УДК 004.4’414

Б. І. Мороз, доктор технічних наук,
декан факультету інформаційних
та транспортних систем і технологій
Університету митної справи та фінансів
Д. Є. Костенко, аспірант кафедри інформаційних
систем та технологій Університету митної
справи та фінансів, провідний фахівець
навчальної лабораторії інформаційних систем
та процесів
В. В. Костенко, старший викладач кафедри
інформаційних систем та технологій
Університету митної справи та фінансів
І. В. Лавренюк, старший викладач кафедри
інформаційних систем та технологій
Університету митної справи та фінансів

ВИКОРИСТАННЯ РЕГУЛЯРНИХ ВИРАЗІВ ДЛЯ ОПТИМІЗАЦІЇ ПОШУКУ В НЕСТРУКТУРОВАНИХ ДОКУМЕНТАХ

Розглянуто проблеми та можливості управління знаннями на основі масивів неструктурованих текстів у документах. Окреслено деякі підходи до застосування регулярних виразів (з використанням метасимволів та квантифікаторів) на базі онтологій і перспективи використання даного підходу для застосування в експертних системах.

Ключові слова: *інформація; квантифікатори; метасимволи; онтології; регулярні вирази.*

There are examine the problems and management possibilities by knowledge on the basis of the unstructured texts arrays in documents. Examines some approaches for using the regular expressions (with using metacharacters and quantifiers) based on ontologies and perspectives of this approach for using in expert systems.

Key words: *information; quantifiers; metacharacters; ontologies; regular expressions.*

Постановка проблеми. Аналіз темпів зростання кількості електронних документів і методів їх обробки наочно показує, що традиційні механізми роботи з електронними документами не задовольняють потреби сучасного користувача [1]. Тому необхідні нові підходи до інтелектуального пошуку й аналізу електронних документів, їх інтеграції в інформаційні системи [1].

Регулярні вирази набули поширення з часів їх упровадження в операційні системи. З їх допомогою розв’язується безліч задач обробки текстової інформації, а саме [2]:

– пошук фрагментів тексту, що відповідають шаблону;
– перевірка тексту на відповідність шаблону;
– заміна тексту за шаблоном.

Нині найактуальніші проблеми такі [1].

1. Експонентне зростання кількості документів є причиною ускладненого пошуку необхідних документів та їх організації у вигляді структурованих за змістом сховищ [3]. Зі збільшенням простору пошуку пропорційно зростає і кількість документів у відгуку пошукової системи.

2. Відсутність стандартизованих механізмів семантичного індексування також згубно впливає на ефективність роботи з електронними документами. Більшість сучасних технологій підготовки і роботи з документами (текстові редактори, HTML) орієнтовані на організацію зручної роботи з інформацією для людини.

3. Неструктурований характер інформації більшості електронних документів не дозволяє застосувати традиційні механізми її обробки й аналізу. Неструктурована інформація становить значну частину сучасних електронних документів, основні знання розташовуються саме в таких документах. Для розв'язання подібного роду проблем необхідно розширити поняття традиційного документа: з документом слід пов'язати метадані, що дозволяють інтерпретувати й обробляти інформацію, яка зберігається в цьому документі, тобто включити в документ інформацію, яка описує структуру і семантику його змісту [1].

Проблема також полягає в тому, щоб зробити пошук динамічним і зручним для користувача. Для будь-якого типу запиту, що виникає в практичній діяльності, має бути знайдено адекватні знання в інформаційному просторі. При цьому мова для формулювання пошукової вимоги не повинна бути занадто складною.

Аналіз останніх досліджень і публікацій. Поняття онтології, яке було запозичене з філософії, зараз активно застосовується в штучному інтелекті й інформатиці. У філософії онтологія вивчає категорії буття, які існують або можуть існувати. У штучному інтелекті онтології згадуються в контексті з такими поняттями, як концептуалізація, знання, подання знань, що ґрунтуються на знаннях системи.

