РОЗПІЗНАВАННЯ ТА ВИПРАВЛЕННЯ ПОМИЛОК У СЛОВАХ ТЕКСТУ З ВИКОРИСТАННЯМ РОЗПОДІЛЕНИХ ПРЕДСТАВЛЕНЬ

Омельченко Руслан

Міжнародний науково-навчальний центр інформаційних технологій та систем
НАН України і МОН України, Київ
ruslan.omelchenko.irtcits@gmail.com

Анотація

При розпізнаванні некоректних слів та виправленні помилок в словах, як правило, програма пропонує користувачу короткий список передбачуваних правильних слів у послідовності від найбільш до найменш вірогідного. В даній статті досліджена можливість застосування бінарних розподілених представлень та методів їх обробки для представлення, розпізнавання та обробки слів з помилками. Проведений порівняльний аналіз з іншими методами та відображені результати експериментів на двох наборах слів з типовими орфографічними помилками.

1. Вступ

Слова можуть містити помилки різного типу. Для їх виправлення була створена система, яка знаходить неправильно написані слова і здійснює корекцію. Вона пропонує користувачеві список слів, які приблизно повинні відповідати вхідному слову з помилками в порядку від найбільш ймовірного до найменш ймовірного. Для представлення слів були використані розподілені розріджені бінарні вектора, а також методи їх обробки.

2. Огляд існуючих підходів до вирішення даної проблеми

У попередніх найбільш відомих алгоритмах використовувалися системи правил замін або ключів [1, 2].

Система Деоровіца та Кіура грунтується на правилах, що моделюють типові орфографічні помилки. Правила застосовуються до слів з помилками таким чином, що вхідний рядок, який знаходиться де завгодно в слові з помилками, замінюється на вихідний рядок [2]. Кожній заміні на новий рядок привласнена вартість. Якщо отримане після заміни слово коректне, воно додається в список кандидатів. Якщо до слова застосовується кілька правил - вартість додається. Після цього всі кандидати сортуються в порядку збільшення вартості. Передбачається, що коректне слово з найменшою вартістю є найбільш ймовірним кандидатом для виправлення слова з помилкою.

У програмі Роджера Міттона [1] використовувалася система ключів. Ключі – це стиснуті версії слів з помилкою та слів зі словника. Вони містять ознаки, які найбільш ймовірно присутні як в слові з помилкою, так і в оригіналі. Ключі будувалися на основі певних правил. Міттон створив алгоритм, який за допомогою цих ключів дає змогу знаходити коректні слова.

Для сортування кандидатів використовувався метод мінімальної відстані корекції. Його суть полягала в знаходженні обсягу змін слова з помилкою для досягнення коректного слова зі словника. Для кожної операції корекції (видалення або вставка букви, заміна однієї букви на іншу, переставлення суміжних букв) визначалася вартість і обчислювалася загальна вартість досягнення коректного слова.

Даний метод також було доповнено методом простого збігу букв, що включає підрахунок не співпадаючих букв і додатковий штраф за розбіжність першої букви. Чим вище результат, тим гірше збіг. Також для додаткової корекції результату кожного кандидата використовувалася частота використання слова (кількість появ в Британській Національній Колекції).

3. Алгоритм корекції слів з помилками

В алгоритмі, що описаний в даній статті, для представлення слів використовувалися розподілені представлення. Розподілене представлення інформації – це форма векторного представлення, де кожен об'єкт представлений сукупністю елементів вектора, а окремий елемент вектора може належати представленням різних об'єктів [3-12]. Такі представлення були впроваджені в систему перевірки орфографії, щоб використовувати їх переваги для перевірки коректності слів.

Розподілені представлення формувалися на основі апріорних знань про ознаки фонем англійської мови. Якщо в попередніх дослідженнях в області представлення образів бінарними розподіленими кодвекторами, для представлення використовувалися випадкові кодвектори (для несхожих образів) [10-12], то в даному алгоритмі з метою відображення подібності букв і слів було використано невипадкове представлення.

Наприклад, графеми «ph» і графема «f» представляють одну і ту ж саму фонему /f/ [13,14]. Відповідно, слова, що містять ці графеми, повинні мати схожість кодвекторів.

