся в виде отдельных коллекций документов и связаны ссылками, которые описывают морфемный состав слов. Разделение морфологии и семантики необходимо, чтобы выделить одну базу данных, посвящённую семантике, к которой потом возможно подключать новые базы данных естественных языков по мере их наполнения.

Упрощённая схема структуры базы знаний представлена на рисунке 2.

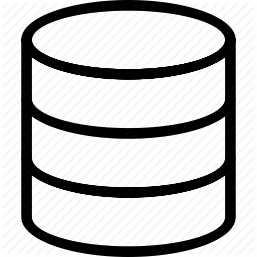
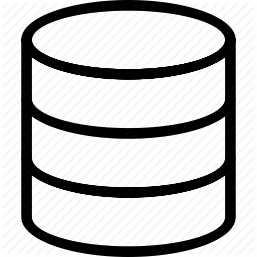
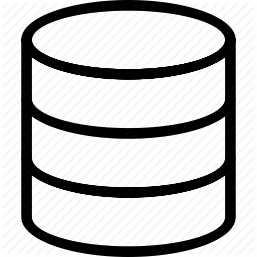
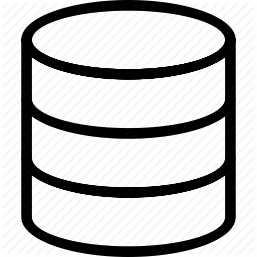
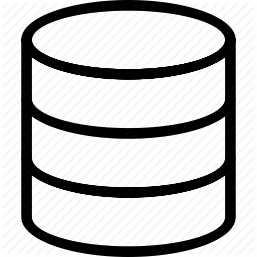


Рисунок 2 – Связи внутри базы знаний

И содержимое баз данных естественных языков, и базы данных универсальных понятий хранится в формате JSON, часто используемом в современной веб-разработке для построения API [3, с.1]. Особенностью хранения объектов является использование идентификаторов универсальных понятий вместо традиционных строковых идентификаторов в качестве свойств и значений: это нужно для максимальной связности данных между собой. В результате получается граф со множеством вершин и рёбер, любая точка которого так или иначе связана со всеми остальными.

Рассмотрим различия между форматом хранения слов на естественном языке и универсальных понятий (рисунок 3).

Рисунок 3 – Связи внутри базы знаний

Для краткости содержимое объектов скрыто. Содержимое объекта, представляющего слово естественного языка, состоит из трёх объектов, каждый из которых отображает определённый уровень языка (в фонетике – транскрипция слова, в морфологии – морфемный состав слова и его морфологические свойства, в семантике – массив значений слова со ссылками на универсальные понятия).

Содержимое объекта, представляющего универсальное понятие, состоит из двух частей: первая является семантическим выражением, по структуре напоминающем дефиниции из толкового словаря, вторая часть представлена программным кодом на языке программирования JavaScript, который может выполнен, когда универсальное понятие является каким-либо сложным действием, включающим в себя набор из более простых действий (например, подсчёт математической формулы или анализ стиля текста).

Результат семантического анализа текста упрощённо изображен на рисунке 4.

Рисунок 4 – Структура данных при семантическом анализе текста

Каждому предложению и каждому слову выдаётся идентификатор соответственно их порядку в тексте. Объект самого верхнего уровня обозначает текст и его свойства (объект «78» в примере), внутри него содержится объект со всеми предложениями, внутри которых расположены объекты слов и свойства предложения, не отображённые в примере. После проведения семантического анализа объекты слов дополняются синтаксическим и прагматическим уровнем.

Подобный семантический анализ позволяет выделить и проанализировать смысловые связи внутри текста и впоследствии производить запросы к его содержимому, аналогичные вопросам к экспертной системе. Кроме того, выявленные семантические связи открывают возможность осуществлять кластеризацию документов по смыслу предложений или текстов в целом.

1.2 Выбор СУБД

Реляционный подход к СУБД зародился в конце 1960-х годов. К концу 1980-х годов реляционные СУБД стали наиболее популярным решением и сохраняют это положение на данный момент [Кузнецов Основы БД].

Несколько десятилетий разработчики программного обеспечения пытались приспособить связанные, полуструктурированные наборы данных к хранению в реляционных СУБД. Но хотя реляционные СУБД были изначально спроектированы для систематизации бланков и табличных структур, они плохо приспособлены для хранения ситуативных, исключительных связей, которые неожиданно возникают на практике [Robinson *Graph Databases*].

Связи между данными являются неотъемлемой частью реляционных СУБД, однако только на уровне моделирования, как средство объединения таблиц. Зачастую необходимо снять неоднозначность семантики связей, связывающих сущности, как и определить их вес или силу.

Реляционные СУБД не располагают подобным функционалом. Помимо этого, по мере накопления резко разнящихся значений и всенаправленного усложнения и размытия набора данных, реляционная модель перегружается соединёнными операцией JOIN таблицами, частично заполненными записями и множеством условий на отсутствие значений.

Рост взаимосвязанности в реляционной СУБД превращается в множество JOIN-операций, которые отрицательно сказываются на производительности и усложняют дальнейшую адаптацию существующего набора данных к дальнейшим возможным изменениям в бизнес-логике [Robinson *Graph Databases*].

…

СУБД только с коммерческой лицензией не соответствуют поставленным требованиям к создаваемой системе, поэтому СУБД Oracle в качестве СУБД для хранения знаний не рассматривается.

…