Графы Знаний

План лекции:

6. Показатели качества графа знаний
7. Добавление данных в граф знаний
8. Публикация графов знаний
9. Практические примеры графов знаний

---

6. Показатели качества графа знаний

Независимо от типа источника(-ов), из которых создается граф знаний, данные, извлеченные для исходного графа знаний, обычно будут неполными и будут содержать повторяющиеся, противоречивые или даже неверные утверждения – особенно если они взяты из нескольких источников. Таким образом, после первоначального создания и обогащения графа знаний из внешних источников, решающим шагом является оценка качества полученного графа знаний. Под качеством мы здесь подразумеваем пригодность к целевому использованию. Показатели качества помогают определить, для каких целей граф знаний может быть надежно использован. Далее мы обсудим измерения качества, охватывающие аспекты качества данных, которые эволюционирует от традиционной области баз данных к области графов знаний. Некоторые из них являются общими, а другие-более специфичными для графов знаний. В то время как измерения качества нацелены на охват качественных аспектов данных, мы также обсуждаем показатели качества, которые обеспечивают способы измерения количественных аспектов этих измерений. Мы обсуждим группировки измерений и метрик, полученные Батини и Сканнапьеко.

Точность

Точность - это степень, с которой объекты и отношения, представленные узлами и ребрами графа, точно отражают реальные явления. Точность может быть подразделена на три измерения: синтаксическая точность, семантическая точность и временная точность.

Синтаксическая точность - это степень точности данных по отношению к грамматическим правилам, определенным для предметной области и/или модели данных. Распространенный пример синтаксических неточностей встречается с узлами типа данных, которые могут быть несовместимы с определенным типом или иметь неправильный формат. Например, если предположить, что свойство start определено с типом xsd:dateTime, то принятие такого значения, "March 29, 2019, 20:00"^^xsd:string которое было бы несовместимо с определенным типом, в то время как значение "March 29, 2019, 20:00"^^xsd:dateTime было бы искажено (так как ожидаемым значением является "2019-11-12T20:00:00"^^xsd:dateTime). Соответствующая метрика синтаксической точности - это отношение между числом неверных значений данного свойства и общим числом значений того же свойства. Такие формы синтаксической точности обычно можно оценить с помощью инструментов валидации.

Семантическая (смысловая) точность - это степень, в которой данные значения правильно представляют явления реального мира, которые могут быть получены неточным извлечением результатов, неточным ограничением владения и т. д. Например, учитывая, что Национальный Конгресс Чили находится в Вальпараисо, это может привести к тому ребру Chile —capital→ Valparaiso, что на самом деле семантически неточно: в Чили столица - Сантьяго. Оценить уровень

семантических неточностей довольно сложно. Хотя одним из вариантов является применение ручной проверки, автоматическим вариантом может быть проверка заявленной связи по нескольким источникам. Другой вариант – проверить качество отдельных процессов, используемых для создания графа знаний, основываясь на таких показателях, как точность, возможно, с помощью человеческих экспертов или стандартов.

Временная точность (своевременность) - это степень, в которой граф знаний в настоящее время соответствует реальному состоянию мира; другими словами, граф знаний может быть семантически точным сейчас, но может быстро стать неточным (устаревшим), если не существует процедур, позволяющих своевременно поддерживать его в актуальном состоянии. Например, представьте себе пользователя, проверяющего граф туристических знаний о рейсах из одного города в другой. Предположим, что расписание рейсов обновляется каждую минуту с текущими статусами рейсов, но граф знаний обновляется только каждый час. В этом случае мы видим, что существует проблема качества, связанная со своевременностью в графе знаний. Своевременность может быть оценена на основе того, как часто граф знаний обновляется по отношению к базовым источникам, что может быть сделано с использованием временных аннотаций изменений в графе знаний, а также контекстуальных представлений, отражающих временную достоверность данных.

Покрытие

Покрытие относится к избеганию пропуска релевантных предметной области элементов, которые в противном случае могут привести к неполным результатам запроса или повлечь за собой предвзятые модели и т. д.

Полнота - это степень, в которой вся необходимая информация присутствует в конкретном наборе данных. Полнота включает в себя следующие аспекты: (I) по схеме полноты - означает степень, в которой классы и свойства схемы представлены на графе данные, (II) в собственною полноту - относится к соотношению пропущенных значений для определенного свойства, а также (III) полнота экземпляров - предусматривает процент от всех реальных сущностей определенного типа, которые представлены в наборах данных, и (IV) полнота связанности - относится к степени, в которой экземпляры в данных комплект взаимосвязаны. Прямое измерение полноты нетривиально, поскольку оно требует знания гипотетического идеального графа знаний, содержащего все элементы, которые рассматриваемый граф знаний должен “идеально” представлять. Конкретные стратегии включают в себя сравнение со стандартами, которые обеспечивают образцы идеального графа знаний (возможно, на основе утверждений о полноте), или измерение отзыва методов извлечения из полных источников и т. д.

Репрезентативность - это связанное измерение, которое вместо того, чтобы фокусироваться на соотношении элементов, относящихся к предметной области, которые отсутствуют, фокусируется на оценке утверждений высокого уровня в том, что включено/исключено из графа знаний. Как таковое, это измерение предполагает, что граф знаний является неполным, т. е. что это образец не идеального графа знаний, и спрашивает, насколько предвзят этот образец. Предубеждения могут возникать в данных, в схеме или во время ризонинга. Примеры предвзятости данных включают географические утверждения, которые недооценивают сущности/отношения из определенных частей мира, лингвистические предубеждения, которые недооценивают многоязычные ресурсы (например, ярлыки и описания) для определенных языков, социальные предубеждения, которые недооценивают людей определенного пола или расы, и т. д. Напротив, смещения схемы могут быть результатом высокоуровневых определений, извлеченных из необъективных данных, семантических определений, которые не охватывают необычные случаи, и т. д. Непризнанные утверждения могут привести к неблагоприятным последствиям; например, если наш граф туристических знаний имеет географический уклон в сторону событий и достопримечательностей, близких к городу Сантьяго, – возможно, из-за источников, используемых для создания, найма

кураторов из города и т. д. – тогда это может привести к непропорциональному продвижению туризма в Сантьяго и вокруг него (потенциально усугубляя будущие утверждения). Меры репрезентативности включают в себя сравнение известных статистических распределений с графом знаний, например, Сравнение геоалоцированных объектов с известными плотностями населения, лингвистических распределений с известными распределениями носителей и т. д. Другой вариант-сравнить граф знаний с общими статистическими законами, например Soulet et al. исследует несоответствие закону Бенфорда для измерения репрезентативности в графах знаний.

Когерентность

Когерентность определяет насколько хорошо граф знаний соответствует – или когерентен – формальной семантике и ограничениям, определенным на уровне схемы.

Непротиворечивость означает, что граф знаний свободен от (логических/формальных) противоречий относительно рассматриваемой конкретной логики. Например, в онтологии нашего графа знаний мы можем определить что flight –range→ Airport –disj. c.→ City, что в сочетании с ребрами Arica –flight→ Santiago –type→ City, порождает непоследовательность, влекущую за собой то, что Santiago является членом непересекающихся классов City и Airport. В более общем случае любой семантический признак в таблицах 3-5 с условием “не” может привести к несоответствиям. Мерой согласованности может быть количество несоответствий, найденных в графе знаний, возможно, подразделенное на количество таких несоответствий, выявленных каждым семантическим признаком.

Валидность означает, что граф знаний свободен от нарушений ограничений, таких как установленные выражения схемы. Мы можем, например, указать класс CITY, целевые узлы которого имеют не более одной страны. Затем, учитывая ребра Chile ←country– Santiago –country→ Cuba, и предполагая , что Santiago становится целью CITY, мы имеем нарушение ограничения. И наоборот, даже если бы мы определили аналогичные ограничения в онтологии, это не обязательно вызвало бы несогласованность, поскольку без UNA мы сначала сделали бы вывод об этом Chile и Cuba ссылались бы на одну и ту же сущность. Простой мерой достоверности является подсчет количества нарушений на одно ограничение.

Лаконичность

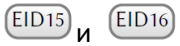
Лаконичность означает включение только релевантного содержания (избегая “информационной перегрузки”), которое представлено в сжатой и понятной форме.

Лаконичность означает отказ от включения элементов схемы и данных, не имеющих отношения к предметной области. Мендес и др. различают интенсиональную краткость (уровень схемы), которая относится к случаю, когда данные не содержат избыточных элементов схемы (свойств, классов, фигур и т. д.), и экстенсиональную краткость (уровень данных), когда данные не содержат избыточных сущностей и отношений. Например, включение событий Santiago de Cuba в наш граф знаний, посвященный туризму в Чили, может повлиять на экстенсиональную краткость графа знаний, потенциально возвращая нерелевантные результаты для данной области. В общем, краткость может быть измерена в терминах соотношения свойств, классов, форм, сущностей, отношений и т. д., имеющих отношение к предметной области, что, в свою очередь, может потребовать стандарта или методов оценки предметной области.

Репрезентативная лаконичность относится к степени, в которой содержание компактно представлено в графе знаний, который также может быть интенсиональным или экстенсиональным. Например, наличие двух свойств полет и мухи, служащие одной и той же цели,

отрицательно повлияло бы на интенсиональную форму репрезентативной краткости, в то время как наличие двух узлов 

представление столицы Чили (причем ни один из них не связан с другим) повлияло бы на экстенсиональную форму репрезентативной краткости. Другим примером краткости представления является ненужное использование сложных моделирующих конструкций, таких как использование овеществления без необходимости или использование связанных списков, когда порядок элементов не важен. Хотя лаконичность представления трудно оценить, можно использовать такие показатели, как количество избыточных узлов.

