

Проблемы семантического поиска

Д.А. Малахов

Московский Государственный Университет им. М. В. Ломоносова

Введение

Семантический поиск – это метод информационного поиска, в котором релевантность документа запросу определяется семантически, а не синтаксически[1].

На протяжении долгого времени ведутся исследования в области семантического поиска. Разрабатываются новые алгоритмы, улучшаются существующие подходы. Но до сих пор остались открытыми некоторые серьезные проблемы, препятствующие повсеместному внедрению технологий семантического поиска.

В качестве примера можно рассмотреть проект Semantic WEB. В рамках него было сделано много исследований, множество организаций выкладывали свои открытые связанные данные, но существуют серьезные проблемы с использованием этих данных. Проблемы Semantic WEB не дают возможности этому проекту развиваться и стать более популярным. Решение части этих проблем позволило бы дать новую жизнь этому проекту.

Бурный рост популярности социальных сетей требует развития технологии поиска информации. Пользователи становятся все более требовательными к качеству услуг, оказываемых в сети. Поддержка семантики при поиске становится все более актуальной проблемой.

Данная работа посвящена анализу проблем, связанных с семантическим поиском. Кроме того, описывается решение ряда проблем.

Semantic WEB

Основной целью проекта Semantic Web является создание WEB 3.0. Для этого документы внутри сети должны быть связаны не только с помощью ссылок, но и семантически[2]. Semantic WEB базируется на нескольких стандартах, разработанных организацией W3C.

Стандарт RDF регламентирует представление знаний в виде троек (субъект, предикат, объект). Кроме RDF используются стандарты описания онтологий RDFS и OWL. Для манипуляции данными в семантическом хранилище был разработан язык запросов SPARQL.

Формат RDF является универсальной моделью данных, подходящей для хранения знаний и использования логического вывода. Возможно, именно это и не дало ему развиваться и стать достаточно востребованным.

Несмотря на кажущуюся простоту, так и не было разработано удобного инструмента для качественного выделения субъектов предикатов и объектов из текста. Кроме того, связывание данных из разных семантических хранилищ может быть осуществлено только в полуавтоматическом режиме, причем качество такого связывания также оставляет вопросы.

Существует проблема масштабирования семантической базы данных. Как правило, семантические запросы накладывают ограничения на связи между понятиями. На каждом узле распределенной семантической базы данных должна быть необходимая запросу информация. Это приводит к значительному дублированию данных, что в некоторых случаях делает использование распределенного хранилища бессмысленным. Или звеньям распределенного хранилища при выполнении запроса приходится общаться друг с другом, что негативно сказывается на производительности.

Отдельной проблемой является построение запроса к семантической базе данных. Далеко не каждый пользователь может легко и просто использовать SPARQL. Люди привыкли к поисковой строке, в которую они вводят описание своей информационно потребности, а машина сама должна ее распознать и выдать результат. Следствием этой проблемы является проблема построения семантического запроса по строке. Также предпринимаются попытки упростить модель языка запросов[3].

Кроме того, существующие механизмы индексирования и поиска данных существенно уступают по производительности полнотекстовому поиску. А логический вывод понижает эту производительность еще более значительно.

Таким образом, можно выделить основные проблемы Semantic WEB:

1. качественное выделение понятий и связей;
2. проблема масштабирования семантического хранилища;
3. построение запроса к семантическому хранилищу.
4. производительность.

Семантическое аннотирование

Семантическое аннотирование является популярным подходом к индексированию документов, для последующего поиска. Существует множество различных алгоритмов семантического аннотирования. В каждом из них документу или части документа приписывается некоторый набор семантически близких документу меток. В дальнейшем, можно искать документы по этим меткам. Кроме того, можно искать документы обычным полнотекстовым поиском, а потом учитывать эти метки при работе с документом, получая больше информации с помощью них[4]. Как правило, в качестве меток используются персоны, места, организации или другие субъекты [5].

С проблемой семантического аннотирования тесно связана проблема разрешения сущностей и проблема поиска сущностей. Проблема разрешения сущностей заключается в определении, что текст описывает одну из заданных сущностей, и составлении общего описания сущности на основе информации из разных источников. Проблема поиска сущностей заключается в поиске упоминаний о заданной сущности в аннотированном наборе документов [6].

