УДК 681.3

Анісімов А. В., Глибовець М. М., Марченко О. О, Кисенко В. К.

МЕТОД ОБЧИСЛЕННЯ СЕМАНТИЧНОЇ БЛИЗЬКОСТІ ДЛЯ СЛІВ ПРИРОДНОЇ МОВИ

Запропоновано метод обчислення семантичної близькості для слів природної мови. Семантична близькість дає змогу побудувати алгоритмічні моделі різних лінгвістичних задач таких, як: розв'язання смислових неоднозначностей, ідентифікація об'єктів тексту, семантичний аналіз текстів тощо. Описаний алгоритм є зваженою модифікацією методу лексичного перетину.

Ключові слова: семантика, неоднозначність, метод лексичного перетину.

Вступ

Ключовим елементом розуміння природномовних процесів є можливість визначення семантичної близькості або відстані між двома поняттями. Обчислення семантичної близькості використовують в розв'язанні різних лінгвістичних задач, зокрема: в автоматичному реферуванні та анотуванні текстів, розв'язанні смислових неоднозначностей, розв'язанні анафор тощо. Є істотна різниця між поняттями **семантична**

схожість і **семантична близькість**, останнє значно ширше. Семантична схожість подібна до синонімії, наприклад: *шофер* і *водій*. Своєю чергою, семантична близькість не вичерпується відношенням синонімії. Міра семантичної близькості – це кількісна величина, яка показує наскільки два поняття близькі (тобто пов'язані або схожі) між собою. Існує багато інших зв'язків між словами (крім синонімії), за наявності яких можна говорити про смислову близькість. Наприклад, антоніми («світле» – «темне») або слова, між якими спостерігаємо відношення «частина» – «ціле» («гілка» – «дерево»). Також є багато пар слів, між якими наявність таких відношень не настільки очевидна, але, незважаючи на це, вони семантично близькі (наприклад, «пустеля» – «спека»). Тут представлено новий метод обчислення семантичної близькості.

1. Наявні методи визначення семантичної близькості–зв'язності слів природної мови

Перш за все слід розглянути деякі вже розроблені методи обчислення величини семантичної близькості. Дослідження у цьому напрямку розпочато з 80-х рр. XX ст. Відтоді розроблено цілу низку методів обчислення міри семантичної близькості. Серед них виділяють дві групи методів [2; 3]:

- методи, основою яких є пошук відстані між концептами у семантичній мережі.
- методи, що базуються на лексичному перетині словарних значень слів.

Опишемо основні ідеї цих двох підходів.

Методи, засновані на відстані між концептами

Методи цієї групи засновано на відшуканні відстані між двома концептами у семантичній мережі (*WordNet*, дерево категорій *Wikipedia*). Між двома поняттями лежить найкоротший шлях і на його основі визначається семантична близькість між словами. Одну з перших таких мір запропонував Резнік [5]:

$$\frac{1}{N_p},$$

де N_p – кількість вершин у найкоротшому шляху. Очевидний недолік цієї міри у тому, що для деяких концептів вкладеність класів, до яких вони належать, є більшою ніж для інших через їхню природу та походження. Для розв'язання цієї проблеми Лікок і Ходоров [7] запропонували метод, який нормалізує довжину шляху з урахуванням глибини загальної ієрархії:

$$-\log\left(\frac{len(c_1,c_2)}{2D}\right),$$

де D – максимальна глибина ієрархії.

Ву і Палмер [8] запропонували інший підхід, що враховує глибину найменшого спільного предка концептів в ієрархії:

$$-\log\left(\frac{depth(LSO(c_1,c_2))}{depth(c_1)+depth(c_2)}\right),$$

де LSO – відстань від найменшого спільного предка концептів до кореня таксономії.

Методи, засновані на лексичному перетині

Перший алгоритм такого типу розробив Леск [1]. Він сконструював алгоритм, який в основі має припущення про те, що пов'язані поняття будуть визначатися (або пояснюватись) однаковими словами. Леск використав цей підхід для розв'язання задачі добору правильного значення слова у деякому контексті. Порядок роботи алгоритму: спочатку шукається лексичний перетин означень заданого слова з означеннями всіх слів тексту. Далі обирається значення слова з найбільшим сумарним перетином, що найбільше відповідає контексту. Лексичний перетин обчислюється як кількість спільних слів двох означень. Недолік такого підходу у тому, що статті у звичайних словниках є досить короткими, а тому можуть погано відображати семантичну близькість деяких слів. Бенаржи і Педерсен розширили метод Леска, додавши до перетину словники інших слів, що пов'язані з початковими, тим чи іншим шляхом. Наприклад, в одній з реалізацій використовувався *WordNet* і додавалися слова, що є безпосередніми предками заданих слів в ієрархії цієї семантичної мережі.

