

Разработка семантических электронных библиотек на основе онтологических моделей

© Ле Хоай,

© А.Ф Тузовский

Национальный исследовательский Томский политехнический университет

Оптимизация систем управления, Институт кибернетики.

lehotomsk@yahoo.com, tuzovskyaf@tpu.ru

Аннотация. Рассматриваются основные функции электронных библиотек на основе использования семантических технологий, делается постановка задач их реализации. Предлагаются методы их решения и демонстрируются примеры реализации функций семантической электронной библиотеки в разработанной системе.

1 Введение

Под электронными библиотеками (ЭБ) понимаются информационные системы, позволяющие автоматизировать работу пользователей с различными электронными ресурсами: документами, изображениями, мультимедиа-файлами. Реализация подобных систем сталкивается с рядом ключевых проблем [1]:

- интеграция разнородной информации (электронных ресурсов, пользовательских профилей, структур знаний предметных областей), представленной разными способами на основе использования различных метаданных;

- обеспечение надёжности результатов поиска и навигации по ресурсам;

- обеспечение точности категоризации электронных ресурсов.

Основной причиной возникновения этих проблем является описание электронных ресурсов (ЭР) в системе в виде набора терминов естественных языков (ключевых слов) и формирование логических выводов об их смысловом содержании без учёта синонимии, полисемии и омонимии. Это влечёт за собой снижение качества функций, предоставляемых ЭБ.

Для решения перечисленных проблем и повышения эффективности работы с ЭР требуется выполнять работу с их семантикой, для чего используются модели описания семантики (таксономии, тезаурусы, онтологии) и технологии Semantic Web (семантические технологии). Использование таких технологий позволяет реализовать работу с ЭР на новом уровне, с учётом содержащегося в них смысла. Электронные библиотеки, разработанные с использованием подобных технологий, обычно называются семантическими ЭБ (СЭБ).

В данной статье предлагается способ реализации

основных функций СЭБ, таких как: семантический поиск, формирование рекомендаций, выполнение навигации по ресурсам и автоматическая категоризация.

2 Онтологический подход к описанию ЭР

Основной идеей данного подхода является использование онтологий предметных областей для аннотирования содержания электронных ресурсов [2]–[4]. В СЭБ описание электронных ресурсов, содержащих знания из разных предметных областей, выполняется с использованием специально разработанных онтологий этих областей, описанных с помощью таких языков, как RDFS или OWL [5],[6].

Под описанием (аннотированием) ресурса понимается его семантическое метаописание, в виде набора простых высказываний (триплетов) на основе онтологической модели, в состав которых могут входить контекстные и контентные семантические метаданные.

Сделаем вначале несколько базовых определений.

Определение 1. Под онтологической моделью (онтологией) O понимается знаковая система $\langle C, P, I, L, T \rangle$, где C – множество элементов, которые называются понятиями; P – множество элементов, называемые свойствами (двуместными предикатами); I – множество экземпляров понятий; L – множество текстовых меток или значений понятий и свойств; T – частичный порядок на множестве C и P .

С помощью набора предикатов P онтологии могут описываться различные отношения между понятиями и экземплярами. Эти отношения задаются с использованием простых утверждений (триплетов) $\langle s, p, o \rangle$, где s и o – это субъект и объект высказывания, а $p \in P$ – это предикат онтологии O .

Считаем, что любому свойству $p \in P$ может быть задан весовой коэффициент (семантический вес) $pv \in [0, 1]$, задающий смысловую близость между субъектом и объектом утверждения (1 – субъект и объект считаются сходными по семантике, а 0 – не похожими), составленного с использованием данного свойства.

Определение 2. Контекстными метаданными ресурса s (заданного с помощью идентификатора

URI) называется набор простых утверждений (триплетов) $M_k = \{t_i = \langle s, p_i, o_i \rangle \mid i = 1, n\}$, где $s \in I$ – это аннотируемый ресурс (субъект), $o \in C \cup I \cup L$ – объект утверждения, $p \in P$ – отношение между субъектом и объектом.

Таким образом, под набором M_k ресурса s в СЭБ понимаются утверждения о его связи с другими объектами, понятиями из общих онтологий библиотеки, таких как онтологии пользователей, онтологии ресурсов и онтологии предметных областей.

Определение 3. *Контентные метаданные ресурса – это набор простых утверждений (кортежей) $M_c = \{t_j = \langle s_j, p_j, o_j, v_j \rangle \mid j = 1, m\}$, где $s \in C \cup I$ – это субъект утверждения, $o \in C \cup I$ – объект утверждения, $p \in P$ – отношение между субъектом и объектом, а v – весовой коэффициент, который оценивает значимость данного утверждения.*

Под набором M_c ресурса s понимаются утверждения о знаниях, содержащихся в самом аннотируемом ресурсе.

