Тема: «Операции над онтологиями семантических поисковых систем»

Выполнила:

Моравецкая В.В.

Группа ДА-52м

Научный руководитель:

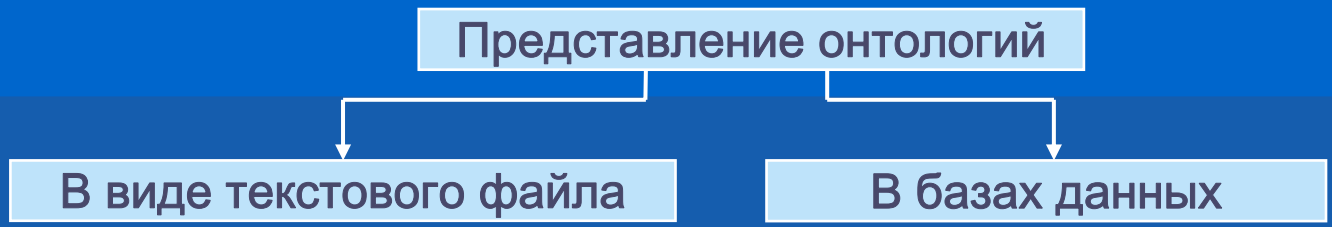
Гемба О.В.

Цель работы: анализ существующих программных средств для выполнения операций над онтологиями с целью выявления способов их улучшения.

В работе ставились следующие задачи:

- Проанализировать способы хранения онтологий.
- Изучить существующие программные средства хранения и выполнения запросов над онтологиями.
- Проанализировать операции над онтологиями при семантическом поиске, выявить среди них наиболее важные и часто используемые.
- На основании анализа алгоритмов отображения предложить подходы к улучшению существующих средств отображения онтологий.

Способы представления онтологий



База данных	Преимущества	Недостатки
Объектная	Объектно-ориентированный подход похож на онтологический.	Не существует конкретной реализации родной объектной БД для модели RDF.
Реляционная	Обеспечивают производительность, надежность, устойчивость и доступность.	Сложность представления всех компонентов онтологии, отсутствие наследования.
Объектно-реляционная	Накопленный опыт работы с реляционными БД, наличие объектных расширений, эффективные SQL3 запросы.	Структура онтологии не полностью соответствует схеме базы данных.

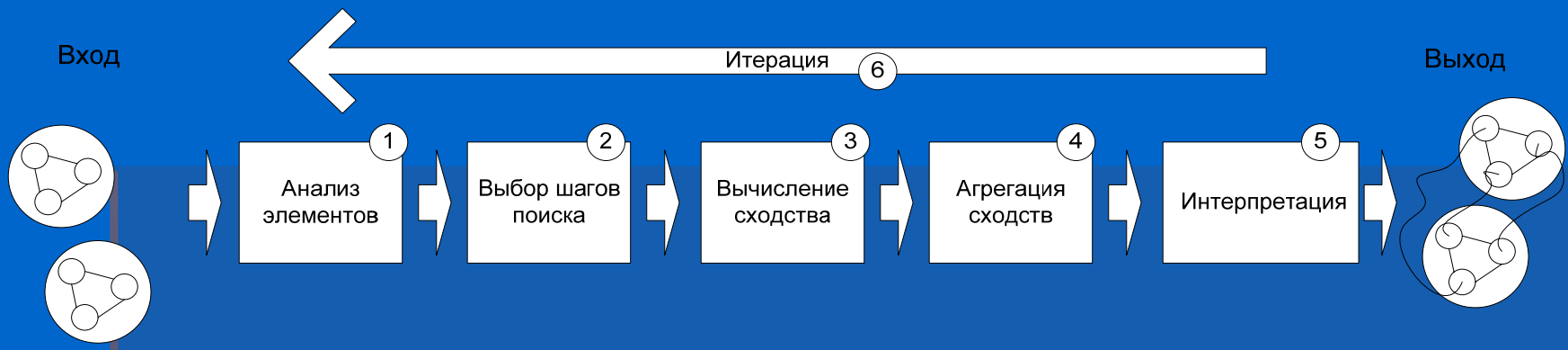
Характеристики средств хранения онтологий и выполнения запросов к ним