У розробці комп'ютерної системи обчислення семантичної близькості дуже важливо обрати збалансоване, коректне та об'ємне джерело смислових даних. У таких дослідженнях використовують різноманітні джерела: від звичайних енциклопедій і тезаурусів до таких сучасних баз знань, як: *WordNet*, *ConceptNet*, *Wikipedia*. Найбільш вагомі результати були отримані на основі *WordNet* та *Wikipedia*.

2. Метод обчислення семантичної близькості–зв'язності слів

Перш за все слід розглянути джерело даних, яке використовується в наших дослідженнях: його структуру, доступність, повноту й об'єм; його недоліки та переваги. Цим джерелом є *Wikipedia* – вільна Інтернет-енциклопедія. На час написання цього тексту, англомовна версія *Wikipedia* містила понад 3 млн статей (для порівняння: *WordNet* містить близько 150 тис. слів). Така кількість забезпечується тим, що кожен може створити нову або змінити вже наявну статтю. Незважаючи на таку свободу редагування, якість статей завдяки модерації досить висока. Дуже важливою для нас є можливість завантажити локальну копію всієї енциклопедії. До недоліків *Wikipedia* належить недостатньо чітка формалізація правил створення та написання статей. Це призводить до того, що загальна структура статей може досить істотно відрізня-

тись. Це значно ускладнює аналіз тексту статей. Часто ще одним недоліком називають недостатню нейтральність статей. Наприклад, стаття може містити особисту думку автора з приводу деякого спірного питання. Проте цей недолік не створює суттєвих незручностей для нашого алгоритму. Таким чином, можна дійти висновку, що *Wikipedia* – унікальне і цінне джерело даних.

Структура *Wikipedia* має ряд властивостей, що можуть бути корисними в обчисленні мір семантичної близькості. Ці властивості дають змогу моделювати певні типи лексикологічних зв'язків між словами. До них належать:

- Синонімія. Моделюється сторінками, що містять перенаправлення (англ. REDIRECT) на іншу сторінку. Наприклад, «кіт» перенаправляється на «кішка».
- Омонімія. Моделюється сторінками «значення» (англ. *Disambiguation*). Наприклад, слово «нота» може означати знак нотного письма, інтонацію голосу, офіційне дипломатичне звернення, розрахунковий документ. У нашій розробці ми використовуємо такі сторінки для розв'язання смислових неоднозначностей. За допомогою таких сторінок можна отримати список слів-кандидатів, що найточніше відповідає значенню іншого слова або контексту.
- Крос-посилання. Моделюються безпосередньо посиланнями на інші статті енциклопедії. Наприклад, у статті «кілометр» є посилання на статті: «метр», «одиниця виміру», «відстань», «ярд», «фут».

Також у нашому алгоритмі використовується *stop-list* – список слів, що не мають власного семантичного значення: прийменники, сполучники, займенники, загальновживані слова.

Мірою семантичної близькості, у цій розвідці використано модифікацію методу лексичного перетину. Для оцінки міри близькості застосовується лексичний перетин текстів статей *Wikipedia* або їх частин. Як було сказано, цей підхід базується на ідеї, що схожі поняття визначаються однаковими словами, а отже, кількість спільних слів в означеннях може бути мірою семантичної близькості цих понять. У нашому методі кожне слово має певне значення, що задається дійсним числом і залежить від певних факторів. Для оцінки вагових параметрів використано алгоритм глобальної оптимізації – метод імітації віджигу.

Для роботи алгоритму потрібно отримати тексти відповідних статей. Пошук та одержання статей виконується за таким алгоритмом:

1. Знайти в індексі файл, в якому знаходиться стаття з потрібною назвою.

2. Якщо ця стаття містить перенаправлення, взяти назву статті та перейти до п. 1. Інакше – повернути текст статті.

Після цього обчислювати міру семантичної близькості. Формально алгоритм буде виглядати так:

нехай задано два слова s_1і s_2, відповідні їм статті (або частини статей) – a_1 і a_2, а також p_1 і p_2 – деякі множини слів (наприклад, усі слова статті).

Алгоритм:

1. a_1, a_2 – розбити на слова. Множини цих слів позначимо W_1 і W_2, відповідно.

2. З множин W_1, W_2 видалити слова, що належать до списку *stop-list*.

3. Словам з множин W_1, W_2 надана вага, яка може залежати від різних факторів: розташування слова у тексті; належність слова до назви статті; слово у посиланні тощо.

4. Знайти ***common*** – суму ваг усіх слів:

$$common = \sum_{w \in W_1} d_1(w) + \sum_{w \in W_2} d_2(w),$$

де $d_i(w)$ – вага слова w у множині W_i.

