АЛГОРИТМ ПОШУКУ ІНФОРМАЦІЇ НА ОСНОВІ МОДЕЛІ АСОЦІАТИВНОЇ ПАМ'ЯТІ ЛЮДИНИ

Олег Бісікало [1]

[1]Вінницький державний аграрний університет
Сонячна, 3, Вінниця, 21008, Україна, тел.: (0432) 43-72-30, E-Mail: inter@vsau.org

Анотація

Розглядається алгебраїчна модель асоціативної пам'яті людини, що поєднує концепти образів та асоціативних зв'язків між ними. В рамках моделі будується семантична мережа образів за допомогою аналізу (читання) синтагматичних конструкцій (речень), які об'єднані у текст. Цим самим фактично створюється індексний файл спеціального типу, який додатково враховує змістовні характеристики тексту, причому в якості критерію синонімії сполучень образів застосовано сумарну силу асоціативного зв'язку. В роботі обґрунтовуються основи алгебраїчної системи і ті складові її сигнатури, які забезпечують побудову ієрархічної структури асоціативної пам'яті та проведення пошуку за допомогою такої структури. Формальний алгебраїчний підхід проілюстровано програмною реалізацією основних функцій пошуку у відомій системі LISP-програмування DrScheme. На відміну від існуючих методів пошуку за послідовністю символів у вигляді окремих слів або словосполучення алгоритм, що пропонується, використовує особливості асоціативних зв'язків між словами тексту та, внаслідок цього, автоматично розширює маску пошуку.

Вступ

Невпинне зростання кількості користувачів Інтернету призводить до збільшення в геометричній прогресії обсягу доступної для них інформації, що надає високого ступеню актуальності алгоритмам пошуку та побудованим на їх основі Інтернет-технологіям. Внаслідок величезного попиту на послуги пошуку в Інтернеті популярні пошукові сервери вже давно вийшли з початкової дослідницької стадії та перетворилися у потужні комерційні підприємства. Більшість спеціалістів прогнозує подальше стрімке розширення цього сегменту ринку інформаційних технологій [1], а тому загальною проблемою можна вважати прискорення пошуку потрібної користувачам інформації в Інтернеті.

Для побудови сучасних інформаційно-пошукових систем використовується відомі математичні методи сортування, в тому числі на основі бінарних дерев, а також класична технологія баз даних з різного типу індексними файлами [2]. Відправною точкою усіх цих методів є пошук послідовності символів, якими найчастіше є слова мови. Звідси випливає обмеженість результатів пошуку, оскільки в них не враховуються важливі для людини змістовні зв'язки між словами. Тому необхідно інтелектуалізувати пошукові методи, що вже певний час є предметом досліджень фахівців в області інформатики. Підтвердженням цього є поява нових напрямків, пов'язаних з пошуком, а саме тематична класифікація, контекстно-залежне анотування, фактографічний пошук, агрегація новин [3]. Значна частина опублікованих праць присвячена створенню синтаксичних аналізаторів тексту, застосування яких навіть без врахування семантики вже зараз відкриває якісно нові можливості для інформаційно-пошукових систем [4]. Проте не вирішеним натепер є саме семантичний аналіз тексту, мету якого можна узагальнити у вигляді – шукати не символи (послідовність знаків), а значення змісту символів.

Відомим методом формалізації семантики предметної галузі є застосування бази знань у вигляді семантично-фреймової мережі, що моделює механізм взаємодії нейронних ансамблів та пірамідних нейронів людини [[5]]. Оскільки асоціативно-проективні структури є природним аналогом бази знань людини, логічною буде спроба побудови моделі асоціативної пам'яті саме на таких засадах. Експериментально доведено, що розуміння фрази найбільш просто здійснюється у випадку прямої комунікації подій, а змістовним ядром тоді можна вважати тему і рему тексту [[5]]. Окрім цього, в моделі необхідно також врахувати об'єктивне існування у людини умовного рефлексу на значущі слова (терміни предметної галузі).

