УДК 05.13.06

КОРРЕКЦИЯ СЛОВ С ОШИБКАМИ С ПОМОЩЬЮ РАСПРЕДЕЛЕННЫХ БИНАРНЫХ РАЗРЕЖЕННЫХ ПРЕДСТАВЛЕНИЙ

Омельченко Р. С.
Международный научно-учебный центр информационных технологий и систем НАН и МОН Украины

Введение

Алгоритм коррекции слов с ошибками позволяет находить и исправлять ошибки в словах. Как правило, такие системы предлагает пользователю короткий список предполагаемых правильных слов в последовательности от самого вероятного к наименее вероятному.

В этой работе использованы распределенные представления, а также методы их обработки (создания, поиска) для коррекции слов с ошибками. Для оценки и сравнения с другими системами использовались два набора слов с типичными орфографическими ошибками.

Цель работы

Цель данной работы состоит в разработке алгоритма коррекции слов с ошибками на основе распределенных представлений, который мог бы выявлять ошибки в словах и исправлять их.

Описание алгоритма коррекции слов с ошибками

Основным отличием описываемого алгоритма коррекции слов с ошибками (проверки орфографии) от предыдущих алгоритмов [1, 2] является представление слов, а также методы проверки. Для представления слов использовались распределенные бинарные разреженные кодвектора.

Распределенное представление информации - форма векторного представления, где каждый объект представлен совокупностью элементов вектора, а отдельный элемент вектора может принадлежать представлениям разных объектов [3]. Такие представления были внедрены в систему проверки орфографии, чтобы использовать их достоинства для проверки корректности слов [3, 4, 5, 6].

В продолжение предыдущих исследований в области представления образов бинарными распределенными кодвекторами, где для представления использовались случайные кодвекторы (для непохожих образов) [7], для данного алгоритма проверки орфографии использовались распределенные представления, формируемые на основе априорных знаний о задаче, а именно используя данные о признаках фонем английского языка. Другими словами, с целью отображения

сходства букв и слов было использовано неслучайное представление. Например, графемы «ph» и графема «f» представляют одну и туже фонему /f/ [8,9]. Соответственно, слова, содержащие эти графемы, должны иметь схожесть кодвекторов.

С целью учета последовательности букв в слове, был предложен метод кодирования слов через кодирование последовательности букв слова. Безусловно, важной характеристикой слова является последовательность букв в нем, а не только их совокупность.

Учитывая это, в работе был предложен новый метод формирования кодвекторов таких отношений. Для этого использовалось кодирование пар букв. Для каждой графемы было выделено поле вектора (972 бита вектора). Это поле делилось на поля связей с другими буквами. Количество таких полей равнялось количеству букв в алфавите. В зависимости от комбинации букв в паре, битам определенного участка вектора присваивались единицы. Таким образом, каждой паре букв присваивался уникальный кодвектор, состоящий из 15 000 битов. Формирование кодвекторов происходило таким образом, что единицы в векторе располагались в виде концентрированных групп – ансамблей.

После того, как каждой паре букв был присвоен кодвектор, следующей задачей являлось формирование словаря, а именно бинарных кодвекторов, которые соответствовали бы словам. Для кодирования слов вектора пар букв соединялись по дизъюнкции. При этом для построения слова использовались биграммы соседних букв в прямом направлении, а также все возможные варианты пар букв в обратном направлении, замыкая связи между буквами в круг.

Например, в слове «came» по дизъюнкции объединялись вектора 9 пар букв: 3 прямых: «ca», «am», «me» и 6 обратных: «ac», «ma», «mc», «ec», «ea», «em».

Кроме этого, чем дальше расположена в слове буква относительно второй буквы в паре, тем меньше количество единиц в векторе этой пары. Таким образом, кодировалось относительное положение букв в слове.

Сравнение слов и поиск корректного слова

После того, как формируется кодвектор слова, он записывается в файл, так называемый словарь. Для поиска кодвекторов слов использовался метод поиска ближайших аналогов [10].

Поиск наиболее близкого аналога в памяти сводится к нахождению (в памяти, где хранятся кодвекторы слов) кодвектора, наиболее похожего на входной.

Поиск ближайших аналогов осуществляется по величине разницы перекрытия единиц и разных битов их кодвекторов из базы $\mathbf{X}_l$ с кодвектором входного аналога $\mathbf{X}_{вх}$

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

где $l=1,L$ – индекс кодвектора в базе, L – число аналогов (эпизодов, слов и т.п.) в БЗ, V(.,.) – величина перекрытия кодвекторов, Z(.,.) – количество отличающихся битов кодвекторов.

В задаче поиска ближайших аналогов для нахождения величины сходства кодвекторов применен метод обратного индексирования [11].

