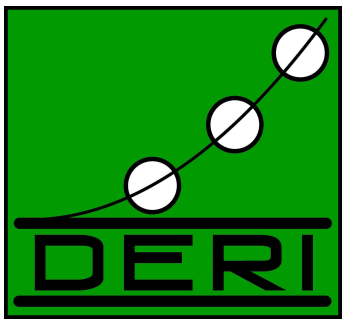


DERI – ІНСТИТУТ ЦИФРОВОГО ДОСЛІДЖЕННЯ ПІДПРИЄМСТВА

TOPDIS: Tensor-based Займає визначне місце для пошуку Даних і Навігаційний

DERI Технічне повідомлення 2009-06-21
червень 2009

DERI Galway
IDA Business Park
Galway, Ірландія
www.deri.ie

DERI – ЦИФРОВИЙ ІНСТИТУТ ДОСЛІДЖЕННЯ ПІДПРИЄМСТВА

DERI Технічне Повідомлення
DERI Технічне Повідомлення 2009-06-21, червень 2009

TOPDIS: Tensor-based, що займає значне місце для пошуку Даних і Навігації

Андреа Харт Шила Кінсела

Абстрактний. Мережеві джерела все більш і більш використовують Формат (RDF) для опису ресурсу як засоби для універсального представлення знань. Багато сайтів зараз доповнюються із структурованим представленням їх вмісту; наприклад, вміст блогів, соціальних мережевих сайтів і Вікіпедія є доступною в RDF і може залучатися. Шукаючи великими масивами структурованої інформації, взятої з Павутини. Нарешті, ми вводимо TOPDIS, як набір алгоритмічних інструментів, щоб визначити видатні елементи у великих семантичних графах. Ми забезпечуємо формалізацію методу, використовуючи поняття мультилінійпої алгебри, оцінючи масштабованість алгоритму і якість результату,показуючи показують, як TOPDIS може поліпшити пошук структурованих даних взагалі.

Визнання: Ця робота підтримується ірландським проектним Науковим Фондом (SFI/02/CE1/I131)

1 Введення

Придатність масивних сум структурованих даних на Павутині, опублікованих в стандартному форматі, має нові шляхи для отримання інформації, що шукається. Все більш і більш Мережеві джерела використовують Формат Опису Ресурсу (RDF) щоб подати структурне представлення їх вмісту. Доступні джерела включають великі монолітні набори даних, як наприклад DBLP (Публікація комп'ютерної інформації), Swissprot (База даних), (Вікіпедіяв RDF), TAP (універсальна база знань), Geonames (географічна інформація), також як і великий ряд маленьких файлів. Видавці даних, можливо, мають різний словесний вигляд і всілякі фони, які призводять строкатий ассортимент ідентифікаторів для зразків, властивостей, і класів. Хоча декілька стандартних словників (як наприклад FOAF для людей, RSS/Atom і SIOC, для оперативних суспільств, і SKOS, для схем класифікації) почало з'являтися через соціальні процеси, простір мережевої інформації. Іншими словами: як приводити замовлення до в значній мірі невідповідних вкладень величезного ряду людей?

Подача лише видатних або доречних альтернатив на розгляд користувача важлива з тих пір, як короткострокова пам'ять в людського мозку є обмеженою. Оскільки мережеві дані не організовується в передбаченій структурі, але є багато альтернатив, щоб використовувати в запиті, семантичні зв'язки, щоб пройти в інформаційному просторі. Тому, двигун пошуку бази буття для структурованих даних повинен забезпечити введену конструкцію запиту, де альтернативи, подані на розгляд користувачів, влаштовані на доцільності їх використання. Отримавши владу поширення і вільного координування моделюючих даних, зазвичай юридичні особи описують через ряд мережевих джерел і тому сума даних, що має відношення до однієї юридичної особи, можливо, стає великою. Результат для зазначеного пошукового терміну або запиту, можливо, є в замовленні сотень або тисячах звітів – що є дуже далеким, для того щоб користувач швидко зорієнтувався. Тому, сторінка з результатами повинна показати лише доречну відповідь. Як засоби скорочують пізнавальний вантаж користувача, так і дозволяють збільшення інформаційний.

Завдання, до якого ми звертаємося, важке із-за трьох характеристик мережевих даних: інезалежності області і масштабу. Дані на Павутині можуть покрити багато області як соціальні мережі (наприклад від соціальних мережевих сайтів), енциклопедичне знання (наприклад Вікіпедія), і наукові набори даних (наприклад наукова література, бази даних). Коли набір даних під експертизою відомі закладу, дослідники мають комфортне знаряддя, вибираючи певні властивості набору даних і розвиваючи алгоритми, які приймають другорядне знання експерта в області уваги. Дані на Павутині не загальноприйнятий юридичним особам, можливо, описуються в сотнях, або в сотнях тисяч звітів. Робота ця стимулююча. Тому, ми переконалися, що наш метод може бути застосований до дуже великих наборів даних.

Проблема, до якої ми звертаємося, абсолютно свіжа і унікальна до інтеграції даних на Павутині. Архітектура Павутини дозволяє нам досягати нових вимірів в термінах масштабу і складності інтеграції даних, яка у свою чергу приносить про нові підходи для ранжирування.

Документально; двигуни мережевого пошуку зазвичай включають алгоритми, що діють на єднальному графікові, як наприклад PageRank [22], і HITS[13], щоб сортувати документи згідно доцільності. Проте методи, що розвиваються для зв'язку документа, не підсилюють повної можливості, місце яких зайняло, на даних, які зазвичай додають здатність використовувати надруковані зв'язки між вузлами. Попередні підходи для ранжирування структурованих даних були застосовані в області бази даних, де схеми, що є посередником, ручних робіт і відносно скромним за розміром; тому, призначаючи вагу різним єднальним типам все ще виконується (наприклад дивляться [12]). Інший аспект, абсолютно новий, інтеграція даних великого масштабу - та походження даних. Дослідження походження повідомлень важливе на Павутині, де хто-небудь може говорити що-небудь про все. Ми показуємо, як включати поняття джерел даних і походженняв нашій процедурі, яка може показатися як спеціальна форма більш загального поняття контексту.

Наш метод діє на даних кодуванням, де контекст використовується, щоб відстежувати походження субграфів в залученому графові. Як перший крок, ми використовуємо аналіз можливості з'єднання, щоб отримати розряди для всіх елементів в графові, тобто, вузли і міток країв, і джерела даних. Інтуіція нашого підходу показує. що всі елементи в направленому графі взаємно впливають один на одного, і алгоритм відкриває стосунки, які представляють вхід комунікабельної людини, який створив дані. Ми використовуємо мультилінійну алгебру, як математичний інструмент, зокрема структура під назвою тензор, який є узагальненням векторів і матриць до n-дименсійного випадку. Ми пристосовуємо енергетичний метод для моделювання тензора, щоб отримати рівневі розряди, і різні методи для об'єднання рівневих розрядів у комбінуванні безлічі комбінацій елементів. Відзначте, що хоча великий ряд свіжої робочої мультилінійної алгебри використань для відкриття знання і гірської промисловості даних, ці методи використовують шифрування основних даних (виміри - “продукт”, “автор”, “ключові слова”, і так далі) поки наш підхід кодує RDF з моделлю контекстних даних в тензорі чотирьох вимірів (суб’єкт-твердження-об’єкт-контекст). Робота Кольда і Бадер надихнула на вивчення і нашого методу – як природного продовження випадку чотирьохвимірного тензора, проте ми діємо на RDF графові даних замість HTML структури документа, забезпечуючи намір залучити індивідуальні розряди елементу, і наші системні масштаби, до наборів даних, які є замовленнями більшої величини.