В СЭБ набор $M(s) = M_k(s) \cup M_c(s)$ будет называться семантическими метаданными ресурса s . Таким образом, все компоненты библиотеки (электронные ресурсы, пользователи, категории и т.п.) описываются метаданными, представленными с помощью RDF-триплетов на основе использования элементов некоторых онтологий. Множество триплетов, описывающих онтологию O и их экземпляры s , формирует базу знаний (БЗ, Knowledge Base) СЭБ. К этой базе знаний могут выполняться запросы, описанные на некотором языке (например, SPARQL или SERQL [8], [9]), и на основе их обработки могут решаться различные задачи, позволяющие реализовать услуги, предоставляемые СЭБ.

3 Постановка задач и их решение

Доступные функции СЭБ (семантический поиск, категоризация и формирование рекомендаций) реализуются с учетом семантики описаний ЭР. В связи с этим необходима некоторая оценка (метрика) семантической близости различных объектов.

3.1 Семантическая близость

Семантическая близость (смысловое сходство) может определяться между разными компонентами триплетов, триплетами и наборами триплетов. Существуют различные методы вычисления таких метрик [4], которые имеют свои сильные и слабые стороны. В рамках данной работы они не анализируются. В ходе выполнения исследований по созданию СЭБ авторами были разработаны новые методы оценки семантической близости, учитывающие специфику разрабатываемой системы.

3.1.1 Метод вычисления семантической близости между компонентами триплетов

Пусть $\text{Sim}(\alpha, \beta)$ – семантическая близость между элементами α и β , где $\alpha, \beta \in C \cup I \cup P \cup T$. Для вычисления $\text{Sim}(\alpha, \beta)$ необходимо построить неориентированный граф GO из всех имеющих триплетов в БЗ. Граф GO создаётся в соответствии со следующими правилами:

- используются только триплеты, у которых значения весовых коэффициентов предикатов не равны нулю ($pv \neq 0$);

- вершинами графа являются субъекты и объекты триплетов, а ребра графа, соединяющие субъекты с объектами имеют веса, равные значению pv предиката того триплета, с использованием которого они были сформированы;

- инверсное отношение (на основе предиката `owl:inverseOf`) между предикатами $p1$ ($pv1$) и $p2$ ($pv2$) добавляет в граф два ребра с весами $pv1$ и $pv2$, например: для триплета $\langle s, p1, o \rangle$ добавляются следующие два ребра: $\langle s, pv1, o \rangle, \langle o, pv2, s \rangle$;

- симметричное отношение добавляет в граф два ребра с равными весами, например: `<owl:sameAs>` добавляет два ребра со значениями $pv = 1.0$.

Под путём $PATH(\alpha, \beta)$ между двумя вершинами α и β графа GO понимается набор рёбер (предикатов) ведущих от вершины α до вершины β , с учётом их направленности. В этом случае значение $\text{Sim}(\alpha, \beta)$ между этими вершинами вычисляется следующим образом:

$$\text{Sim}(\alpha, \beta) = \max_{i=1 \rightarrow k} (\text{Sim}_{PATH_i}(\alpha, \beta)), \quad (1)$$

где k – число возможных путей графа GO от вершины α до вершины β . Значение семантической близости между элементами α и β по направлению пути i $\text{Sim}_{PATH_i}(\alpha, \beta)$ определяется по следующей формуле:

$$\text{Sim}_{PATH_i}(\alpha, \beta) = \prod_{j=1}^{h_i} pv_{i,j}, \quad (2)$$

где h_i – число семантических отношений между элементами α и β на пути i , $pv_{i,j}$ – значение веса ребра на основе j -ого семантического предиката на пути i . На основе формул 1 и 2 можно получить окончательную формулу для определения семантической близости между вершинами α и β :

$$\text{Sim}(\alpha, \beta) = \max_{i=1 \rightarrow k} (\text{Sim}_{PATH_i}(\alpha, \beta)) = \max_{i=1 \rightarrow k} \left(\prod_{j=1}^{h_i} pv_{i,j} \right) \quad (3)$$

Величина $\text{Sim}(\alpha, \beta)$ удовлетворяет следующим свойствам: $\text{Sim}(\alpha, \beta) \in [0,1]$; $\text{Sim}(\alpha, \beta) = 0$ при отсутствии пути от α к β ; $\text{Sim}(\alpha, \alpha) = \text{Sim}(\beta, \beta) = 1$. В исключительном случае $\text{Sim}(\alpha, \beta)$ может равняться 1, при условии существования инверсного отношения между элементами α, β .

3.1.2 Метод вычисления семантической близости между триплетами

Пусть $\text{Sim}(t_1, t_2)$ – семантическая близость между триплетом t_1 и t_2 . Близость между триплетом вычисляется на основе близостей между их компонентами. В данной работе учитываются свойства инверсного отношения между предикатами (Если два предиката p_1 и p_2 имеют отношение $\langle p_1, owl:inverseOf, p_2 \rangle$, то при наличии триплета $\langle s, p_1, o \rangle$ подразумевается триплет $\langle o, p_2, s \rangle$) [7].