Понятность относится к легкости, с которой данные могут быть интерпретированы без двусмысленности людьми – пользователями, что включает, по крайней мере, предоставление удобочитаемых для человека ярлыков и описаний (предпочтительно на разных языках), которые позволяют им понять, о чем идет речь. Возвращаясь к рисунку 1, хотя узлы 

используются для обеспечения уникальных идентификаторов событий, они также должны быть связаны с такими метками, как 

. В идеале читаемая человеком информация достаточна для того, чтобы устранить неоднозначность конкретного узла, например, связать описание 

тем, чтобы исключить город из синонимичных. Меры понятности могут включать в себя соотношение узлов с удобочитаемыми метками и описаниями, уникальность таких меток и описаний, поддерживаемые языки и т. д.

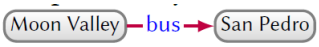
Другие Качественные Метрики

Мы обсудили некоторые ключевые аспекты качества, которые были рассмотрены для графов знаний и в целом применимы к ним. Дальнейшие измерения могут быть уместны в контексте конкретных областей, конкретных приложений или конкретных моделей графовых данных.

7. Добавление данных в граф знаний

Помимо оценки качества графа знаний, существуют методы уточнения графа знаний, в частности для (полу)автоматического завершения графа знаний и коррекции графа знаний. В отличие от задач создания и обогащения, описанных в разделе 6, уточнение обычно не включает в себя применение методов извлечения или картирования по внешним источникам с целью включения их содержимого в локальный граф знаний. Вместо этого уточнение обычно нацелено на улучшение заданного графа локальных знаний (но потенциально с использованием внешних источников для проверки локального контента).

Завершение (Completion)

Графы знаний характеризуются неполнотой. Таким образом, завершение графа знаний направлено на заполнение недостающих ребер (или недостающих звеньев) графа знаний, т. е. ребер, которые считаются правильными, но не заданы и не связаны графом знаний. Эта задача часто решается с помощью методов предсказания связей, предложенных в области статистического реляционного обучения, которые предсказывают существование – или, иногда в более общем плане, предсказывают вероятность правильности – отсутствующих ребер. Например, можно предсказать, что ребро 

является вероятным недостающим ребром для графа на рис. 24, учитывая, что большинство наблюдаемых автобусных маршрутов являются возвратными. Ребра предсказания могут ориентироваться на три параметра: общие с одним из ребер с произвольными названиями, например, автобус, рейс, тип и т. д.; тип ссылки на этикетку типа, указывающую тип объекта, и связи идентичности с участием ребра с такой же этикеткой, указывая, что два узла, относятся к одной и той же сущности. В то время как связи типа и идентичности могут быть определены с помощью общих методов прогнозирования связей, конкретная семантика связей типа и идентичности может быть решена с помощью пользовательских методов.

Общее предсказание связи. Предсказание связей, в общем случае, часто рассматривается с помощью индуктивных методов, как описано в разделе 5, и, в частности, вложений графа знаний и интеллектуального анализа правил/аксиом. Например, на рис. 24, используя встраивания графа знаний, мы можем обнаружить, что заданное ребро $(x)\text{–bus}\rightarrow(y)$, (отсутствующее) ребро $(y)\text{–bus}\rightarrow(x)$ имеет высокую правдоподобность, а используя символьные подходы, мы можем узнать правило высокого уровня $(?x)\text{–bus}\rightarrow(?y) \Rightarrow (?y)\text{–bus}\rightarrow(?x)$. Любой такой подход поможет нам предсказать недостающее звено (Moon Valley)–bus→(San Pedro).

Предсказание типизированной связи. Типизированные связи имеют особое значение для графа знаний, где выделенные методы могут быть использованы с учетом конкретной семантики таких связей. В случае прогнозирования типа существует только одна граничная метка (тип) и, как правило, меньше различных значений (классов), чем в других случаях, так что задача может быть сведена к традиционной задаче классификации, модели обучения для идентификации каждого семантического класса на основе на таких функциях, как исходящие и / или входящие метки ребер на их экземплярах в графе знаний. Например, предположим, что на рисунке 24 мы также знаем, что (Arica), (Calama), (Puerto Montt), (Punta Arenas) и (Santiago) относятся к типу (City). Затем мы можем предсказать, что (Iquique) и (Easter Island) также имеют тип (City), на основе наличия ребер, помеченных как полет к / из этих узлов, которые (мы предполагаем) стали хорошей функцией для прогнозирования этого класса (первое прогнозирование является правильным, а второе - неверным). Графовые нейронные сети также могут использоваться для классификации / прогнозирования типа узлов.

Предсказание идентификационной связи. Прогнозирование связей идентичности включает поиск узлов, которые относятся к одному и тому же объекту; это аналогично задаче сопоставления сущностей (также известной как связывание записей, дедупликация и т. д.), рассматриваемой в более общих настройках интеграции данных. Такие методы обычно основаны на двух типах сопоставителей: сопоставители значений определяют, насколько похожи значения двух сущностей в данном свойстве, что может включать показатели сходства для строк, чисел, дат и т. Д .; в то время как средства сопоставления контекста учитывают сходство сущностей на основе различных узлов и ребер. Иллюстративный пример приведен на рисунке 35, где средства сопоставления значений будут вычислять сходство между значениями, такими как (7400) и (7500), в то время как средства сопоставления контекста будут вычислять сходство между (Easter Island) и (Rapa Ñui) на основе информации об окружающей среде, например, о том, что у них схожие широта, долгота, население и одно и то же место (для сравнения, сопоставитель значений на этой паре узлов будет измерять сходство строк между «Островом Пасхи» и «Rapa ui») . Основной проблемой в этой настройке является эффективность, где для попарного сопоставления потребуется $(n2)$ сравнений для количества узлов. Чтобы решить эту проблему, можно использовать блокировку для группировки похожих объектов в (возможно, перекрывающиеся, возможно, непересекающиеся) «блоки» на основе ключей, сохраняющих сходство, с сопоставлением, выполняемым внутри каждого блока; например, при сопоставлении мест на основе широты / долготы блоки могут представлять географические регионы. Альтернативой дискретной блокировке является использование окон над объектами в порядке, сохраняющем сходство, или рассмотрение поиска похожих объектов в многомерных пространствах (например, пространстве-времени, пространствах с расстояниями Минковского, ортодромных пространствах и т. д.). Результатами могут быть либо пары узлов с вычисленной достоверностью того, что они относятся к одному и тому же объекту, либо четкие идентификационные связи, извлеченные на основе фиксированного порога, двоичной классификации и т. д. Для надежных идентификационных связей затем могут быть объединены связи узлов; например, мы можем выбрать (Easter Island) в качестве канонического узла и объединить

связи Rapa Nui на него, что позволит нам найти, например, объекты всемирного наследия в Тихом океане из рисунка 35 на основе (консолидированного) подграфа.

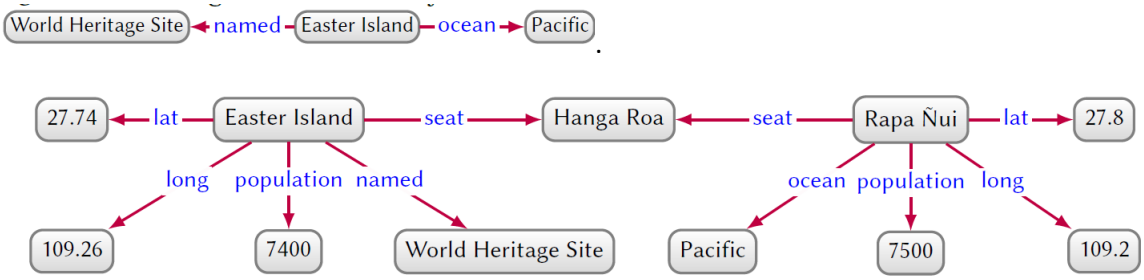


Рис. 35. Пример тождественности, где Rapa-Nui и остров Пасхи относятся к одному и тому же острову

Исправление

В отличие от завершения, которое находит новые ребра в графе знаний, исправление выявляет и удаляет существующие неправильные ребра в графе знаний. Здесь мы разделяем основные подходы к исправлению графа знаний на две основные линии: проверка фактов, которая присваивает оценку правдоподобности заданному краю, обычно со ссылкой на внешние источники; и устранение несоответствий, которые направлены на устранение несоответствий, обнаруженных в графе знаний, с помощью онтологических аксиом.

