УДК 004.4'414(045)

Сич М.Ю.

АЛГОРИТМ СЕМАНТИЧНОЇ ОБРОБКИ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Національний авіаційний університет

Приведено опис алгоритму аналізу текстової інформації на основі дворівневого кодування слів. Приведено основні концепції семантичної обробки інформації та процесів визначення контекстних змістів слів і зв'язків між ними

Вступ

Розробка систем, що мають справу з природною людською мовою є важливим та перспективним напрямком досліджень у сфері інформаційних технологій. Адже з моменту появи перших комп'ютерів люди прагнули спростити процес взаємодії людини та машини. Перші комп'ютерні системи були складними у використанні механізмами, робота з якими потребувала великих знань, зусиль та професіоналізму. Саме тому комп'ютери використовувались лише людьми, що мали відповідну кваліфікацію, навички та уміння. Сьогодні ситуація кардинально змінилась. Комп'ютери міцно закріпились у нашому житті. Вони використовуються скрізь і ми не розлучаємось з ними ні на роботі, а тим паче під час відпочинку. Саме тому пошук нових, більш зручних засобів спілкування і взаєморозуміння між людиною та комп'ютером є дуже важливою справою на сьогоднішній день.

Постановка задачі

Створити алгоритм, основним завданням якого є обробка вхідної словарної конструкції з метою встановлення смислів слів у контексті їх використання, а також знаходження семантичних зв'язків між словами з чітким виділенням типу цього зв'язку.

Після закінчення обробки ми повинні отримати структуру, що буде містити у собі змісти слів та характер зв'язків між цими словами. На цьому робота алгоритму закінчується.

Опис алгоритму

Алгоритм займається обробкою введеної послідовності слів. Він оперує словами, намагаючись знайти зв'язки між ними, встановити залежності та основні смислові лінії у введеній словарній конструкції.

У якості вхідних даних алгоритму виступає простий текстовий рядок. Цей рядок проходить набір послідовних обробок, зміст яких буде розкритий нижче.

Будь-яке речення є набором слів, що пов'язані між собою відповідними смисловими лініями. Для того, щоб почати розробляти алгоритм знаходження цих зв'язків, необхідно для початку визначитись з тим, яким чином описувати їх, і на якому етапі обробки вважати, що знайдені взаємодії слів є достатньо визначеними та однозначними.

У словосполученні є головне та залежне слово чи їх сукупність. Встановленим зв'язком будемо вважати пару «головне слово – залежна конструкція слів», у якій всі слова мають чіткий однозначний зміст та є зрозумілим характер їх взаємодії, тобто вид зв'язку.

Тепер визначимо самі типи можливих зв'язків. Для цього спочатку усі слова доцільно розділити на групи в залежності від їх змісту:

1) носій ознаки чи дії. Це може бути сутність, жива чи нежива, що може виступати як об'єкт певної дії чи носій певної ознаки;

2) ознака. Група, до складу якої входять слова, що визначають певні властивості для слів 1-ї групи. До цієї групи входять всі слова, що характеризують предмети чи істоти;

3) дії. Група слів, що вказує на активні чи пасивні дії для слів 1-ї групи;

4) ознаки дії. Слова, що характери-

зують особливості та умови виконання дій. Слова цієї групи виражають час, місце, умови та інше для слів 3-ї групи;

5) числівники. Слова вказують на числа, кількість для слів 1-ї групи.

Поділ на такі групи є умовним, проте тісно пов'язаний з подальшою обробкою цих слів.

Як вже було сказано, у нашій парі одне слово є головним, а інші – залежними, то опираючись на приведену вище класифікацію слів ми можемо сказати, що слова 2, 3, 5 груп є залежними від слів 1-ї групи, слова групи 4 – від слів групи 3.

Отже, будемо розглядати наступні види зв'язків слів у парі:

1) носій – ознака;

2) носій – дія;

3) носій – кількість;

4) дія – ознака дії;

5) дія – суб'єкт дії.