Помічено, що в дослідженнях деякі вчені намагаються дати неформальні визначення, а інші описують онтології на основі понять і конструкцій логіки й математики. Але, незважаючи на те, що побудовано безліч різних онтологій і збільшується сфера їх застосування, дотепер немає точного визначення цього поняття стосовно області штучного інтелекту [4].

Основу онтології становить безліч представлених у ній термінів. Утім, не тільки термінів. До онтологічної сукупності входять також відомості про предметні області, про області визначень і т. д.

Х. Такеда ставить онтології в центр проблеми організації знань, тому що в кожній області можуть існувати різні розуміння тих самих термінів. У цьому випадку онтологія використовується для структурування інформації, залишаючись посередником між людино-машинно-орієнтованим і машинно-машинно-орієнтованим рівнем подання інформації. Тоді онтологія визначається як "угода", "контракт" про деяку сферу інтересів для досягнення певних цілей [5].

Трохи інший підхід декларує Н. Гуаріано. Для встановлення взаєморозуміння про знання, подані деякою мовою, зокрема логічною, на думку Н. Гуаріано, онтологія має характеризувати концептуалізацію, обмежуючи можливі значення предикатів і функцій. У цьому розумінні онтологія – це логічна теорія, аксіоми якої обмежують інтерпретації нелогічних символів мови [6].

На думку Т. А. Гаврилової, онтологія – це структурна специфікація деякої предметної області, її формалізоване подання, яке включає словник (або імена) покажчиків

на терміни предметної області та логічні зв'язки, що описують співвідношення термінів. Онтології утворюють словник для подання та обміну знаннями про деяку предметну область і множину зв'язків, установлених між термінами в цьому словнику [7].

Важливість підходу, пов'язаного з онтологіями, обумовлена також тим, що знання, яке не описано і не тиражовано, насамкінець стає застарілим і непотрібним. Навпаки, знання, яке поширюється, є генератором нових знань [7].

Регулярні вирази – засіб пошуку в тексті на основі шаблонів. Шаблон описує закономірність, якій мають підкорятися шукані послідовності символів у тексті.

Використання регулярних виразів дозволяє враховувати різні структурні перестановки всередині тексту, різні варіанти написання одних і тих самих понять, а також відношення синонімії. Однак у класичних регулярних виразах немає знань про онтологію, до якої належить текст, що перевіряється. Як наслідок, ручне складання класичного регулярного виразу для перевірки відкритих тестів практично не можливе, оскільки в підсумку воно зводиться до повного перебирання всіх можливих варіантів відповіді, що навіть з урахуванням використання регулярних конструкцій для скорочення виразу і перевірки лише найімовірніших варіантів відповіді у роботі з текстом на природній мові потребує величезних затрат [2].

У разі зміни знань про предметну область може знадобитися перегляд усіх регулярних виразів, порушених цими змінами [2].

Одне з можливих розв'язань зазначеної проблеми – привнесення семантичної складової в регулярні вирази, що дозволить автоматично та гнучко перебудовувати їх у разі зміни знань про предметну область [2].

Мета статті – аналіз, дослідження та покращання можливостей ефективного пошуку інформації у сукупності неструктурованих документів. Дослідження перспектив створення спеціалізованої програмної системи з використанням спеціальних методів і алгоритмів морфологічного й синтаксичного аналізу неструктурованих текстів у службових документах, оптимізація пошуку інформації та її статистичної обробки. Розглядається питання використання онтологій і регулярних виразів.

Виклад основного матеріалу. У величезних масивах інформації мають працювати пошукові системи, які б забезпечували користувачу швидкий і точний результат. Вважається, що системи, які ґрунтуються на онтологічному підході, досконаліші та відповідають потребам користувачів.

Онтологія являє собою деякий опис погляду на світ стосовно конкретної сфери інтересів. Цей опис складається з термінів і правил використання цих термінів, що обмежують їхні значення в рамках конкретної області. На формальному ж рівні онтологія – це система, що складається з набору понять і тверджень про ці поняття, на основі яких можна будувати класи, об'єкти, відносини, функції й теорії. Онтологія як приклад загальної уяви про семантику області сприяє встановленню коректних зв'язків між значеннями елементів області, створюючи умови для їх спільного використання.