Проверка фактов. Задача проверки фактов (также известная как facts validation) включает в себя присвоение оценок правдоподобия или правдивости фактам / граням, обычно между 0 и 1. Идеальная функция проверки фактов предполагает наличие гипотетического эталонного универсума (идеального графа знаний) и возвращает 1 для факт Santa Lucía city Santiago (будучи истинным), возвращая 0 для Sotomayor city Santiago (будучи ложным). Существует четкая связь между проверкой фактов и прогнозированием связей - и то и другое основано на оценке правдоподобия границ / фактов / связей - и действительно, в обоих случаях могут применяться одни и те же методы, основанные на числах и символах. Однако проверка фактов часто рассматривает онлайн-оценку заданных ребер в качестве входных данных, тогда как прогнозирование связей часто является автономной задачей, которая генерирует новые кандидатные ребра для оценки из графа знаний. Более того, работы по подтверждению фактов характеризуются рассмотрением внешних справочных источников, которые могут быть неструктурированными или структурированными источниками. Подходы, основанные на неструктурированных источниках, предполагают, что им дана функция вербализации - с использованием, например, основанных на правилах подходов, архитектур кодировщика-декодера и т. д. - которая способна переводить ребра на естественный язык. После этого можно напрямую использовать подходы для вычисления правдоподобия фактов на естественном языке - так называемые средства поиска фактов. Многие алгоритмы поиска фактов строят n-разделенный (часто двудольный) граф, узлы которого являются фактами и источниками, где источник связан с фактом, если источник «свидетельствует» об этом факте, т.е. если он содержит соответствующий фрагмент текста - с достаточной уверенностью - вербализация входного ребра. Затем на основе этого графа вычисляются две взаимозависимые оценки, а именно достоверность источников и достоверность фактов, где специалисты по поиску фактов различаются по способам вычисления этих оценок. Здесь мы упоминаем три оценки, предложенные Пастернаком и Ротом:

- *Sums* адаптирует алгоритм HITS, определяя источники как узлы (с 0 баллами авторитетности) и факты как авторитетные источники (с 0 баллами узлов).

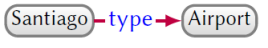
- Журнал усреднений расширяет HITS с помощью коэффициента нормализации, который не позволяет одному источнику получить высокую оценку надежности за счет подтверждения множества фактов (которые могут быть ложными).
- Инвестиции позволяют множеству фактов расти с нелинейной функцией, основанной на «инвестициях», поступающих из подключенных источников. Оценка, которую источник получает за факт, основана на отдельных фактах в этом конкретном источнике по сравнению с другими связанными источниками.

Затем Пастернак и Рот показывают, что эти три алгоритма могут быть обобщены в единую многослойную структуру на основе графов, в которой (1) источник может подтвердить факт с весом, выражающим неопределенность, (2) аналогичные факты могут подтвердить каждый другие, и (3) источники могут быть сгруппированы вместе, что приводит к неявной поддержке между источниками одной и той же группы. Другие подходы к проверке фактов графов знаний позже расширили эту структуру. Также появились альтернативные подходы, основанные на классификаторах, где обычно используемые функции включают в себя оценки доверия к источникам информации, совпадение фактов в источниках и т. д. Подходы к проверке фактов, основанные на структурированных данных, обычно предполагают внешние графы знаний в качестве справочных источников и основаны на поиске путей, свидетельствующих о том, что входное ребро проверяется. Неконтролируемые подходы ищут неориентированные или направленные пути до заданной пороговой длины, которые свидетельствуют о входном крае. Связь между входными ребрами и путями вычисляется с использованием функции взаимной информации, такой как нормализованная точечная взаимная информация. Контролируемые подходы скорее извлекают признаки для входных ребер из внешних графов знаний и используют эти особенности для обучения модели классификации маркировке ребер как истинных или ложных. Важным набором функций являются метапути, которые кодируют последовательности предикатов, которые положительно коррелируют с меткой входного ребра. Среди таких работ PredPath автоматически извлекает мета-пути на основе информации о типе. Некоторые подходы скорее кодируют опорные узлы и ребра, используя вложения графа, которые затем используются для оценки правдоподобия проверяемого входного ребра.

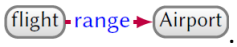
Устранение несогласованностей. Онтологии могут содержать аксиомы, такие как дизъюнктность, которые приводят к несогласованности. Хотя такие аксиомы могут быть предоставлены экспертами, их также можно вывести посредством символического обучения. Такие аксиомы затем можно использовать для обнаружения несоответствий. Однако что касается исправления графа знаний, обнаружения несоответствий недостаточно: также требуются методы для исправления таких несоответствий, что само по себе не является тривиальной задачей. В простейшем случае у нас может быть экземпляр двух непересекающихся классов, например, 

имеет типы 

и 

, которые заявлены или признаны не пересекающимися. Чтобы устранить несоответствие, было бы предпочтительнее удалить только «неправильный» класс, но какой мы должны удалить? Это нетривиальный вопрос, особенно если учесть, что одно ребро может быть связано со многими несогласованностями, а одно несоответствие может включать много ребер. Проблема вычисления ремонта становится более сложной, когда рассматривается влечение, когда нам необходимо удалить не только указанный тип, но также все способы, которыми он мог бы быть вызван; например, удаление ребра 

недостаточно, если мы далее имеем ребро 

в сочетании с аксиомой 

. Töpper et al. предлагают возможные способы устранения таких нарушений - удалить ограничение домена / диапазона, удалить ограничение непересекаемости, удалить границу типа, удалить границу с ограничением домена / диапазона - где одно выбирается вручную. Напротив, Bonatti et al. предлагают автоматизированный метод устранения несоответствий, основанный на минимальных наборах совпадений, где каждый набор является минимальным объяснением несоответствия. Ребра для удаления выбираются на

основе оценок надежности их источников и того, сколько минимальных наборов совпадений они являются элементами или помогают повлечь за собой элемент, при этом граф знаний пересматривается, чтобы избежать повторного возникновения удаленных ребер. Другой вариант - не восстанавливать данные, а оценивать запросы в соответствии с семантикой, учитывающей несогласованность, например возвращать согласованные ответы, действительные при каждом возможном исправлении.

Другие задачи уточнения

По сравнению с характеристиками качества, обсуждаемыми в разделе 7, обсуждаемые здесь методы уточнения обращаются к конкретным аспектам точности, охвата и когерентности. Помимо этого, можно было бы придумать дополнительные методы уточнения для решения дополнительных проблем качества графов знаний, таких как лаконичность. В целом, однако, до сих пор наибольшее внимание уделялось уточняющим задачам завершения графа знаний и исправлению графов знаний.

8. Публикация графов знаний

Хотя может быть нежелательно публиковать, например, графы корпоративных знаний, которые предлагают компании конкурентное преимущество, может быть желательно - или даже требуется - публиковать другие графы знаний, например, созданные добровольцами, финансируемыми государством. исследования, правительственные организации и т. д. Публикация - это обеспечение публичного доступа к графу знаний (или его части), часто через Интернет. Графы знаний, опубликованные как открытые данные, в таком случае называются графами открытых знаний. Далее мы сначала обсудим два набора принципов, которые были предложены для публикации данных в Интернете. Затем мы обсудим протоколы доступа, которые составляют интерфейсы, с помощью которых публика может взаимодействовать с содержимым графа знаний. Наконец, мы рассматриваем методы ограничения доступа или использования (частей) графа знаний, в зависимости от ситуации.

Лучшие Практики

Теперь мы обсудим два ключевых набора принципов публикации данных, а именно принципы FAIR, предложенные Уилкинсоном и др., и принципы связанных данных, предложенные Бернерсом-Ли.

Принципы FAIR. Принципы FAIR были первоначально предложены в контексте публикации научных данных - особенно мотивированных максимизацией воздействия исследований, финансируемых государством, - но эти принципы обычно применимы к другим ситуациям, когда данные должны публиковаться таким образом, чтобы облегчить их повторное использование внешними агентами, с особым упором на машиночитаемость. FAIR сам по себе является аббревиатурой четырех основополагающих принципов, каждый из которых имеет определенные цели, которые могут применяться к данным, метаданным или и тем, и другим, причем последние являются обозначенными (мета) данными. Теперь мы опишем принципы FAIR (немного перефразируя исходную формулировку в некоторых случаях для краткости).

- Под возможностью поиска понимается легкость, с которой внешние агенты, которые могут извлечь выгоду из набора данных, могут изначально найти набор данных. Необходимо достичь четырех подцелей:

  - F1: (мета) данным назначается глобальный уникальный и постоянный идентификатор.

  - F2: данные описываются с помощью расширенных метаданных (см. R1).

- F3: метаданные четко и явно включают идентификатор данных, которые они описывают.

- F4: (мета) данные регистрируются или индексируются в доступном для поиска ресурсе.

- Доступность означает легкость, с которой внешние агенты (после обнаружения набора данных) могут получить доступ к набору данных. Определены две цели, первая с двумя подцелями:

  - A1: (мета) данные можно получить по их идентификатору с использованием стандартного протокола.

  - A1.1: протокол открытый, бесплатный и универсальный.

  - A1.2: протокол позволяет при необходимости аутентификацию и авторизацию.

  - A2. метаданные доступны, даже если данные больше не доступны.

- Функциональная совместимость означает легкость использования набора данных (в сочетании с другими наборами данных) с помощью стандартных инструментов. Определены три цели:

  - I1: (мета) данные используют формализм доступного, совместно используемого и общего представления знаний.

  - I2: (мета) данные используют словари, соответствующие принципам FAIR.

  - I3: (мета) данные включают квалифицированные ссылки на другие (мета) данные.

- Возможность повторного использования означает легкость, с которой набор данных можно повторно использовать в сочетании с другими наборами данных. Определена одна цель (с тремя подцелями):

  - R1: мета (данные) подробно описаны множеством точных и релевантных атрибутов.

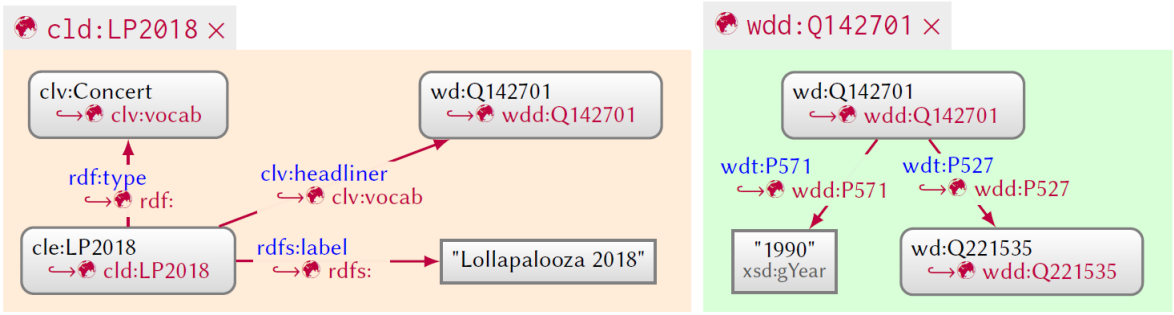
  - R1.1. (мета) данные выпускаются с четкой и доступной лицензией на использование данных.