Для описания меток часто используются онтологии, содержащие набор понятий и отношения между ними. Некоторые методы используют Википедию, как масштабный источник знаний[7]. В последнее время, методы семантического аннотирования все чаще обращаются к использованию массивного, взаимосвязанного облака Linked Open Data [5][6]. Например, с помощью средства семантического аннотирования GATE был проаннотирован Национальный Архив Великобритании (42 TB)[8].

Основная проблема использования результатов аннотирования - это поиск по понятиям. Здесь сказывается проблема Semantic WEB получения информации из семантического хранилища, ее решение могло бы помочь лучше использовать результаты аннотирования.

Поиск по хэштегам

С приходом социальных сетей, ориентированных на медиа-контент, более популярными становятся метки, описывающие контент, называемые хэштегами [9].

Хэштег — тематическая метка, вставляемая в записи Твиттера, ВКонтакте и прочих социальных сетей и содержащая ключевое слово или фразу, о которой идёт речь в записи.

Хэштеги становятся незаменимы при описании медиа-контента. Кроме того, короткие сообщения в микроблогах социальных сетей могут быть помечены хэштегами, включая

в себя как одно, так и несколько объединённых слов. Хэштеги дают возможность группировать подобные сообщения, таким образом можно найти хэштег и получить набор сообщений, которые его содержат.

Существует ряд проблем использования хэштегов:

1. ориентированность на ручное тегирование;
2. отсутствие семантики при поиске [10].

Решение проблемы использования семантики в хэштегах является ключевой задачей в информационном поиске по хэштегам.

Сильно упростить поиск по семантическому хранилищу могло бы использование семантического поиска по хэштегам или семантическим тегами. Каждому объекту из хранилища можно сопоставить набор семантических тегов, который должен быть проиндексирован с учетом семантики. В дальнейшем, для поиска информации о понятиях семантического хранилища можно использовать индекс тегов.

В следующем разделе будет представлена формализация понятия семантического тега. Также будут рассмотрены разные версии алгоритма поиска сужений семантического тега. Будет представлено теоретическое обоснование этого алгоритма, на базе него можно разработать алгоритм семантического поиска.

Семантический тег

Базовые определения

В этом разделе вводятся основные определения, связанные с семантическими тегами. Определяется связь семантических тегов с объектами, которые описываются тегами, эквивалентность тегов, множество объектов семантического тега.

Определение. Алфавитом будем называть любое конечное непустое множество. Элементы этого множества называются символами данного алфавита. Будем считать, что рассматриваемые дальше алфавиты не содержат символы собок (ϵ).

Пусть задан алфавит A .

Определение. Семантическим тегом алфавита A будем называть конечный набор символов, соответствующий следующей БНФ:

$\langle \text{семантический тег алфавита } A \rangle ::= \langle \text{тег} \rangle$

$\langle \text{тег} \rangle ::= \langle \text{простой тег} \rangle \mid \langle \text{сложный тег} \rangle$

$\langle \text{простой тег} \rangle ::= (\langle \text{набор символов } A \rangle)$

$\langle \text{набор символов } A \rangle ::= \langle \text{символ алфавита } A \rangle [\langle \text{набор символов } A \rangle]$

$\langle \text{сложный тег} \rangle ::= \langle \text{объединение} \rangle \mid \langle \text{пересечение} \rangle \mid \langle \text{дополнение} \rangle$

$\langle \text{объединение} \rangle ::= (\langle \text{тег} \rangle \cup \langle \text{тег} \rangle)$

$\langle \text{пересечение} \rangle ::= (\langle \text{тег} \rangle \cap \langle \text{тег} \rangle)$

$\langle \text{дополнение} \rangle ::= C(\langle \text{тег} \rangle)$

Для упрощения в дальнейшем будем называть семантические теги просто тегами.

В определении тега даются определения простого тега, сложного тега, объединения, пересечения и дополнения тега. В дальнейшем будем использовать эти понятия.

Пусть заданы множество тегов T и множество простых тегов ST .

Будем считать, что теги предназначены для описания некоторых объектов. Пусть задано множество таких объектов O .

Определение. Описанием объекта $o \in O$ тегом $t \in T$ будем называть пару (t, o) .

