УДК 681.3

Глибовець А. М., Марченко О. О., Циганок Д. В., Бабіч О. М.

АЛГОРИТМИ ОБРОБКИ ТЕКСТІВ ВІЛЬНОЇ ФОРМИ ДЛЯ ОТРИМАННЯ ФАКТІВ І ЗВ'ЯЗКІВ МІЖ НИМИ

Інтернет є найбільшим джерелом знань у світі, але сучасні інформаційні технології далекі від ефективного використання цього потенціалу. У статті розглянуто сучасний стан проблеми побудови методів автоматичного та автоматизованого наповнення баз знань шляхом аналітичної обробки простих неструктурованих природномовних текстів.

Ключові слова: база знань, обробка інформації, «макрочитання».

Вступ

У всесвітній мережі Інтернет накопичений величезний обсяг інформації, різні знання – все змішалося у значну кількість неструктурованих, низької якості текстів та медіа-контенту. Якби усю інформацію, опубліковану в Інтернеті, можна було б отримати у структурованому вигляді – більшість проблем, пов'язаних з інформаційним пошуком, були б неактуальні. Тоді внаслідок такого пошуку користувачі отримували б не множину посилань, а одразу відповідь. Постає запитання: чи можливо систематично збирати факти з мережі та компілювати їх у повноцінну машинно-орієнтовану базу знань про сутності світу, їх семантичні властивості, та зв'язки одне з одним?

Цей напрям інженерії знань існує вже протягом певного часу. Універсальні бази знань були об'єктом досліджень у напрямку штучного інтелекту від початку, наприклад CYC [11; 12] або проект *Semantic Web* [4; 14]. На цей час найпомітніші з них – *freebase.com* і *trueknowledge.com* – компіляції величезної кількості сутнісно-та зв'язково-орієнтованих фактів; проект *Dbpedia* [3] (*dbpedia.org*), який зусиллями спільноти збирає RDF трійки суб'єкт-властивість-об'єкт із Вікіпедії та інших подібних джерел; проект *KnowItAll* [10] і його модуль – *TextRunner* [22]; *Kylin/KOG* [19; 20; 21] і *Omnivore* [5], метою яких є отримання довільних зв'язків із природномовних текстів; проект *ReadTheWeb* [7] з амбіційною метою «макрочитання» довільної веб-сторінки; система *sig.ma* [17], яка використовує дані у форматі *RDF* з Інтернету; проект *YAGO* [16; 15], який інтегрує знання з Вікіпедії та *WordNet* (поєднуючи спеціальними зв'язками сторінки з Вікіпедії та вузли-синсети WordNet між собою).

Крім того, з'являються нові пошукові системи, орієнтовані на семантичний пошук і виявлення знань, які зосереджені навколо великих баз знань. Серед таких систем – *wolframalpha.com*, яка обчислює відповідь на наборі баз даних, *entitycube.research.microsoft.com*, що забезпечує динамічне збирання фактів про іменовані сутності (на основі методології *StatSnowball* [23]); *opencalais.com*, яка надає послуги зі структурування документів та веб-сторінок; *kosmix.com* (придбаний *Walmart*), що використовує великі онтології для класифікації запитів і виявлення сутностей, пов'язаних із запитом користувача. Також нещодавно компанія *Google* оголосила, що має намір вводити в обіг семантичний пошук і на частину запитів відповідати безпосередньо на сторінках результатів, без потреби переходити на сайти.

1. Виділення фактів і зв'язків між ними

Збір та обробка інформації з Інтернету прийнято називати *web content mining* [1]. Цей напрямок є частиною більшого напрямку *web mining* (складається з *web content mining, web structure mining* та *web usage mining* – див. таблицю). Своєю чергою, *web mining* є частиною *data mining* (див. рис. 1).

Таблиця. **Напрямки у Web Mining**

Напрямок	***Web usage mining***	***Web structure mining***	***Web content mining***
На вході алгоритму	Логи веб-серверів	Посилання	*Html*-сторінки
На виході	Вподобання користувачів	Взаємозв'язок між сторінками	Інформація та знання

Web content mining у літературі також трапляється як *wrapper induction, web harvesting, web scraping, information extraction, web data extraction*. Найчастіше *web content mining* діляють на такі завдання, як отримання структурованих даних та виділення фактів і зв'язків.

