A Survey of Spelling Error Detection and Correction Techniques

Ritika Mishra, Navjot Kaur

Обзор подготовил Краснов Станислав.

Настоящая статья аккумулирует сведения о значительном количестве используемых на данный момент подходов к созданию спеллчекера – программы, способной помечать неверные с точки зрения орфографии слова в тексте и предлагать варианты их исправления.

Как известно, ошибки спеллинга делятся на две большие группы – real-word errors и non-word errors. Первая группа объединяет ошибки, в результате которых вместо целевого слова было получено другое слово, присутствующее в языке (например, 'rat' вместо 'cat'). Вторая группа имеет дело со словами, чья орфография была искажена настолько, что найти подобное слово в словаре едва ли возможно (скажем, 'xat' вместо 'cat').

Прежде всего спеллчекеру требуется обнаружить ошибку в тексте. Здесь используется два основных метода – обращение к словарю (dictionary lookup) и анализ n-грамм (n-gram analysis). Первый метод просто проверяет наличие слова в словаре и помечает слово как ошибку, если ничего не обнаружил. Чаще всего для более быстрого доступа к словарю используют хэш-таблицы, содержащие хэш-индексы отдельных слов, с которыми сравниваются подсчитанные хэш-индексы слов в анализируемом тексте. Анализ n-грамм работает схожим образом уже не с отдельными словами, а с коллокациям из n слов (монограммами, биграммами, триграммами и т. д.) и их частотной встречаемостью (чем ниже вероятность встретить ту или иную n-грамму, тем выше вероятность, что спеллчекер посчитает её за ошибку).

Когда с ошибками в тексте определились, приходит время выдвигать предположения, как их можно исправить. Здесь в дело вступают следующие методы:

- *Минимальная дистанция редактирования (minimal edit distance)*, или минимальное кол-во базовых операций (вставка, удаление или замена символа), требуемых для перехода от одного слова (строки) к другому. Обычно для редактирования сравниваются исходное слово с ошибкой и m близких слов из словаря: слово, находящееся на минимальной (по сравнению с другими) дистанции редактирования к исходному, выбирается в качестве исправленной альтернативы. Известны три основных алгоритма подсчёта данной дистанции: алгоритм Левенштейна (каждая базовая операция имеет стоимость в 1 у. е.), алгоритм

Хэмминга (подсчитывает количество отличий в позициях элементов одинаковых по длине строк) и алгоритм наибольшей общей подпоследовательности (как видно из названия, считает пересечение множеств элементов исходных строк).

- *Методы, основанные на принципах схожести звукового/буквенного состава слов (similarity key techniques).* Это алгоритмы, одинаково индексирующие в строках либо схожие звуки (Soundex), либо одинаковые буквы (SPEEDCOP) и, соответственно, приближающие слова со схожей транскрипцией или написанием. Благодаря реорганизации формы исходного слова, алгоритмы этого класса используют куда более компактные по объёму словари. Кроме того, SPEEDCOP записывает все слова в виде буквенной последовательности, из первой буквы слова, затем всех согласных (без повторений) и в конце всех гласных (тоже без повторений). Полученные последовательности сравниваются с достаточно обширным словарём. Это позволяет исправить весомую долю ошибок (например, задвоений букв), допущенных при быстрой печати.

- *Правиловые методы (rule based techniques).* Данный подход ничем не отличается от других правиловых подходов: есть набор правил с чётко прописанным входом и выходом, и эти правила в заданном порядке применяются к слову с ошибкой.

- *Вероятностные методы (probabilistic techniques)* рассматривают вероятности перехода (transition probabilities) и вероятности ошибок (confusion probabilities). Первые на основе корпусных данных оценивают зависимость вероятности появления элементов в последовательности от наличия предыдущих. Вторые измеряют частоту подмены одного элемента в последовательности другим, довольно зависимы от типа источника анализа и часто применяются в таких устройствах, как OCR (оптического распознавания символов).

- *N-граммные методы* аналогичны своим братьям и сёстрам из блока обнаружения ошибок. Также весьма любимы устройствами OCR.

- *Нейронные сети.* Куда же сейчас без них. Отличительно хороши, когда входные данные неполны или зашумлены. Особенно успешным себя показывает метод обратного распространения ошибки (back propagation algorithm) – сеть, состоящая из трёх слоёв – входного, промежуточного и выходного. Количество узлов в слое может достигать больших значений. Каждый узел во входном слое связан весами с каждым узлом в слое выходном. Сигналы ошибки распространяются от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Информация на входах и

выходах представлена моделью «включён» («1») / «выключен» («0») в соответствии с поведением узлов на входах и выходах.

Таким образом, данная статья охватывает довольно широкий спектр методов, используемых в современном спеллчекинге, и даёт первичное представление о том, как эти методы работают на реальных примерах. Из минусов хотелось бы отметить присущую современной компьютерной лингвистике «англоцентричность»: то, что хорошо работает с английским языком, не всегда адекватно справляется с задачами других языков, особенно когда речь идёт о языках – условно – экзотических, со слоговой или иероглифической системой письма. Однако это – уже предмет отдельного исследования.