Имеются следующие две ситуации:

- если $t_1, t_2 \in Mk$: то $\text{Sim}(t_1, t_2)$ вычисляется по следующей формуле:

$$\text{Sim}(t_1, t_2) = \text{Sim}(p_1, p_2) \times \text{Sim}(o_1, o_2) \quad (4)$$

- если $t_1, t_2 \in Mc$: то $\text{Sim}(t_1, t_2)$ определяется следующим образом:

$$\text{Sim}(t_1, t_2) = \begin{cases} |k| \frac{\text{Sim}(s_1, s_2) + \text{Sim}(o_1, o_2)}{2} \omega(t_1, t_2), \forall k > 0 \\ |k| \frac{\text{Sim}(s_1, o_2) + \text{Sim}(o_1, s_2)}{2} \omega(t_1, t_2), \forall k \leq 0 \end{cases} \quad (5)$$

где $\omega(t_1, t_2) = v_1 \times v_2$ – функция весовых коэффициентов значимости двух триплетов; $k = \text{Sim}(p_1, p_2)$ и $\text{Sim}(t_1, t_2) \in [0, 1]$; $\text{Sim}(t_1, t_2) = 0$ при $k = 0$. $\text{Sim}(p_1, p_2)$ также вычисляется по формуле 3.

3.1.3 Метод вычисления семантической близости между наборами триплетов

Семантическая близость между наборами триплетов может быть вычислена с использованием двух методов: на основе метода предложенного в работе [10] и метода косинусной меры в модели векторного пространства [11]. Для вычисления семантической близости между наборами триплетов $T_1 = \{t_i = \langle s_i, p_i, o_i \rangle \mid i = 1, k\}$ и $T_2 = \{t_j = \langle s_j, p_j, o_j \rangle \mid j = 1, h\}$ используются следующие формулы:

$$\text{Sim}(T_1, T_2) = \frac{\sum_{t_i \in T_1} \max(\text{Sim}(t_i, T_2))}{|T_1|} \quad (6)$$

$$= \frac{\sum_{i=1}^k \max_{j=1 \rightarrow h}(\text{Sim}(t_i, t_j))}{k} \text{ и}$$

$$\text{Sim}(T_1, T_2) = \frac{T_1 \times T_2}{\sqrt{T_1^2} \times \sqrt{T_2^2}} = \frac{\sum_{i=1}^k \sum_{j=1}^h \text{Sim}(t_i, t_j)}{\sqrt{\sum_{i=1}^k \sum_{j=1}^k \text{Sim}(t_i, t_j) \times \sum_{i=1}^h \sum_{j=1}^h \text{Sim}(t_i, t_j)}}, \quad (7)$$

где $\text{Sim}(T_1, T_2) \in [0, 1]$ и $\text{Sim}(t_i, t_j)$ – семантическая близость между триплетом, вычисляемая по формуле 4 (если $\forall t_i, t_j \in Mk$) или формуле 5 (если $\forall t_i, t_j \in Mc$).

3.2 Семантический поиск

Как отмечалось ранее, метаописание ЭР на основе онтологической модели O рассматривается с точки зрения его контекста и контента. На основе

контекстных метаданных и контентных метаданных, запрос семантического поиска q представляется в виде набора $M(q) = (Mk(q), Mc(q))$, а объект кандидата d (возможный результат на данный запрос) – $M(d) = (Mk(d), Mc(d))$. Тогда, задачу семантического поиска можно переформулировать следующим образом: Имеется запрос q с его набором $M(q)$, тогда результатом выполнения данного запроса q будет конечный набор из k объектов знаний (ЭР) $D = \{d_i \mid i = 1, k\}$, где каждый d с набором $M(d)$ удовлетворяет следующему условию:

$$\text{Sim}(M(q), M(d)) = \alpha \times \text{Sim}(Mk(q), Mk(d)) + \beta \times \text{Sim}(Mc(q), Mc(d)) > \varepsilon, \quad (8)$$

где $\varepsilon \in (0, 1)$ – пороговое значение, α и β – коэффициенты близости по контексту и по контенту, соответственно и $\alpha + \beta = 1.0$. Значения α и β настраиваются так, чтобы значение семантической близости корректно определяло важность контента и контекста искомого объекта. Если более важным является содержание искомого объекта (контента), тогда значение β превышает значение α и наоборот.

Кроме семантического поиска, в СЭБ также доступен и поиск по графу, который получает широкое применение в социальных сетях (например: *Facebook*) [13].

3.3 Формирование рекомендаций

В данной работе под формированием рекомендаций понимается предоставление пользователям набора ресурсов, релевантных рассматриваемому ЭР. Пусть в СЭБ имеется ЭР d с метаданными $M(d) = (Mk(d), Mc(d))$. Задача формирования рекомендаций для документа d может рассматриваться как выполнение семантического поиска с использованием запроса $q = d$ с наборами $Mk(d)$ и $Mc(d)$. Набор кандидатов $R = \{dr_i \mid dr_i \neq d \wedge i = 1, h\}$ на запрос q считаются рекомендуемыми для документа d если для любого $dr \in R$ выполняется условие:

$$\alpha \times \text{Sim}(Mk(d), Mk(dr)) + \beta \times \text{Sim}(Mc(d), Mc(dr)) > \varepsilon \quad (12)$$

Выборы значений α , β и порогового значения ε делаются вручную или автоматически на основе количества триплетов метаописаний ресурса d .