Перевагами невипадкового представлення є те, що комбінації природних ознак підвищують їх дискримінаційні властивості. В такому випадку кожна ознака зазвичай характеризує не унікальний об'єкт, а цілий ряд об'єктів (одна і та ж частота може бути включена в різні фонеми), а збільшення набору виділених ознак дозволяє більш точно описувати конкретний об'єкт.

Безумовно, важливою характеристикою слова є послідовність букв в ньому, а не тільки їх сукупність. Тобто різним словам, які містять однакові букви, повинні відповідати різні кодвектори, що враховують різні послідовності букв в слові. Такі відношення виду R (A, B, ..., де R - ідентифікатор відносини, A, B ... - аргументи), у яких важлива послідовність їх аргументів, є направленими. З метою врахування цього, був запропонований метод формування кодвекторів таких відносин.

Для цього використовувалося кодування пар букв. Для кожної графеми було виділено поле вектора (972 біта вектора). Це поле ділилося на поля зв'язків з іншими буквами. Кількість таких полів дорівнювала кількості букв в алфавіті. Залежно від комбінації літер у парі, бітам певної ділянки вектора присвоювалися одиниці. Таким чином, кожній парі букв присвоювався унікальний кодвектор, що складається з 15 000 бітів. Формування кодвекторів відбувалося таким чином, що одиниці в векторі розташовувалися у вигляді концентрованих груп - ансамблів.

Таким чином, на основі кодвекторів пар букв будувалися вектора слів. Для цього кодвектори пар букв з'єднувалися по диз'юнкції. Бінарні кодвектори слів, які утворювались на виході, записувалися в словник.

4. Пошук коректного слова

Для пошуку кодвекторів слів у словнику використовувався метод пошуку найближчих аналогів [15-17].

Пошук найбільш близького аналога в пам'яті зводиться до знаходження (у пам'яті, де зберігаються кодвектори слів) кодвектору, найбільш схожого на вхідний.

Пошук найближчих аналогів здійснюється за величиною різниці перекриття одиниць і різних бітів їх кодвекторів з бази X_l з кодвектором вхідного аналога $X_{вх}$:

$$l^*(x_{вх}) = \arg\max_{l=1,L}\left(V(X_{вх}, X_l) - Z(X_{вх}, X_l)\right) \qquad (1)$$

де l = 1, L - індекс кодвектора в базі, L - число аналогів (епізодів, слів і т.п.) в БЗ, V (.,.) - величина перекриття кодвекторів, Z (.,.) - кількість відмінних бітів кодвекторів.

У задачі пошуку найближчих аналогів для знаходження величини подібності кодвекторів застосований метод зворотного індексування [18].

Таким чином, для кожного кандидата (вектору зі словника) обчислюються величини подібності, на основі яких вектори-кандидати можна впорядкувати.

Вектор з найбільшою величиною подібності повинен бути найбільш підхожим кандидатом і, відповідно, вектором слова, яке потрібно знайти.

Для подальшого порівняння з іншими системами формувався список з 10-и найбільш схожих векторів-кандидатів на вхідний.

5. Результати експериментів

Таким чином, був створений описаний алгоритм пошуку коректних слів, відповідних вхідним словам з помилками.

Працездатність алгоритму перевірена на двох наборах слів, що містять реальні орфографічні помилки, які і використовувалися для експериментів: aspell і Wikipedia [19, 20].

Для обчислення точності визначення коректних слів-кандидатів була використана наступна формула:

$$Top\_n = \frac{t_n}{t_t} * 100\% \qquad (2)$$

де t_t - кількість пар слів (коректне слово і слово з помилками) в базі. Всі лічильники t_n (n = 1,2,3,5,10 - кількість слів-кандидатів) збільшувалися, якщо коректне слово з'являлося в межах n позицій кандидатів.