Обробка речення приводить на кінцевому етапі до набору словосполучень, що мають перетини різними словами, тобто це просто група взаємопов'язаних пар слів, або точніше сказати змістів слів. Для кожного слова ми маємо чітко визначений зміст, а також всі зв'язки і їх характер для всіх слів, з яким це слово взаємодіє. Все це зручно представити графом, вершини якого є словами, а ребра – зв'язками. Таким чином, коренем дерева є слово, що має залежні слова, але саме не є таким, а листя дерева – слова, що не мають залежних слів, а лише самі залежні від інших. Таким чином, результатом обробки вхідного речення має стати дерево - граф. У такому графі вершина утворює пару з кожною зі своїх дочірніх вершин. А ребро є зв'язком, значення якого належить множині можливих значень зв'язків між словами.

Первинне кодування слів

Для обробки слів необхідно відмовитись від простого кодування у вигляді простих послідовностей символів. Такий спосіб є зручним та простим для зберігання слів у пам'яті комп'ютера, проте є абсолютно неприйнятним з точки зору обробки слів як змістовних одиниць інформації. Кодування слів має нести у собі смислове навантаження слова. Новий вид кодування слів буде зручним для зберігання інформації про слово, його структуру та основні властивості. Таким чином, першим кроком роботи алгоритму є приведення слів речення у новий формат, заснований на морфологічних та синтаксичних особливостях слів.

На даний момент кодування слів відбувається на основі 64-розрядних чисел, у які закладаються основні морфологічні властивості слова та деяке додаткове смислове навантаження, що може бути досить компактно вміщене у код слова. Первинне кодування слів опирається на їх розподіл по частинам мови. Виділяються іменники, прикметники, дієслова, дієприкметник, дієприслівники, прийменники, прислівники та займенники.

Група бітів, що будуть представляти певні властивості слова назвемо *полем*.

Кількість біт, необхідних для поля вираховується за наступною формулою:

$$N = \log K,$$

де N – необхідна кількість біт для поля,
K – кількість можливих значень, що необхідно кодувати полем.

При кодуванні слова всі біти числа, що буде представляти код слова, розбиваються на поля, що будуть нести у собі певні дані про слово. Код слова буде містити обов'язкові поля та додаткові поля, що містять дані, які є, як правило, результатом морфологічного аналізу слова як певної частини мови, а також деякі додаткові смислові дані про слово.

Нижче приведений перелік обов'язкових полів, що будуть присутні у коді кожного слова.

1) зміст. Розкриває суть слова, несе смислове навантаження, тобто однозначно визначає суть слова. Займає 30 біт, що дозволяє закодувати 16.777.215 словарних змісти;

2) мова. Визначає мову, до якої належить слово. Це поле витікає з символьного написання слова, що кодується у новий формат, оскільки одні і ті ж слова

присутні у різних мовах. Займає 4 біти, тобто можливо закодувати 16 мов;

3) номер слова у синонімічному ряду. Оскільки слова, що пишуться порізному, можуть мати однаковий зміст, то повинна бути можливість однозначно ідентифікувати кожне слово такого синонімічного ряду. Займає 5 біт, тобто кожен синонімічний ряд може вмістити до 32 слів.

4) частина мови. Це поле вказує на частину мови слова. Займає 5 біт, тобто 32 частини мови;

5) поле можливої частини мови. Це поле необхідне для додаткового кодування слова, якщо його частина мови не може бути визначена. Проте в основному це поле буде використовуватись для віднесення слова до певної групи. Інакше кажучи, тут будуть записуватися смисли слова, тобто носій ознак, дія чи інше.

Поля з 1-го по 4-ий мають послідовне наскрізне кодування, тобто їх значення можуть приймати всі можливі значення у діапазоні від 0 до максимуму, що дозволяє розрядність, при чому це значення однозначне та єдине при даному кодуванні. Поле 5 має паралельне кодування, а отже містить одиниці на певних позиціях, чим вказує можливі сутності для даного слова.

Поля, наведені вище присутні у кожному слові, оскільки відображають інформацію, якою може володіти кожне слово. Таке базове кодування може бути основою для обробки слів, оскільки дозволяє розділяти їх за ознаками, необхідними для будь-яких подальших операцій.

Зрозуміло, що для коректного кодування слів по частинам мови, необхідно, щоб однакові поля розміщувались на однакових позиціях. Це дозволить усунути можливі проблеми, пов'язані з отриманням цих полів при обробці.