Определение. Множеством описаний будем называть множество $D \subset \{(t, o) \mid o \in O, t \in T\}$, удовлетворяющее следующим правилам:

1. $\forall o \in O, \forall t_1 t_2 \in T, t = t_1 \cap t_2 ((t, o) \in D \Leftrightarrow (t_1, o) \in D \wedge (t_2, o) \in D)$
2. $\forall o \in O, \forall t_1 t_2 \in T, t = t_1 \cup t_2 ((t, o) \in D \Leftrightarrow (t_1, o) \in D \vee (t_2, o) \in D)$
3. $\forall o \in O, \forall t_1 t_2 \in T, t_1 = C(t_2) ((t_1, o) \in D \Leftrightarrow \neg((t_2, o) \in D))$

В дальнейшем, если $(t, o) \in D$, то будем считать, что t описывает o .

Определение. Базисом множества D на T и O является множество $B_D \subseteq D$ такое, что после применения правил 1-3 к B_D получается множество D , причем $\forall b \in B_D : B_D \setminus b$ базисом множества D не является.

Будем считать, что если $(t, o) \in D \vee \neg((t, o) \in B_D)$, то (t, o) выводима из B_D , и обозначать $B_D \mapsto (t, o) \in D$.

Теорема. B_D однозначно определяет D .

Доказательство. Предположим, что существует два разных множества D_1 и D_2 , имеющих один базис B . Так как $D_1 \neq D_2$: $\exists(t, o) \in D_1, \neg((t, o) \in D_2) \Rightarrow (t, o) \in B \vee B \mapsto (t, o) \in D_1 \Rightarrow B \mapsto (t, o) \in D_2 \Rightarrow (t, o) \in D_2 \Rightarrow$ противоречие. ■

Замечание. Базис не обязан быть единственным.

Пусть задано некоторое множество описаний D на множестве T и O .

Определение. Пусть задан тег $t \in T$. Множеством объектов тега (классом) t будем называть $class_t = \{o \mid o \in O \wedge (t, o) \in D\}$.

Согласно определению множества D справедливы следующие утверждения:

1. $\forall t_1 t_2 \in T, t = t_1 \cap t_2 (class_t = class_{t_1} \cap class_{t_2})$
2. $\forall t_1 t_2 \in T, t = t_1 \cup t_2 (class_t = class_{t_1} \cup class_{t_2})$
3. $\forall t_1 t_2 \in T, t_1 = C(t_2) (class_{t_1} = C(class_{t_2}))$

Определение. Теги $t_1 t_2 \in T$ будем называть эквивалентными $t_1 = t_2$, если $class_{t_1} = class_{t_2}$

Определение. Тег $t \in T$ будем называть пустым $t = \emptyset$, если: $class_t = \emptyset$

Теорема. $\forall t \in T (t \cap \emptyset = \emptyset \wedge t \cup \emptyset = t)$.

Доказательство.

1. $class_{t \cap \emptyset} = class_t \cap class_{\emptyset} \Rightarrow class_{t \cap \emptyset} = class_t \cap \emptyset = \emptyset \Rightarrow t \cap \emptyset = \emptyset$.

2. $class_{t \cup \emptyset} = class_t \cup class_{\emptyset} \Rightarrow class_{t \cup \emptyset} = class_t \cup \emptyset = class_t \Rightarrow t \cup \emptyset = t$.

Теорема. $\forall t t_1 t_2 \in T, t = t_1 \cup t_2 (C(t) = C(t_1) \cap C(t_2))$

Доказательство. $class_t = class_{t_1} \cup class_{t_2} \Rightarrow C(class_t) = C(class_{t_1}) \cap C(class_{t_2}) \Rightarrow class_{C(t)} = C(class_{C(t_1)}) \cap class_{C(t_2)} \Rightarrow C(t) = C(t_1) \cap C(t_2)$ ■

Теорема. $\forall t t_1 t_2 \in T, t = t_1 \cap t_2 (C(t) = C(t_1) \cup C(t_2))$

Доказательство. $class_t = class_{t_1} \cap class_{t_2} \Rightarrow C(class_t) = C(class_{t_1}) \cup C(class_{t_2}) \Rightarrow class_{C(t)} = class_{C(t_1)} \cup class_{C(t_2)} \Rightarrow C(t) = C(t_1) \cup C(t_2)$ ■

Теорема. $\forall t \in T (C(C(t)) = t)$

Доказательство. $class_t = C(C(class_{t_1})) \Rightarrow class_t = C(class_{C(t_1)}) \Rightarrow class_t = class_{C(C(t_1))} \Rightarrow t = C(C(t))$ ■

Подведем итог, было дано определение семантического тега, введено множество семантических тегов T . Определено множество описаний D , рассмотрены некоторые свойства множества описаний D , введен базис множества описаний B_D . Введено понятие класса тега, изучены его свойства.