5. Для слів з W_1 знайти суму:

$$total_1 = \sum_{\substack{w \in W_1 \\ w \in p_2}} d_1(w).$$

6. Для слів з W_2 знайти суму:

$$total_2 = \sum_{\substack{w \in W_2 \\ w \in p_1}} d_2(w).$$

7. Обчислити міру семантичної близькості за формулою:

$$relatedness = \frac{total_1 + total_2}{common}.$$

Деякі слова можуть мати різні значення, але однакове написання. Наприклад, «ягуар» може означати велику кішку або марку автомобіля. Тоді залежно від другого слова з пари, потрібно обрати правильне значення і відповідну йому статтю. Наприклад, якщо задана пара («ягуар», «лев»), то правильне значення слова «ягуар» – велика кішка, а для пари («ягуар», «мерседес») – марка автомобіля. Опишемо алгоритм розв'язання таких неоднозначностей.

Як і в попередньому алгоритмі, на вхід задано слова s_1 і s_2. Для кожного з цих слів, тим чи іншим способом, отримуємо список кандидатів-значень. Після цього для кожної пари значень, де одне значення взято з першого списку кандидатів, а інше – з другого, обчислюємо семантичну близькість й обираємо пару з максимальною мірою. Формально алгоритм можна записати так:

- Для обох слів отримати список статей-кандидатів:

  1) з індексу отримати список статей, з назвою виду: <w> (<*>), де w – задане слово, a* – деяка послідовність символів;

  2) (не обов'язково) зі статті <w> (*disambiguation*) отримати список можливих значень.

• Для всіх пар статей (*X*, *Y*), де стаття *X* належить до списку неоднозначностей слова s_1, а стаття *Y* належить списку – s_2, обчислити міру семантичної близькості.

• Обрати пару (*X*, Y**) з максимальною семантичною близькістю.

Для оцінки вагових параметрів використовується алгоритм імітації віджигу [9; 10] – ймовірнісна метаевристика для глобальної дискретної оптимізації. Цей метод оперує точками простору розв'язків, на кожній ітерації алгоритму зберігається одна поточна точка. Поточна точка може бути змінена за певним ймовірнісним законом. Цей алгоритм схожий на метод градієнтного спуску, але зміна точок за ймовірнісним законом дає можливість не застрягати в точках локального максимуму.

У запропонованому алгоритмі ваги визначаються на основі таких факторів:

• Слово належить до назви статті.

• Слово є посиланням на іншу статтю.

• Слово належить першому параграфу статті.

• Інші слова.

Кожний фактор дає слову різну вагу. Таким чином, простір розв'язків є чотиривимірним – за кількістю факторів. Як функцію для максимізації використано ранговий коефіцієнт кореляції Спірмена.

3. Програмна реалізація та тестування

Створено програмну реалізацію описаного методу, для цього використано мову програмування *Scala* [12; 13]. *Scala* – сучасна, добре розвинена мова, і є дуже зручною для розроблення програм, пов'язаних з обробкою текстової інформації. Окремою її перевагою є те, що код програм компілюється у байт-код для JVM (*java virtual machine*), завдяки чому програма здатна працювати на будь-яких системах з платформою JVM (зокрема, *Windows* та *GNU/Linux*). Як джерело даних використовується локальний архів *Wikipedia*, що був завантажений з відповідного розділу сайту проекту. Розмір архіву становить близько 5,5 Гб, а тому для ефективного пошуку та одержання статей ми виконали його попередню обробку. В загальних рисах алгоритм попередньої обробки архіву *Wikipedia* виглядає так:

1. Для кожної статті з локальної копії *Wikipedia* виконати такі дії:

а) отримати назву та вміст статті;

б) обробити вміст статті, відкинувши все, що непотрібне для роботи алгоритму (наприклад, посилання на зовнішні веб-сайти, коментарі, описи зображень тощо);

в) записати у текстовий файл назву статті та оброблений вміст;

г) додати пару <назва статті; номер файлу в якому вона зберігається у БД> .

2. Після обробки всіх статей, побудувати індекс БД за полем з назвою статті.

Статті зберігаються у звичайних текстових файлах (назва, текст статті). В якості БД ми використовуємо *MongoDB*. *MongoDB* – це сучасна, нереляційна база даних, яка за багатьма тестами одна з найшвидших. Окрім цього, важливою є можливість ефективного пошуку за регулярним виразом, яка використовується для отримання списку варіантів неоднозначностей слова. У БД зберігаються пари виду <назва статті, номер файлу, в якому знаходиться текст статті>. Індекс БД будується за першим ключем, тобто за полем «назва статті». Все згадане дає змогу шукати та отримувати текст статті за долі секунди.