Вкладення цієї примітки є як вказано нижче:

• Ми представляємо алгоритм, заснований на фіксованому обчисленні, щоб автоматично визначити ранжирування для кожного елементу в збільшених у четверо кількостях (вищий метод).
• Ми показуємо кореспонденцію між семантичним графом з контекстом і чотирьохвимірним тензором і описуємо структуру даних, щоб кодувати чотирьохвимірний тензор.
• Ми представляємо набір методів для об'єднання індивідуальних розрядів в складеній розкішній відмітці для збільшених учетверо кількостей або частин цього (Метод ЗНИЖКИ).
• Ми забезпечуємо експериментальну оцінку на великому RDF наборі даних, що складається з понад 200k RDF файлів, зібраних з Павутини, у тому числі зображення даних Вікіпедія.

Залишок паперу організовується як вказано нижче: Секція 2 вводить попередні вибори, семантично пошуковий сценарій, і одну графу від реальних Мережевих даних, разом з результатами, місце яких зайняло.

Секція 3 забезпечує короткий огляд підходу, Секція 4 і 5 описують TOP і алгоритм ЗНИЖКИ, відповідно. Секція 6 забезпечує експериментальну динамічну роботу і якісну оцінку на великому наборі даних, стягнутому з Павутини. Порівняння секції 7 наш підхід із зв'язаною роботою,і Секція 8 закінчується.

2 Семантичний навігатор

Тут ми визначаємо атомну одиницю інформації, в RDF, і контексті, продовження, яке може відстежувати походження вправ тріплексу. І, ми забезпечуємо мотивуючий сценарій, де наше ранжирування і метод розкішного агрегату звертається.

2.1 Тріплекс і Контекст

Стандартна RDF модель даних описується в різних Рекомендаціях W3C. Для зручності і, щоб зберегти простір, ми використовуємо простори імен в Таблиці 1, щоб скоротити URIs, аналог до синтаксису Notation31. Ми використовуємо юридичну особу терміну, щоб послатися на будь-яку річ близько URI, як наприклад особа, протеїн, тема або розташування. Наприклад, юридична особа Тіма Бернера Лі ідентифікує URI http://www.w3.org/People/Berners-Lee/card#i , скоротив як timbl:i.

Префікс	Назва
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
foaf	http://xmlns.com/foaf/0.1/
pim	http://www.w3.org/2000/10/swap/pim/contact#
timbl	http://www.w3.org/People/Berners-Lee/card#
aifb	http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL

Таблиця 1: Скорочення простору імен

(Трійка RDF, Вузол RDF) Дано набір URI, що посилається на R, набір порожнього вузли B, і набір друкарської помилки L, потрійний (s, p, емульсія типу масло у воді) ∈ (R ∪ B) ? R ? (R ∪ B ∪ L) названий трійкою RDF

У такій трійці, s названий темою, p предикат, і емульсія типу масло у воді об'єкт. Приклад для трійки timbl:i foaf:homepage <<http:// <http://www.w3.org/People/Berners-Lee > . Хоча специфікація RDF безпосередньо не визначає поняття контексту, зазвичай додатки вимагають, щоб контекст запам'ятав різні види для даного набору звітів RDF.

(Потрійний за формою і змістом) Пара (t, c) з t бути трійкою і c ∈ (R ∪ B) названий трійкою за формою і змістом c.

Трійка ((s, p, емульсія типу масло у воді), c) за формою і змістом c еквівалентний до четвірки (s, p, емульсія типу масло у воді, c). Інтерпретація контексту залежить від додатка. У нашому випадоку використання інформаційної інтеграції, контекст означає файлу URI або сховища, з якого трійка відбувалася. Завоювання походження одне з фундаментальної потреби в отворі поширювали оточення подібно до Павутини, де якимсь даним доведеться оцінити його походження.

Ми використовуємо звіт, щоб посилатися на трійку або збільшену у четверо кількість, і звертаються до частини звіту. Відзначте, що та ж трійка може відбуватися в багаторазових контекстах.

---

[1] http://www.w3.org/DesignIssues/Notation3.html

2.2 Сценарій

Розглядують систему, що залучає велику кількість доступних структурованих даних на Павутині. Пошукач має наступну функціональність для допиту граф агрегованих значень:

- Представлення пошукових результатів. Отримали набір ключових слів, система повертає юридичним особам, відповідні вказані ключові слова,які потрібно розташувати по пріоритетах.

- Показ інформації юридичної особи. Користувач, який ідентифікував юридичну особу може запитати сторінку, що містить всю інформацію, що має відношення до юридичної особи. Двигун, можливо, повернув би сотні або навіть тисячі звітів, що описують юридичну особу, щоб користувач швидко знайшов. Тому, сторінка юридичної особи повинна лише містити top-n доречні звіти.

У наступному ми ілюструємо проблему вибору найдоречніших top-n збільшених у четверо кількостей даних для даної юридичної особи, в даному випадку Тім Berners-Lee. Проблема схожа до типових графічних сценаріїв, місце яких зайняло, бо ми вимагаємо в розрядів повні збільшені учетверо кількості замість розрядів для індивідуальних вузлів. Дані, представлені тут, є спрощена версія реального набору даних, як стягнутий з Павутини. Для властивостей повного субграфа погляньте на Секцію 6.

Наш приклад заснований на типовому сценарії використання сьогоденної Семантичної Павутини, включаючи соціальні дані мережі, наприклад: Друг за Друга (ДзД)[2]. Фігура 1 це частина показів різного графа, яке отримане від трьох різних джерел, які посилаються на вузол центру. Ми показуємо лише декількох звітів, проте, повне число збільшених у четверо кількостей, що описують Тіма в нашому наборі даних, складає 197 від 28 різних джерел. Для того, щоб забезпечити стислу характеристику всієї доступної інформації про Тіма, ми хочемо показати лише найвидатніші звіти. Об'єднуючи різні субграфи, походження інформації - важливий аспект, який потрібно взяти до уваги в подальших оброблювальних кроках. Щоб записати походження даної трійки, ми використовуємо поняття контексту, який означає, що ця наша модель даних складається із збільшених у четверо кількостей: суб’єкт, предикат, об'єкт, контекст.