Порівняння результатів кількох систем обробки орфографії на базах слів з помилками Aspell та wikipedia показані в табл. 1 та 2:

Таблиця 1: Порівняння результатів програм перевірки орфографії (база aspell)

	aspell Деоровіц та Кіура	aspell Міттон	aspell (ця програма)
Перший	66,3%	71,1%	58,6%
Топ два	75,5%	83,2%	69,7%
Топ три	79,6%	88,6%	77,7%
Топ п'ять	83,6%	91,4%	82,4%
Топ десять	85,5%	94,4%	88,9%
Всього = 100%	511	499	488

Таблиця 2: Порівняння результатів програм перевірки орфографії (база wikipedia)

	wikipedia Деоровіц та Кіура	wikipedia Міттон	wikipedia (ця програма)
Перший	94,1%	92,9%	80,0%
Топ два	97,4%	97,2%	89,4%
Топ три	98,3%	97,9%	92,8%
Топ п'ять	98,9%	98,6%	95,7%
Топ десять	99,0%	99,0%	97,5%
Всього = 100%	2196	2154	2284

6. Висновки

В даному підході не розроблялись спеціальні правила для конкретних мов, а використовувалися загальні методи на основі бінарних розподілених представлень. Тим не менше, результати показують можливість застосувати розподілені представлення з метою корекції орфографічних помилок у словах, а використані методи дозволяють отримати при цьому якісні результати.

Недоліком такої системи є обробка ізольованих слів. В подальшому можливості таких систем можна розширити, впроваджуючи в вектори слів синтаксичні та семантичні ознаки [21-23]. Такі ознаки можуть допомогти як в поліпшенні вибору кола найбільш ймовірних кандидатів, так і в їх більш коректному ранжируванні.

Перспективними напрямками подальших досліджень є також:

- Використання для знаходження найближчих кодвекторів нейромережевої асоціативної пам'яті матричного типу [24-30];
- Застосування парсерів для виділення більшого обсягу інформації з тексту;
- Розподілене представлення виділеної інформації за допомогою методів розподіленого представлення послідовностей та структур [9,10-12,15-17,31].

7. Література

[1] Roger Mitton, «Ordering the suggestions of a spellchecker without using context», Natural Language Engineering, N. 15 (2), 2009, p. 173-192.