  - R1.2. (мета) данные связаны с подробным происхождением.

  - R1.3. (мета) данные соответствуют отраслевым стандартам сообщества.

В контексте графов знаний было предложено множество словарей, инструментов и сервисов, которые как прямо, так и косвенно помогают удовлетворить принципы FAIR. Что касается возможности поиска, как обсуждалось в разделе 2, IRI встроены в модель RDF, обеспечивая общую схему для глобальных идентификаторов. Кроме того, такие ресурсы, как Словарь взаимосвязанных наборов данных (VoID), позволяют представлять метаданные о графах, а такие службы, как DataHub, предоставляют центральное хранилище описаний таких наборов данных. Протоколы доступа, обеспечивающие доступность, будут обсуждаться в Разделе 9.2, а механизмы авторизации - в Разделе 9.3. Что касается интероперабельности, как обсуждалось в разделе 4, онтологии служат в качестве общего формализма представления знаний и, в свою очередь, могут использоваться для описания словарей, которые следуют принципам FAIR. Наконец, что касается возможности повторного использования, лицензирование будет обсуждаться в Разделе 9.3, в то время как модель данных PROV, обсуждаемая в Разделе 3, позволяет фиксировать подробное происхождение. Был опубликован ряд графов знаний с использованием принципов ЧЕСТНОЙ ДЕЯТЕЛЬНОСТИ, в которых Wilkinson et al. [555] прямо упоминают Open PHACTS [200], платформу интеграции данных для открытия лекарств, и UniProt [518], большой набор данных по

последовательностям и аннотациям белков, как соответствующие принципам FAIR. Оба набора данных предлагают графовые представления своего содержимого через модель данных RDF.



На фиг. 36. Два примера связанных данных документов с двух сайтов, каждый из которых содержит файл RDF графа, где компания WD:Q142701 относится к Перл джем в Викиданных а WDD по:Q142701 в PCO графе о "Перл Джем", и где WD:Q221535 относится к Эдди Веддер, а WDD по:Q221535 относится к PCO графе об Эдди Веддер; именованная связь компании:571 означает “inception” в Викиданных, в то время как компании wdt:527 относится к “has part”

Принципы Связанных Данных. Wilkinson et al. заявляют, что принципы FAIR «предшествуют выбору реализации», что означает, что принципы не охватывают, как они могут или должны быть достигнуты. Практически на десять лет впереди принципов FAIR находятся Принципы связанных данных, предложенные Бернерсом-Ли, которые обеспечивают техническую основу для одного из возможных способов достижения принципов FAIR. В частности, Принципы связанных данных заключаются в следующем:

(1) Используйте IRI в качестве имен для вещей.

(2) Используйте HTTP IRI, чтобы можно было найти эти имена.

(3) При поиске HTTP IRI предоставьте полезный контент об объекте, который именуется IRI.

с использованием стандартных форматов данных.

(4) Включите в возвращаемый контент ссылки на IRI связанных сущностей.

Эти принципы были предложены в настройке семантической сети, где для принципа (3) в настоящее время рекомендуются для использования стандарты, основанные на RDF (включая RDFS, OWL и т. д.), особенно потому, что они позволяют именовать объекты с использованием HTTP IRI, что дополнительно открывает путь к соблюдению всех четырех принципов. Таким образом, эти принципы определяют способ, которым (RDF) данные с графовой структурой могут быть опубликованы в сети таким образом, чтобы эти графы были связаны друг с другом, чтобы сформировать то, что Бернерс-Ли называет «Сетью данных», цель которой состоит в том, чтобы повысить автоматизацию в Интернете, сделав контент доступным не только в (HTML) документах, предназначенных для человеческого потребления, но и в виде структурированных данных (RDF), которые машины могут находить, извлекать, комбинировать, проверять, анализировать, запрашивать и т. д. для решения задач автоматически. Концептуально сеть данных состоит из графов данных, опубликованных на отдельных веб-страницах, где можно щелкнуть узел или ребро - или, точнее, выполнить HTTP-поиск по IRI графа - для переноса в другой граф где-нибудь в Интернете с релевантным контентом на этом узле или ребре, и так далее рекурсивно. На рисунке 36 мы показываем простой пример с двумя документами связанных данных, опубликованными в Интернете, каждый из которых содержит граф RDF. Как обсуждалось в разделе 3.2, такие термины, как clv: Concert, wd: Q142701, rdfs: label и т. д, являются сокращениями для IRI, где, например, wd:

Q142701 заменяется на http://www.wikidata.org/entity. / Q142701. Префиксы, начинающиеся с cl, - это вымышленные префиксы, которые, как мы предполагаем, были созданы чилийским советом по туризму. IRI с префиксом ⮌→m указывает на документ, возвращаемый при поиске узла. Самый левый документ публикуется туристическим советом и описывает Lollapalooza 2018 (идентифицируется узлом cle:LP2018); который ссылается на акт Pearl Jam (wd:Q142701), описываемый внешним графом знаний, а именно Викиданными. Просматривая узел wd:Q142701 в крайнем левом графе, IRI разыменовывает (то есть возвращает через HTTP) документ с графом RDF справа, описывающим эту сущность более подробно. Из самого правого документа узел wd:Q221535 может быть просмотрен, в свою очередь, чтобы найти граф об Эдди Веддере (не показан в Примере). IRI для объектов и документов различаются, чтобы гарантировать, что мы не путаем данные об объекте и документе; например, в то время как wd: Q221535 относится к Эдди Веддеру, IRI wdd: Q221535 относится к документу об Эдди Веддере; если бы мы должны были назначить документу дату последнего изменения, мы должны использовать wdd:Q221535 не wd:Q221535. На рис. 36 мы также видим, что метки ребер (которые также являются радужной оболочкой) и узлы, представляющие классы (например, clv:Concert), также могут быть разыменованы, как правило, возвращая семантические определения соответствующих терминов. Ключевой проблемой является четвертый принцип – включение ссылок на связанные сущности – как показано на рис. 36, где wd:Q221535 в крайнем левом графе представляет собой ссылку на связанный контент о Pearl Jam во внешнем графе знаний. В частности, задача обнаружения ссылок рассматривает добавление таких ссылок из одного графа знаний в другой, что может включать включение IRI, разыменованных на внешние графы (как показано на рисунке 36), или ссылок со специальной семантикой, такой как идентификационные ссылки. По сравнению с задачей прогнозирования ссылок, обсуждаемой в разделе 8.1, которая используется для завершения ссылок в графе знаний, обнаружение ссылок направлено на обнаружение связей в графах знаний, что включает уникальные аспекты: во-первых, обнаружение ссылок обычно рассматривает непересекающиеся наборы источников (локальных) узлы и целевые (удаленные) узлы; во-вторых, графы знаний могут часто использовать разные словари; в-третьих, в то время как при прогнозировании ссылок уже существуют локальные примеры ссылок для прогнозирования, при обнаружении ссылок часто отсутствуют существующие связи между графами знаний, на которых можно было бы учиться. Распространенным методом является определение созданных вручную правил связывания (также называемых спецификациями ссылок), которые применяют эвристику для определения ссылок, которые потенциально включают меры сходства. Обнаружение ссылок значительно ускоряется за счет предоставления стандартных схем идентификаторов в графах знаний, таких как ISBN для книг, коды альфа-2 и альфа-3 для стран (например, cl, clp) или даже ссылки на общие графы знаний, такие как DBpedia или Викиданные (которые сами включают стандартные идентификаторы). Мы ссылаемся на обзор обнаружения связей, проведенный Nentwig et al. для более подробной информации. Были предложены дополнительные руководящие принципы, которые предоставляют более подробные рекомендации по публикации связанных данных, касающиеся, среди прочего, того, как лучше всего реализовать разыменование, какие виды связей включать, как публиковать и связывать словари. Мы ссылаемся на книгу Хита и Байзера для более подробного обсуждения того, как публиковать связанные данные в Интернете.

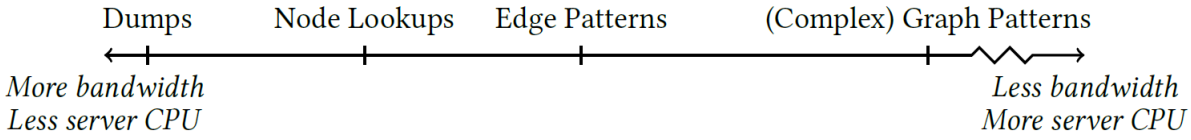
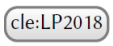


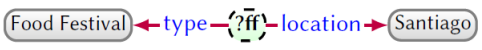
Рис. 37. Протоколы доступа к графам знаний: от простых протоколов (слева) к более сложным протоколам (справа)

Протоколы Доступа

Публикация включает в себя предоставление возможности общественности взаимодействовать с графом знаний, что подразумевает предоставление протоколов доступа, которые определяют запросы, которые могут делать агенты, и ответ, который они могут ожидать в результате. Согласно принципу доступности FAIR (в частности, A1.1), этот протокол должен быть открытым, бесплатным и универсально реализуемым. В контексте графов знаний, как показано на рисунке 37, существует ряд протоколов доступа на выбор: от простых протоколов, которые позволяют пользователям просто загружать весь контент, до протоколов, которые принимают и оценивают все более сложные запросы. В то время как более простые протоколы требуют меньше вычислений на сервере, который публикует данные, более сложные протоколы позволяют агентам запрашивать более конкретные данные, тем самым уменьшая пропускную способность. Граф знаний может также предлагать множество протоколов доступа, обслуживающих разных агентов с разными требованиями. Теперь мы обсудим такие протоколы доступа.