Наш метод було тестовано на спеціальному наборі зважених пар слів *Finkelstein WordSimilarity-353* [14]. Він містить 353 пари слів, що були оцінені експертами-людьми. До кожної пари слів експерти дібрали число від 0 до 10. Для перевірки відповідності міри близькості до тестових даних було використано ранговий коефіцієнт кореляції Спірмена.

Таблиці 1, 2, 3 містять результати вимірювань швидкодії та якості запропонованого алгоритму.

Таблиця 1. **Без розв'язання смислових неоднозначностей**

Тип алгоритму	Кореляція	Середній час запиту, с
Простий	0,39	0,05
Зважений	0,63	0,05

Таблиця 2. **З частковим розв'язанням смислових неоднозначностей**

Тип алгоритму	Кореляція	Середній час запиту, с
Простий	0,45	0,25
Зважений	0,68	0,25

Таблиця 3. **З повним розв'язанням смислових неоднозначностей**

Тип алгоритму	Кореляція	Середній час запиту, с
Простий	0,49	0,4
Зважений	0,74	0,41

Висновки

В цій статті запропоновано новий метод обчислення семантичної близькості для слів природної мови. За джерело даних використана вільна інтернет-енциклопедія *Wikipedia*. Для оцінки вагових параметрів застосовано метод імітації віджигу. Такий підхід дає можливість значно підвищити кореляцію з тестовими даними, що були отримані від людей-експертів.

Також запропоновано метод розв'язання смислових неоднозначностей, що використовує специфічні властивості для *Wikipedia*.

Майбутня робота над проектом буде направлена на підвищення швидкодії обчислення семантичної близькості. Також для покращення якості оцінки планується побудувати інтегральну оцінку, що використовує запропонований метод та метрики, що засновані на обчислення шляхів в графах (Резнік, Лікок і Чодоров, Ву і Палмер).

Література

1. Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In SIGDOC'86: Proceedings of the 5th annual international conference on Systems documentation, pages 24–26, New York, NY, USA. ACM. – 1986.
2. Wubben S. Using free link structure to calculate semantic relatedness. ILK Research Group Technical Report Series no. 08-01. – 2008.
3. Ponzetto S. P., Strube M. Knowledge deriver from Wikipedia for computing semantic relatedness. EML Research gGmbH, Natural language processing group. – 2007.
4. Gabrilovich E., Markovitch S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. Department of Computer Science Techion – Israel Institute of Technology. – 2006.
5. Resnik P. Using information content to evaluate semantic similarity in a taxonomy. In International Joint Conference for Artificial Intelligence (IJCAI-95). – P. 448–453. – 1995.
7. Leacock C., Chodorow M., and Miller G. A. Using corpus statistics and wordnet relations for sense identification. Computational Linguistics, 24(1): 147–165. – 1998.
8. Wu Z. and Palmer M. Verb semantics and lexical selection // 32nd Annual Meeting of the Association for Computational Linguistics. – New Mexico State University, Las Cruces, New Mexico. – 1994. – P. 133 – 138.
9. Strube M., Ponzetto S. P. WikiRelate! Computing Semantic Relatedness Using Wikipedia // Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), pp.1419–1424.
10. Kirkpatrick, S.; C. D. Gelatt, M. P. Vecchi (1983-05-13). "Optimization by Simulated Annealing". Science. New Series 220 (4598): 671-680.
11. Sean Luke, 2009. Essentials of Metaheuristics. – Режим доступу: http://cs.gmu.edu/!sean/book/metaheuristics/. – Назва з екрана.
12. Odersky M. Scala by example. Programming methods laboratory, EPFL, Switzerland. – 2009.
13. Odersky M., Spoon L., Venners B. Programming in Scala. Artima Press, Inc. – 2008.
14. Finkelstein. L., Gabrilovich E., Matias Y., Rivlin E., Solan Z., Wolfman G., and Ruppin E.. Placing Search in Context : The Concept Revisited. ACM Transactions on Information Systems, 20(1) : 116-131, January 2002.

A. Anisimov, M. Glybovets, O. Marchenki, V.Kysenko

THE METHOD FOR CALCULATING OF SEMANTIC CLOSENESS OF NATURAL LANGUAGE WORDS MEANINGS

The paper concerns methods of calculating semantic relatedness and similarity measures for evaluating closeness of words meanings in tasks of computational linguistics. The semantic relatedness and similarity measures allows to implement algorithmic models of linguistic context analysis to solve such problems as words meaning ambiguity, entity recognition, semantic analysis of natural language texts etc. The work describes one method for calculating measure of semantic closeness of natural language words meanings. This method is a weighted modification of overlap based metrics, as a data source we use Wikipedia. For estimation of weighting parameters we use a simulate annealing.

Keywords: semantics, ambiguity, the method of lexical intersection.

Матеріал надійшов 18 травня 2011 р.