[2] Sebastian Deorowicz, Marcin G. Ciura, «Correcting Spelling Errors by Modeling their Causes», International Journal of Applied Mathematics and Computer Science, N. 12(2), 2005, p. 275-285.
[3] Thorpe S., «Localized Versus Distributed Representations», Arbib M. The Handbook of Brain Theory and Neural Networks – Cambridge, MA: MIT Press, 2003, p.643-646.
[4] Browne A., Sun R., «Connectionist Inference Models», Neural Networks, Vol. 14, N 10, 2001, p. 1331-1355.
[5] Eliasmith C., Thagard P., «Integrating Structure and Meaning: A Distributed Model of Analogical Mapping», Cognitive Science, Vol. 25, N 2, 2001, p. 245-286.
[6] Amosov, N. M., Baidyk, T. N., Goltsev, A. D., Kasatkin, A. M., Kasatkina, L. M., Kussul, E. M., Rachkovskij D. A., «Neurocomputers and intelligent robots», Kiev: Naukova dumka, 1991, p. 269.
[7] Rachkovskij D.A., «Representation and Processing of Structures with Binary Sparse Distributed Codes», IEEE Transactions on Knowledge and Data Engineering, Vol. 13, N 2, 2001, p. 261-276.
[8] Rachkovskij D.A., Misuno I.S., Slipchenko S.V., «Randomized projective methods for construction of binary sparse vector representations», Cybernetics and Systems Analysis, N 1, 2012, p. 146-156.
[9] Gritsenko V.I., Rachkovskij D.A., Goltsev A.D., Lukovych V.V., Misuno I.S., Revunova E.G., Slipchenko S.V., Sokolov A.M., «Neural distributed representation for intelligent information technologies and modeling of thinking», Cybernetics and computer engineering, 2013, p. 7-24.
[10] Rachkovskij D.A., Kussul E.M., Baidyk T.N., «Building a world model with> structure-sensitive sparse binary distributed representations», Biologically Inspired Cognitive Architectures, 2013, p. 64-86.
[11] Rachkovskij D.A., Slipchenko S.V., Kussul E.M., Baidyk T.N., «Binding procedure for distributed binary data representations», Cybernetics and Systems Analysis, N3, 2005, p. 319-331.
[12] Rachkovskij D.A., Kussul E.M., «Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning», Neural Computation, Vol. 13, N 2, 2001, p. 411-452.
[13] Letters and Sounds: Principles and Practice of High Quality Phonics Notes of Guidance for Practitioners and Teachers Andrew Adonis Rt Hon Beverly Hughes 11 May 2010. Файл доступний на: http://www.education.gov.uk/
[14] Nima Mesgarani, Stephen David, Shihab Shamma, «Representation of phonemes in primary auditory cortex: how the brain analyzes speech», Electrical and Computer Engineering Department University of Maryland, College Park, MD 20742, ICASSP, 2007.
[15] Rachkovskij D.A, Slipchenko S.V., «Similarity-Based Retrieval with Structure-Sensitive Sparse Binary Distributed Representations», Computational Intelligence, Vol. 28, Issue 1, 2012, p. 106-129.
[16] Rachkovskij D.A., «Some approaches to analogical mapping with structure sensitive distributed representations», Journal of Experimental and Theoretical Artificial Intelligence, N 3, 2004, p. 125-145.
[17] Slipchenko S.V., Rachkovskij D.A., «Analogical mapping using similarity of binary distributed representations», International Journal Information Theories and Applications, N 3, 2009, p. 269-290.
[18] Слипченко С.В., Рачковский Д.А., Мисуно И.С., «Декодирование разреженных бинарных распределенных кодов скалярных и векторных величин», Компьютерная математика, № 3, 2005, с. 108-120.
[19] Atkinson K., «Spell Checker Test Kernel Results», 2011, Файл доступний на: http://aspell.net/test/cur/
[20] «Wikipedia Corpora of misspellings», Файл доступний на: http://www.dcs.bbk.ac.uk/~roger/wikipedia.dat.
[21] Misuno I. S., Rachkovskij D.A., Slipchenko S.V., «Vector and distributed representations reflecting semantic relatedness of words», Mathematical machines and systems, N 3, 2005, p. 50-67.
[22] Sokolov A., «LIMSI: learning semantic similarity by selecting random word subsets», Proceedings of the Sixth International Workshop on Semantic Evaluation (SEMEVAL'12). – Association for Computational Linguistics, 2012, p. 543-546.
[23] Sokolov A., Riezler S., «Task-driven greedy learning of feature hashing functions», Proceedings of the NIPS'13 Workshop "Big Learning: Advances in Algorithms and Data Management", Lake Tahoe, USA, 2013, p. 1–5.
[24] Frolov A.A., Rachkovskij D.A., Husek D., «On informational characteristics of Willshaw-like auto-associative memory», Neural Network World 12 (2), 2002, p. 141-158.
[25] Frolov A.A., Husek D., Rachkovskij D.A., «Time of searching for similar binary vectors in associative memory», Cybernetics and Systems Analysis, N 5, 2006, p. 615-623.
[26] Frolov A., Kartashov A., Goltsev A., Folk R., «Quality and efficiency of retrieval for Willshaw-like autoassociative networks. I. Correction», Network: Computation in Neural Systems, N 4, 1995, p. 513-534.
[27] Frolov A., Kartashov A., Goltsev A., Folk R., «Quality and efficiency of retrieval for Willshaw-like autoassociative networks. II. Recognition», Network: Computation in Neural Systems, N 4, 1995, p. 535-549.
[28] Frolov A.A., Husek D., Polyakov P.Yu., «Recurrent-neural-network-based boolean factor analysis and its application to word clustering», IEEE Transactions On Neural Networks, N 7, 2009, p. 1073-1086.
[29] Nowicki D.W., Dekhtyarenko O.K., «Averaging on Riemannian manifolds and unsupervised learning using neural associative memory», Proc. ESANN 2005. – Bruges, Belgium, April, 27-29, 2005, p. 181-189.
[30] Nowicki D., Siegelmann H., «Flexible kernel memory», PLoS ONE, N6. – e10955. doi:10.1371/journal.pone.0010955, 2010.
[31] Kussul E.M., Rachkovskij D.A., «Multilevel assembly neural architecture and processing of sequences», In A.V. Holden & V. I. Kryukov (Eds.), Neurocomputers and Attention: Vol. II. Manchester and New York: Manchester University Press, 1991, p. 577-590.