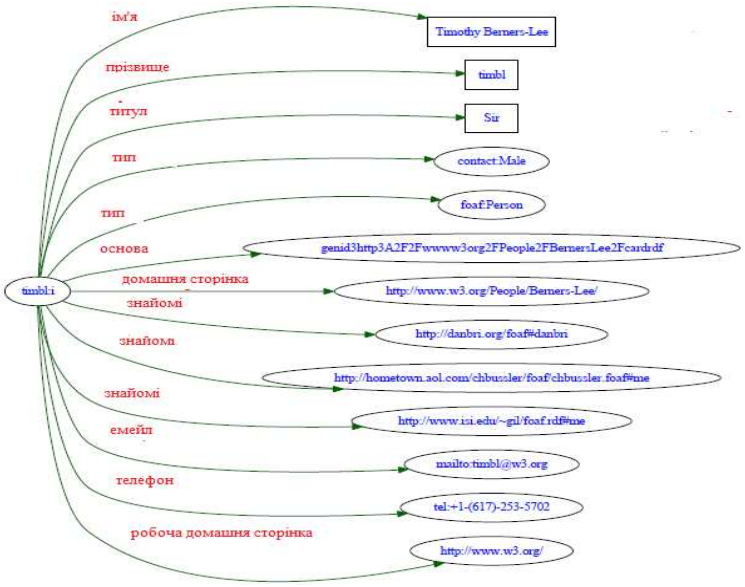
Поки більшість даних в прикладі виражаються в FOAF, ми не можемо зробити жодних гарантій. Зокрема, збираючи файли, ми також отримуємо дані вираженим в інших словниках, які не обов'язково відомі закладу. Будь ласка також відзначте, що хоча ми
Показуємо в нашому прикладі, всі види юридичних осіб (як наприклад протеїни, розташування, публікації, і т.п.) могли бути повернені, кожен з них описує використання словників.

3 Короткий огляд TOPDIS

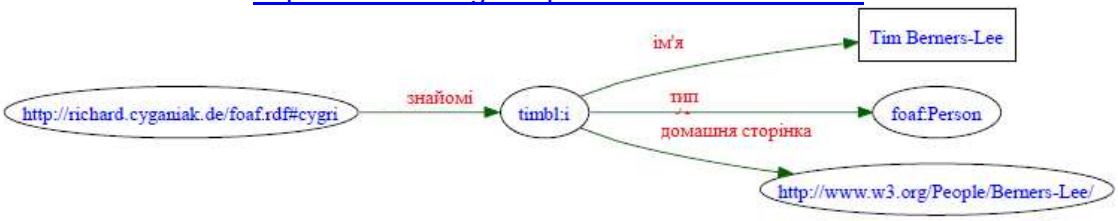
Метод TOPDIS використовує алгоритм (під назвою ВЕРШИНА) єднального аналізу, щоб обчислити безліч важливостей для кожного елементу в графові. Другий крок під назвою ЗНИЖКА бере це “безліч” як вхід до методу агрегату, щоб класифікувати кожен звіт згідно важливості його засновницьких елементів. Нарешті, top-k звіти - відібраний, щоб отримати ущільнений вид вхідного графа.

---

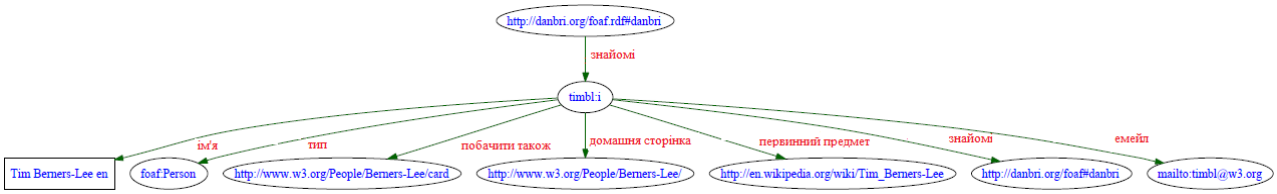
[2] http://www.foaf-project.org/



http://www.w3.org/People/Berners-Lee/card.rdf



http://richard.cyganiak.de/foaf.rdf



http://danbri.org/foaf.rdf

Фігура 1: Три джерела, що нагадують вузол центру URI. Кожне джерело даних представляє один контекст.

Мета методу, - ідентифікувати найдоречнішу інформацію в графові і ущільнити це в одній послідовній виставі. По-перше, нам, можливо, потрібно ідентифікувати частину приналежності графа до нашого запиту, якщо є один. Потім ми оцінюємо важливість кожної теми, предиката об'єкт і контекст, використовуючи алгоритм аналізу можливості з'єднання, який бере до уваги взаємозалежні властивості графа. Нарешті, ми комбінуємо ці тури ранжирування в твердження ранжирування, щоб розташувати по пріоритетах звіти і витягувати найголовніші для здобуття ущільнення вхідного графа.

Іншими словами, метод складається з наступних кроків:

• Виберіть повний граф, як вхід до алгоритму.

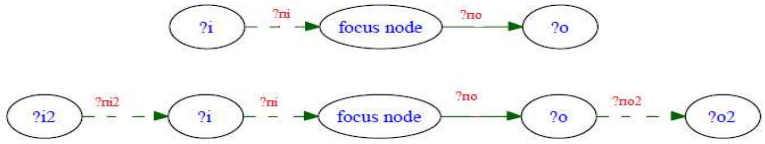
• Здійснити обчислення для кожного елементу (тема, предикат, об'єкт, контекст) у відібраному графові, використовуючи метод аналізу для можливості з'єднання (ВИЩИЙ алгоритм).

• Визначте твердження для кожного субграфа об'єднуючи безліч елементів у відмітці звіту (Алгоритм ЗНИЖКИ).

• Застосуйте евристику, щоб вибрати представницьку підмножину найдоречніших звітів.

3.1 Вибір субграфа

Ми можемо вплинути на результат алгоритму, вибираючи різні стратегії для здобуття вхідного графа, зокрема змінюючи параметр □, це визначає широту виділення. Ми можемо також використовувати маленький граф, який благоволить центральним вузлам (□ = 1), або більшому графові (наприклад, □ = 2), з більшою відстанню, щоб отримати більш загальні ранжирування. Фігура 2 показує графічно запити використання, щоб вибрати субграф з e = 1 і e = 2 від вузла центру. Вибираючи більшого графа ми можемо бачити ефекти дрейфу теми – метод, можливо, надзвичайно класифікував би елементи, які не доречні для початкового запиту. Саме спільне ранжирування отримує відбір повного графа як входу до алгоритму TOP, тому обчислюючи глобальний розряд кожного елементу в графові незалежно від будь-якого запиту. Розгляньте Секцію 6 для експериментальних результатів на графах із зміною розміру виділення сусідства.



Фігура 2: Ставить під сумнів використання, субграфів з □ = 1 і □ = 2

У відібраномусубграфі, ми нехтуємо звітами, що містять друкарські помилки, які можуть виглядати як властивості вузла замість частини топології графа. Крім того, ми не розглядаємо rdf:type і owl:sameAs звіти, з тих пір, як ці звіти мають справу з представленням даних замість зразків безпосередньо.

3.2 Обчислювальний Елемент Класифікує

Мета методу, представленого тут, - ідентифікувати найдоречнішу інформацію в графові і довільно ущільнювати це в одній послідовній виставі. Спершу, ми оцінюємо важливість кожної теми, предиката об'єкту і контекст, використовуючи алгоритм аналізу можливості з'єднання, який бере до уваги властивості графа. Метод TOPDIS використовує алгоритм (під назвою ВЕРШИНА) єднального аналізу, щоб обчислити безліч важливостей для кожного елементу в графові. Результат елементу, що займає місце для мережі, показаної у Фігурі 1, забезпечені в Таблиці 2.