В [[6]] показано, що формально окреслене коло задач може бути представлене як послідовність відомих моделей та алгоритмів оброблення ієрархічних структур в рамках функціонального програмування. Тому актуальною будемо вважати задачу побудову моделі асоціативної пам'яті людини та алгоритму пошуку на її основі, в якому враховуються семантичні зв'язки між словами тексту. Конкретизація семантичного пошуку у тексті за змістом словосполучення, наприклад, з двох слів означає знаходження таких речень, в яких:

- два слова повністю співпадають з маскою пошуку (базовий варіант);
- одне чи два слова відрізняються відмінками, відмінами, часом чи іншими граматичними особливостями від базового варіанту;
- одне чи два слова відрізняються частинами мови від базового варіанту;
- головне і підлегле слово у словосполученні можуть помінятися місцями;
- між потрібними словами у реченні можуть знаходитися інші слова.

Формалізація

Для вирішення поставленої задачі пропонується використати алгебраїчну систему:

$$Algebra = \langle B;\Omega \rangle, \quad (1)$$

що складається з основ $$B = \{Image, Links - syntagma, Long - memory\} \quad (2)$$

та операторів $$\Omega = \{IF, OP\}. \quad (3)$$

Основи B відповідають текстовому вигляду інформації, що є предметом пошуку, та складаються з списку образів (слів) Image та списку речень Long-memory. До списку Image в першу чергу додаються ті слова, з яких складається текст, а структура Long-memory, як модель довготермінової пам'яті людини, представляє собою формальне відображення синтагм (речень), послідовність яких будує текст. Можливість перетворення речення мови у ієрархічний список забезпечується за допомогою врахування концептів слова як символьного позначення образу:

$$Image = \{OQ, O, N, M, MQ\}, \quad (4)$$

де OQ – якість об'єкту; O – об'єкт; N – поняття; M – метод; MQ – якість методу. Врахувавши, що

$$O = \{Ob, Su\}, \quad (5)$$

$$OQ = \{ObQ, SuQ\}, \quad (6)$$

$$M = \{Wh, T, H\}, \quad (7)$$

де Ob – об'єкт дії; Su – суб'єкт дії; ObQ – якість об'єкту; SuQ – якість суб'єкту; Wh – обставина місця; T – обставина часу; H – обставина дії, можна формально представити синтагму як елемент списку Long-memory наступного дерева (рис.1):

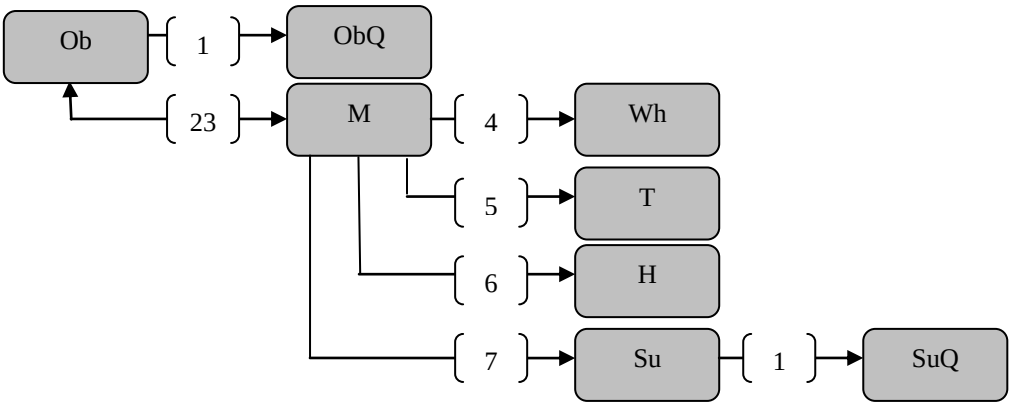

Рис.3. Граф дерева взаємозв'язку слів у складі синтагми.