Подальше кодування вже опирається на конкретну частину мови. В залежності від неї у слово кодується додаткова інформація, що містить дані по морфологічних властивостях слова.

Приведемо поля, що будуть присутні у кодах слів залежно від частини мови:

1) іменник. Його додаткові поля: число, відмінок, рід;

2) прикметник. Додаткові поля: число, відмінок, рід, ступінь порівняння, тип прикметника;

3) дієслово. Додаткові поля: число, рід, час, спосіб дії, особа, форма;

4) дієприкметник. Ця частина мови є перетином властивостей таких частин мови як прикметник та дієслово, отже має додаткові поля тих і інших;

5) дієприслівник. Додаткові поля: час, спосіб дії;

6) займенник. Поля: тип, особа, число, відмінок, рід;

7) числівник. Поля: тип, рід, число, відмінок.

Додаткові поля по частинам мови будуть нести допоміжну інформацію по словам, проте у процесі зв'язування слів вони, як правило, не використовуються.

Наведене кодування слів є досить потужним та достатньо об'ємним для того, щоб визначити основні характеристики конкретного слова. На основі цих даних можна спробувати визначити взаємозв'язки слів, тобто зв'язати предмети з ознаками та діями, дії – з ознаками дій(час, місце і т.п.) і т.д.

Алгоритм пошуку зв'язків між словами є ітераційним, тобто представляє собою набір послідовних повторюваних кроків, на кожному з яких обробляється певна група слів. На першому кроці у реченні виділяються усі ознаки і для них шукаються носії, на другому кроці – дії, потім ознаки дій. Інакше кажучи, на певній ітерації, для певного слова ми намагаємось знайти головне слово. Для прикладу нехай n – загальна кількість ітерацій обробки речення. Якщо певне слово обробляється на i-ій ітерації, то на ітераціях з номером (1, i-1), дане слово виступає як головне слово при приєднанні до нього слів з інших груп, на ітерації і – як залежне, а на ітерація (i, n) – взагалі не приймає участі в обробці. Більш детально алго-

ритм знаходження зв'язків між словами описаний нижче:

1) кожне слово вхідного речення кодується у новий формат, тобто позначається 64-розрядним числом;

2) відбувається послідовний обхід всіх слів речення зліва направо;

3) якщо слово належить групі змістів, що обробляється на даній ітерації, то для слово відбувається пошук можливого головного слова;

4) якщо головне слово знайдене, то встановлюється зв'язок, якщо ні, то поточне слово мітиться як необроблене або незначуще;

5) якщо обхід по словам речення на ітерації закінчено, то розпочинаємо нову ітерацію, якщо ні, то продовжуємо прохід;

6) якщо поточна ітерація остання, то вихід.

Таким чином, даний алгоритм передбачає кількість ітерацій по кількості смислових груп слів. На кожній ітерації між головним та залежними словами може бути встановлений тільки певний характер зв'язку.

Тепер розглянемо більш детально умови відповідності головного слова залежному, при якому можливо зв'язати дані слова. Необхідні умови будемо будувати на основі характеру встановлюваного зв'язку.

1) носій – ознака. Така пара слів являє собою пару типу іменник – прикметник, займенник – прикметник і т.д. При цьому умовою відповідності є: однакове число, однаковий рід, однаковий відмінок слів. При встановленні відповідного характеру зв'язку необхідно проаналізувати дані поля і якщо вони задовольняють умовам – встановити зв'язок;

2) носій – дія. Пара слів типу іменник – дієслово. Умовою встановлення зв'язку є відповідність у числі, роді слів. Якщо поля задовольняють умовам – встановити зв'язок;

3) носій – числівник. Це пара слів типу іменник – числівник. Умовою відповідності тут є відмінок та рід;

4) дія – ознака дії. Зв'язок при первинному кодуванні може бути встановлений тільки на основі послідовності слів, тобто у разі коли ознака дії знаходиться у реченні перед дією або після неї, то встановлюється зв'язок;

5) дія – суб'єкт дії. Зв'язок не може бути встановлений на основі даного способу кодування.