Метою цього дослідження є аналіз завдання отримання фактів та зв'язків між ними для формування бази знань.

Формування бази знань передбачає її проходження через різні стадії життєвого циклу:

1) побудова великої колекції фактів про сутності, класи та зв'язки, отримані з веб-джерел (або з корпоративної інформації, цифрових документів, бібліотеки тощо);
2) фільтрація бази знань на основі оцінки достовірності фактів і видалення недійсних частин або додавання фрагментів, яких не вистачає;
3) посилання запитів до колекції фактів, задля отримання відповідей на розширені запити від модулів знань;
4) ранжування відповідей на комплексні запити або логічні виведення за допомогою статистики інформативності, достовірності, стислості й інших критеріїв;
5) логічні виведення стосовно сутностей та зв'язків для отримання додаткових знань, які не завжди представлені в екстенсіональній формі;
6) супровід та доповнення бази знань новими фактами, які з'являються в Інтернеті.

Рис. 1. Співвідношення напрямків аналізу даних між собою

Нас цікавить 1, 2 та 6 стадія, що безпосередньо стосуються наповнення бази [18].

Ключова технологія для збору знань відома як *information extraction* (IE). IE містить моделі, алгоритми та інструменти для отримання з вебсторінок, текстових джерел, таких слабо структурованих даних, як HTML–таблиці або Вікіпедія, конкретних фактів (унарних, бінарних, або *n*-арних зв'язків). Поширеними методами виділення фактів є пошук відповідності деякому шаблону (*rule-based pattern matching*), обробка природної мови (NLP), а також статистичне машинне навчання (ML). За останні роки в IE зроблено багато великих кроків, методи стали більш масштабованими, а також менше залежати від людського контролю (наприклад, див. : [2; 8; 9; 13] і наведені там посилання).

1.1. Алгоритм CPL (університет Карнегі-Мелоун)

Алгоритм CPL [7] навчає тому, як отримувати категорії та зв'язки між ними з неструктурованого тексту. На першому кроці, на початкових множинах категорій та зв'язків між цими категоріями алгоритм знаходить найімовірніші текстові шаблони (*textual patterns*). Далі використовує ці шаблони для пошуку нових імовірних категорій та зв'язків. Ці знайдені категорії та зв'язки використовують на наступних ітераціях роботи алгоритму для пошуку або уточнення текстових шаблонів.

Алгоритм Coupled Pattern Learner (CPL)
Вхід: *онтологія O, корпус текстів C*
Вихід: *сутності та текстові шаблони для кожного предикату*

for i = 1, 2, … , ∞ **do**
  for_each предикат $p \in O$ **do**
    Видобути нові імовірні сутності / текстові шаблони, використовуючи останні обрані шаблони / сутності;
    відфільтрувати ті, що порушують зв'язок;
    відранжувати ці сутності / шаблони;
    обрати сутності / шаблони з найбільшим рангом;
  end
end

Імовірні сутності ранжуються, використовуючи оцінки точності кожного шаблону *p*:

$$\textbf{точність } (p) = \frac{\sum_{i \in x} кількість(i, p)}{кількість(p)},$$

де *x* – це множина обраних сутностей для предикату, що розглядається. Кількість (*i, p*) – це число скільки разів сутність ***i*** була отримана за допомогою шаблону *p*. Кількість (*i, p*) – це загальна кількість спрацьовувань предикату *p*.

1.2. Алгоритм TextRunner (Вашингтонський університет)

За [22, 6], алгоритм *TextRunner* має дві особливості:

1. Цей алгоритм спроектовано так, щоб одразу на першій ітерації використовувались усі доступні документи, тобто робота здійснюється у «пакетному» режимі (*batch mode*). У цьому

полягає його відмінність від інших систем, що використовують на початку лише частину документів, які містять початкові категорії та зв'язки, тобто запит-орієнтовані системи «на вимогу» (*on-demand query-driven systems*). На перший погляд, підхід «на вимогу» може здаватися привабливішим через використання лише релевантних документів. Проте пакетно-орієнтована техніка дає змогу заздалегідь обчислити деякі шаблони, що підвищує швидкість навчання. Також слід зауважити, що пошукові системи використовують аналогічний підхід.