Процес аналізу текстової інформації поділяється на такі рівні:

1) графематичний аналіз (токенізація, визначення іменованих сутностей);

2) морфологічний аналіз (нормалізація, стеммінг);

3) синтаксичний аналіз (побудова дерева синтаксичного розбору);

4) семантичний аналіз (побудова семантичного графа).

Онтології необхідно застосовувати в ролі посередника між користувачем і процесом пошуку, між процесом пошуку і пошуковою системою. Для побудови онтології потрібне формальне декларативне подання чітко організованих конструкцій, які міс-

тять словник термінів тематичної області, опис визначень цих термінів, наявні взаємозв'язки між ними і взагалі теоретично можливі й неможливі взаємозв'язки.

Взаємодія з онтологією відбувається на таких етапах:

1) обробка та аналіз запиту;

2) розкладання запиту на складові частини;

3) перевірка інформації на старіння та відповідність;

4) перевірка на старіння інформації, яка була обрана зі сховища даних.

Робота з регулярними виразами має проводитися в ті моменти, коли запит аналізується та розкладається на складники.

У регулярних виразах використовуються звичайні (літерали) та спеціальні символи (метасимволи). Більшість символів у регулярному виразі позначає самих себе, за винятком спеціальних символів (табл. 1).

Таблиця 1

Метасимволи

[ ]	^	/	$.	|
?	*	+	( )	{ }

Перед ними може стояти символ \ (обернений слеш) для подання їх як символів тексту. Це так званий процес "екранування".

Одним з найвикористовуваніших метасимволів є крапка. Він позначає один будь-який символ, включно із символом нового рядка в деяких реалізаціях регулярних виразів.

Наведемо приклад:

/пол.с/

Даний регулярний вираз відповідатиме словам "полюс" та "поліс".

Але таким підходом не можна обмежувати роботу з регулярними виразами. Треба також розуміти, що в регулярних виразах широко використовується квантифікація, тобто пошук послідовностей символів. При цьому застосовуються квантифікатори, які розташовані після символу, символьного класу або групи і визначають, скільки разів попередній вираз може зустрічатися.

Слід ураховувати, що квантифікатор може стосуватися більш ніж одного символу в регулярних виразах, тільки якщо це символьний клас або група (табл. 2).

Таблиця 2

Квантифікатори

Подання	Кількість повторень	Приклад	Відповідність
{*n*}	Рівно *n* разів	colou{3}r	colouuur
{*m*,*n*}	Від *m* до *n* включно	colou{2,4}r	colouur, colouuur, colouuuur
{*m*,}	Не менше *m*	colou{1,}r	colour, colouur, colouuur тощо
{,*n*}	Не більше *n*	colou{,2}r	color, colour, colouur

Крім цифрових квантифікаторів, також треба використовувати символьні (табл. 3).

Таблиця 3

Символьні квантифікатори

Подання	Кількість повторень	Еквівалент	Приклад	Відповідність
*	Нуль або більше	{0,}	colou*r	color, colour, colouur тощо
+	Одне або більше	{1,}	colou+r	colour, colouur тощо (але не color)
?	Нуль або одне	{0,1}	colou?r	color, colour

Часто використовується послідовність /.*/ для позначення будь-якої кількості будь-яких символів між двома частинами регулярного виразу. Символьні класи в поєднанні з квантифікаторами дозволяють установлювати відповідності з реальними текстами: стовпцями цифр, телефонами, поштовими адресами, елементами HTML-розмітки тощо. Якщо символи { } не утворюють квантифікатор, їх спеціальне значення ігнорується. В деяких реалізаціях квантифікаторам у регулярних виразах відповідає максимально довгий рядок із можливих. В окремих випадках це може бути проблемою. Наприклад, часто очікують, що вираз (<.*>) знайде в тексті теги HTML. Однак якщо в тексті є більше одного HTML-тега, то цьому виразу відповідає цілий рядок, що містить множину тегів:

<*p*> Сторінка <*b*> Курс </*b*> Матеріал </*p*>

Можна запропонувати такий підхід. Є сенс визначити тип такого квантифікатора як “ледачий” – більшість реалізацій регулярних виразів дозволяють це зробити, додавши після нього знак питання (табл. 4). Але використання “ледачих” квантифікаторів може спричинити зворотну ситуацію, коли виразу відповідає занадто короткий і (що найкритичніше) порожній рядок.