Наведене кодування дає можливість обробити приблизно 70% взаємозв'язків між словами. Проте у значній кількості випадків аналізу на основі вище приведених властивостей слів недостатньо. Основні проблеми, що виникають при аналізі семантичних зв'язків між словами, пов'язані перш за все у тому, що неможливо визначити істинний зміст слова, не враховуючи контекст його використання. Зустрічаються слова, що пишуться однаково, але можуть мати кілька змістів, бути кількома частинами мови і т.п. У цих випадках неможливо призначити слову якийсь певний смисл, не аналізуючи його у контексті його використання у певній групі слів. Саме у таких випадках застосовується обробка слів на основі змістів самих слів, а не їх граматичних та морфологічних властивостях чи синтаксичних характеристик. Це і є пряма смислова обробка слів.

Додаткове кодування слів

Механізм первинного кодування є простим, але недостатнім для встановлення семантичних зв'язків між словами на необхідному рівні. Залишаються невирішеними наступні проблеми:

1) слова можуть мати невизначеності у родах, числах, відмінках і т.д. Тому ці поля у коді слів є незаповненими або нульовими, що не дає можливості коректно перевірити умови відповідності між словами для встановлення зв'язків. Такі ситуації виникають, коли слово при відмінюванні по певній ознаці може писатись однаково. При цьому виникають невизначеності у межах однієї частини мови, що має слово;

2) слова можуть мати кілька змістів, при цьому писатись однаково. Це так зва-

ні слова-пароніми. У цьому випадку структура слів абсолютна різна і морфологічні властивості слів теж. У цьому випадку взагалі на основі приведеного кодування слів неможливо задати дані про слова. Невизначеності вже переходять у межі різних частин мови, тобто слово може містити різні набори полів у своєму числовому коді.

3) проблемою залишається також зв'язок типу «дія – суб'єкт дії», оскільки ми на основі приведеного вище кодування не маємо механізмів для встановлення взаємодії між словами такого характеру.

Слід додати, що перший та другий з перерахованих пункти є значно складнішим при вирішенні.

Отже, вирішення даних задач потребує розширення кодування слів, оскільки закодованих даних по слову недостатньо.

Вирішенням проблеми стає кодування слів додатковими частками, що будуть нести смислове навантаження на слова, а також вказувати на додаткові дані по морфологічній структурі слова, його синтаксичним особливостям використання, містити дані по можливій взаємодії слів. Таким кодування можна вирішити проблеми, перераховані вище.

Частки містять дані про слово у вигляді можливих взаємодій слова, у які він може вступати. Характер цих часток дуже схожий на питання, що ставляться до слів. Це питання типу хто?, який?, що робить?, де?, скільки? і т.п. Тому у подальшому ці частки ми і будемо називати *питаннями*.

Питання розбиваються на кілька груп в залежності від характеру слів та типу взаємодії, що вони несуть у собі.

Кожному слову ставиться у відповідність 2 набори питань:

1) так звані «вхідні» питання. Вони несуть у собі характер взаємодії слова для пар, у яких саме слово може виступати тільки як залежне слово;

2) «вихідні» питання. Кожне питання несе інформацію про взаємодію, у якій дане слово буде виступати як головне.

Отже, кожне слово тепер буде кодуватись не тільки 64-розрядним числом, а й отримувати 2 набори питань: вхідних та вихідних.

Оскільки, кожному слову ставиться у відповідність не одне питання, а саме група, то у нас з'являється можливість кодувати дані про слово, шляхом переліку усіх його характеристик, залежно від можливого контексту та конкретного смислу слова, у якому воно буде використовуватись.

Кодування питань відбувається також на основі розрядних полів у числах. Для зручності будемо використовувати 64-розрядні числа, як і для слів. Щоб мати можливість розрізняти коди питань і слів, ми встановимо старший біт коду слів рівним 1, а у питань – 0.

Питання матимуть обов'язкові поля, тобто такі, що присутні у коді кожного питання. У даному випадку це тип питання. Вказує на групу, до якої належить питання. Займає 4 біти.

Це обов'язкове поле усіх питань.