Утилита	Язык реализации	Формат экспорта	База данных хранилища
ICS-RDFSuite	Java/C++	RDF	ORDBMS (SQL3 совместимый; например PostgreSQL)
Sesame	Java	RDF	ORDBMS (PostgreSQL)
Inkling	Java	триплеты в ASCII	в памяти/персистентная (поддержка JDBC; например SQL, PostgreSQL)
rdfDB	C	триплеты в ASCII	персистентная (SleepyCat)
RDFStore	C, Perl	N-триплеты, RDF	в памяти/персистентная (напр. file, BerkeleyDB, SDBM)
EOR	Java	триплеты с XSL	персистентная (SQL БД; например MySQL)
Redland	C	триплеты	в памяти/персистентная (SleepyCat, BerkeleyDB)
Jena	Java	триплеты в ASCII	в памяти/персистентная (например BerkeleyDB, Interbase, PostgreSQL)
RDF Gateway	?	триплеты в ASCII	RDBMS
TRIPLE	Java	Lisp, XML для DOT, AML, ASCII	в памяти
KAON	Java, Python	?	в памяти/персистентная (KAON сервер, file, RDBM)
Cerebra	Java	?	распределенные данные (CORBA)
Empolis K42	Java	Topic Maps (XTM)	персистентная (K42 Generic Store, DBMS)
Ontopia KS	Java	XTM, XML версия ISO 13250	в памяти/персистентная (RDBMS, OODB)

Операции над онтологиями

- Слияние (merging)
- Отображение (mapping)
- Выравнивание (alignment)
- Уточнение (refinement)
- Унификация (unification)
- Интеграция (integration)

Канонический процесс операции отображения



1. **Анализ элементов** преобразовывает начальное представление онтологий в подходящий формат.
2. **Выбор следующих шагов поиска**. Может отбрасывать неподходящие пары для сравнения.
3. **Вычисление сходства** определяет сходство каждой пары понятий.
4. **Агрегация сходств**. Если пара имела несколько значений сходства, они объединяются в одно итоговое значение.
5. **Интерпретация**. Формируются отображения между сущностями на основании сходств.
6. **Итерация**. Повторение некоторых шагов алгоритма.

Сравнительный анализ основных алгоритмов отображения

Алгоритм	Характеристика	Участие пользователя	Временная сложность
NOM	Полное сравнение, много функций сходства	Автоматический	$O(n^2 \cdot log^2(n))$
PROMPT	Полное сравнение, только сходство меток	Полуавтоматический	$O(n \cdot log(n))$
Anchor-PROMPT	Полное сравнение, структурное сходство	Полуавтоматический	$O(n^2 \cdot log^2(n))$
GLUE	Полное сравнение, сходство на основе изученных правил и ослабления меток	Машинное обучение	$O(n^2)$
QOM	Эвристика в выборе пар для сравнения, много функций сходства	Автоматический	$O(n \cdot log(n))$

Критерии оценки результатов

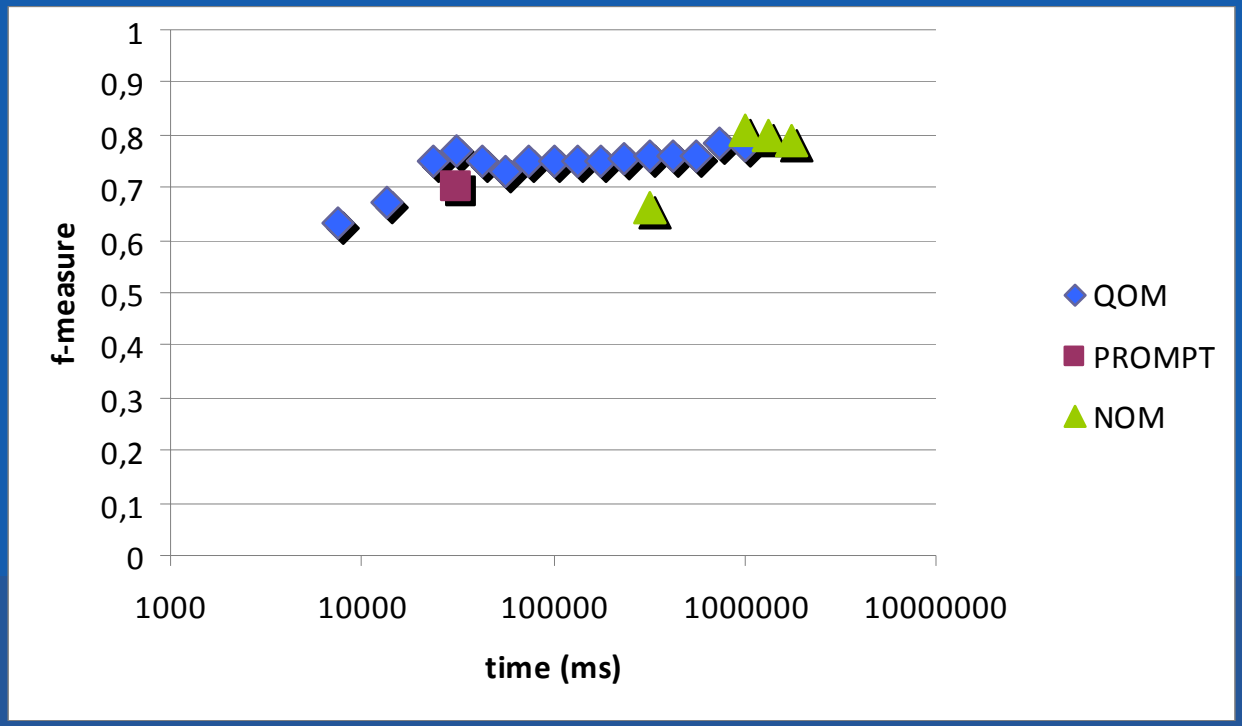
Для сравнения алгоритмов были проведены практические эксперименты. В качестве оценки были использованы стандартные метрики информационного поиска:
Точность (p) – отношение числа найденных корректных отображений к общему количеству полученных отображений.
Эффективность (r) – отношение числа найденных корректных отображений к общему количеству существующих отображений.