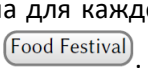
Дамп - это файл или набор файлов, содержащих содержимое графа знаний, доступных для загрузки. Запрос в этом случае предназначен для файла (ов), а ответ - это содержимое файла (ов). Для публикации дампов, прежде всего, требуются конкретные - и в идеале стандартные - синтаксисы для сериализации графа. Хотя для графов RDF доступны различные стандартные синтаксисы, основанные на XML, JSON, пользовательских синтаксисах и т. д., В настоящее время для графов свойств доступны только нестандартные синтаксисы. Во-вторых, для уменьшения пропускной способности могут применяться методы сжатия. В то время как стандартное сжатие, такое как GZIP или BZip2, можно напрямую применить к любому файлу, для графов были предложены специальные методы сжатия, которые не только предлагают лучшие степени сжатия, чем эти стандартные методы, но также предлагают дополнительные функции, такие как компактные индексы для выполнения эффективного поиска. как только файл будет загружен. Наконец, для дальнейшего уменьшения пропускной способности при обновлении графа знаний можно вычислить и опубликовать «различия», чтобы агентам не нужно было загружать все данные с нуля. Тем не менее, дампы подходят только для определенных случаев использования, в частности для агентов, которые хотят поддерживать полную локальную копию графа знаний. Если бы агента интересовали только, например, все кулинарные фестивали в Сантьяго, загрузка всей свалки потребовала бы передачи и обработки большого количества нерелевантных данных.

Node lookups. Протоколы для выполнения поиска узлов принимают запрос узла (id) (например, 

на рис. 36) и возвращают (суб)граф, описывающий этот узел (например, документ cld:LP2018). Такой протокол является основой для принципов связанных данных, описанных ранее, где поиск узлов осуществляется через HTTP-разыменование, что в дальнейшем позволяет ссылаться на узлы в удаленных графах через Интернет. Хотя существуют различные определения того, какое содержимое должно быть возвращено для узла, общим соглашением является возврат подграфа, содержащего либо все исходящие ребра для этого узла, либо все инцидентные ребра (как исходящие, так и входящие) для этого узла. Несмотря на кажущуюся простоту, механизмы для ответа на графовой модели может быть реализован на вершине поиска узла интерфейса, двигаясь от одного узла к другому в зависимости от конкретной графовой схемы; например, найти все фестивали еды в Сантьяго – представлены графом 

– мы можем выполнить узел поиска 

, впоследствии выполнении поиска узла для каждого узла соединены расположение ребра 

, возвращая эти узлы объявляется тип 

. Однако, такой подход не может быть осуществима, если нет начального узла (например, если все узлы

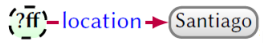
переменные), если узел службы поиска не возвращает входящих ребер и т. д. Кроме того, клиентскому агенту может потребоваться запросить больше данных, чем необходимо, где документ, возвращаемый Для 

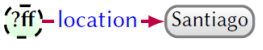
, может вернуть много нерелевантных данных, и где узлы с местоположением в 

, которые не представляют экземпляры 

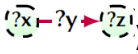
, все еще нуждаются в поиске, чтобы проверить их тип. С другой стороны, поиск узлов относительно недорог для поддержки серверов.

Паттерны ребер. Паттерны ребер - также известные как тройные паттерны в случае ориентированных графов с именованными ребрами - представляют собой одноэлементные графовые паттерны, то есть паттерны графа с одним ребром. Примеры паттернов ребер: 

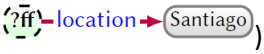
или 

и т. д., где любой член может быть переменной или константой. Протокол для паттернов ребер принимает такой шаблон и возвращает все решения для шаблона. Паттерны ребер обеспечивают большую гибкость, чем поиск узлов, где шаблоны графа легче разлагаются на паттерны ребер, чем поиск узлов. Что касается агента, заинтересованного в фестивалях еды в Сантьяго, они могут сначала, например, запросить решения для паттерна ребер 

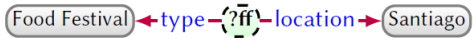
и локально соединить/пересечь эти решения с решениями других стран 

. Учитывая, что некоторые паттерны ребер (например, 

) можгут возвращать множество решений, протоколы для паттернов ребер могут предлагать дополнительные практические функции, такие как итерация или разбиение на страницы результатов. Как и при поиске узлов, стоимость ответа сервера на запрос относительно невысока и ее легко предсказать. Однако серверу часто может потребоваться передать клиенту нерелевантные промежуточные результаты, что в предыдущем примере может включать возвращаемые узлы, расположенные в Сантьяго, которые не являются фестивалями еды. Эта проблема еще больше усугубляется, если у клиента нет доступа к статистике о графе знаний, чтобы спланировать, как лучше всего выполнить соединение; например, если есть относительно немного фестивалей еды, но много вещей расположены в Сантьяго, вместо того, чтобы пересекать решения двух вышеупомянутых шаблонов границ, должно быть более эффективным отправить запрос для каждого фестиваля еды, чтобы узнать, находится ли он в Сантьяго, но для этого требуется статистика о графе знаний. Таким образом, были предложены расширения протокола паттернов ребер для обеспечения более эффективных объединений, таких как разрешение отправки пакетов решений вместе с паттерном ребер, возвращение только решений, совместимых с решениями в запросе (например, отправка пакета решений для 

объединения с решениями для запроса 

).

(Сложные) графовые паттерны. Другая альтернатива-позволить клиентским агентам делать запросы на основе (сложных) графовых паттернов, а сервер возвращает (только) окончательные решения. В нашем запущенном примере это включает в себя отправку клиентом запроса 

и напрямую получать соответствующие результаты. По сравнению с предыдущими протоколами этот протокол намного эффективнее с точки зрения пропускной способности: он позволяет клиентам делать более конкретные запросы, а серверу возвращать более конкретные ответы. Однако это сокращение использования полосы пропускания происходит за счет того, что серверу приходится оценивать гораздо более сложные запросы, где, кроме того, гораздо труднее предвидеть стоимость одного запроса. Хотя существует множество оптимизированных механизмов для оценки (сложных) графовых шаблонов, проблема оценки таких запросов, как известно, является неразрешимой. Возможно, по этой причине было обнаружено, что общедоступные службы, предлагающие такой протокол (чаще всего поддерживающие запросы SPARQL), часто демонстрируют простои, тайм-ауты, частичные результаты, низкую производительность и т. д. Однако даже с учетом таких проблем популярные службы продолжают

получать - и успешная оценка - миллионы запросов / запросов в день, при этом сложные (наихудшие) случаи на практике встречаются редко.

Другие протоколы. Хотя на рис. 37 явно указаны некоторые из наиболее часто встречающихся протоколов доступа, встречающихся на практике для графов знаний, можно, конечно, представить другие протоколы, лежащие в диапазоне от более простых до более сложных интерфейсов. Справа от (сложных) графовых шаблонов можно было бы рассмотреть возможность поддержки еще более сложных запросов, таких как запросы со следствием, запросы, допускающие рекурсию, федеративные запросы, которые могут объединять результаты из удаленных сервисов, или даже (гипотетически) поддерживающие запросы, завершенные по Тьюрингу, которые разрешить запуск произвольного процедурного кода на графе знаний. Как упоминалось вначале, сервер также может поддерживать несколько дополнительных протоколов.

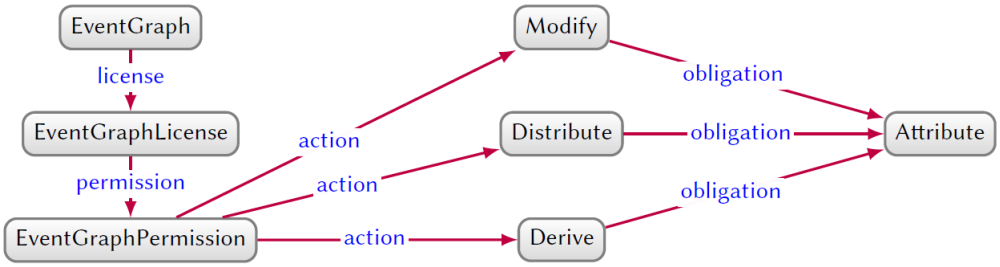


Рис. 38. Связывание лицензий с данными о событиях, а также с разрешениями, действиями и обязательствами

Контроль Использования

Рассматривая наш гипотетический граф знаний о туризме, на первый взгляд, можно предположить, что знания, необходимые для предоставления предполагаемых услуг, являются общедоступными и, следовательно, могут использоваться как советом по туризму, так и туристами. Однако при более внимательном рассмотрении мы можем увидеть необходимость контроля использования в различных формах: (i) и совет по туризму, и его партнеры должны связать соответствующую лицензию для знаний, которые они вносят в граф знаний, таким образом, чтобы условия использования были понятен всем сторонам; (ii) турист может выбрать установку на свой мобильный телефон приложения, которое можно использовать для рекомендации туристических достопримечательностей в зависимости от их местоположения, что может повлечь за собой потенциальные проблемы с конфиденциальностью; (iii) от совета по туризму может потребоваться сообщить в полицию о преступной деятельности и, следовательно, может потребоваться зашифровать личную информацию; и (iv) совет по туризму потенциально может обмениваться информацией, касающейся демографии туризма, в анонимном формате, что позволит улучшить транспортную инфраструктуру на стратегических маршрутах. Таким образом, в этом разделе мы исследуем состояние дел с точки зрения лицензирования графа знаний, политик использования, шифрования и анонимности.