Згідно нашим ранжируванням елементу, найголовніша тема в межах стрибка 1 із захисту Тіма Бернерса Лі - timbl:i. Наступний, вище всього класифікував теми - двох документів, створених Тімом Беонерсом Лі, завершеним двома подіями, в яких він брав участь. У об'єктних ранжируваннях, чотири з п'яти позицій займають юридичні особи, що представляють людей, з timbl:i знову, займаючи місце вище всього. Лише інша об'єктна юридична особа зайняла місце в межах п'яти вершин, - початкова сторінка Тіма Бернерса Лі (http://www.w3.org/People/Berners-Lee/).

Найвищий предикат, місце якого зайняло в графові, обчислюється як foaf:knows, який дає звіт про найбільш важливі зв'язки в мережі. Інші важливі предикати в мережі включають rdfs:seeAlso, який використовується, щоб вказати ресурс, що забезпечує додаткову інформацію, що має відношення до підвладного ресурсу властивості, що описують особисту інформацію від словника (наприклад foaf:homepage) FOAF, і ПЕРЕСТАВЛЯЮТЬ Особисті предикати Інформаційного Підвищення (pim:participant).

Суб’єкти	Рахунок
timbl:i	627432
http://dig.csail.mit.edu/breadcrumbs/blog/4	356325
http://dig.csail.mit.edu/2007/01/camp/data#course	356325
http://www.ecs.soton.ac.uk/~dt2/dlstuff/www2006 data#panel-panelk01	353858
http://wiki.ontoworld.org/index.php/ IRW2006	353858

Предикати	Рахунок
foaf:knows	619492
foaf:homepage	424084
rdfs:seeAlso	422884
foaf:maker	40776 7
pim:participant	392859

Об’єкти	Рахунок
timbl:i	166413
http://www.w3.org/People/Berners-Lee/	153487
timbl:amy	144570
http://www.w3.org/People/karl/karl-foaf.xrdf#me	135552
http://www.w3.org/People/EM/contact#me	135552

Контекст	Рахунок
http://www.w3.org/People/Berners-Lee/card	587689
http://danbri.org/foaf.rdf	401159
http://kuruman.org/foaf.rdf	375509
http://richard.cyganiak.de/foaf.rdf	345754
http://www.w3.org/People/EM/contact	342809

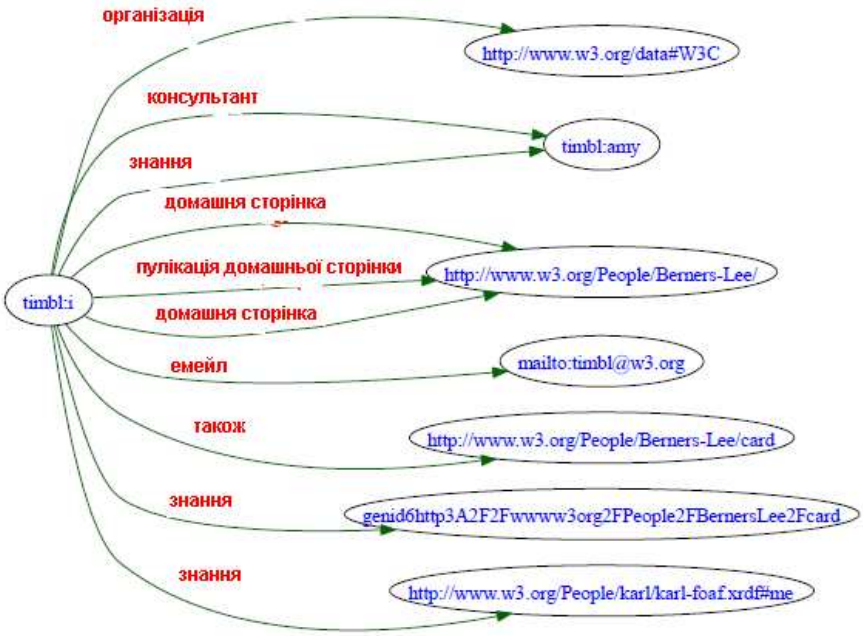
Таблиця 2: Походить від ранжирування графів, представлених у Фігурі 1
Контекст, який розрядив спочатку, - головне джерело інформації про Тіма.

3.3 Розряд елементу об'єднання

Обчисливши чотири набори розрядів, ми можемо комбінувати розряди елементу, щоб отримати в складеній відмітці, місце, згідно важливості його засновницьких елементів, необхідних до того, наприклад щоб показати найвидатніші вправи. На додаток до обчислення комбінованої безлічі ми забезпечуємо евристику, щоб вибрати зразок представлення найпопулярніших звітів. Фігура 3 показує завершальний результат алгоритму TOPDIS в комбінації з евристикою: граф, що дистилює, походив від вхідного графа у Фігурі 1. Відзначте, що графічна вистава лише один шлях передачі результуючого графа, інші види можуть також проводитися ідеально, зважаючи на моделювання інформації, яка конкретизує, як певні властивості які потрібно показати.

4 Обчислювальні розряди

В наступній секції, яку ми обговорюємо детально, як отримати розряди елементу, використовуэмо алгоритм TOP. Алгоритм TOP - метод аналізу заснування зв’язку можливості з'єднання в розряді традиційної сторінки.



Фігура 3: Граф, що дистилює, містить вищі-10 звіти.

ХІТИ (TOPHITS), застосовані до чотирьох вимірних даних. До формалізації описують наш алгоритм, ми вводимо поняття тензора, математична структура використовувала в мультилінійній алгебрі. Тензор - узагальнення векторів і матриць до n- вимірного випадку. Тобто, вектор може виглядати як тензор розряду 1 і матриця як тензор розряду 2.

4.1 Важливості обчислення

Важливі теми вказують на важливі об'єкти і на важливі об'єкти вказують важливі теми. Єднальні важливі теми важливих предикатів і об'єкти. Важливі вправи утрьох відбуваються у важливих контекстах. Зважаючи на ці взаємні асоціації, ми прагнемо дистилювати зрозумілий контур найістотніших особливостей від
залученого графа.

Безліч важливості обчислюється як вказано нижче:

$$s_i^{t+1} = \sum_{i \xrightarrow{k} j@l} p_k^t o_j^t c_l^t \quad \text{для i=1,…,n} \qquad (1)$$

$$p_j^{t+1} = \sum_{i \xrightarrow{k} j@l} s_i^{t+1} o_j^t c_l^t \quad \text{для j=1,…,n} \qquad (2)$$

$$o_k^{t+1} = \sum_{i \xrightarrow{k} j@l} s_i^{t+1} p_k^{t+1} c_l^t \quad \text{для k=1,…,n} \qquad (3)$$

$$c_l^{t+1} = \sum_{i \xrightarrow{k} j@l} s_i^{t+1} p_k^{t+1} o_j^{t+1} \quad \text{для l=1,…,n.} \qquad (4)$$

де $i \xrightarrow{k} j@l$ засіб зв'язоку між вузлом і та вузлом j з предикат k і контекстний l. Підвладна відмітка оновлюється з продуктом предиката, об'єкту і звіту. Відмітка предиката оновлюється з продуктом теми, об'єкту.