Множество $SubClasses_t$

В этом разделе вводится иерархия семантических тегов, рассматриваются ее свойства, ставится задача поиска тегов, приводится первое приближение алгоритма, решающего эту задачу.

Определение. Пусть задан тег $t \in T$.

Множеством сужений тега t будем называть множество:

$$SubClasses_t = \{t_1 \in T \mid class_{t_1} \subseteq class_t\}.$$

Определение. Пусть задан тег $t \in T$.

Множеством расширений тега t будем называть множество:

$$SuperClasses_t = \{t_1 \in T \mid class_t \subseteq class_{t_1}\}$$

Задачей поиска сужений заданного тега $t \in T$ на заданном множестве описаний D тегами T объектов O будем называть нахождение множества $SubClasses_t$.

Теорема. $\forall t t_1 t_2 \in T, t = t_1 \cap t_2 (t \in SubClasses_{t_1} \wedge t \in SubClasses_{t_2})$

Доказательство. $class_t = class_{t_1} \cap class_{t_2} \Rightarrow class_t \subseteq class_{t_1} \wedge class_t \subseteq class_{t_2} \Rightarrow t \in SubClasses_{t_1} \wedge t \in SubClasses_{t_2}$ ■

Теорема. $\forall t t_1 t_2 \in T, t = t_1 \cup t_2 (t_1 \in SubClasses_t \vee t_2 \in SubClasses_t)$

Доказательство. $class_t = class_{t_1} \cup class_{t_2} \Rightarrow class_{t_1} \subseteq class_t \wedge class_{t_2} \subseteq class_t \Rightarrow t_1 \in SubClasses_t \vee t_2 \in SubClasses_t$ ■

Теорема. $\forall t_1 t_2 \in T, t_1 = C(t_2) (t_2 \notin SubClasses_{t_1} \wedge t_1 \notin SubClasses_{t_2})$

Доказательство. $class_{t_1} = C(class_{t_2}) \Rightarrow class_{t_1} \not\subseteq class_{t_2} \wedge class_{t_2} \subseteq class_{t_1} \Rightarrow t_1 \notin SubClasses_{t_2} \wedge t_2 \notin SubClasses_{t_1}$ ■

Теорема о сужениях объединения.

$\forall t \in T, t = \bigcup_{i=1}^n t_i, \forall M \subseteq \{1..n\} (\bigcup_{i \in M} t_i \in SubClasses_t)$

Доказательство.

$\bigcup_{i \in M} class_{t_i} \subseteq class_t \Rightarrow class_{\bigcup_{i \in M} t_i} \subseteq class_t \Rightarrow \bigcup_{i \in M} t_i \in SubClasses_t$ ■

Теорема о сужениях пересечения.

$\forall t \in T, t = \bigcap_{i=1}^n t_i, \forall M \subseteq \{1..n\} (t \in SubClasses_{\bigcap_{i \in M} t_i})$

Доказательство.

$class_t \subseteq \bigcap_{i \in M} class_{t_i} \Rightarrow class_t \subseteq class_{\bigcap_{i \in M} t_i} \Rightarrow t \in SubClasses_{\bigcap_{i \in M} t_i}$ ■

Следствие. $\forall t_1 t_2 \in T (t_1 \cap t_2 \in SubClasses_{t_1} \wedge t_1 \cap t_2 \in SubClasses_{t_2})$

Замечание. Теоремы о сужениях объединения и сужениях пересечений не позволяют полностью определить $SubClasses_t$, но позволяют приблизиться к решению и определить подмножество $SubClasses_t$.

Теорема о сужениях дополнения.

$\forall t_1 t_2 \in T, t_1 = C(t_2) (SubClasses_{t_1} = T \cap C(\bigcap_{k \in SubClasses_{t_2}} SuperClasses_k))$

Доказательство.