2. Алгоритм *TextRunner* не намагається поповнити цільові зв'язки чи схеми, проте виявляє їх в процесі обробки. Тоді як системи *KnowItAll, Dipre, Snawball* потребують цільових початкових даних (наприклад, множину початкових пар або множину шаблонів), цей підхід дає можливість отримати цінні дані з мережі Інтернет за відсутності чіткої моделі веб-контенту (наприклад, деякої загальної онтології). Крім того, такий підхід дає змогу екстракторам автоматично отримувати абсолютно нові зв'язки.

Причина цих відмінностей у тому, що на початку роботи алгоритму використовують кілька загальних шаблонів (наприклад, «такий X, як Y» або «X, що містить Y»). Далі виявляють правильно або помилково отримані дані завдяки тренуванню класифікаторів, яким на вхід дається частота, з якою зустрічається та чи інша отримана фраза (категорія або зв'язок).

2. Використання алгоритмів

Частина інформації в Інтернет-просторі представлена у напівструктурованому форматі (таблиці, списки). Одна з робіт (Etzioni, 2004) демонструє, що методи отримання фактів на основі шаблонів та на основі списків (таблиць) можуть бути об'єднані для досягнення істотних покращень в ітераційних методах. *Downey, Etzioni* та *Soderland* (2005) представили імовірнісну модель для використання та навчання множини екстракторів, де вони роблять некорельовані помилки.

Розглянемо на прикладі розробки університету Карнегі-Мелоун, на яких базових принципах можна побудувати систему автоматизованого навчання. Проект NELL – це система безперервного навчання, яка ітераційно виконує два завдання: читання та навчання. Під завданням читання розуміють отримання нових фактів із неструктурованних (текстів) або напівструктурованних джерел. Завдання навчання полягає у формуванні нових шаблонів на основі отриманих фактів для ефективнішого «читання».

Найважливішим принципом при реалізації такого підходу до навчання системи є використання підсистем, які роблять некорельовані помилки. Тоді ймовірність того, що вони всі зроблять одну й ту саму помилку, – дуже низька. Це дає змогу звести до мінімуму втручання людини у процес навчання такої системи.

NELL використовує такі підсистеми для отримання фактів та зв'язків:

- *Coupled Pattern Learner* (CPL) – для обробки тексту довільної форми;
- *Coupled SEAL* (CSEAL) – підсистема для обробки напівструктурованих даних, яка задає запити до веб-сторінок і знаходить для множини фактів із кожної категорії чи зв'язків у таблицях та списках нові екземпляри для відповідних предикатів;
- *Coupled Morphological Classifier* – модуль використовує морфологічні особливості (частину мови, великі літери і т. д.);
- *Rule Learner* – модуль використовує логічні виведення для отримання нових фактів.

В умовах розвитку Інтернет-простору та появи у ньому спільнот певної спрямованості, об'єднаних якоюсь спільною метою (прагнення придбати товари та послуги, спілкування на основі належності до певних соціальних груп тощо), простежуються своєрідні тренди щодо появи у мережі коротких текстів (наприклад «твітів»), які не улягають стандартним алгоритмам. Для розв'язання завдань пошуку та обробки текстової інформації, беручи до уваги нові особливості Інтернет-контенту, з'являється дедалі більше спеціалізованих пошукових систем, які використовують для пошуку напівструктуровані джерела з конкретної тематики (наприклад, оголошення, кіно, товари). Пошукові ж системи загального призначення (*google*, *bing*, *yandex*) більше зорієнтовані на тексти довільної форми. Тому за останніми науковими дослідженнями для розв'язання нових завдань обробки текстової інформації у мережі Інтернет при побудові бази знань доцільно використовувати тексти довільної форми і напівструктуровані джерела (таблиці, списки, сайти регулярної структури).

Висновки

Таким чином, використання алгоритмів виявлення фактів та зв'язків із неструктурованої текстової інформації дає можливість автоматизованого виділення знань за заданим напрямом із величезної кількості різноманітних текстів у мережі Інтернет.