3.4 Автоматическая категоризация

В СЭБ пользователь может создавать свои категории (рубрики), а система автоматически будет относить релевантные ресурсы к заданным категориям. Одна категория может включать в себя другие категории, и они структурируются в виде какой-либо иерархии (например: категория – подкатегория).

Категория k может описываться конечным набором шаблонных ресурсов $Dk = \{tr_i \mid i = 1, h\}$ и каждый tr_i имеет своё метаописание $M(tr_i)$. Любой ресурс $dr \notin Dk$ считается релевантным к заданной категории k при выполнении следующего условия:



Рис. 1. Программная архитектура системы SemDL

$$\alpha \times \text{Sim}(M_k(Dk), M_k(dr)) + \beta \times \text{Sim}(M_c(Dk), M_c(dr)) > \varepsilon, \quad (13)$$

где $M_k(Dk) = M_k(tr1) \cup \dots \cup M_k(trh)$, $M_c(Dk) = M_c(tr1) \cup \dots \cup M_c(trh)$. В качестве шаблонных ресурсов могут служить существующие ЭР или наборы триплетов (контекст и контент), созданные вручную для описания созданной категории. Кроме метода категоризации по набору шаблонных ресурсов в СЭБ выполняется категоризация по элементам используемых в СЭБ онтологий, описывающих предметные области.

3.5 Метод просмотра

В СЭБ все ЭР (экземпляры), понятия (элементы) онтологии (кроме лексических и числовых данных) имеют уникальные идентификаторы (*URIs*). Такими идентификаторами могут быть субъекты, объекты и предикаты триплетов в БЗ. В связи с этим имеется возможность просмотра содержания СЭБ по субъектам или объектам описаний ЭР. Реализация функции просмотра позволяет пользователям находить и просматривать все допустимые отношения (триплеты), связанные с рассматриваемым ресурсом. Например: по ссылке автора ресурса, можно найти все его публикации, его интересы и т.п., а далее по этим публикациям и интересам могут быть выполнены следующие переходы.

4 Программная реализация в SemDL

Предложенные методы решения задач по реализации описываемых функций СЭБ применены в разработанной системе *SemDL* [10].

4.1 Программная архитектура SemDL

Программная архитектура системы *SemDL* показана на рис. 1. Она логически разделена на четыре уровня. В соответствии с этим разделением, пользователи взаимодействуют с системой с помощью веб-интерфейса и под контролем системы могут получить доступ к интересующим их функциям системы. Системные компоненты (пакеты) группируются по четырём категориям на основе их функциональности:

- **ПРЕДСТАВЛЕНИЯ (VIEWS):** включают в себя различные разработанные теги (например, HTML-теги) и сервлеты (*servlets*), которые непосредственно обрабатывают полученные запросы и возвращают ответы системы.

- **АННОТАЦИЯ (ANNOTATION):** включает функции для работы с текстовыми документами (извлечение, индексирование). В основном, данный пакет выполняет поиска кандидатов для аннотирования ресурсов (создание метаописания ЭР). Индексирование и поиск ключевых слов возлагается на библиотеку с открытым исходным

кодом *LUCENE ENGINE* [14], [15].

• **ГРАФ ОЦЕНОК (WG)**: данный пакет предназначен для создания и индексирования графа семантических оценок *GO*, и вычисляет семантическую близость между компонентами триплетов, наборами триплетов по контенту и контексту. Элементарные операции с графом осуществляется с помощью библиотеки с открытым исходным кодом *JGRAPH* [16]-[18].

• **СЕРВИСЫ SEMDL**: данный пакет осуществляет доступ к базе знаний (*SESAME*) и выполняют различные операции с хранимыми данными, а также обеспечивают фильтрацию ресурсов по наборам триплетов контента и контекста. Для работы с *SESAME* используется библиотека с открытым исходным кодом *OPENRDF-SESAME-API* [19], [20].

4.2 Реализация

Примеры интерфейса системы SemDL показаны на рис. 2–3. Составление запроса с использованием набора триплетов по контексту или контенту для семантического поиска выполняется с помощью рекомендации субъектов, предикатов и объектов на основе вводимых терминов (слов). Результаты поиска, рекомендации или категоризации ранжируются в порядке уменьшения оценок семантической близости. Ресурсы в результатах кратко показываются с основными свойствами (авторы, ключевые слова), по которым можно выполнять переходы к другим ресурсам. При работе с категориями и рекомендациями для получения желаемых результатов можно настраивать значения параметров α , β и ε . Значения параметров α и β по умолчанию определяются на основе количества триплетов для описания контекста и контента ресурса.