Протягом кожного повторення, після обчислення кожного результату вектора, ми нормалізуємо таким чином, що відмітка мінімального елементу складає 1 і виконуються обчислення

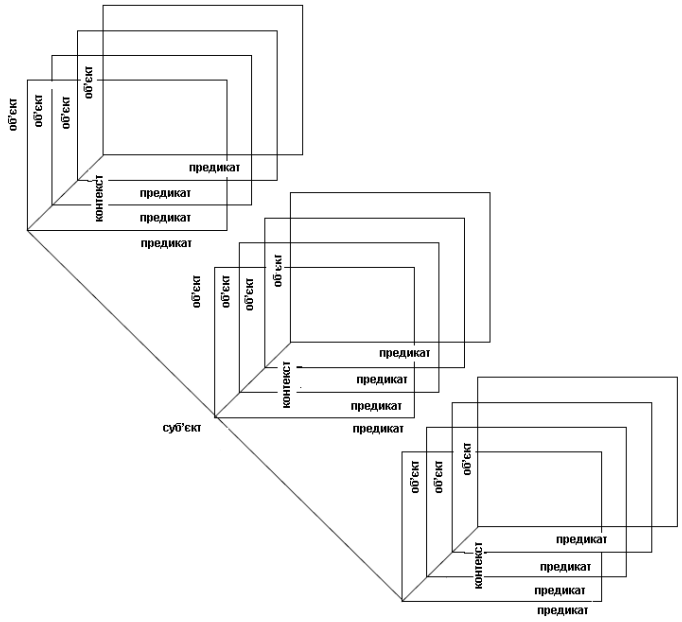
$$a_i = 1 + \log(a_i) \quad \text{для i = 1 . . ., n} \qquad (5)$$

, щоб перешкоджати декільком видатним елементам домінувати над результатами.

4.2 Зберігання Тензора

Наш алгоритм вимагає послідовного доступу до кожного з вимірів тензора. Кодуючи тензор з виміру i,j,k,l то це вимагало б простору розміром i*j*k*l, який ясно дуже дорогий для будь-якого додатка до проблем. Для ефективного представлення для розкладання тензора ми винаходимо структуру даних, яка дозволяє доступ до всіх звітів, що мають відношення до даної теми, предиката, об'єкту, або контексту, через послідовне сканування. Для угрупування звітів разом з даними елементами, ми проводимо чотири індексні файли, які сортуються згідно кожному з чотирьох вимірів. Ми використовуємо структуру дискових баз даних, яка може обробити великі тензори, але структуру дискових баз може легко замінити подібна структура даних в оперативній пам'яті. Індексні файли створюються через багатоканальне сортування злиттям з складністю Емульсія(n колода n) типу масло у воді, і стискував використання Huffmann, що закодував, щоб зберегти дисковий простір.

Фігура 4 ілюструє індексну структуру. Індексний файл - сортований акордом до теми, який означає, що ми можемо звернутися до всього предиката, об'єкту, і контекстних вузлів, лише, скануючи послідовно через індексні файли. З метою алгоритму, ми використовуємо чотири індексні файли, що групується акорди до різного виміру. Для більш маленьких вхідних файлів, ми також винайшли еквівалентну структуру даних в оперативній пам'яті.



Фігура 4: Індекс тензора з чотирма тема вимірів, предикат, об'єкт, і контекст.

4.3 Обчислювальні розряди

Як тільки вхідний тензор вибраний і індексні файли готові, ми застосовуємо повторний Алгоритм алгоритму 1) щоб обчислити індивідуальну ВИЩУ безкінечність. По суті, ми використовуємо адаптацію Енергетичного Методу [9] до багатовимірного випадку. Спершу вектори елементу ініціалізувалися до 1. Потім, для кожного з теми, предиката, об'єкту, і контекстних векторів, ми оновлюємо вектор розряду елементу в термінах векторів інших елементів в звіті, і нормалізуємо і вимірюємо вектори.

Алгоритм 1 ВИЩИЙ алгоритм для здобуття розрядів від топології графа на направленному графі з контекстом.

Вимагайте: Граф G кодував в темі, предикаті, об'єкті, контекстні індекси
$s \leftarrow 1$, $p \leftarrow 1$, $\leftarrow 1$, $c \leftarrow 1$
при дуже великій помилці число повторень нижче, роблять
для всього збільшеного учетверо $sp_ko_jc_l$ з підвладним $s \in G$
якщо, повторюючи s потім
$s[s] \leftarrow s[s] + p[p_k] * [o_j] * c[c_l]$
інший
$s[s] \leftarrow p[p_k] * [o_j] * c[c_l]$
кінець, якщо
закінчуються
нормалізують і масштаб s
обчислюють o p, і c передача
закінчуються, поки
повертають розряди елементу s, p, c

Ми передбачаємо, що цей наш алгоритм сортував доступ до всіх звітів з даною темою, предикатом, об'єктом, або контекстом, використавши багатовимірну структуру доступу, яка може повторювати над кожним з чотирьох вимірів і здійснювати обчислення. Є багато можливих виконань; ми здійснили як дискова база, так і структура доступу в оперативній пам'яті.

Ми знайшли, що безпосередньо, застосовуючи обчислення без операційних результатів в усіх явищах, де декілька елементів нагромаджують весь розряд в системі і превалюють над результатами, не дивлячись на їх доцільність.

Складність алгоритму лінійна, якщо ми переймаємо на себе сортований доступ, що вирішує обчислення уздовж специфічного виміру через індексне сканування. Час прогону алгоритму є переважно визначеним числом повторень.

4.4 Математична характеристика

В цій секції ми описуємо свій метод в термінах мультилінійної алгебри. Ми використовуємо зображення тензора знаками як описано в [19]. Тензор $I_1*I_2*I_3*I_4$, відповідний темі, предикату, об'єкту і контекстним вимірам. Вектори s, p і c оновлюються протягом кожного повторення приблизно

$$s^{t+1} = A \times_2 p^t \times_3 o^t \times_4 c^t \quad (6)$$

$$p^{t+1} = A \times_1 s^{t+1} \times_3 o^t \times_4 c^t \quad (7)$$

$$o^{t+1} = A \times_1 s^{t+1} \times_2 p^{t+1} \times_4 c^t \quad (8)$$

$$c^{t+1} = A \times_1 s^{t+1} \times_2 p^{t+1} \times_3 o^{t+1} \quad (9)$$

5 ЗНИЖКИ: безкінечності

Об'єднання безкінечності, отримана методом TOP, місце якої дозволяє розташовувати по пріоритетах індивідуальні елементи (теми, предикати, об'єкти, і контекст). Проте, додатки вимагають, щоб розташувати по пріоритетах не лише єдині елементи, але і пари, вправи утрьох, або навіть збільшені учетверо кількості. Наприклад показ лише top-k вправ утрьох, що мають відношення до юридичної особи, вимагає, щоб розташувати по пріоритетах вправи утрьох, не лише єдині елементи. У цій секції, ми описуємо крок агрегату, так званого TOP, який дозволяє комбінувати індивідуальну безкінечність в залученій відмітці.

