Идентификация языка UNICODE-текста по N-граммам длиной до 4-х включительно (квадрограммам)

С.Л. СОТНИК

Днепродзержинский государственный технический университет

В статье предлагается алгоритм определения языка текстов, написанных на языках индоевропейской группы. Особенностью алгоритма является возможность работы с многоязыковыми текстами, и возможность более скоростной реализации по сравнению с алгоритмом предложенным Cavnar W. B. и J. M. Trenkle.

У статті пропонується алгоритм визначення мови текстів, що написані на мовах індоєвропейської групи. Особливістю алгоритму є можливість роботи з багатомовними текстами, і можливість більш швидкісної реалізації в порівнянні з алгоритмом запропонованим Cavnar W. B. та J. M. Trenkle.

This article describes algorithm of language recognition for texts written in Indo-European group languages. The main features of algorithm are working with multi-language texts and capability of more rapid realization compare to Cavnar W.B. and J.M. Trenkle algorithm.

Введение в проблему

Порядок обработки информации индексирующей частью поисковых машин можно представить следующей схемой:

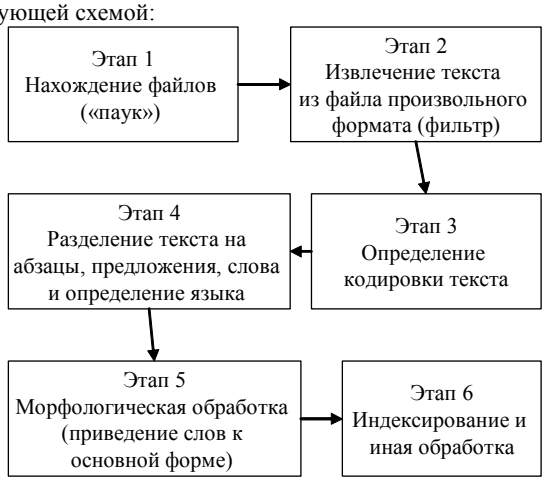

Рис. 1. Этапы обработки текстовой информации поисковыми машинами.

На практике часто объединяют этап 2 и 3, поскольку кодировка часто входит как составная часть во многие форматы документов. Этап 4 и 5 также иногда объединяют, определяя язык документа по принципу – слов какого языка распознано больше, тот и является языком текста. Современные компьютеры позволяют держать базы для десятков языков в оперативной памяти, поэтому не так уж важно, что такой подход требует выделения большого количества памяти, однако факт существенного (пропорционально количеству тестируемых языков) замедления обработки, будет иметь место и для компьютеров в десятки и тысячи раз более быстрых. На этапе 4 необходим алгоритм автоматической идентификации языка (индоевропейской языковой группы) предложений текста в кодировке UTF16 (двухбайтовые символы). Поэтому разработка быстрых, не требующих для своей работы больших словарей и позволяющих легко добавлять новые идентифицируемые языки алгоритмов, является актуальной.

В работе [1] был предложен метод для определения языка и кодировки документа по его содержимому, основываясь на статистиках документов, для которых язык и кодировка известны заранее. Подсчитываются частоты N-грамм (сочетаний символов или подстрок, длиной не более N) и предполагается, что примерно 300 самых часто используемых N-грамм сильно зависят от языка.

Алгоритм заключается в нахождении частот N-грамм для всех тестовых документов, для которых известен язык, а также для каждого документа, язык которого пытаемся определить. После этого среди всех тестовых документов находим тот, для которого расстояние от его N-граммной статистики до статистики тестируемого документа минимально. После этого языком тестируемого документа считается язык найденного тестового документа.

Расстояние между статистиками подсчитывается следующим образом: все N-граммы сортируются в порядке убывания частоты их появления, затем для каждой N-граммы вычисляется разница её позиций в отсортированном списке N-грамм тестового и тестируемого документов. Расстояние между статистиками определяется как сумма разностей позиций каждой N-граммы:

$$l = \sum_{i=1}^{300} \left| P_i - \widetilde{P}_i \right|, \qquad (1)$$

где P_i, $\widetilde{P}_i$ – позиции i-й N-граммы в тестовом и тестируемом документах соответственно.