Лицензирование. Когда речь заходит об ассоциировании машиночитаемых лицензий с графами знаний, язык открытых цифровых прав W3C (ODRL) предоставляет информационную модель и связанные с ней словари, которые можно использовать для определения разрешений, обязанностей и запретов в отношении действий, связанных с активами. ODRL поддерживает детальные описания цифровых прав, которые представлены как – и, таким образом, могут быть встроены в – графы. Рисунок 38 иллюстрирует лицензию, предоставляющую цессионарию разрешение Modify, Distribute и Derive работу EventGraph (например, рисунок 1); однако цессионарий обязан Attribute правообладателю. С точки зрения моделирования ODRL может использоваться для

моделирования нескольких известных семейств лицензий, например Apache, Creative Commons (CC) и Berkeley Software Distribution (BSD), и это лишь некоторые из них. Кроме того, Cabrio et al. предлагают методы автоматического извлечения машиночитаемых лицензий из неструктурированного текста. С точки зрения рассуждения, методы проверки совместимости лицензий и композиции могут использоваться для объединения графов знаний, управляемых различными лицензиями. Такие методы используются центром оформления лицензий на доступ к данным (DALICC), который включает библиотеку стандартных машиночитаемых лицензий и инструменты, позволяющие пользователям как составлять произвольные пользовательские лицензии, так и проверять совместимость различных лицензий.

Политика использования. Политики управления доступом на основе граничных шаблонов могут использоваться для ограничения доступа к частям графа знаний. WebAccessControl (WAC) - это структура управления доступом для графов, которая использует WebID для аутентификации и предоставляет словарь для определения политик управления доступом. Расширение этого словаря WAC было предложено для учета предпочтений конфиденциальности и учета контекстуальных ограничений. Хотя ODRL в основном используется для определения лицензий, профили для определения политики доступа и нормативных обязательств также были предложены в последние годы, как описано в обзоре Kirrane et al.

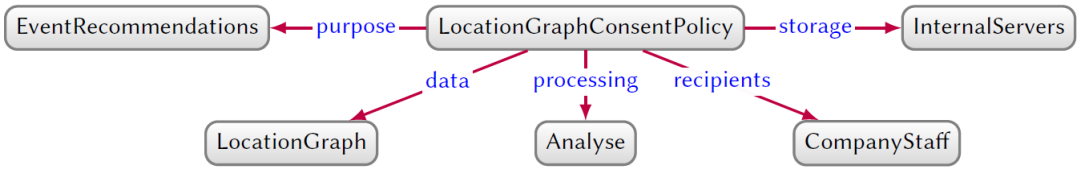


Рис. 39. политика использования подграфа данных о местоположении в графе знаний

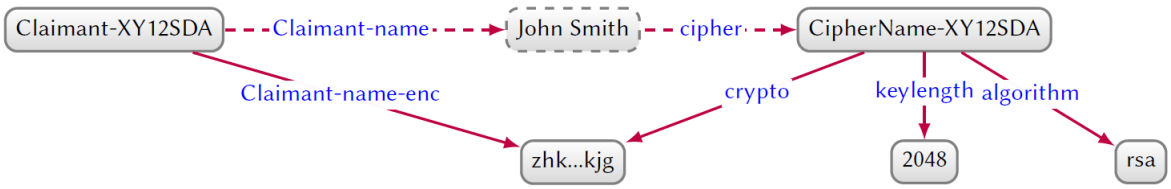


Рис. 40. Направленный граф ребер с зашифрованным именем заявителя; элементы открытого текста пунктирны и могут быть опущены из опубликованных данных (возможно, вместе с деталями шифрования)

В качестве обобщения политик доступа, политики использования определяют, как данные могут быть использованы: какие виды обработки могут быть применены, кем, с какой целью и т. д. В примере политики использования, представленном на рисунке 39, говорится, что процесс Analyse для LocationGraph может быть выполнен InternalServers членами CompanyStaff для обеспечения EventRecommendations. Словари для политик использования были предложены специальным проектом H2020 и группой сообщества W3C Data Privacy Vocabularies and Controls (DPVCG). После уточнения политики использования могут быть использованы для проверки соответствия обработки данных правовым нормам и согласию, предоставленному субъектами.

Шифрование. Вместо внутреннего контроля за использованием, совет по туризму может использовать механизмы шифрования на части опубликованного графа знаний, например, касающиеся сообщений о преступлениях, и предоставлять ключи партнерам, которые должны иметь доступ к открытому тексту. В то время как простой подход заключается в шифровании всего графа (или подграфов) одним ключом, более тонкое шифрование может быть выполнено для

отдельных узлов или ребер в графе, потенциально предоставляя разным клиентам доступ к различной информации через разные ключи. Криптоантология может быть использована для внедрения подробных сведений о механизме шифрования, используемом в графе знаний. На рис. 40 показано, как это может быть использовано для шифрования имен заявителей из рис. 34, хранения зашифрованного zhk...kjg текста, а также используемой длины ключа и алгоритма шифрования. Чтобы предоставить доступ к открытому тексту, один из подходов заключается в шифровании отдельных ребер симметричными ключами, чтобы позволить определенным типам шаблонов ребер выполняться только клиентами с соответствующим ключом. Этот подход может быть использован, например, для того, чтобы позволить клиентам, которые знают идентификатор заявителя (например, Claimant-XY12SDA) и имеют соответствующий ключ, найти (только) имя заявителя через шаблон edge Claimant-XY12SDA -Claimant-name→ ?name. Ключевым ограничением этого подхода, однако, является то, что он требует попыток расшифровать все ребра, чтобы найти все возможные решения. Более эффективной альтернативой является сочетание функционального шифрования и специализированного индексирования для извлечения решений из зашифрованного графа без попыток расшифровать все ребра.

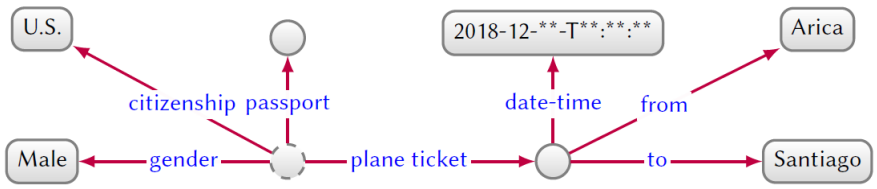


Рис. 41. Анонимизированная выборка направленного графа ребер, описывающего пассажира (пунктир) рейса

Анонимизация. Учтите, что совет по туризму получает информацию о транспорте, которым пользуются люди внутри страны, что может быть использовано для понимания траекторий, используемых туристами. Однако с точки зрения защиты данных было бы целесообразно удалить любые личные данные из графа знаний, чтобы избежать утечки информации о поездках каждого человека. Первый подход к анонимности состоит в том, чтобы скрыть и обобщить знания в графе таким образом, чтобы люди не могли быть идентифицированы на основе k-анонимности, l-разнообразия и т. Д. Подходы к применению-анонимности на графах идентифицируют и подавляют «квазиидентификаторы», позволит отличить данного человека от менее чем k - 1 других людей. На рисунке 41 показан возможный результат k-анонимизации для подграфа, описывающего пассажира рейса, где квазиидентификаторы (паспорт, билет на самолет) были преобразованы в пустые узлы, гарантируя, что пассажира (пунктирный пустой узел) невозможно отличить. от k - еще 1 человек. Однако в контексте графа атаки по соседству - с использованием информации о соседях - также могут нарушить k-анонимность, где мы также подавляем день и время полета, что, хотя и не является конфиденциальной информацией как таковой, в противном случае могло бы нарушить-анонимность для пассажиров (если, например, на конкретном рейсе было меньше мужчин из США). Более сложные атаки на соседство могут полагаться на более абстрактные графовые шаблоны, учитывая, что люди могут быть деанонимизированы исключительно на основании знания структуры графа, даже если все узлы и метки ребер оставлены пустыми; например, если мы знаем, что команда из - 1 игроков совершает полеты вместе для определенного количества выездных игр, мы могли бы использовать эту информацию для атаки по соседству, которая показывает набор игроков на графе. Поэтому был предложен ряд гарантий, характерных для графов, в том числе анонимность-степени, которая гарантирует, что злоумышленники не могут деанонимизировать людей, зная степень анонимности. Подход основан на минимальной модификации графа, чтобы гарантировать, что каждый узел имеет не менее k −1 других узлов с той же степенью. Более строгая

