

Метод вероятностного морфологического анализа для задач полнотекстового индексированного поиска

© Артемьев К.

Астраханский государственный технический университет
konstantin@vistaglance.com

Аннотация

Рассматривается традиционный подход к задаче морфологического анализа слова, указываются его недостатки при применении в системах полнотекстового поиска. Предлагается алгоритм морфологического анализа слов для целей построения обратного индекса в задаче полнотекстового поиска, основанный на вероятностном подходе. Вводится понятие морфологической эквивалентности, рассматривается способ и формула для вычисления морфологической эквивалентности пары слов. Описывается эксперимент, доказывающий работоспособность алгоритма. Особенности предложенного автором алгоритма являются его высокая скорость и нечувствительность к языковым и тематическим неоднородностям текстов в корпусе.

1. Введение

В настоящее время особое значение приобретают системы «мгновенного» локального полнотекстового поиска по содержимому документов различных типов. Это связано с повсеместным переходом предприятий на электронный документооборот. Количество документов, с которыми сотрудник должен работать за день, увеличивается с каждым годом в геометрической прогрессии. Поэтому особенно важной становится задача

поиска по содержимому за конечное время (равное нескольким секундам) в большом объёме документов (50 000 и больше).

Результаты поиска должны быть нечувствительны к морфологическим формам слов в поисковом запросе. Кроме того, поиск зачастую производится по базе, содержащей документы на двух и более различных языках (вот лишь некоторые часто используемые сочетания: английский, русский; английский, русский, французский; английский, русский, иврит). При этом несколько разных языков могут чередоваться даже в рамках одного документа.

Для задач полнотекстового поиска документов с учётом морфологии их языка перед построением обратного индекса слов необходимо преобразовать каждое слово документа к некоей общей основе. Таким образом, в обратном индексе будет храниться только одна форма слова. Алгоритм морфологического анализа применяется дважды: в момент индексирования документов, чтобы преобразовать все возможные формы одного слова к одной единственной, и во время поиска, чтобы преобразовать слова запроса именно к тем их морфологическим формам, которые хранятся в обратном индексе.

Классическим подходом к задаче приведение разных форм слова к единой основе является стемминг. Стеммером называется конечный автомат, который итеративно отбрасывает изменяемые части слова по правилам морфологии конкретного языка, например русского. Давно и успешно применяется стеммер Портера [1,2], реализованный на момент написания данной работы для 16 языков (включая русский). Стеммер Портера качественно приводит слова к общей морфологической основе, однако у него есть ряд существенных недостатков.

Первый из таких недостатков – алгоритмическая сложность и скорость работы. Для среднестатистического языка каждое слово необходимо проверить с помощью нескольких десятков, а то и сотен правил из набора конечного автомата. Именно скорость работы такого алгоритма становится камнем преткновения, когда необходимо за разумный срок проиндексировать корпус текстов, генерирующий обратный индекс из нескольких миллионов слов.

Второй недостаток заключается в том, что при появлении нового языка в корпусе текстов придётся добавлять новый алгоритм стемминга для данного конкретного языка, причём вручную. Кроме того, во многих случаях стеммер не способен сам определить, что слово было написано на неизвестном для него языке (это актуально для языков с совпадающими или пересекающимися алфавитами). Стеммер попытается применить к этому слову правила чужого языка, что приведёт к серьёзным ошибкам в работе системы.

Альтернативным подходом к морфологическому разбору слов является вероятностный анализ. Он базируется на том предположении, что в корпусе текстов в неявном виде уже содержится вся информация о морфологии используемых языков. Вероятностные морфологические анализаторы не нуждаются в предварительно написанных правилах морфологического разбора и нечувствительны к добавлению в корпус текстов новых языков.

Автор данной работы предлагает простой и эффективный алгоритм вероятностного морфологического анализа слов, специально адаптированный по скорости к задаче полнотекстового поиска по многоязычной коллекции документов.

2. Описание алгоритма

На вход морфологического анализатора обычно поступает несортированный список слов от парсера, т.е. модуля, читающего с диска содержимое документа с учётом его внутреннего формата и разбивающего документ на отдельные слова. В целях увеличения быстродействия такой список целесообразно превратить в сортированный ещё на этапе парсинга, так как вставка в уже отсортированный список новых слов в лексикографическом порядке займёт меньше времени, чем сортировка готового списка.