4.3 Обоснование полученного решения

Процесс семантического поиска. Схема процесса выполнения семантического поиска показана на рис. 4. Пользователь выполняет семантический поиск с помощью составленных триплетов контекста и контента $M(q)$. На основе $M(q)$ выполняется фильтрация возможных кандидатов (ЭР) d , которые могут быть ответом на запрос q . В результате чего получают наборы: S_{PRK} – набор возможных кандидатов по контексту и S_{PRC} – набор по контенту. В дальнейшем между метаописаниями $M(q)$ и $M(d)$ вычисляются близости по формуле 8. Ресурсы d , которые удовлетворяют заданному запросу q , ранжируются по степени убывания значений их близости обрабатываемому запросу.

Фильтрация возможных кандидатов. Для повышения эффективности работы данного алгоритма имеется возможность выполнять фильтрацию возможных кандидатов с использованием описанных ниже методов.

Метод фильтрации по набору триплетов контекста. Пусть задан набор триплетов контекста некоторого объекта s : $T_K = \{t_i = \langle s, p_i, o_i \rangle \mid i = 1, k\}$. Для любого элемента e триплета имеется непустой список связанных элементов по семантике Exs : $Exs(e) = \{e, e_i \mid i = 0, h \wedge \text{Sim}(e, e_i) > \varepsilon\}$, где ε – пороговое значение близости и $\text{Sim}(e, e_i)$ вычисляется по формуле 3.

Пусть $Exs_P(T_K)$ – список всех предикатов из набора T_K и их связанных элементов по семантике, $Exs_P(T_K) = Exs(p_1) \cup \dots \cup Exs(p_k)$; $Exs_O(T_K) = Exs(o_1) \cup \dots \cup Exs(o_k)$ – список всех объектов из набора T_K и их связанных элементов по семантике.

На основе списков связанных элементов по семантике предложенный метод фильтрации по набору T_K допускает только тот ресурс $prk \in S_{PRK}$ с набором триплетов его контекста $T_{PRK} = \{t_j = \langle prk, p_j, o_j \rangle \mid j \in [1, h]\}$, который удовлетворяет следующему условию:

$$(\exists t_j \in T_{PRK}) \wedge (p_j \in Exs_P(T_K)) \wedge (o_j \in Exs_O(T_K)) \quad (14)$$

Метод фильтрации по набору триплетов контента. Пусть заданный набор триплетов контекста некоторого объекта s : $T_C = \{t_i = \langle s_i, p_i, o_i \rangle \mid i = 1, k\}$. Для набора T_C имеются следующие списки связанных элементов по семантике для компонентов всех триплетов: $Exs_S(T_C) = Exs(s_1) \cup \dots \cup Exs(s_k)$; $Exs_P(T_C) = Exs(p_1) \cup \dots \cup Exs(p_k)$; $Exs_O(T_C) = Exs(o_1) \cup \dots \cup Exs(o_k)$.

На основе списков связанных элементов по семантике предложенный метод фильтрации по набору T_C допускает только тот ресурс $prc \in S_{PRC}$ с набором триплетов его контента $T_{PRC} = \{t_j = \langle s_j, p_j, o_j \rangle \mid j = 1, h\}$, который удовлетворяет следующему условию:

$$(\exists t_j \in T_{PRC}) \wedge (s_j \in Exs_S(T_C)) \wedge (p_j \in Exs_P(T_C)) \wedge (o_j \in Exs_O(T_C)) \quad (15)$$

Условия фильтрации (14 и 15) могут описываться на языках запросов *SPARQL* или *SERQL*, которые эффективно обрабатываются сервером БЗ.

Пример шаблона запроса на языке *SERQL* для фильтрации возможных кандидатов некоторого объекта S по заданным наборам триплетов показан ниже.

```
/*Запрос на фильтрацию возможных кандидатов по набору T_K*/
```

```
SELECT S FROM
```

```
{S} P {O}
```

```
WHERE P IN Exs_P(T_K) AND O IN Exs_O(T_K)
```

```
USING NAMESPACE
```

```
...
```

```
/*Запрос на фильтрацию возможных кандидатов по набору T_C*/
```

```
/* CONTEXT S – триплеты контента объекта S*/
```

```
SELECT S FROM CONTEXT S
```

```
{Sc} Pc {Oc}
```

```
WHERE Sc IN Exs_S(T_C) AND Pc IN Exs_P(T_C) AND Oc IN Exs_O(T_C) USING NAMESPACE ...
```

Семантический поиск

Контентные триплеты для поиска

[Ле Хоай, Знает, Тузовский А. Ф.]