1. Пусть $t \in T \cap C(\bigcap_{k \in SubClasses_{t_2}} SuperClasses_k) \Rightarrow \nexists subt \in T (subt \in SubClasses_{t_2} \wedge t \in SuperClasses_{subt}) \Rightarrow class_t \cap class_{t_2} = \emptyset \Rightarrow class_t \in C(class_{t_2}) \Rightarrow class_t \in class_{t_1} \Rightarrow t \in SubClasses_{t_1}$

2. Пусть $t \in SubClasses_{t_1} \Rightarrow class_t \in class_{t_1} \Rightarrow class_t \in C(class_{t_2}) \Rightarrow class_t \cap class_{t_2} = \emptyset \Rightarrow \nexists subt \in T (subt \in SubClasses_{t_2} \wedge t \in SuperClasses_{subt}) \Rightarrow t \in T \cap C(\bigcap_{k \in SubClasses_{t_2}} SuperClasses_k)$

На основании теорем о сужениях расширений, объединений, дополнений можно определить алгоритм решения задачи поиска сужений:

1. Пусть $t \in ST$, тогда $SubClasses_t = \{t\}$.

2. Пусть $t = \bigcup_{i=1}^n t_i$, тогда задача сводится к поиску сужений для тегов $\bigcup_{i \in M} t_i, \forall M \subseteq \{1..n\}$.

3. Пусть $t = \bigcap_{i=1}^n t_i$, тогда задача сводится к поиску сужений для тегов $t \cap extt, \forall extt \in T$.

4. Пусть $t = C(t_1), t_1 \in ST$, тогда $SubClasses_t = T/SuperClasses_{t_1}$, по теореме о сужениях дополнения, так как $SubClasses_{t_1} = \{t_1\}$.

5. Если $t = C(t_1), t_1 \notin ST$, то используем правила
 $C(t_1 \cap t_2) = C(t_1) \cup C(t_2), C(t_1 \cup t_2) = C(t_1) \cap C(t_2)$.

Подведем итог, были даны определения сужения и расширения семантического тега, рассмотрены их свойства. Был приведен алгоритм, решающий задачу поиска тегов на основании доказанных теорем о сужениях объединений, пересечений, дополнений.

Синонимы

В алгоритме решения задачи поиска сужений есть ряд недостатков, один из основных недостатков в том, что в нем не учитываются семантические связи между простым тегами. Этот алгоритм рассчитан на то, что сужением некоторого тега $t_1 \in T$ может быть только тег $t_2 = t_1 \cap anyt$, где $anyt \in T$. Также этот алгоритм рассчитан на то, что расширением некоторого тега $t_1 \in T$ может быть только тег $t_2 = t_1 \cup anyt$, где $anyt \in T$. В общем случае эти допущения не верны.

Определение. Пусть задан тег $t \in T$.

Множество синонимов тега t :

$$Synonyms_t = \{syn \mid syn \in ST \wedge t \in SubClasses_{syn} \wedge syn \in SubClasses_t\}.$$

Множество гипонимов тега t :

$$Hyponyms_t = \{hyp \mid hyp \in ST \wedge hyp \in SubClasses_t\}.$$

Множество гиперонимов тега t :

$$Hyperonyms_t = \{hyp \mid hyp \in ST \wedge t \in SubClasses_{hyp}\}.$$

Из определения следует, что $Synonyms_t = Hyponyms_t \cap Hyperonyms_t$.

Множества синонимов, гипонимов, гиперонимов задаются с помощью множества описаний D тегами множества T объектов множества O .

Рассмотрим теги $t_1, t_2 \in T$.

Если $class_{t_1} = class_{t_2}$, то t_1 и t_2 синонимы.

Если $class_{t_1} \subseteq class_{t_2}$, то t_1 гипоним t_2 , а t_2 гипероним t_1 .

Учитывая то, что сужения некоторого тега t могут быть определены с помощью множества описаний D , открывается возможность улучшения алгоритма решения задачи поиска сужений:

1. Пусть $t \in ST$, тогда $SubClasses_t = Hyponyms_t$.
2. Пусть $t = \bigcup_{i=1}^n t_i$, тогда задача сводится к поиску сужений для тегов $\bigcup_{i \in M} t_i, \forall M \subseteq \{1..n\}$. Кроме того решением являются теги из множества $Hyponyms_t$.