Таблиця 4

“Жадібний” та “ледачий” квантифікатори

Жадібний	Ледачий
*	*?
+	+?
{*n*,}	{*n*,}?

Також спільною проблемою як “жадібних”, так і “ледачих” виразів є точки повернення для перебирання варіантів виразу. Точки ставляться після кожного повтору, або ітерації квантифікатора. Якщо інтерпретатор не знайшов відповідності після квантифікатора, то він починає повертатися за всіма встановленими точками, перераховуючи звідти вираз по-іншому.

Нарешті, існує так звана “ревнива” квантифікація, яка, на відміну від звичайної “жадібної”, не тільки намагається знайти максимально довгий варіант, але ще й не дозволяє алгоритму повертатися до попередніх кроків пошуку для того, щоб знайти можливі відповідності для решти регулярного виразу. Використання “ревнивих” квантифікаторів (табл. 5) збільшує швидкість пошуку особливо в тих випадках, коли рядок не відповідає регулярному виразу. Крім того, “ревниві” квантифікатори можуть бути використані для виключення небажаних збігів.

Таблиця 5

“Жадібний” і “ревнивий” квантифікатори

Жадібний	Ревнивий
*	*+
?	?+
+	++
{*n*,}	{*n*,}+

Тепер спробуємо розглянути приклади виконання типових завдань з опрацювання тексту з використанням регулярних виразів. Наприклад: пошук рядків, в яких є або, навпаки, немає певних слів. Якщо потрібно знайти рядок, в якому зустрічається одне з кількох слів, то достатньо використовувати регулярні вирази з альтернативами, а саме такий вираз:

/^.**b*(один|два|три)*b*.*$/

Будуть знайдені всі рядки, що міститимуть будь-яке зі слів “один”, “два” або “три”. Першим зворотним викликом буде знайдене в рядку слово.

Якщо рядок міститиме два або всі три слова, то тільки останнє з них (крайнє справа) вважатиметься першим зворотним викликом. Це відбувається тому, що зірка * вмикає “жадібний” пошук.

Можна зробити першу зірку “ледачою”:

/^.*?*b*(один|два|три)*b*.*$/,

тоді зворотний виклик міститиме перше слово зліва.

Якщо рядок має містити всі три потрібні слова, то слід використати інструмент перегляду вперед.

/^(?=.*?*b*один*b*)(?=.*?*b*два*b*)(?=.*?*b*три*b*).*$/.

Такий регулярний вираз знайде тільки такі рядки з тексту, що містять усі три слова: “один”, “два” і “три”.

Усі три перегляди вперед мають бути позитивними, щоб рядок загалом відповідав умові регулярного виразу.

Якщо потрібно, щоб рядок не містив певного тексту, слід використовувати негативний перегляд вперед.

/^((?!*b*один*b*).)*$/.

Завдяки цьому регулярному виразу буде знайдено цілий рядок, що не містить слова “один”.

Можна помітити цікаву річ: водночас повторено негативний перегляд вперед і крапку, оскільки для позитивного перегляду вперед потрібно лише знайти місце в тексті, де регулярний вираз спрацює, а для негативного перегляду вперед треба перевірити всі без винятку позиції символу в рядку. Тобто слід переконатися, що регулярний вираз не дає позитивного результату пошуку всюди, а не тільки в якомусь одному місці. Нарешті, можна об'єднати кілька позитивних і негативних переглядів вперед.

Наприклад:

/^(?=.*?*b*обов'язковий*b*)(?=.*?*b*необхідний*b*)((?!непотрібний|факультативний).)*$/.

Цей регулярний вираз знаходитиме рядки, в яких є слова "обов'язковий" і "необхідний", але немає слів "непотрібний" чи "факультативний".