Зображені на рис.1 цифрами зв'язки між термінами в синтагмі означають можливу роль слова у реченні, а саме:

$$Links - syntagma = \{1,2,3,4,5,6,7\}, \quad (8)$$

де 1 – визначення; 2 – присудок; 3 – підмет; 4 – обставина місця; 5 – обставина часу; 6 – обставина; 7 – додаток.

До складу операторів Ω входять предикати IF та операції OP, які дозволяють з відомих компонентів трьох основ побудувати семантичну мережу Assoc-memory з метою накопичення інформації щодо асоціативних зв'язків між словами тексту. Формалізація моделі асоціативної пам'яті людини Assoc-memory забезпечується за допомогою спискового представлення ускладненої матриці інцидентності, в якій кожному образу з Image відповідає відсортований по зменшенню список асоційованих з ним образів у вигляді підсписків тих синтагм з Long-memory, в яких даний зв'язок між образами мав місце. Операціями, що забезпечують оновлення бази знань ЕС за рахунок зовнішніх фреймів є:

$$OP = \begin{Bmatrix} Convolution, Add - event, Deconvolution, Add - image, \\ Add - assoc, Find - word, \mathrm{Print} - assoc, \mathrm{Print} - assoc - memory \end{Bmatrix}, \quad (9)$$

де Convolution – запис тексту предметної галузі у вигляді нелінійного списку Long-memory; Add-event – додавання інформації з синтагми до семантичної мережі Assoc-memory; Deconvolution – розгортка формалізованого списку синтагми за допомогою вербальних концептів Image; Add-image – додавання нового образу в список Image; Add-assoc - додавання нової асоціації до семантичної мережі ; Find-word – пошук певного образу в синтагмах Long-memory; Print-assoc – виведення ряду асоційованих образів з певним образом; Print-assoc-memory – виведення всієї семантичної мережі Assoc-memory.

Програмна реалізація алгоритму

В якості тестового прикладу для відлагодження програмного коду було використано текст з [8]: «Иванов, работающий в НИИ, был командирован с целью согласования ТЗ, чему он был рад. Остаток командировочного фонда после указанной командировки составил 1000 рублей, что Иванову известно. Следствием рассматриваемой командировки явилось начало работ».

В системі LISP-програмування DrScheme ці речення та потрібні для них образи формалізовано у вигляді функцій *long-memory* та *image* (додаток А). Третю основу *links-syntagma* та головні операції ОР сигнатури алгебраїчної системи також реалізовано у вигляді функцій DrScheme та представлено у додатку В. Загальний результат роботи функції *Print-assoc-memory* для тестового прикладу у вигляді повного списку асоціативної пам'яті представлено у додатку С, а у вигляді відповідного графу – у додатку D. Можна вважати, що поставлену задачу розв'язано, оскільки пошук слова «Командирован» призвів до знаходження трьох речень - e21, e23 та e24.

Висновки

Таким чином, підхід до побудови алгоритму пошуку на основі моделі асоціативної пам'яті, що пропонується, дозволяє досягти вищої якості пошуку окремих слів та словосполучень. Алгебра (1÷9) дозволяє генерувати алгоритми знаходження образів, які моделюють процес взаємодії довготермінової та асоціативної пам'яті людини. Якщо експерти-розробники бази знань експертної системи попередньо визначають склад тезаурусу в прикладній галузі, то використання алгебри (1÷9) забезпечує в процесі розробки та функціонування цієї системи накопичення асоціативних зв'язків як в семантичній мережі тезаурусу, так і в межах окремих текстів фахового призначення. Характерною особливістю підходу, що пропонується, є поліморфність операторної частини стосовно елементів списку Image незалежно від джерела їх походження. Цим самим забезпечується включення до семантичної мережі Assoc-memory не тільки фахової термінології, але й понять, що виражають представлення «здорового глузду».

Широке впровадження семантичного підходу до використання в інформаційно-пошукових системах може бути досягнуто за умови автоматизації процесу введення інформації з нових текстів в основи *long-memory* та *image* алгебраїчної системи. Таким чином, основними перспективами досліджень в окресленому напрямку є врахування синтаксичних особливостей кожної із мов у функціях Convolution та Add-event.