3. Пусть $t = \bigcap_{i=1}^n t_i$, тогда задача сводится к поиску сужений для тегов $t \cap extt$, $\forall extt \in T$. Кроме того решением являются теги из множества $Hyponymst$.

4. Пусть $t = C(t_1), t_1 \in ST$, тогда $SubClasses_t = T / \bigcup_{h \in Hyponymst} SuperClasses_h$, по теореме о сужениях дополнения, так как $SubClasses_{t_1} = Hyponymst$.

5. Если $t = C(t_1), t_1 \notin ST$, то используем правила
 $C(t_1 \cap t_2) = C(t_1) \cup C(t_2)$, $C(t_1 \cup t_2) = C(t_1) \cap C(t_2)$.

В улучшенной версии алгоритма в шагах 1-4 учитываются $Hyponymst$. Это позволяет учитывать связи между простыми тегами, задавая их с помощью множества описаний D. Эти связи в итоге переносятся и на сложные теги.

Подведем итог, были введены множества синонимов, гипонимов, гиперонимов, рассмотрены связи между ними. Улучшен алгоритм решения задачи поиска сужений.

Заключение

В данной работе были рассмотрены проблемы семантического поиска. Был представлен обзор проблем проекта Semantic WEB. Было предложено решение ряда проблем связанных с Semantic WEB, семантическим аннотированием, поиском в социальных сетях.

Было введено понятие семантического тега, были изучены его свойства и представлены несколько вариантов алгоритма поиска сужений для семантического тега с теоретическим обоснованием.

Алгоритм поиска сужений семантического тега можно применить для поиска объектов семантического тега. Если объекты описываются сужением тега, значит эти объекты описываются и самим тегом. Таким образом, для поиска объектов достаточно найти все сужения тега, а для каждого сужения нужно получить его объекты.

Было показано, что, решив проблему семантического поиска для тегов, мы также можем использовать поиск для решения проблем в Semantic Web и в семантическом аннотировании. В частности, за счет тегирования понятий семантического хранилища можно организовать поиск по этим понятиям, фильтровать понятия при поиске, повышая производительность системы. Кроме того, с помощью тегов может быть решена проблема масштабируемости. Можно выделять родственные понятия, расположив их на одной машине, повысить качество распределенного хранилища.

В рамках дальнейшей работы планируется реализация приведенного алгоритма и построения, на основании него, системы индексирования и поиска для семантических тегов.

Литература

1. *Guha R.* Semantic search // Proceedings of the 12th international conference on World Wide Web. – N.Y. ACM Press, 2003. – P. 700–709.
2. *Berners-Lee T. et al.* The semantic web //Scientific american.–2001.–Т. 284.– С. 28-37.
3. *Styperek A., Ciesielczyk M., Szwabe A.* Semantic search engine with an intuitive user interface //Proceedings of the companion publication of the 23rd international conference on World wide web companion. – International World Wide Web Conferences Steering Committee, 2014. – С. 383-384.
4. *Giannopoulos G. et al.* GoNTogle: a tool for semantic annotation and search //The Semantic Web: Research and Applications. – Springer Berlin Heidelberg, 2010. – С. 376-380.
5. *Bontcheva K., Tablan V., Cunningham H.* Semantic search over documents and ontologies //Bridging Between Information Retrieval and Databases. – Springer Berlin Heidelberg, 2014. – С. 31-53.
6. *Alahmari F., Magee L.* Linked Data and Entity Search: A Brief History and Some Ways Ahead //Proceedings of the 3rd Australasian Web Conference (AWC 2015). – 2015. – Т. 27. – С. 30.
7. *Berlanga R., Nebot V., Pérez M.* Tailored semantic annotation for semantic search //Web Semantics: Science, Services and Agents on the World Wide Web. – 2015. – Т. 30. – С. 69-81.
8. *Maynard D., Greenwood M. A.* Large Scale Semantic Annotation, Indexing and Search at The National Archives //Lrec. – 2012. – С. 3487-3494.
9. *Efron M.* Hashtag retrieval in a microblogging environment //Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2010. – С. 787-788.
10. *Posch L. et al.* Meaning as collective use: predicting semantic hashtag categories on twitter //Proceedings of the 22nd international conference on World Wide Web companion. – International World Wide Web Conferences Steering Committee, 2013. – С. 621-628.