ПОИСК Пусто

Создание триплетов для поиска

Удалить Semantic searches Ключевое слово Тузовс

Контекст content search context search

Объект

Добавить

Тузовский А. Ф.
type: Author add

другие результаты [1/1]

Контекстные триплеты для поиска

Выбор типа искомого объекта Другие

type: Автор Ключевое слово

Документ Ле Хоай Тузовский А. Ф. СЭБ Семантические технологии

ПОИСК Пусто

Результат поиска

11 результатов за 1.03 секунд с установленным порогом:0.1

РАЗРАБОТКА ЭЛЕКТРОННЫХ БИБЛИОТЕК НА ОСНОВЕ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ
Год:2012 [84.2%]
Автор:Тузовский А. Ф, Ле Хоай,



Рассматривается ряд проблем в электронных библиотеках, анализируются новые технологии, предоставляющие средства их решения. Появляются семантические электронные библиотеки, их архитектура электронных ресурсов, а также задачи

Ключевые слова:СЭБ, Семантические технологии, Электронная библиотека, Semantic search, Система управления знаниями,
Домены:

Рис. 2. Составление триплетов для поиска



7 результатов за 0.89 секунд с $\varepsilon=0.3$

Разработка семантических электронных библиотек Год: [99.0%]
Автор:Ле Хоай, Тузовский А. Ф,



Рассматривается подход к созданию электронных библиотек и их разработке с использованием семантических технологий. Появляются функции электронных библиотек, для автоматизации которых требуется использовать семантику и

Ключевые слова:Семантические технологии, СЭБ, Электронная библиотека, Semantic search,
Домены:

Рис. 3. Интерфейс категоризации или рекомендации на основе метаописаний

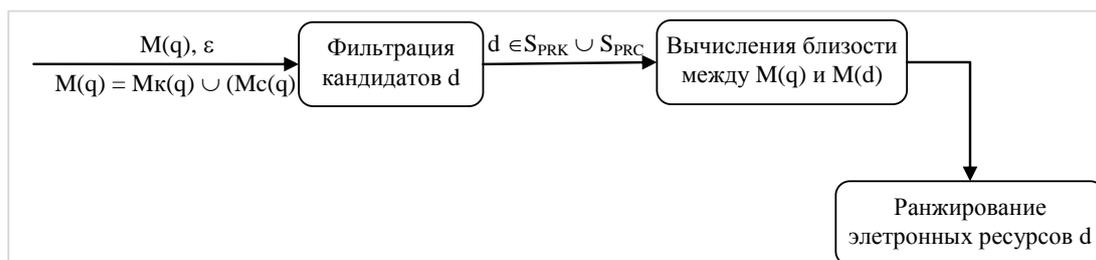


Рис. 4. Схема процесса семантического поиска

	S	P	O	№
Mк(q)	?S	_:type	_:Document	1
	?S	_:hasKeyword	_:Semantic_Technology	2
	?S	_:hasKeyword	_:Semantic_Digital_Library	3
	?S	_:hasAuthor	_:Ле Хоай	4
	?S	_:hasAuthor	_:Тузовский	5
Количество триплетов контекста (k): 5				
Mк(d)	_:2013051402540193	_:title	РАЗРАБОТКА ЭЛЕКТРОННЫХ БИБЛИОТЕК НА ОСНОВЕ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ	1
	_:2013051402540193	rdf:type	_:Article	2
	_:2013051402540193	_:hasKeyword	_:Semantic_Technology	3
	_:2013051402540193	_:hasKeyword	_:Semantic_Digital_Library	4
	_:2013051402540193	_:hasKeyword	_:Digital_Library	5
	_:2013051402540193	_:hasKeyword	_:Semantic_Search	6
	_:2013051402540193	_:hasKeyword	_:Knowledge_System	7
	_:2013051402540193	_:hasAuthor	_:Ле Хоай	8
	_:2013051402540193	_:hasAuthor	_:Тузовский	9
Количество триплетов контекста (h): 9				

Рис. 5. Пример метаописаний q и d

Пример численных вычислений. Пример вычисления делается для семантического поискового запроса, показанного на рис. 2 и его первого результата. Их метаописания показаны на рис. 5.

Вначале вычисляются близости между $M(q)$ и $M(d)$ по формуле 8 где $\varepsilon = 0.1$ $\alpha = 1$ и $\beta = 0$, так как семантический поиск выполняется по контексту.

На следующем шаге значение $\text{Sim}(M(q), M(d)) = \text{Sim}(M_k(q), M_k(d))$ вычисляется по формуле 6 или 7. и с целью простоты пояснения используется формулы 6. Для заданных $M_k(q)$ и $M_k(d)$ значение $k = 5$ (число триплетов $M_k(q)$), а значение $h = 9$ (число триплетов $M_k(d)$). Тогда значение $\text{Sim}(M_k(q), M_k(d))$ вычисляется следующим образом:

$$\text{Sim}(M_k(q), M_k(d)) = \frac{\sum_{t_i \in M_k(q)} \max(\text{Sim}(t_i, M_k(d)))}{|M_k(q)|} = \frac{\sum_{i=1}^5 \max(\text{Sim}(t_i, t_j))}{5} = \frac{\sum_{i=1}^5 \max(\text{Sim}(p_i, p_j) \times \text{Sim}(o_i, o_j))}{5}$$

Для $i = 1$ и $j = 1 \rightarrow 8$ величина $p(q)_1 = \text{rdf:type}$, а $p(d)_{1 \rightarrow 8} \in \{ _:title, \text{rdf:type}, _:hasKeyword, _:hasAuthor \}$. Так как в используемой онтологической модели нет семантических отношений между предикатами $p(q)_1$ и $p(d)_{1 \rightarrow 8}$ (но могут быть семантические отношения между другими предикатами), то $\max(\text{Sim}(t_1, M_k(d))) = \max_{i=1, j=1 \rightarrow 8} (\text{Sim}(p_i, p_j) \times \text{Sim}(o_i, o_j)) = (\text{Sim}(p_1, p_2) \times \text{Sim}(o_1, o_2))$, так как $\text{Sim}(p_i, p_j) = 0$ при $j \neq 2$.