гарантия, называемая k-изоморфной анонимностью соседа, позволяет избежать атак на окрестности, когда злоумышленник знает, как человек связан с узлами в своем районе; это делается путем изменения графа, чтобы гарантировать, что для каждого узла существует по крайней мере k - 1 узел с изоморфными (т. е. идентично структурированными) окрестностями в другом месте графа. Оба подхода защищают от злоумышленников только со знанием ограниченных окрестностей. Еще более сильным понятием является понятие k-автоморфизма, которое гарантирует, что для каждого узла он структурно неотличим от k - 1 других узлов, что позволяет избежать любой атаки, основанной на структурной информации (как тривиальный пример, k-клика или -цикл удовлетворяют k-автоморфизму). Многие из этих методов анонимизации данных графа изначально были мотивированы социальными сетями, хотя они также могут быть применены к графам знаний, в соответствии с работой Лин и Трипунитара, которые адаптировали k-автоморфизм для ориентированных графов с метками ребер (в частности, графов RDF). В то время как вышеупомянутый подход к анонимности данных, второй подход заключается в применении анонимности при ответах на запросы, таких как добавление шума к решениям таким образом, чтобы сохранить конфиденциальность. Один из подходов заключается в применении ε-дифференциальной конфиденциальности для запросов к графам. Такие механизмы обычно используются для агрегированных запросов (например, подсчета), куда добавляется шум, чтобы избежать утечки информации о людях. Чтобы проиллюстрировать, дифференциальная конфиденциальность может позволить подсчитать количество пассажиров определенных национальностей, выполняющих определенные рейсы, добавив (ровно столько) случайного шума к подсчету, чтобы гарантировать, что мы не можем сказать с определенной вероятностью (контролируемой), действительно ли конкретный человек совершил полет, при котором нам потребовалось бы (пропорционально) меньше шума для представителей обычных национальностей, но больше шума, чтобы «спрятать» людей из более необычных национальностей. Эти подходы требуют потери информации для более надежных гарантий конфиденциальности; выбор, таким образом, сильно зависит от приложения. Если анонимные данные должны быть опубликованы в полном объеме «на свалке», то подход, основанный на-анонимности, может использоваться для защиты отдельных лиц, а l-разнообразие может использоваться для защиты групп. С другой стороны, если данные должны быть доступны частично через интерфейс запросов, то ε-дифференциальная конфиденциальность является более подходящей структурой.

9. Практические примеры графов знаний

В этом разделе мы обсудим некоторые из наиболее известных графов знаний, появившихся за последние годы. Мы начнем с обсуждения графов открытых знаний, которые были опубликованы в Интернете в соответствии с руководящими принципами и протоколами, описанными в разделе 9. Позже мы обсудим графы корпоративных знаний, которые были созданы компаниями для различных приложений.

Открытые Графы Знаний

Под графами открытых знаний мы, в частности, ссылаемся на графы знаний, опубликованные в соответствии с философией открытых данных, а именно, что «открытые» означают, что любой может свободно получать доступ, использовать, изменять и делиться для любых целей (в лучшем случае с учетом требований, сохраняющих происхождение и открытость). Многие графы открытых знаний были опубликованы в форме связанных открытых наборов данных, которые представляют собой (RDF) графы, опубликованные в соответствии с принципами связанных данных и философией открытых данных. Многие из наиболее известных графов открытых знаний, включая DBpedia, YAGO, Freebase и Wikidata, охватывают несколько доменов, представляя широкий спектр сущностей и взаимосвязей; мы сначала обсудим их по очереди. Позже мы обсудим некоторые другие

(конкретные) области, для которых в настоящее время доступны графы открытых знаний. Большинство графов открытых знаний, которые мы обсуждаем в этом разделе, смоделированы в RDF, опубликованы в соответствии с принципами связанных данных и предлагают доступ к своим данным через дампы (RDF), поиск узлов (связанные данные), шаблоны графов (SPARQL) и, в некоторых случаях, футляры, кромочные узоры (фрагменты тройного узора).

DBpedia. Проект DBpedia был разработан для извлечения графового представления полуструктурированных данных, встроенных в статьи Википедии, что позволяет интегрировать, обрабатывать и запрашивать эти данные унифицированным способом. Полученный граф знаний дополнительно обогащается за счет ссылок на внешние открытые ресурсы, включая изображения, веб-страницы и внешние наборы данных, такие как DailyMed, DrugBank, GeoNames, MusicBrainz, New York Times и WordNet. Среда извлечения DBpedia состоит из нескольких компонентов, соответствующих абстракциям источников статей Википедии, местам хранения и сериализации графов, экстракторам вики-разметки, синтаксическим анализаторам и менеджерам извлечения. Конкретные экстракторы предназначены для обработки меток, рефератов, межъязыковых ссылок, изображений, перенаправлений, страниц с разрешением неоднозначности, внешних ссылок, внутренних ссылок на страницы, домашних страниц, категорий и геокоординат. Контент в графе знаний DBpedia является не только мультидоменным, но и многоязычным: по состоянию на 2012 год DBpedia содержала метки и аннотации на 97 различных языках. Сущности в DBpedia классифицируются с использованием четырех различных схем, чтобы соответствовать различным требованиям приложений [48]. Эти схемы включают в себя представление категорий Википедии в простой системе организации знаний (SKOS), схему классификации еще одной великой онтологии (YAGO) (обсуждается ниже), схему категоризации онтологии верхнего уровня сопоставления и связывания (UMBEL) и настраиваемую схему. Схема называется онтологией DBpedia с такими классами, как Person, Place, Organization и Work. DBpedia также поддерживает синхронизацию в реальном времени, чтобы соответствовать динамической статье в Википедии.

Yet Another Great Ontology. YAGO также извлекает из Википедии данные с графовой структурой, которые затем объединяются с иерархической структурой WordNet для создания «легкой и расширяемой онтологии с высоким качеством и охватом». Этот граф знаний предназначен для применения в различных задачах информационных технологий, таких как машинный перевод, устранение неоднозначности слов, расширение запроса, классификация документов, очистка данных, интеграция информации и т. д. В то время как более ранние подходы автоматически извлекали структурированные знания из текста с использованием сопоставления с образцом, естественного языковой обработки (НЛП) и статистического обучения, получаемый в результате контент имел тенденцию быть не качественным по сравнению с тем, что было возможно при ручном построении. Однако создание вручную обходится дорого, что затрудняет достижение широкого охвата и поддержание актуальности данных. Чтобы извлечь данные с высоким охватом и качеством, YAGO (как и DBpedia) в основном извлекает данные из информационных ящиков Википедии и страниц категорий, которые содержат основную информацию об объектах и списки статей для конкретной категории, соответственно; они, в свою очередь, объединены с иерархическими концепциями WordNet. Схема, называемая моделью YAGO, предоставляет словарь, определенный в RDFS; эта модель позволяет представлять слова как сущности, улавливая синонимию и двусмысленность. Модель также поддерживает реификацию, n-арные отношения и типы данных. Механизмы уточнения, используемые в YAGO, включают канонизацию, при которой каждое ребро и узел сопоставляются с уникальным идентификатором, а повторяющиеся элементы удаляются, и проверку типов, при которой удаляются узлы, которые не могут быть присвоены классу дедуктивными или индуктивными методами. В последующие годы YAGO будет расширен для поддержки пространственно-временного контекста и многоязычных Википедий.

Freebase. Freebase представляла собой общий набор человеческих знаний, направленных на решение некоторых крупномасштабных проблем интеграции информации, связанных с децентрализованной природой семантической сети, таких как неравномерное принятие, проблемы реализации и ограничения производительности распределенных запросов. В отличие от DBpedia и YAGO, которые в основном взяты из Википедии / WordNet, Freebase запрашивала материалы непосредственно у редакторов-людей. В платформу Freebase было включено масштабируемое хранилище данных с механизмами управления версиями; хранилище больших объектов данных (LOB) для хранения текстовых, графических и мультимедийных файлов; API, который можно запрашивать с помощью языка запросов Metaweb (MQL); пользовательский веб-интерфейс; и легкая система набора текста. Последняя система набора текста была разработана для поддержки совместных процессов. Вместо того, чтобы навязывать онтологическую корректность или логическую последовательность, система была реализована как свободный набор механизмов структурирования, основанных на типах данных, семантических классах, свойствах, определениях схем и т. д., Которые позволяли несовместимым типам и свойствам сосуществовать одновременно. Контент может быть добавлен в Freebase в интерактивном режиме через веб-интерфейс пользователя или автоматически, используя функции записи API. Freebase была приобретена Google в 2010 году, а содержимое Freebase стало важной частью сети знаний Google, анонсированной в 2012 году. Когда Freebase стала доступной только для чтения в марте 2015 года, граф знаний содержал более трех миллиардов ребер. Большая часть этого контента впоследствии была перенесена в Викиданные.

Wikidata. Используемая DBpedia и YAGO, Википедия содержит множество полуструктурированных данных, встроенных в информационные блоки, списки, таблицы и т. д. Однако эти данные традиционно собирались и обновлялись вручную в разных статьях и на разных языках; например, гол, забитый чилийским футболистом, может потребовать ручного обновления статьи игрока, статьи турнира, статьи команды, списков лучших бомбардиров и т. д. на сотнях языковых версий. Ручное курирование привело к множеству проблем с качеством данных, включая противоречивые данные в разных статьях, на разных языках и т. д. Фонд Викимедиа, таким образом, предложил Викиданные в качестве централизованного, совместно редактируемого графа знаний для снабжения Википедией - и произвольными другими клиентами - данными. Согласно этому видению факт может быть добавлен в Викиданные один раз, что приведет к автоматическому обновлению потенциально множества статей в Википедии на разных языках. Как и Википедия, Викиданные также считаются вторичным источником, содержащим утверждения, которые должны ссылаться на первичные источники, хотя утверждения также могут быть изначально добавлены без ссылки. Викиданные также допускают различные точки зрения с точки зрения потенциально противоречивых (упоминаемых) утверждений. Викиданные являются многоязычными, где узлам и ребрам назначаются не зависящие от языка коды Qxx и Pxx (см. Рисунок 36), которые впоследствии связываются с метками, псевдонимами и описаниями на различных языках, что позволяет отображать утверждения на этих языках. Совместное редактирование разрешено не только на уровне данных, но и на уровне схемы, позволяя пользователям добавлять или изменять упрощенные семантические аксиомы, включая подклассы, подсвойства, обратные свойства и т. Д., А также формы. Викиданные предлагают различные протоколы доступа и получили широкое распространение, используются Википедией для создания информационных ящиков в определенных доменах, поддерживаются Google и используются в качестве источника данных для известных приложений, таких как Siri от Apple и других.