F-мера: $f = (b^2+1)pr / (b^2p + r)$, $b = 1$.

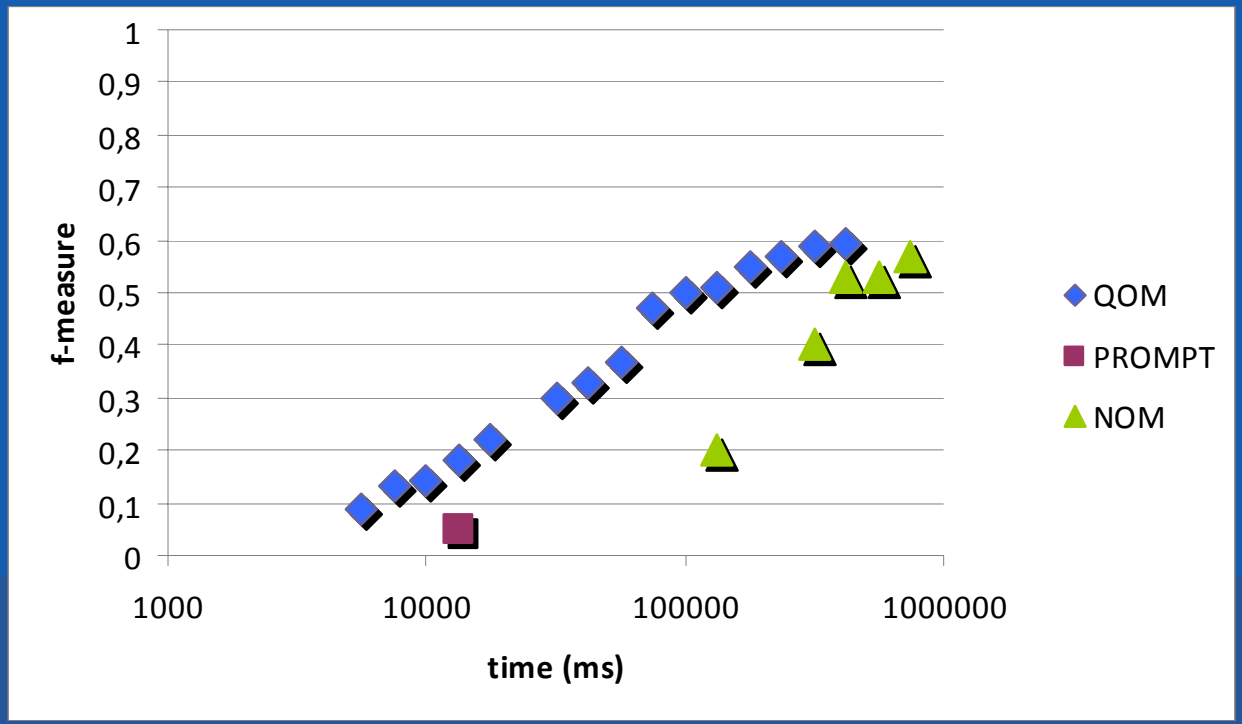
Сценарий тестирования

1. Загрузка набора тестовых онтологий.
2. Выполнение операции отображения.
3. Сравнение полученного результата с эталонным.
4. Вычисление метрик результата.

Результаты тестирования 1



Результаты тестирования 2



Полученные результаты

- - Оптимизация с целью повышения эффективности уменьшает общее качество отображения.
- - Поиск сходства по меткам сам по себе уже дает удовлетворительный результат.
- - Комбинирование многих элементов для определения сходства приводит к значительному повышению качества отображения.
- - Подход QOM показывает очень хорошие результаты за счет использования эвристик.

Выводы

В настоящее время наиболее оптимальным способом представления онтологий являются реляционные базы данных благодаря своей доступности, надежности, производительности и устойчивасти.

Было показано, что операция отображения наиболее часто используется при семантическом поиске и является базовой для других более сложных операций над онтологиями.

Были протестированы наиболее известные алгоритмы отображения онтологий в результате чего был написан плагин QOM для Protégé, который является более эффективной альтернативой существующему плагину PROMPT.

Спасибо за внимание!