Значение N предлагается использовать равным 5 (включительно). Вариант данного алгоритма (с несколько иной процедурой вычисления расстояния) реализован в поисковой машине MnogoSearch (www.mnogosearch.org).

Основными недостатками описанного выше алгоритма являются: неустойчивость определения языка малых текстовых фрагментов — отдельные предложения на похожих языках распознаются неуверенно, и часто с ошибками. Кроме того, использование сочетаний с длиной 5 не позволяет поместить их в 64 бита (ширина регистров современных процессоров и наибольший целочисленный тип данных, поддерживаемый современными компиляторами с большинства языков).

Иногда поступают еще проще. Например, в ABBYY RME, как минимум до четвертой версии, морфологическая машина при словарной лемматизации перебирала все загруженные языковые словари.

Следует также отметить, что процесс морфологической обработки является сравнительно трудоемким, сами словари занимают в памяти от единиц до десятков мегабайт на один язык, поэтому перебор даже по десятку возможных языков становится весьма медленным.

Постановка задачи

Необходим алгоритм автоматической идентификации языка (индоевропейской языковой группы) текста в кодировке UTF16 (двухбайтовые символы), который бы работал быстро, для своей работы не требовал больших словарей, позволял легко добавлять новые идентифицируемые языки.

Корпус документов

Для получения статистических характеристик каждого языка, использовался корпус документов, построенный следующим образом. В папке, соответствующей языку, находится файл, содержащий в себе все символы, встречающиеся в языке. Здесь надо отметить, что это не всегда алфавит языка, поскольку, во-первых, сюда включаются и строчные и заглавные буквы, а, во-вторых, в текстах очень часто встречаются символы, отсутствующие в официальные алфавитах. Таковыми, например, являются символы с акцентами в испанском и греческом языках.

Если язык использует уникальный набор символов, гарантированно не встречающихся в остальных языках (таковыми являются арабский, иврит, грузинский, армянский, греческий, если не принимать во внимание копский язык, документы на котором вряд ли кому-то встретятся), то такового списка символов будет вполне достаточно. Для остальных же языков дополнительно подбираются текстовые файлы, которые наиболее ярким образом представляют свой язык.

По мнению автора, лучше всего отвечают поставленной задаче литературные произведения, написанные носителем языка (не переводные). Именно такие произведения и легли в основу использованного корпуса документов.

Кроме того, на этапе разбора дополнительно осуществляется отсеивание слов, в которых обнаруживаются символы не из текущего алфавита – даже в литературных произведениях достаточно часто встречаются вставки на других языках.

Предварительный разбор

На этапе предварительного разбора производится разбор корпуса документов и выявление в нем:

1. Уникальных для языка N-грамм с длиной до 4-х символов включительно (включая и одиночные символы).
2. Часто встречающихся (но не уникальных по всему корпусу) в языке N-грамм с длиной до 4-х символов включительно.

Из тонкостей разбора можно отметить следующие моменты:

1. Тексты предварительно разбиваются на слова (монолитные последовательности из символов, соответствующих данному языку).
2. Все слова приводятся к одному регистру (например, к верхнему).
3. Сочетания кодируются следующим образом – если мощность N-граммы меньше, чем 4, то отсутствующие символы заменяются на символ с кодом 0.
4. К словам слева и справа добавляется по пробелу (или любой другой символ, гарантировано не являющийся буквой из какого-то алфавита). Это позволяет в дальнейшем отличить сочетание с мощностью до 3-х включительно, стоящее в конце, от такового же, стоящего в середине и в конце слова. Такая мелочь позволила увеличить точность распознавания языка примерно в 1.5 раза и более.
5. Уникальные N-граммы для того, чтобы попасть в окончательный список N-грамм, должны превысить определенный порог встречаемости (на практике использовался уровень в 10-15 раз). Это позволяет убрать случайные сочетания, нехарактерные для данного языка.
6. Часто встречающиеся сочетания сортировались по частоте, и выбиралось определенное их количество - N_{freq}. Эта процедура выполнялась отдельно для сочетаний разной длины. Т.е., при N_{freq}=16, в конечную таблицу сочетаний попадало 16 часто встречающихся квадрограмм, 16 триграмм, 16 биграмм. Для одиночных символов сделано исключение – для них количество выбирается как меньшее значение из N_{freq}, либо 1/8 от размера алфавита (учитывается количество символов одного регистра). Эксперименты показали, что оптимальное значение N_{freq} (для разных текстов и языковых наборов), лежит, как правило, в промежутке 64…128.