Тогда в данном случае получается следующая окончательная формула: $\max(t_1, M_k(d)) = \text{Sim}(\text{rdf:type}, \text{rdf:type}) \times \text{Sim}(_:Document, _:Article) = \text{Sim}(_:Document, _:Article)$.

Для $i = 2$ и $j = 1 \rightarrow 8$ получаем: $\max(\text{Sim}(t_2, M_k(d)) = \text{Sim}(_:Semantic_Technology, _:Semantic_Technology) = 1$.

Аналогичным образом выполняются вычисления и для $i = 3 \rightarrow 5$.

В конечном результате значение $\text{Sim}(M_k(q), M_k(d))$ определяется следующим образом:

$$\text{Sim}(M_k(q), M_k(d)) = \frac{\text{Sim}(_:Document, _:Article) + 1 + 1 + 1 + 1}{5} = \frac{\text{Sim}(_:Document, _:Article) + 4}{5}$$

Для определения величины $\text{Sim}(_:Document, _:Article)$ используется граф семантических отношений GO (рис. 6). Здесь символы « $_$ » обозначают некоторое пространство имен понятий.

В показанном фрагменте графа GO используются четыре семантических отношений:

subClassOf (*подКласс*), *subPropertyOf* (*ПодСвойство*), *hasBroader* (*Уже*) и *hasNarrower* (*Шире*). Отметим, что все эти предикаты отображают таксономические отношения (на практике могут использоваться и другие предикаты). В таксономической иерархии имеются два отношения, которые либо обобщают (*subClassOf*, *subPropertyOf*, *hasBroader*), либо детализируют (*hasNarrower*) понятия.

Предпочтение всегда отдаётся детализации, это означает, что при поиске некоторого понятия более важными являются те понятия, которые детализуют рассматриваемое понятие. В связи с этим можно задать значения предикатом следующим образом:

- Если предикат p описывает отношение в направлении детализации, то $p_v > 0.5$. Например, в данной работе $p_v(\text{hasBroader}) = 0.8$.
- Если предикат p описывает отношение в направлении обобщения, то $p_v < 0.5$. Например, в данной работе $p_v(\text{hasNarrower}) = 0.4$.

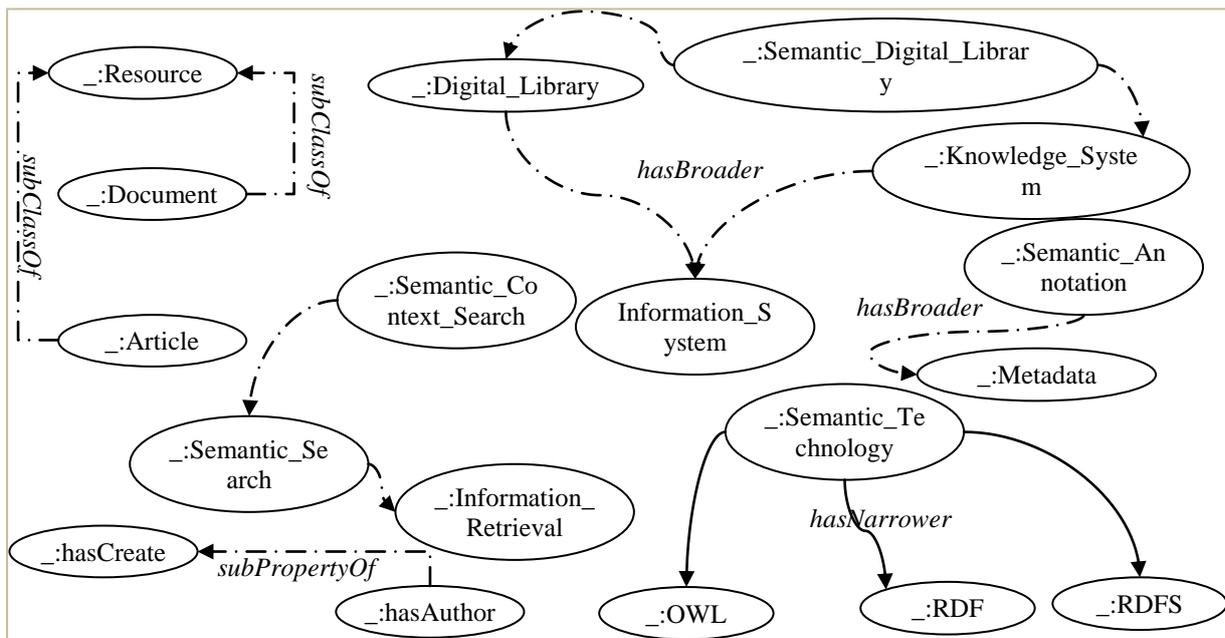


Рис. 6. Фрагмент графа семантических отношений GO

Все значения веса предикатов могут быть заданы специалистом, поддерживающим работу СЭБ, в соответствии с его пониманием онтологии предметной области.