Для пары слов введём понятие морфологической эквивалентности. Неформально под этим термином будем понимать вероятность, с которой два слова имеют одну и ту же семантику (что зачастую означает, что они имеют одинаковые основы). Очевидно, что морфологическая эквивалентность двух слов должна выражаться вещественным числом в интервале $[0,1]$, при этом ноль должен выражать отношение между абсолютно различными словами (имеющими разные начальные буквы), а единица – между одинаковыми словами.

Попробуем эмпирически вывести формулу для вычисления морфологической эквивалентности слов $S1$ и $S2$. Допустим, длины этих слов равны соответственно $L1$ и $L2$. Найдём длину общей основы этих слов L , т.е. количество совпадающих символов в строках $S1$ и $S2$, начиная с их начала. Тогда отношение $L/L1$ покажет, какую часть слова $S1$ составляет общая основа длины L , а отношение $L/L2$ – какую часть слова $S2$ составляет эта же общая основа. Для нахождения степени взаимной эквивалентности M слов логично перемножить эти два отношения (1):

$$M = \frac{L^2}{L1 * L2}. \quad (1)$$

Взглянув на полученную формулу, нетрудно заметить, что она выражает квадратичную зависимость величины M от той величины, которую мы подразумевали под морфологической эквивалентностью двух слов. Для удобства дальнейших расчётов и сравнения разных слов между собой сделаем эту величину линейной (2). Мы получили финальную формулу для вычисления морфологической эквивалентности двух слов:

$$M = \frac{L}{\sqrt{L1 * L2}} . \quad (2)$$

Пройдёмся по сортированному списку слов документа и найдём для каждой пары стоящих рядом в этом списке слов их морфологическую эквивалентность. В Таблице приведёна выдержка из списка слов типового документа, участвовавшего в тестировании алгоритма.

Слово	M(текущее слово, следующее слово)
полуприцеп	0,953462600708008
полуприцепа	0,870388269424438
полуприцепов	0,480384469032288
полупроводник	0,846153855323792
полупроводной	0,859337866306305
полупроводность	0,544949233531952
полупродукт	0,957427084445953
полупродукты	0,540061712265015
полупрозрачный	0,45374259352684
полупроницаемость	

Нетрудно заметить, что для пар слов, имеющих одинаковую семантику, но находящихся в разных формах, функция M возвращает значительно большие значения, чем для пар слов, имеющих разную семантику, несмотря даже на то, что все слова в списке начинаются практически одинаково. Можно предположить, что существует некое граничное значение K функции M , до которого абсолютное большинство пар слов будут иметь разную семантику, а после которого – одинаковую. Задача нахождения оптимального K и проверки этого предположения на практике будет рассмотрена в следующем разделе данной статьи, а пока посмотрим, какую выгоду даёт нам такое предположение.

После того, как для каждой пары соседних по сортированному списку слов найдена морфологическая эквивалентность M , можно удалить из списка все рядом стоящие слова, для которых $M > K$, и заменить их каким-

то одним, наиболее общим словом для удалённой группы. Автор предлагает пойти ещё дальше и заменить два прохода алгоритма по списку слов одним, используя следующий подход: в первом и единственном проходе для каждого текущего и следующего за ним слова вычисляется значение функции M , и если оно превышает K , оба слова удаляются из списка, а вместо них в позицию текущего слова вставляется их общая основа длины L .

Таким образом, алгоритм преобразует список слов в список основ, причём каждая основа в большинстве случаев несёт свою семантику, отличную от семантик рядом стоящих основ. Эксперименты, проведённые автором, показали, что этот подход позволяет уменьшить объём обратного индекса в среднем в три раза для корпуса текстов, состоящего из нескольких тысяч документов разных тематик, и в десятки раз для корпуса текстов с однородной тематикой. Пропорционально уменьшению объёма обратного индекса увеличивается и скорость поиска в нём. Использование сортированного списка делает возможным применение бинарного поиска. А это значит, что в худшем случае для нахождения нужной основы в обратном индексе, состоящем из миллиона элементов, потребуется не больше 20 обращений к индексу (во-первых, $2^{20} > 1000\ 000$, а во-вторых, мы предполагаем, что распределение основ по индексу однородное).