Другие открытые междоменные графы знаний. За прошедшие годы был разработан ряд других междоменных графов знаний. BabelNet - аналогично YAGO - основан на объединении WordNet и Википедии, но с интеграцией дополнительных графов знаний, таких как Викиданные, и сосредоточен на создании графа знаний многоязычных лексических форм (организованных в многоязычные синсеты) путем преобразования лексикографических данных. ресурсы, такие как

Wiktionary и OmegaWiki, в графы знаний. По сравнению с другими графами знаний, лексикализованные графы знаний, такие как BabelNet, объединяют энциклопедическую информацию, содержащуюся в Википедии, с лексикографической информацией, обычно содержащейся в одноязычных и двуязычных словарях. Проект Cyc направлен на кодирование здравого смысла в машиночитаемом виде, в котором с 1986 года было потрачено более 900 человеко-лет на создание 2,2 миллиона фактов и правил. Хотя Cyc является частной собственностью, было опубликовано открытое подмножество OpenCyc, в котором мы ссылаемся на сравнение Färber et al. DBpedia, Freebase, OpenCyc и YAGO для получения дополнительных сведений. Проект «Never Ending Language Learning» (NELL) с 2010 года извлек граф из 120 миллионов ребер из текста веб-страниц с помощью методов МЭБ (см. Раздел 6). Каждый такой граф открытых знаний применяет разные комбинации языков и методов, обсуждаемых в этой статье, в разных источниках с разными результатами.

Отраслевые графы открытых знаний. Графы открытых знаний были опубликованы в различных конкретных областях. Schmachtenberg et al. идентифицируют наиболее известные домены в контексте связанных данных следующим образом: СМИ, относящиеся к новостям, телевидению, радио и т. д. (Например, BBC World Service Archive); правительство, связанное с публикацией данных для обеспечения прозрачности и развития (например, правительствами США и Великобритании); публикации, относящиеся к академической литературе по различным дисциплинам (например, OpenCitations, SciGraph, Microsoft Academic Knowledge Graph); географические, относящиеся к интересующим местам и регионам (например, LinkedGeo-Data); науки о жизни, относящиеся к белкам, генам, лекарствам, болезням и т. д. (например, Bio2RDF); а также пользовательский контент, относящийся к обзорам, проектам с открытым исходным кодом и т. д. (например, Revyu). Графы открытых знаний также были опубликованы в других областях, включая культурное наследие, музыку, право, теологию и даже туризм. Предполагаемые приложения для таких графов знаний столь же разнообразны, как и домены, из которых они исходят, но часто связаны с интеграцией, рекомендациями, прозрачностью, архивированием, децентрализацией, многоязычной поддержкой, соблюдением нормативных требований и т. Д.

Графы Корпоративных Знаний

Различные компании объявили о создании проприетарных «графов корпоративных знаний» для различных целей, в том числе: улучшение возможностей поиска, предоставление рекомендаций пользователям, внедрение диалоговых / личных агентов, улучшение целевой рекламы, расширение возможностей бизнес-аналитики, подключение пользователей , расширение многоязычной поддержки, содействие исследованиям и открытиям, оценка и снижение рисков, отслеживание новостных событий и повышение автоматизации транспорта, среди (многих) других. Хотя эти графы корпоративных знаний весьма разнообразны, они соответствуют некоторым общим тенденциям, что отражено в обсуждении Ноя и др.: (1) данные обычно интегрируются в граф знаний из множества как внешних, так и внутренних источников (часто включающих текст); (2) граф корпоративных знаний часто бывает очень большим, с миллионами или даже миллиардами узлов и ребер, что создает проблемы с точки зрения масштабируемости; (3) уточнение исходного графа знаний - добавление новых ссылок, объединение повторяющихся сущностей и т. Д. - важно для повышения качества; (4) методы поддержания графа знаний в актуальном состоянии в предметной области часто имеют решающее значение; (5) сочетание онтологических представлений и представлений машинного обучения часто комбинируется или используется в разных ситуациях, чтобы сделать выводы из графа корпоративных знаний; (6) используемые онтологии имеют тенденцию быть легковесными, часто простыми таксономиями, представляющими иерархию классов или концепций. Теперь мы обсудим основные отрасли, в которых были развернуты графы корпоративных знаний.

Поиск в интернете. Системы веб-поиска традиционно сосредоточены на сопоставлении строки запроса с подстроками в веб-документах. Сеть знаний Google скорее продвигала парадигму «вещи, а не строки» - по аналогии с семантическим поиском - где поисковая система теперь будет пытаться идентифицировать сущности, к которым может проявлять интерес конкретный поиск. Сам граф знаний описывает эти сущности и то, как они взаимосвязаны. Одним из основных пользовательских приложений Сети знаний Google является «Панель знаний», которая представляет собой панель с правой стороны (некоторых) результатов поиска, описывающую основную сущность, которую, по-видимому, ищет поиск, включая некоторые изображения, пары атрибут-значение и список связанных сущностей, которые также ищут пользователи. Сеть знаний Google сыграла ключевую роль в популяризации современного использования фразы «граф знаний» (см. Приложение А). Другие основные поисковые системы, такие как Microsoft Bing, позже объявят о графах знаний аналогичным образом.

Коммерция. Графы корпоративных знаний также были анонсированы компаниями, которые в основном занимаются продажей или арендой товаров и услуг. Ярким примером такого графа знаний является тот, который используется Amazon, который описывает продукты, продаваемые на их онлайн-рынке. Одна из основных заявленных целей этого графа знаний - включить более продвинутые (семантические) функции поиска для продуктов, а также улучшить рекомендации по продуктам для пользователей его онлайн-рынка. Еще один граф знаний для торговли был анонсирован eBay, который кодирует описания продуктов и модели покупательского поведения и используется для поддержки диалоговых агентов, которые помогают пользователям находить соответствующие продукты через интерфейс на естественном языке. Airbnb также описал граф знаний, который кодирует жилье для аренды, места, события, впечатления, окрестности, пользователей, теги и т. д., поверх которого определяется таксономическая схема. Этот граф знаний используется для предложения потенциальным клиентам рекомендаций о достопримечательностях, событиях и мероприятиях, доступных в окрестностях конкретного дома для аренды. Uber также анонсировал диаграмму знаний о еде и ресторанах для своей службы доставки Uber Eats. Цели снова заключаются в том, чтобы предложить функции семантического поиска и рекомендации пользователям, которые не уверены, какую именно еду они ищут.

Социальные сети. Графы корпоративных знаний также появились в контексте сервисов социальных сетей. Facebook собрал воедино граф знаний, описывающий не только социальные данные о пользователях, но и объекты, которые им интересны, включая знаменитостей, места, фильмы, музыку и т. д., чтобы связывать людей, понимать их интересы и давать рекомендации. LinkedIn анонсировала граф знаний, содержащий пользователей, должности, навыки, компании, места, школы и т. д., поверх которого определена таксономическая схема. Граф знаний используется для обеспечения многоязычного перевода важных концепций, улучшения целевой рекламы, предоставления расширенных функций для поиска работы и поиска людей, а также для предоставления рекомендаций по подбору вакансий для людей (и наоборот). Pinterest был создан еще один граф знаний, описывающий пользователей и их интересы, причем последние организованы в таксономию. Основные варианты использования графа знаний - помочь пользователям легче находить интересующий их контент, а также увеличить доход за счет целевой рекламы.

Финансы. В финансовом секторе также были внедрены графы корпоративных знаний. Среди них Bloomberg предложил граф знаний, который поддерживает аналитику финансовых данных, включая анализ настроений компаний на основе текущих новостных отчетов и твитов, службу ответов на вопросы, а также обнаружение новых событий, которые могут повлиять на стоимость акций. Thompson Reuters (Refinitiv) аналогичным образом анонсировала граф знаний, кодирующий «финансовую экосистему» людей, организаций, долевых инструментов, отраслевых классификаций, совместных предприятий и альянсов, цепочек поставок и т. д., С использованием

таксономической схемы для организации этих структур. Некоторые из приложений, которые они упоминают для графа знаний, включают мониторинг цепочки поставок, оценку рисков и инвестиционные исследования. Графы знаний также были исследованы в академической среде с помощью Banca d'Italia, с использованием основанных на правилах аргументов для определения, например, доли владения компанией различными заинтересованными сторонами. К другим компаниям, изучающим графы финансовых знаний, относятся Accenture, Capital One, Wells Fargo и другие.

Другие отрасли. Предприятия также активно разрабатывают графы знаний, чтобы использовать новые приложения во множестве других отраслей, включая: здравоохранение, где IBM изучает варианты использования для открытия лекарств и извлечения информации из вкладышей в пакеты, а AstraZeneca использует граф знаний для продвигать исследования в области геномики и понимание болезней; транспорт, где Bosch изучает граф знаний сцен и мест для автоматизации вождения; нефть и газ, где Maana использует графы знаний для интеграции данных для снижения рисков, связанных с нефтяными скважинами и бурением; и многое другое кроме того.