Вектор с наибольшей величиной сходства должен быть наиболее подходящим кандидатом и соответственно, вектором слова, которое требовалось найти. Для дальнейшего сравнения с другими системами формировался список из 10-и наиболее похожих векторов-кандидатов на входной.

Результаты работы

Таким образом, был создан описанный алгоритм поиска корректных слов, соответствующих входным словам с ошибками.

Работоспособность алгоритма проверена на двух наборах слов, содержащих реальные орфографические ошибки, которые и использовались для экспериментов: aspell и Wikipedia [12, 13].

Для вычисления точности определения корректных слов-кандидатов использовалась следующая формула:

$$Top\_n = \frac{t_n}{t_t} * 100\%,$$

где t_t – количество пар слов (корректное слово и слово с ошибками) в базе. Все счетчики t_n (n=1,2,3,5,10 – количество слов-кандидатов) увеличивались, если корректное слово появлялось в пределах n позиций кандидатов.

Сравнения результатов нескольких систем обработки орфографии на примере базы слов с ошибками Aspell показаны в таблице:

	aspell Деорович и Киура	aspell Миттон	aspell (эта программа)
First	66,3%	71,1%	58,6%
Top two	75,5%	83,2%	69,7%
Top three	79,6%	88,6%	77,7%
Top five	83,6%	91,4%	82,4%
Top ten	85,5%	94,4%	88,9%
Total = 100%	511	499	488

Несмотря на то, что в данном подходе не разрабатывались специальные правила для конкретных языков, а использовались общие методы на основе бинарных распределенных представлений, результаты показывают возможность применить распределенные представления с целью исправления орфографических ошибок в словах, а также при этом получить качественные результаты.

Выводы

Недостатком такой системы является обработка изолированных слов. Тем не менее, в дальнейшем возможности таких систем можно расширить, внедряя в такие вектора слов синтаксические и семантические признаки. Причем в данном случае такие признаки могут помочь в улучшении как выбора круга наиболее вероятных кандидатов, так и в их более корректном ранжировании.

Литература

1. Roger Mitton. Ordering the suggestions of a spellchecker without using context / Roger Mitton // Natural Language Engineering – 2009. – N. 15 (2). – P. 173-192.
2. Sebastian Deorowicz. Correcting Spelling Errors by Modeling their Causes / Sebastian Deorowicz , Marcin G. Ciura // International Journal of Applied Mathematics and Computer Science – 2005. – N. 12(2). – P. 275-285.
3. Thorpe S. Localized Versus Distributed Representations / Thorpe S. // Arbib M. The Handbook of Brain Theory and Neural Networks – Cambridge, MA: MIT Press, 2003. – P.643-646.
4. Browne A. Connectionist Inference Models / Browne A., Sun R. // Neural Networks – 2001. – Vol. 14, N 10. – P. 1331-1355.
5. Eliasmith C. Integrating Structure and Meaning: A Distributed Model of Analogical Mapping / Eliasmith C., Thagard P. // Cognitive Science – 2001. – Vol. 25, N 2. – P. 245-286.
6. Rachkovskij D.A. Representation and Processing of Structures with Binary Sparse Distributed Codes / Rachkovskij D.A. // IEEE Transactions on Knowledge and Data Engineering. – 2001. – Vol. 13, N 2. – P. 261-276.
7. Rachkovskij D.A. Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning / Rachkovskij D.A., Kussul E.M. // Neural Computation. – 2001. – Vol. 13, N 2. – P. 411-452.
8. 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. [Электронный ресурс]. Режим доступа: http://www.education.gov.uk/
9. Nima Mesgarani. «Representation of phonemes in primary auditory cortex: how the brain analyzes speech». / Nima Mesgarani, Stephen David, Shihab Shamma. // Electrical and Computer Engineering Department University of Maryland, College Park, MD 20742, ICASSP. – 2007.

10. Dmitri A Rachkovskij «Similarity-Based Retrieval with Structure-Sensitive Sparse Binary Distributed Representations» / Dmitri A Rachkovskij, Serge V Slipchenko // Computational Intelligence. – 2012. – Vol. 28. – Issue 1. – P: 106-129.
11. Слипченко С.В. Декодирование разреженных бинарных распределенных кодов скалярных и векторных величин / Слипченко С.В., Рачковский Д.А., Мисуно И.С. // Компьютерная математика. – 2005. – № 3. – С. 108-120.
12. Atkinson K. Spell Checker Test Kernel Results. / Atkinson K. [Электронный ресурс]. – 2011. Режим доступа: http://aspell.net/test/cur/
13. Wikipedia Corpora of misspellings [Электронный ресурс]. Режим доступа: http://www.dcs.bbk.ac.uk/~roger/wikipedia.dat.