Література:

[1] http://research.metric.ru

[2] *Joachims T.* Making large-scale support vector machine learning practical // Advances in Kernel Methods: Support Vector Machines / B.Scholkopf, C. Burges, A. Smola (eds.) - MIT Press: Cambridge, MA – 1998.

[3] Плешко В.В., Ермаков А.Е., Голенков В.П., Поляков П.Ю. Российский семинар по Оценке Методов Информационного Поиска. Труды второго российского семинара РОМИП'2005. (Ярославль, 6 октября 2005г.) – Санкт-Петербург: НИИ Химии СПбГУ, 2005, - 226 с.

[4] http://www.osp.ru/pcworld/2002/09/086.htm

[5] Амосов Н.М., Куссуль Э.М., Касаткин А.М., Касаткина Л.М. Стохастические нейроподобные сети с ансамблевой организацией. – Киев, 1989. – 30 с. – (Препринт./ АН УССР. Институт кибернетики им. В.М. Глушкова; 89 – 25)

[6] Лурия А.Р. Язык и сознание. Под редакцией Е.Д.Хомской. – М., Издательство Московского университета, 1979. – 320 с.

[7] Бісікало О.В. Проектування процесів дистанційного навчання на основі формалізації пізнавальної діяльності людини // Інформаційні технології та комп'ютерна інженерія – 2005. - № 3 – с.274-280.

[8] Искусственный интеллект: В 3-х кн. Кн.1. Системы общения и экспертные системы: Справочник / Под ред. Э. В. Попова. – М.: Радио и связь, 1990. – 464 с.

Додаток А.

```
(module complex-sentences-data mzscheme
(define *image* '(
i50 (() ИВАНОВ () () ())
i51 (() РАД РАДОСТЬ РАДОВАТЬСЯ РАДОСТНО)
i52 (ЦЕЛЕВОЙ () ЦЕЛЬ ЦЕЛИТЬ ЦЕЛЬНО)
i53 (() () СОГЛАСОВАНИЕ СОГЛАСОВЫВАТЬ СОГЛАСОВАННО)
i54 (() () РАБОТА РАБОТАТЬ ОТРАБОТАННО)
i55 (() НИИ () () ())
i56 (ИЗВЕСТНЫЙ () ИЗВЕСТИЕ ИЗВЕСТИТЬ ИЗВЕСТНО)
i57 (ОСТАТОЧНЫЙ () ОСТАТОК ОСТАВИТЬ ОСТАТОЧНО)
i58 (ФОНДОВЫЙ () ФОНД () ())
```

```
i59 (() () ПОСЛЕ () ПОСЛЕ)
i60 (СОСТАВЛЕННЫЙ ()  СОСТАВИЛ  СОСТАВИЛ СОСТАВЛЕННО)
i61 (УКАЗАННЫЙ  ()  УКАЗАНИЕ  УКАЗАТЬ УКАЗАННО)
i62 (()  ()  НАЧАЛО  НАЧИНАТЬ НАЧАЛЬНО)
i63 (()  ()  ЯВИЛЕНИЕ  ЯВИЛОСЬ ЯВНО)
i64 (СЛЕДСТВЕННЫЙ ()  СЛЕДСТВИЕ  СЛЕДИТЬ СЛЕДСТВЕННО)
i65 (РАССМОТРЕННЫЙ () РАССМОТРЕНИЕ РАССМАТРИВАТЬ РАССМОТРЕННО)
i66 (КОМАНДИРОВОЧНЫЙ () КОМАНДИРОВКА КОМАНДИРОВАТЬ ())
i67 (() ТЗ () () ())
i68 (()  РУБЛЬ  () () ())
i69 (() ТЫСЯЧА ТЫСЯЧА () ())
True (ДЕЙСТВИТЕЛЬНЫЙ () ДЕЙСТВИТЕЛЬНОСТЬ () ДЕЙСТВИТЕЛЬНО)
))
 (define *long-memory*
  (make-immutable-hash-table '(
   (e20((i50 (2 i51))(i51 (3 i50) (Чему? e21))))
   (e21((True (2 i66))(i66 (3 True) (7 i50) (7 i52)) (i52 (7 i53)) (i53 (7 i67)) (i50 (1 i54)) (i54 (4 i55))))
   (e22 ((True (2 i56))(i56 (3 True) (Кому? i50) (Что? e23))))
   (e23 ((i57 (2 i60) (Чего? i58)) (i58 (1 i66)) (i60 (7 i57) (Когда? i59) (Сколько? i69)) (i69 (Чего? i68)) (i59
(Чего? i66)) (i66 (1 i61))))
   (e24((i62 (2 i63) (Чего? i54)) (i63 (3 i62) (7 i64)) (i64 (Чего? i66)) (i66 (Какой?  i65))))
)))
(provide *image* *long-memory*))
```