Не менш важливим завданням автоматичного опрацювання тексту може бути пошук двох близько розташованих слів. Деякі пошукові інструменти, крім використання логічних операторів "і/або", також мають спеціальний оператор з умовною назвою "поряд", "біля".

Пошукова умова "слово1 біля слово2" знаходить усі входження "слово1" і "слово2", які перебувають на певній відстані у тексті одне від одного. Відстань і є кількістю слів. Ця кількість залежить від інструменту пошуку і часто може бути налаштована користувачем. Подібне завдання можна реалізувати і за допомогою регулярних виразів. У цьому випадку регулярний вираз досить нескладний: перше слово, певна кількість невідомих слів і друге слово. Невідомі слова можна описати символьним класом *w*+. Пробіли та інші символи між словами можна описати символьним класом *W*+.

Загальний вигляд регулярного виразу:

/*b*слово1*W*+(?:*w*+*W*+){1,6}?слово2*b*/.

Квантифікатор {1,6}? вказує, що відстань між словами може становити від 1 до 6 слів. Якщо слова можуть зустрічатися й у зворотному порядку, то це слід зазначити, а саме:

/*b*(?:слово1*W*+(?:*w*+*W*+){1,6}слово2|слово2*W*+(?:*w*+*W*+){1,6}слово1)*b*/.

Якщо потрібно знайти пару з двох слів з певного списку, то РВ може мати вигляд:

/*b*(слово1|слово2|слово3)(?:*W*+*w*+){1,6}?*W*+(слово1|слово2|слово3)\b/.

Цей регулярний вираз також знайде повтор того самого слова.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку. Використання регулярних виразів дозволяє гнучко враховувати різні структурні перестановки всередині неструктурованого документа. Звісно, що це має відбуватися з урахуванням онтологічної складової частини.

Завдяки цьому можна використовувати і складати прості регулярні вирази в рамках конкретної предметної області.

Перспектива використання регулярних виразів на базі онтологій полягає не тільки в пошуку окремих слів або словосполучень, але й у можливості здійснення пошуку точних фраз, фраз зі списку, у пошуку слова в різних варіантах написання або зі спеціальними нестандартними символами, у пошуку слова зі змінними символами. Звісно, що це не є повним списком сфери застосування регулярних виразів.

Список використаних джерел:

1. Ланин В. Онтологии как основа функционирования систем обработки электронных документов / В. Ланин // Материалы конференции с международным участием “Знания-Онтологии-Теории (ЗОНТ-09)”. – Новосибирск, 2009. – Т. 2. – С. 173–177.

2. Мерзляков Д. А. Генерация регулярных выражений для автоматизации проверки тестов открытого характера [Электронный ресурс] : материалы 5-й международной студенческой электронной научной конференции “Студенческий научный форум” – 15.02–31.03, 2013. / Д. А. Мерзляков. – Режим доступа : http://www.scienceforum.ru/2013/147/2470

3. Ефремов В. Search 2.0: огонь по “хвостам” / В. Ефремов // Открытые системы. СУБД. – 2007. – № 8. – С. 72–74.

4. Захарова И. В. Об одном подходе к реализации семантического поиска документов в электронных библиотеках / И. В. Захарова // Вестник УГАТУ. – 2009. – Т. 12. – № 1 (30): Серия “Управление, вычислительная техника, информатика”. – С. 133.

5. Takeda H. Collaborative development and Use of Ontologies for Design / H. Takeda, M. Takaai, T. Nishida // Proceedings of the Tenth International IFIP WG 5.2/5.3 Conference PROLAMAT 98. – Trento, Italy. – September 9–12.

6. Guarino N. OntoSeek: Content-BasedAccess to the Web / N. Guarino, C. Masolo, G. Vetere // IEEE Intelligent Systems. – May/June 1999. – P. 70–80.

7. Гаврилова Т. А. Онтологический подход к управлению знаниями при разработке корпоративных информационных систем / Т. А. Гаврилова // Новости искусственного интеллекта. – 2003. – № 2. – С. 24–30.

✧ ✧ ✧ ✧ ✧ ✧ ✧