Як і у слів, у питаннях необхідно закодувати додаткову інформацію по питанню, а точніше по слову та характеру взаємодії для цього питання.

Спираючись на набір можливих змістів слів, тобто носій, ознака, дії, суб'єкти дії та числа, ми можемо створити набір питань для кожної групи таких слів. У авторській версії кодування питань налічується 109. Кодування відбувалось для слів української мови. Проте перевірки груп питань для інших мов вказують на те, що їх кількість може змінюватись залежно від особливостей конкретної мови. В більшій частині випадків це стосується різних словарних конструкцій, що по суті означають те саме, проте реально мають різну структуру слів: частини мови, роди, звороти і т.д. Проте, незважаючи на це, таке кодування дає можливість закласти у нього велику кількість даних по слову, більше того, необхідність у подальшій обробці бітових полів питань взагалі відпадає, тому що для встановлення факту взаємодії слів, а також характеру взаємо-

дії ми можемо на основі того, якими питаннями перетинаються відповідні групи слів між собою.

Важливим моментом у роботі алгоритму є наявність саме двох груп питань у слова. Для встановлення факту та характеру взаємодії між словами необхідно просто перевірити наявність перетину між групою вихідних питань головного слова та вхідних питань залежного слова. Якщо перетин є, то відразу відомо і характер взаємодії.

Таким чином, базуючись на вторинному кодуванні слів ми маємо змогу вирішити описані на початку розділу проблеми. Адже тепер слово має всі вичерпні необхідні дані про всі свої можливі смисли. Саме це і є головним завданням груп додатків у вигляді питань. Тепер встановлення зв'язків між словами відбувається на основі цих груп додатків-питань. Після знаходження зв'язків між словами ми можемо визначити і зміст самого слова через тип взаємодії та зміст слова, з яким дане пов'язане.

Слід додати, що робота алгоритму потребує бази слів, що закодовані у новий формат. База має містити всі слова у всіх своїх словоформах. У авторській версії такої бази для української мови налічується приблизно 1,5 мільйона слів. Є можливість створення бази автоматично з вже існуючих словників різних мов.

Особливістю описаного алгоритму є те, що він є повністю уніфікованим по відношенню до обраної мови інформації. Алгоритм працює однаково для мов різних груп. Проблемою залишаються мови з ієрогліфами.

З одного боку, кожен ієрогліф має зміст. Проте, самі мови мають особливості змістовного характеру розуміння цих ієрогліфів. Певна послідовність ієрогліфів має зміст, відмінний від суми змістів ієрогліфів окремо. Іншими словами, вираз може містити зміст, абсолютно не пов'язаний зі змістами окремих його частин.

Проте у подальшому алгоритм встановлення семантичних зв'язків можна вдосконалити. Можна спробувати кодувати кожен ієрогліф окремим змістом, тобто надати йому за стандартним механізмом обробки 64-розрядне число у відповідність. Також будуть надані і 2 групи питань до цього ієрогліфа. Однак «розуміння» даних комп'ютером буде низькими і недостатнім для практичного використання.

Висновки

Від моменту створення перших комп'ютерних систем людина робить спроби створити загальні механізми для подальшої можливості закладення даних у пам'ять машини на природній людській мові. З цією метою розроблено безліч алгоритмів семантичної обробки інформації. Проте потужність, точність та надійність цих алгоритмів не достатньою для широкого їх використання та створення систем, що дадуть високий рівень смислової обробки інформації комп'ютером.

Запропонований у статті алгоритм дозволяє переступити недосяжний раніше поріг «розуміння» інформації комп'ютером у 80%, необхідний для створення широкого спектру потужних систем обробки природної мови, включаючи експертні системи, системи «питання-відповіді», інформаційні системи та системи штучного інтелекту. Подальші його розробки ведуться.

Список літератури

1. *Джарратано Д.* Экспертные системы: принципы разработки и программирование, 4-е издание.: Пер. с англ. – М.: ООО «Вильямс», 2007. – 1152 с.

2. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений: Пер. с англ. – М.: Мир, 1976. – 358 с.

3. *Мальковский М.Г.* Диалог с системой искусственного интеллекта. – М.: Изд-во МГУ, 1985. – 214 с.