5.1 Комбінації векторної норми

n елементів можуть бути інтерпретовані як n-вектор. Наприклад, пари як наприклад суб’єкт/ комбінації об'єкту або предикат/об’єкт комбінації розглядуються як 2-вектор, і вправи утрьох як наприклад суб’єкт/предикат/об’экт розглядуються як 3-вектор, і так далі. Ми використовуємо довжину вектора, щоб обчислити комбінований розряд.

Повна процедура дистиляції є для збільшених учетверо кількостей:

$$\|x\| = \sqrt{s^2 + p^2 + o^2 + c^2} \quad (10)$$

Для інших комбінацій елементу, що дистилюють класифіковано, обчислюється аналогічно.

5.2 Вибір представницьких звітів

Завершальний крок відсутній, щоб прибути в субграф, що містить лише top-k самі представницькі вправи утрьох. Шлях мав би лише вибрати top-n звіти після обчислення векторної норми. Проте, початкові дані зазвичай один або два предикати превалюють над вищими розрядами. В результаті, top-k звіти, можливо, цілком включали б той же предикат, який зазвичай повертає дуже загальноприйнятий субграф, які не дають лише характеристику вузла центру. Щоб здолати цю поведінку ми винаходимо алгоритм, який вибирає top-k звіти, підтримуючи деяку різноманітність в повернених результатах. Евристики для здобуття графів, що дистилюють, для комбінацій елементу показується в Алгоритмі 2.

6 Експериментів і Оцінки

Ми зараз представляємо результати вивчення, щоб затверджувати методи і алгоритми, описані в цьому папері. Ми починаємося характеризацією набору даних, отриманого від Павутини, потім описують експериментальну установку і продуктивні характеристики методу і остаточно вводимо метод якісної оцінки і результати. Ми використовуємо набір базових алгоритмів для для нашої ВЕРШИНИ і кроків ЗНИЖКИ.

Алгоритм 2 Здобуття графів, що дистилюють.

Вимагайте: G
Вимагають: s, p, c з безліччю розряду елементу
Вимагають: Складений м.
Вимагають: Пороговий k
для кожного збільшеного учетверо spoc роблять
c = складений розряд для spoc
кінець для
сорт c
для кожного входу e в c I менш ніж k кінчається надруковано роблять
якщо там існує багаторазові комбінації елементу з таким же значенням, потім
сколупують випадковий одного і друкують комбінацію елементу і інша
комбінація елементу друку значення
і значення
закінчуються, якщо
закінчуються, бо

Ми порівнюємо ВЕРШИНУ з частотним графом (ЧАСТОТА) у роботі і УДАРЯЄТЬСЯ, щоб перевірити ще раз за різними джерелами тему і об'єктну безліч, і, щоб порівняти динамічну роботу. Ми порівнюємо якість кроку DIS з підсумовуванням (СУМА) і множенням (MUL) безлічі елементів.

6.1 Набір даних

Ми зібрали набір даних з 72,462,443 звітами від 222,469 джерела RDF, використовуючи MultiCrawler [11]. Великі бази даних лише частково увійшли до набору даних. Набір даних містить широку різноманітність юридичних осіб (27,399 різних класів) і зв'язків між ними (107487 чітких предикатів). Розмір файла даних складає 17 ГБАЙТ, сортовані і стислі індекси між 1.8 і 2.0 ГБАЙТ. Різниця за розміром для індексних файлів може бути приписана співвідношенням різної компресії залежно від сортуючого замовлення.

6.2 Виконання

Ми здійснили прототип алгоритму TOPDIS в Яві. Експерименти здійснювалися на машинах AMD Opteron 2.2GHz з 4 ГБАЙТ оперативної пам'яті, бігом Debian Linux.

Для експериментів на кроці TOP ми виправили число повторень. Зазвичай, десяти повторень досить, щоб повернути розумно стійкі вектори результату. Установка числа повторень має додаткову вигоду, що нам не потрібно запам'ятовувати результати попередніх повторень в пам'яті, тому ділячи два на суму оперативної потрібної пам'яті. Ми використовуємо чотири hashtables в оперативній пам'яті, щоб тримати вектори результату що виходить протягом кожного повторення.

Для кроку DIS, ми послідовно читаємо всі звіти, обчислюємо завершальну відмітку для кожного, і тримаємо результуючі top-k звіти в пам'яті.

[3]livejournal.com, tribe.net або vox.com

6.3 Закінчення оботи

Для продуктивної оцінки, ми отримали два набори субграфів. Таблиця 3 складає список маленьких субграфів, що мають відношення до юридичної особи (наприклад timbl:i), і Таблиця 4, широкий субграф, що містить юридичні особи даного типу (наприклад всі юридичні особи rdf:type foaf:Person) щоб продемонструвати масштабованість алгоритмів. Для субграфа, що описує юридичну особу, ми вибираємо два набори субграфів з □ = 1 і □ = 2, щоб показати ефекти вибору субграфа змінюючи розміри на часі алгоритму.

URI	є	Ніщо з stmt	ВЕРШИНИ	не ПРИГОЛОМШУЄ	БАГАТОКРАТНИЙ
dbpedia:Beijing	1	393	536	432	339
dbpedia:Beijing	2	1383	1212	786	715
dbpedia:DNA	1	41	317	58	39
dbpedia:DNA	2	50	262	62	43
dbpedia:Galway	1	79	332	90	49
dbpedia:Galway	2	322	700	384	317
dbpedia:Republic Ірландії	1	916	842	748	518
dbpedia:Republic Ірландії	2	5008	2856	1624	1196
aifb:id57instance	1	1085	869	668	513
aifb:id57instance	2	8260	4479	2283	1347
timbl:i	1	197	490	329	71
timbl:i	2	3371	1709	1174	982

Таблиця 3: Часи, місце яких зайняло, для різного юридичного субграфа (часи в ms, десяти повтореннях індексів в оперативній пам'яті).

Обмеження	Ні з stmt	Індексує ЧАСТОТА	ХІТІВ	ВЕРШИНИ	часу
rdf:type foaf:Person	5201263	1738406	21077732	6885006	397181
rdf:type skos:Concept	62358	23694	41078	18587	6295

Таблиця 4: Часи, місце яких зайняло, для субграфів, що містить всі юридичні особи за класом.

6.4 Якісна оцінка

Щоб оцінити якість методу, ми проводили призначене для користувача, в якому ми просили учасників уручну оцінити масштаб з 1 до 5 важливих звітів, що мають відношення до чотирьох юридичних осіб в субграфі. Дев'ять чоловік брало участь у вивченні, привівши від чотирьох до шести ручних рейтингів для кожного з перевіреного субграфа. Залучені рейтинги представили звіти, як виражено причетною групою.

Ми вирішували використовувати Кендалівську τ відстань для порівняння альтернативних алгоритмічних підходів з ручними рейтингами. Другий вибір мав використовувати точність і відзивання, яке, вимагає двійкового вибору доцільності. В таблиці 5 ми перевіряємо Кендалівську τ для класифікації списків (де 1 повна кореспонденція , 0 немає кореспонденції, і -1 останнього).