3. Методика нахождения оптимального значения K

Для проверки выдвинутого выше предположения, а также для нахождения оптимального значения K автором была проведена серия экспериментов. За основу брался корпус текстов, состоящий из 5000 документов с примерно равномерным распределением по ним следующий тематик: computer science, юриспруденция, медицина, философия, офисные документы (отчёты, ведомости, приказы и т.д.). Треть документов была на английском языке, остальные на русском. Большинство русскоязычных документов содержали некоторое количество английских слов. С использованием вышеприведённого алгоритма по корпусу текстов был построен обратный индекс основ, состоявший, в конечном счете, из чуть более миллиона элементов.

Проводилось 50 равномерных выборок по 20 основ в каждой из списка основ обратного индекса. Для каждой выборки вручную подсчитывалось количество ошибок первого и второго рода.

К ошибкам первого рода мы относим появление в индексе рядом стоящих основ, имеющих одинаковую семантику. К ошибкам второго рода мы относим пропадание из списка основ независимой по своей се-

мантике основы слова, присутствовавшего в исходном списке. В таких случаях алгоритм ошибочно считал, что два слова имеют одинаковую семантику, и на этом основании заменял их одной основой.

Очевидно, что последствием ошибок первого рода является появление избыточности в обратном индексе. Это ведёт так же к небольшому увеличению времени поиска, но не ведёт к ухудшению качества поиска. Последствия ошибок второго рода гораздо опаснее. Они приводят к тому, что из результатов поиска исчезают релевантные документы, либо наоборот, появляются нерелевантные. Причина в том, что одна основа начинает нести две или более семантических нагрузок, что неприемлемо.

Очевидно также, что задачи минимизации ошибок первого и второго рода противоречат друг другу. Следовательно, нужно найти между ними здоровый компромисс. Автору представляется логичным решение найти оптимальное значение K , стараясь свести к нулю ошибки второго рода, и при этом оставить допустимый минимум ошибок первого рода. Лучше недолечить больного, чем залечить его насмерть.

В результате вышеописанного эксперимента было найдено оптимальное значение K , равное 0,7. После работы алгоритма с $K=0,7$ в 50 выборках было обнаружено ноль ошибок второго рода и 9 ошибок первого рода (то есть менее одного процента от числа проверенных в результате выборов основ). Очевидно, что оптимальное значение K зависит как от тематик(и) корпуса текстов, так и от используемых языков. Для задач поиска с заранее известными языково-тематическими характеристиками корпуса текстов целесообразно повторить эксперимент для нахождения оптимального значения K . Тем не менее, автор считает, что найденное им значение K обеспечивает достаточно высокое качество индексации и поиска для большинства сфер применения этого алгоритма.

4. Заключение

В последнее время наметилась тенденция заменять точные, математически обоснованные, но медленные или трудоёмкие в подготовке алгоритмы их вероятностными модификациями. Коснулся этот подход и задачи морфологического анализа слов для полнотекстового поиска. Особенности предложенного автором алгоритма, решающего подобную задачу вероятностным методом, являются его скорость и нечувствительность к языковым и тематическим неоднородностям текстов в корпусе.

Был проведён эксперимент, доказывающий высокую эффективность алгоритма на практике. Однако следует понимать, что предложенный алгоритм неприменим в случаях, когда точность морфологического анализа

оказывается для разработчика на порядок важнее, чем скорость индексирования и поиска.

В настоящий момент, данный алгоритм применяется автором в рамках разрабатываемой им системы интеллектуального полнотекстового поиска среди документов, расположенных в локальных сетях.

Литература

- [1] C.J. van Rijsbergen, S.E. Robertson and M.F. Porter, 1980. New models in probabilistic information retrieval. London: British Library. (British Library Research and Development Report, no. 5587).
- [2] M.F. Porter, An algorithm for suffix stripping, Program, 14(3), 1980, pp. 130–137.

Method of probabilistic morphologic analysis for the purpose of full-text index-supported search

Artemev K.

Traditional approach to the task of morphologic analysis is reviewed, its disadvantages in conjunction with full-text search systems are outlined. Algorithm of morphologic analysis for the purpose of building reverse index in the task of the full text search based on probabilistic model is proposed. Concept of morphologic equivalency is introduced, the method and formula for its calculation for the pair of words is considered. Experiment proving workability of the algorithm is described, the method of finding the boundary value of K is outlined. The advantages of proposed algorithm are its speed, reliability and insensibility for language and thematic heterogeneity of the text corpus.