Подстройка весовых коэффициентов

Для выявления языка документа, к нему применялся разбор, аналогичный тому, который велся на предварительном этапе, только теперь никакие слова не отфильтровывались, а выявляемые N-граммы сравнивались с таблицей (в практической реализации использовалась структура, известная как хэш-таблица) N-грамм, построенной на предварительном этапе. Если находилось соответствие, то весовой коэффициент языка, которому принадлежала табличная N-грамма увеличивался по следующему правилу.

Если найденная N-грамма являлась уникальной по отношению ко всем остальным языкам, то весовой коэффициент увеличивался на W_U. В текущей реализации W_U=10. Если N-грамма просто являлась часто встречающейся в соответствующем языке, то весовой коэффициент увеличивался на α_N. В текущей реализации было принято, что α_N=N. Данный выбор весовых коэффициентов является эмпирическим, и на практике он показал себя надежно.

Было произведено исследование 3-х вариантов алгоритма оценки текстов:

1. Учитываются все найденные сочетания – как уникальные, так и неуникальные.
2. Учитываются все уникальные сочетания. Неуникальные учитываются только в том случае, если в других языках такие сочетания не прошли в часто встречающиеся (т.е., остались за пределами, которые очерчиваются параметром N_{freq}).
3. Учитываются только уникальные сочетания.

Эксперименты показали, что результаты 1-го алгоритма, учитывающего все, как уникальные, так и часто встречающиеся сочетания, самые неубедительные.

Лучше всего проявляет себя алгоритм 3, который работает только с уникальными сочетаниями. Однако у него имеется один изъян – из-за очень большой требовательности к используемым данным, на коротких последовательностях он может просто не успеть выбрать главный язык. Т.е., если встречается предложение из двух-трех слов, то очень часто все языки имеют весовой коэффициент 0. Кроме того, исследование показало, что имеются языки, имеющие мало уникальных сочетаний (например, испанский), что также затрудняет его применение.

В связи с этим наиболее надежной видится следующая схема работы алгоритма распознавания языка – на первом этапе текст обрабатывается по 3-му алгоритму. Если отношение коэффициентов 1-го и 2-го по значимости языков больше, чем 2, а абсолютные значения весовых коэффициентов выше некоторого порога (например, 30-50), то на этом и останавливаемся. В противном случае, обрабатываем текст повторно, используя алгоритм 2. В этом случае, для принятия решения о принадлежности текста определенному языку, достаточно будет только превышение весовым коэффициентом определенного порога.