URI	FREQ/SUM	FREQ/MUL	FREQ/DIS	TOP/SUM	TOP/MUL	TOP/DIS
timbl:i	0.51111	0.51111	0.51111	0.51111	0.51111	0.51111
aifb:id57instance	-0.066667	-0.066667	-0.066667	-0.066667	-0.066667	-0.066667
dbpedia:Beijing	-0.022222	-0.28889	0.022222	0.15556	-0.33333	0.15556
dbpedia:Galway	0.15556	0.11111	0.066667	0.15556	0.11111	0.15556

Таблиці 5 Кендалівська τ відстань, яка вимірює перекриття вищих-10 результатів між алгоритмічним і ручним ранжируванням.

6.5 Обговорення

TOP/SUM і TOP/DIS виконуються краще всього на чотирьох вибрав субграфах. Проте, якісний результат для субграфа значно відмінний в відстані. Можливо одне поясненя знаходиться в природі субграфа (і джерела). URI timbl:i використовується в діапазоні даних джерела, що охоплюють багато людей і організації. Результуючий subgraph містить декілька надлишкових вправ утрьох, що походять від різноманітності призначених для користувача вкладень. В деякому відношенні, видавці даних голосують за важливі звіти і елементи, використовуючи їх URIs. Він видається, що дано досить різноманітність в призначених для користувача вкладеннях, прості частотні графи, щоб визначити популярні елементи і звіти.

У контрасті, URI aifb:id57instance, який означає навантаження, не розділяється через багато сайтів. URI відбувається в багатьох контекстах, проте, більшість їх, видається, є databasegenerated від центрального сховища.

Вікіпедія суьграфа, задовільняє нашу процедуру, місце якої зайняло, дано вправи утрьох походять від одного джерела, результатів немає, як добре щодо timbl:i.

Протягом експериментування, ми стикалися з єднальним спамом особливо з даними від павутини HTML. Використання імен провідного вузла замість повного URIs для контексту полегшує цю специфічну проблему, з тих пір, як єднальні ферми потім не набувають дуже великої безкінечності. Хоча цей простий метод працював на нашому наборі даних.

7 Робота зв'язаних алгоритмів

Основну цитату було дослідженно в соціології в області соціального аналізу [24]мережі, який заявляється як результат невідповідність між теорією двовимірних графів і склані вимірні соціальними мережами.

Ми розширили алгоритми можливості з'єднання як наприклад PageRank [22], ХІТИ [13], і TOPHITS [14], щоб діяти на графах даних. При масштабах PageRank дуже добре, це лише діє на двовимірних матрицях. HITS також діє на двосторонніх матрицях, проте це отримує два результатуючі вектори, які відповідають центру і повноваженням, і застосовує процес, подібний до єдиного розкладання значення, щоб отримати більш ніж один набір векторних (по суті, обчислюючи не лише домінуючу векторну пару, але і всі векторні пари) результатів. Так само, TOPHITS використовує як математичну модель розкладання PARAFAC,
яке є продовженням SVD до багатовимірного випадку. У контрасті до TOPHITS,

який працює на представленні документів, наш підхід приймає чотирьохвимірні дані, кодував як збільшені учетверо кількості, зважаючи на походження звітів.

Барат і Гензінгер [4] обговорюють різні проблеми XITIB як наприклад сімейність і дрейфу теми, і запропонували методи, щоб виправити ці проблеми. Альтернативний алгоритм, SALSA[16], виконує випадкові прогулянки на веб-сторінках, метод, який уникає проблеми, викликаної суспільствами. Безліч властей визначає випадкова прогулянка, наступна за заднім зв'язком, а потім переднім зв'язком додатково і визначає випадкова прогулянка. Знайшли, що алгоритм HITS нестійкий при деяких обставинах[20] і до цієї проблеми звернувся випадковий побій моделювання в змішаних XITAX[21], [23]і Змішав SALSA[15]. Для хорошого короткого огляду цих методів, місце яких зайняло, погляньте [6].

Для нашого прикладного сценарію, ми лише вимагаємо першого набору векторів, в контрасті до TOPHITS, який обчислює всі розкладання тензора. Приховані семантичні подібні методи індексуючих [7] використань до XITIB, також на двовимірні рівні, щоб проаналізувати взаємини між термінами і документами.

Ми продемонстрували свої масштаби алгоритму до великих наборів даних, поки найбільш робочий на багатовимірній виставі поки що лише зосередився на тензорах малого розміру. TOPDIS міг використовувати дворівневий алгоритм, подібний [5], щоб комбінувати глобальні ранжирування, що займає місце для скорочення внизу розмір проміжних результатів в поширюваній архітектурі пошукача.

Об'кт [12] описує підхід, щоб зайняти місце направленого графа, використавши сторінку. Робота включає поняття під назвою повноваження перенесення схеми графів , який визначає надбавки для передачі поширення через різневі види зв'язків. Об'єкт залежить від призначеного для користувача входу зв'язків між вузлами, щоб описати їх семантичну вагу, таким чином, що триколійна вистава може бути зруйноване в двосторонній матриці, на якій алгоритм стилю сторінки застосований. Також, жоден розгляд не наданий походженню даних.

SemRank [1] стосунки і шляхи на Семантичних Мережевих даних, використовуючи інформаційно-теоретичні заходи. У контрасті, ми класифікуємо всі елементи графа даних з контекстом, використовуючи математичну модель , упроваджену в лінійну алгебру.

8 Закінчення

У цьому вивченні, ми показали, як прикласти значення до звітів RDF на Павутині, і використовують значення, щоб витягувати реквізити потенційно дуже великі графи. Ми застосували метод лінійної алгебри до великих наборів даних, зібраних з тисяч джерел. Наш підхід розширює алгоритми єднального аналізу, відомі від додатків гіпертексту до королівства семантичних графів з контекстом. Ми дали математичну характеристику нашого алгоритму, і перевірили експериментально інтуїцію позаду методу. Ми застосовуємо отримані ранжирування сконструювавши версію потенційно великих даних графів, що дистилюють, прибуваючи в стислу характеристику вузлів центру. Проте, алгоритм може бути застосований до широкого діапазону інших прикладних областей, що включають графічну структуру даних.

Як ВЕРШИНА, так і алгоритми ЗНИЖКИ, упроваджуються в чисту модель алгебри і тому можуть легко бути розширені. Наприклад, додаючи вимірів як наприклад безліч зворотних зв'язків доцільності, членство групи або відстань в соціальній мережі для ранжирувань просте.

що відкриває структуру, представлену в спільно редагованих семантичних графах, має істотний науковий і комерційний потенціал. По-перше, Павутина, найбільший експонат людського знання, стає темою до наукового аналізу [3]. Розуміння зв'язків, що маються на увазі, і структури в графові Мережевих даних може допомагати виявити нове розуміння зразків співпраці і процесів, якими мережі формують і еволюціонують. По-друге, маючи сенс поза науковими виданими даними на Павутині може допомогти ученим, щоб придбати інтуїції для їх дослідження. Здатність пройти і знайти фактично через магазин знання, об'єднаного тисячами джерел, може розширити накопичувач інформації і ідей, доступних до наукового суспільства. По-третє, роблячи графа Мережевих даних доступним для діалогових відносний, розглядуючи і навігація має додатки в областях як наприклад e-commerce і e-health, і має потенціал, щоб поліпшити якість і користь Мережевого пошуку взагалі.