Визначені у цій статті алгоритми дають можливість ефективно здійснювати моніторинг

мережі Інтернет. Це алгоритм CPL (який на основі знайдених найбільш імовірних текстових шаблонів шукає нові ймовірні категорії та зв'язки) й алгоритм TextRunner (у якому реалізовано пакетно-орієнтований підхід, що значно прискорює обробку природномовних текстів та збільшує швидкість навчання цього алгоритму). Наведені приклади більш ранніх робіт на цю тему й посилання на сучасні проекти, що займаються «структуруванням» Інтернет-простору, відображають загальну тенденцію зростання інтересу до ведення досліджень у цій сфері.

З огляду на сказане стають очевидними подальші наукові дослідження з визначення нових підходів до обробки природномовних текстів із врахуванням їхніх особливостей, а також розроблення нових алгоритмів за цим напрямом.

Література

1. Яковлева Е., Батыгин В. Извлечение информации из полуструктурированных веб-источников. ШАД, 2011.
2. Agichtein E. Scaling information extraction to large document collections // IEEE Data Eng. Bull. – 2005. – Vol. 28(4).
3. Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z. DBpedia : A nucleus for a web of open data. – ISWC, 2007.
4. Berners-Lee T., Hendler J., Lassila O. The semantic web. Scientific American, 2001.
5. Cafarella M. J. Extracting and querying a comprehensive web database. – CIDR, 2009.
6. Cafarella M. J., Madhavan J., Halevy A. Web-Scale Extraction of Structured Data, 2008.
7. Carlson A., Betteridge J., Wang R. C., Hruschka Jr. E. R., Mitchell T. M. Coupled semi-supervised learning for information extraction. – WSDM, 2010. – Режим доступу: http://rtw.ml.cmu.edu/readtheweb.html. – Назва з екрана.
8. Cunningham H. An Introduction to Information Extraction, Encyclopedia of Language and Linguistics – 2nd Ed. – Elsevier, 2005.
9. Doan A., Gravano L., Ramakrishnan R., Vaithyanathan S. Special issue on information extraction / Vaithyanathan S. (Eds.). – SIGMOD Record. – 2008. – Vol. 37 (4).
10. Etzioni O., Cafarella M., Downey D., Kok S., Popescu A.-M., Shaked T., Soderland S., Weld D. S., Yates A. Web-scale information extraction in KnowItAll. WWW, 2004.
11. Lenat D. B. CYC : a large-scale investment in knowledge infrastructure. – Commun. ACM. – 1995. – Vol. 38(11).
12. Lenat D. B., Guha R. V. Building Large Knowledge-Based Systems. Representation and Inference in the CYC Project.
13. Sarawagi S. Information extraction. Foundations and Trends in Databases, 2008.
14. Staab S., Studer R. Handbook on Ontologies. – 2nd ed. – Springer, 2009.
15. Suchanek F.M., Kasneci G., Weikum G. YAGO: A large ontology from Wikipedia and WordNet. – J. Web Sem. – 2008. – Vol. 6 (3).
16. Suchanek F.M., Kasneci G., Weikum G. YAGO: acoreofsemantic knowledge. WWW, 2007.
17. Tummarello G. SIG.MA : Live views on the web of data. WWW, 2010.
18. Weikum G., Theobald M. From Information to Knowledge : Harvesting Entities and Relationships from Web Sources. – PODS, 2010.
19. Weld D. S., Hoffmann R., Wu F. Using Wikipedia to bootstrap open information extraction. – SIGMOD Record. – 2008. – 37 (4).
20. Wu F., Weld D. S. Autonomously semantifying Wikipedia. – CIKM, 2007.
21. Wu F., Weld D. S. Automatically refining the Wikipedia infobox ontology. WWW, 2008.
22. Yates A., Banko M., Broadhead M., Cafarella M. J., Etzioni O., Soderland S. TextRunner : Open information extraction on the web. – HLT-NAACL, 2007.
23. Zhu J., Nie Z., Liu X., Zhang B., Wen J.-R. StatSnowball : a statistical approach to extracting entity relationships. WWW, 2009.

A. Glybovets, O. Marchenко, D. Tsyganok, O. Babich

FREE-FORM TEXT PROCESSING ALGORITHMS FOR EXTRACTING ENTITIES AND RELATIONS

Internet is the largest source of knowledge in the world, but modern information technologies are far from effective use of this potential. We consider the current state of the problem of constructing methods for automatic filling knowledge bases by analytical processing unstructured natural language texts.

Keywords: knowledge base, information retrieval, «macroreading».

Матеріал надійшов 11.04.2012