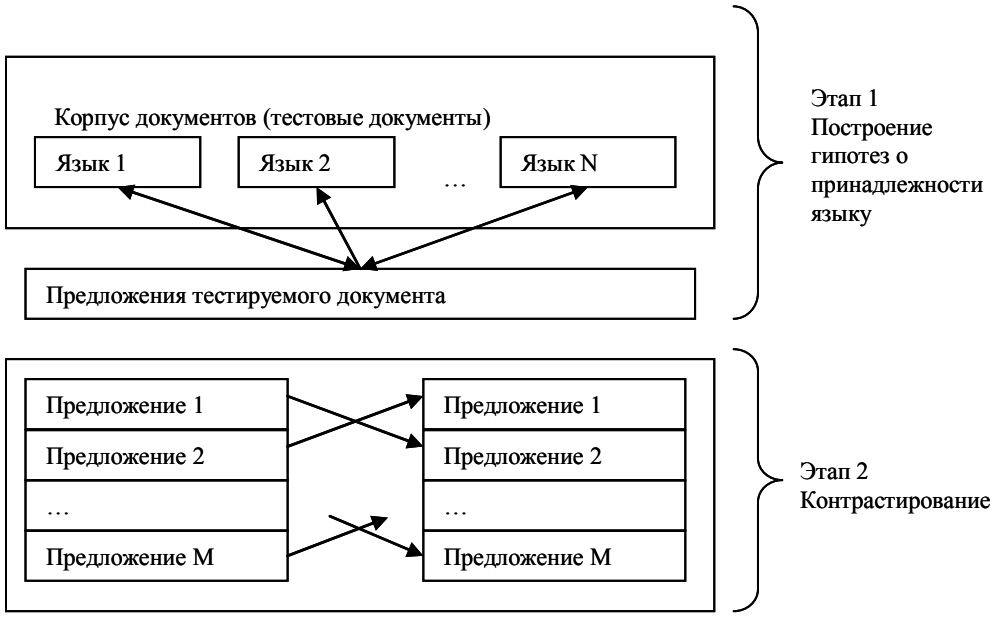

Рис. 2. Два прохода алгоритма

Работа с многоязычными текстами

В современном мире очень большое количество документов является многоязычными. Определить за один шаг языки данных документов представляется достаточно сложным, поскольку реально имеющийся в документе язык, в большом количестве случаев, будет "прятаться" за языками, родственными основному языку документа. Чтобы преодолеть данную проблему, рекомендуется производить определение языка для каждого предложения отдельно. Поскольку предложения представляют собой некие законченные мысли, то, за исключением некоторых малораспространенных случаев, их можно считать самыми крупными моноязыковыми единицами текста. И в тех случаях, когда в предложение на каком-то языке попадают отдельные слова на другом, их вполне можно считать инвариантами для основного языка.

Несмотря на достаточно надежную работу алгоритма, его оригинальных характеристик часто бывает недостаточно в случае, когда обрабатывается текст, содержащий части на похожих языках, например, на русском и украинском. Для того, чтобы сделать его работу надежной, алгоритм становится двух проходным (смотри рис. 2) – первый проход выполняется как описано выше, но каждое предложение помечается самым вероятным языком только в случае уверенного распознавания (либо детектируется только один язык с весовым коэффициентом не ниже определенного порога, либо самый вероятный язык опережает следующего претендента не менее, чем в 2 раза). Оставшиеся же предложения подвергаются обработке на втором проходе – так называемой процедуре контрастирования. Она заключается в том, что языком предложения становится неуверенно распознанный язык, если окружающие его предложения уже помечены, как предложения на этом же языке.

Такое контрастирование позволило практически избежать ошибок при работе с многоязычными текстами.

Результаты работы

Результатом работы над алгоритмом распознавания языка стали две библиотеки (одна написана на Delphi для платформы Win32, другая – на C# для платформы .NET Framework). Обе библиотеки распознают следующие языки: английский, немецкий, русский, украинский, французский, польский, испанский, греческий, арабский, иврит, армянский, грузинский, белорусский, болгарский, итальянский, португальский, ирландский. Данный список ограничен только исходным корпусом эталонных документов, и может быть легко расширен.

Обе реализации работают в десятки раз быстрее методов, основанных на проверке словарей морфологических машин, и при этом занимают в памяти во много раз меньше места (точнее сказать сложно, поскольку автор не смог найти морфологическую машину, работающую именно с таким набором языков). Это позволяет рекомендовать данные, либо подобные библиотеки для предварительного определения, какой набор языков загружать в морфологическую машину, позволяя ей работать быстрее, и занимать меньше оперативной памяти.

Выводы

Работа над алгоритмом идентификации языка по n-граммам показала, что для точной работы вполне достаточно N-грамм, с длиной до 4-х символов включительно (квадрограммы). Это дает возможность очень эффективно реализовать алгоритм на современных процессорах, как правило, имеющих регистры шириной 32 и 64 бита. Второй проход алгоритма обеспечил надежную работу с короткими текстовыми фрагментами – предложениями, что дало возможность работать с многоязыковыми текстами. Применение подстройки весовых коэффициентов языка вместо последовательного вычисления расстояния к каждому из тестовых текстов, позволило использовать на этом этапе такую структуру, как хэш-таблицу, содержащую словарь сразу для всех языков – соответственно, скорость определения языка перестала зависеть от количества определяемых языков.

Литература

1. Cavnar, W. B. and J. M. Trenkle, “N-Gram-Based Text Categorization” In Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, UNLV Publications/Reprographics, pp. 161-175, 11-13 April 1994