Додаток B

```
(define *links-syntagma* (make-immutable-hash-table '(
(Какой? 1) (Какая? 1) (Какое? 1) (Какие? 1) (Каком? 1)
(Что_делать? 2) (Что_делает? 2) (Что_делают? 2) (Что_делаешь? 2) (Что_сделал? 2)
(Что_будешь_делать? 2) (Что_делал? 2)
(Кто? 3) (Что_? 3)
(Где? 4) (Откуда? 4) (Куда? 4)
(Когда? 5) (В_какое_время? 5) (Сколько_времени? 5) (Как_долго? 5)
(Как? 6) (Каким_образом? 6) (Сколько? 6) (Отчего? 6)
(Кого? 7) (Что? 7) (Чего? 7) (Кому? 7) (Чему? 7) (Кем? 7) (Чем? 7) (О_ком? 7) (О_чем? 7)
)))
(define (find-word img link-type)
  (if (syntagma? img)
      (list link-type (syn->sentence img))
      ((link->find (normalize-link link-type)) (find-sublist img *image*))))
(define (print-assoc l)
  (let ((assoc-s ""))
        (for-each
         (lambda (assoc)
           (set! assoc-s
                 (string-append assoc-s (format "~a[~a] "
                                        (find-word (first assoc) 3)
                                        (second assoc)))))
         l)
    assoc-s))
(define (print-assoc-memory a-m)
  (hash-table-for-each a-m
    (lambda (k v)
      (display (format "~a -> ~a~%" (find-word k 3) (print-assoc v))))))
```

Додаток C

```
ИВАНОВ -> РАД[e20] РАБОТА[e21]
ФОНД -> КОМАНДИРОВКА[e23]
КОМАНДИРОВКА -> РАССМОТРЕНИЕ[e24] УКАЗАНИЕ[e23] ДЕЙСТВИТЕЛЬНО[e21] ИВАНОВ[e21]
ЦЕЛЬ[e21]
```

РАД -> ИВАНОВ[e20] КОМАНДИРОВКА[e20]
ПОСЛЕ -> КОМАНДИРОВКА[e23]
ЦЕЛЬ -> СОГЛАСОВАНИЕ[e21]
РАБОТА -> НИИ[e21]
СОСТАВИЛ -> ОСТАТОК[e23] ПОСЛЕ[e23] ТЫСЯЧА[e23]
СОГЛАСОВАНИЕ -> ТЗ[e21]
НАЧАЛО -> ЯВИЛОСЬ[e24] РАБОТА[e24]
ТЫСЯЧА -> РУБЛЬ[e23]
ЯВИЛОСЬ -> НАЧАЛО[e24] СЛЕДСТВИЕ[e24]
ДЕЙСТВИТЕЛЬНО -> ИЗВЕСТНО[e22] КОМАНДИРОВКА[e21]
ИЗВЕСТНО -> ДЕЙСТВИТЕЛЬНО[e22] ИВАНОВ[e22] СОСТАВИЛ[e22]
СЛЕДСТВИЕ -> КОМАНДИРОВКА[e24]
ОСТАТОК -> СОСТАВИЛ[e23] ФОНД[e23]

Додаток D