Список літератури

[1] К.Анави. Мадуко, і. Шет. Семранк: складний пошук взаємовідношення, місце якого зайняло, кінчається на семантичній павутині. У WWW ’05: Слухання 14-ої міжнародної конференції зі Світової Широкої Павутини, сторінки 117–127, Нью-Йорк, НЬЮ-ЙОРК, США, 2005. Преса АСМ.

[2] J. . Аслам і М.. Монтегю. Моделює. У SIGIR ’01: Слухання 24-ої міжнародної ACM SIGIR конференції щорічника по Дослідженню і розвитку в пошуку інформації , сторінки 276–284, Нью-Йорк, НЬЮ-ЙОРК, США, 2001. Преса АСМ.

[3] Т. Бернер-Лі, В. Зал, Дж. Хендлер, Н. Шадболд, і Д. Дж. Вейтзней. Створення наукової павутини. Наука 313(11), 2006.

[4] К. Барат і М.. Р. Гейзінгер. Вдосконалені алгоритми для дистиляції теми в оточенні. У Слуханнях 21-ої Міжнародної ACM SIGIR Конференції Щорічника по Дослідженню і Розвитку в Пошуку Інформації, сторінки 104–111, Мельбурн, AU, 1998.

[5] . З. Броднр, Д. Кармел, М.. Герсковісі. Софер, і Дж. Зієн. Оцінка запиту чинника, використовуючи дворівневий пошуковий процес. У CIKM ’03: Слухання дванадцятої міжнародної конференції з Інформації і управління знання, сторінки 426–434, Нью-Йорк, НЬЮ-ЙОРК, США, 2003. Преса АСМ.

[6]С. Чакрабарті. Гірська промисловість Павутини. Видавці Моргана Кауфман, 2003.

[7] С. Дірвестер, С. Думайс, Г. Фурнарс, Т. Ландаєр, і Р. Хершман. Індексація прихованим семантичним аналізом. Журнал Американського Суспільства для Інформаційної Науки, 41(6):391–407 1990.

[8] С. Дворк, Рю Кумар, М.. Наор, і Д. Сівакумар. Методи агрегату для павутини.
У WWW ’01: Слухання 10-ої міжнародної конференції зі Світової Широкої Павутини, сторінки 613–622, Нью-Йорк, НЬЮ-ЙОРК, США, 2001. Преса АСМ.

[9] Г. Н. Голуб і С. Ф. В. Позика. Матричні Обчислення. Преса Університету Джона Хопкінса, третєвидання, 1996.

[10] Р. В. Гах, Р. МакКул, і Р. Фікс. Контексти для Семантичної Павутини. У Слуханнях третьої Міжнародної Семантичної Мережевої Конференції, Хіросіми, 2004 листопада.

[11] . Харт Дж. Умбіч, і С. Деккер.: Прокладена трубопроводом архітектура для повзання і індексації семантичних мережевих даних. У 5-ій Міжнародній Семантичній Мережевій Конференції, Афінах, GA, США., 2006.

[12] Н. Гван, В Хрістідіс, і Ю. Конструктивні: система для основного пошуку на базах даних. У Слуханнях 2006 ACM SIGMOD Міжнародна Конференція Управління Даними, сторінки 796–798, Нью-Йорк, НЬЮ-ЙОРК, США, 2006. Преса ACM.

[13] Дж. М.. Кленберг. Авторитетні джерела в оточенні. Журнал ACM, 46(5) :604–632, 1999.

[14] Т. Колда, В. Бадер, і Дж. Кенні. Higher-order аналіз зв'язку павутини, використовуючи лінійну алгебру. У Слуханнях П'ятої Міжнародної Конференції IEEE по гірській Промисловості Даних, сторінки 242–249. IEEE Вашингтон Суспільства Комп'ютера, DC, США, 2005.

[15] Н. Ліі і. Бородін. Хвилювання оточення. Слухання Міжнародного Обчислення дев'ятої Частини і Конференції Конбінаторики, 2003.

[16] Р. Лемпел і С. Моран. Стохастичний підхід для аналізу (SALSA) і ефект TKC. Обчислювальні Мережі, 33(1-6) :387–401, 2000.

[17] М.. В. Магоні, М. Магіні, і Р.Дінес. Розкладання тензора для основного тензора даних. У KDD ’06: Слухання 12-го ACM SIGKDD міжнародна конференція з відкриття Знання і гірської промисловості даних, сторінки 327–336, Нью-Йорк, НЬЮ-ЙОРК, США, 2006. ACM.

[18] Ф. Манола і Е. Мірошник. Буквар RDF. Рекомендація W3C, 2004. лютого http://www.w3.org/TR/rdf-primer/.

[19] С. Д. М.. Мартін. Примітки обговорення цеху розкладань, 2004.

[20] . Н. Жен, і М.. Йорданія. Єднальний аналіз, власні вектори і стабільність. У Слуханнях 17-а Міжнародна Погоджувальна Комісія на Штучному Інтелекті, сторінки 903–910, 2001.

[21] . Н. Жен, і М.. Йорданія. Стійкі алгоритми для єднального аналізу. У Слуханнях 24-ої Міжнародної ACM SIGIR Конференції Щорічника по Дослідженню і Розвитку в Пошуку Інформації, сторінки 258–266. ACM Нью-Йорк Преси, НЬЮ-ЙОРК, США, 2001.

[22] Л. Пейдж, С. Брін, R. Мотвані, і Т. Виноград. Сторінка ранжирування цитати: Приведення замовлення до павутини. Технічне повідомлення, Стенфорд Цифровий Проект Технологій Бібліотеки, 1998.

[23] Д. Раффі і. Емульсія типу масло у воді. Мендензол. Для чого ця сторінка відома? Обчислення репутацій веб-сторінки. Обчислювальні Мережі, 33:823–835, 2000.

[24] Дж. Скотт. Повідомлення тенденції: Соціальний аналіз мережі. Соціологія, 22(1) :109–27, 1988.

[25] Дж. Сан, Д. Тао, і С. Фелутсос. Після потоків і графів: динамічний аналіз тензора. У KDD ’06: Слухання 12-го ACM SIGKDD міжнародна конференція з відкриття Знання і гірської промисловості даних, сторінки 374–383 Нью-Йорк, НЬЮ-ЙОРК, США, 2006. ACM.

[26] Дж.-Т. Сан, Н.-Дж. Зен, Н. Луі, У. Лютецій, і З. Чен.: новий підхід до персонального мережевого пошуку. У WWW ’05: Слухання 14-ої міжнародної конференції зі Світової Широкої Павутини, сторінки 382–390, Нью-Йорк, НЬЮ-ЙОРК, США, 2005. ACM.

[27] Е. Ян Дж. Хан, і П. С. Ю. Лінклас: ефективне групування через різнорідні семантичні зв'язки. У VLDB ’06: Слухання 32-ої міжнародної конференції з Дуже великих баз даних, сторінки 427–438. Вклад VLDB, 2006.