В рассматриваемом случае $pv(subClassOf) = 0.7$ в направлении детализации, а в направления обобщения $pv(subClassOf) = 0.3$, а это значит, что $Sim(_:Resource, _:Article) = 0.7$, а $Sim(_:Resource; _:Article) = 0.3$ и тогда

$$Sim(M_k(q), M_k(d)) = \frac{Sim(_:Document, _:Article) + 4}{5} = \frac{0,3 * 0,7 + 4}{5} = 0,842.$$

Для контентного семантического поиска процессы вычисления выполняются аналогично. Однако при контентном поиске для вычисления близости между триплетами контента учитывается ещё и близость между субъектами триплетов.

5 Заключение

Повышение эффективности электронных библиотек в значительной степени связано с использованием в их работе описания семантики электронных ресурсов. Для создания таких ЭБ требуется решение целого комплекса новых задач. В данной статье предложены методы описания семантики ЭР и вычисления семантической близости, рассмотрена реализация стандартных функций электронных библиотек с их использованием. Выполнение тестирования показало высокие результаты со средними значениями критерий: Точность = 100%, Полнота = 94%.

Литература

- [1] Тузовский, А. Ф. Разработка семантических электронных библиотек / А. Ф. Тузовский, Х. Х. Ле // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011 – №. 2 (24) – С. 195–199.
- [2] Ле Х. Х. Разработка электронных библиотек на основе семантических технологий // Научно–технический вестник Поволжья. – Казань, 2012. №. 3. С. 138–145.
- [3] Тузовский А.Ф. Формирование семантических метаданных для объектов системы управления знаниями. //Известия Томского политехнического университета. 2007. Т. 310. №. 3. С. 108 – 112.
- [4] Нгуен, Б. Н. Модели и методы поиска информационных ресурсов с использованием семантических технологий: дис. канд. техн. наук / Нгуен Ба Нгок. – Томск, 2012. – 198с.
- [5] Hendler, A. J. Handbook of Semantic Web Technologies. – Springer, 2011.
- [6] OWL Web Ontology Language Overview // Доступ осуществлен 03.04.2013 по адресу <http://www.w3.org/TR/owl-features/>.
- [7] OWL:inverseOf // Доступ осуществлен 03.04.2013 по адресу <http://www.infowebml.ws/rdf-owl/inverseOf.htm>.
- [8] SPARQL 1.1 Federated Query // Доступ осуществлен 03.04.2013 по адресу <http://www.w3.org/TR/sparql11-federated-query/>.
- [9] System documentation for Sesame 2.x // Доступ осуществлен 03.04.2013 по адресу <http://www.openrdf.org/doc/sesame2/system/>.
- [10] Тузовский, А. Ф. Онтолого-семантические модели в корпоративных системах управления знаниями: дис. д-тр. тех. наук / А. Ф.

Тузовский. – Томск, 2007. – 342с.

- [11] Vector space model // Доступ осуществлен 26.07.2013 по адресу http://en.wikipedia.org/wiki/Vector_space_model.
- [12] Хоай, Л. Программная система «SemDL – система управления хранилищем электронных ресурсов с использованием семантических технологий» / Ле Хоай, А.Ф. Тузовский // Свидетельство о государственной регистрации программы для ЭВМ № 2013613266. М.: Федеральная служба по интеллектуальной собственности (Роспатент). – 2013.
- [13] Facebook graph search // Доступ осуществлен 03.04.2013 по адресу http://en.wikipedia.org/wiki/Facebook_Graph_Search.
- [14] Thomas Paul. The Lucene Search Engine // Доступ осуществлен 12.05.2013 по адресу <http://www.javaranch.com/journal/2004/04/Lucene.html>.
- [15] Welcome to Apache Lucene // Доступ осуществлен 12.05.2013 по адресу <http://lucene.apache.org/>.
- [16] Java Graph Visualization Library // Доступ осуществлен 12.05.2013 по адресу <http://www.jgraph.com/jgraph.html>.
- [17] JGraph Diagram Component // Доступ осуществлен 12.05.2013 по адресу <http://sourceforge.net/projects/jgraph/>.
- [18] JGraphT Visualizations via JGraph // Доступ осуществлен 12.05.2013 по адресу <http://jgrapht.org/visualizations.html>.
- [19] The Sesame API // Доступ осуществлен 12.05.2013 по адресу <http://www.openrdf.org/doc/sesame/users/ch07.html>.
- [20] Sesame distribution // Доступ осуществлен 12.05.2013 по адресу <http://sourceforge.net/projects/sesame/files/Sesame%20/>.