DDC - ПРОГРАММА ПОИСКА ПО МОРФОЛОГИЧЕСКИ И СИНТАКСИЧЕСКИ РАЗМЕЧЕННОМУ МАССИВУ

А. В. Сокирко
Берлинская и Бранденбургская Академия наук
sokirko@yandex.ru

Обоснование

Главным предназначением системы DDC является поиск определенных слов и словосочетаний в корпусе, то есть, по своей сути DDC – это конкорданс - программа поиска контекстов слова или словосочетания в некотором корпусе. Необходимо объяснить, почему мы стали разрабатывать собственную систему конкорданса. Существующие системы, которые можно использовать как конкорданс, можно поделить на два больших класса:

1.Коммерческие системы информационного поиска (Oracle Text, SQL Server Full-Text Indexing AltaVista, Yandex и т.д.)

2. Академические разработки для лингвистического поиска (CQP, BNCweb и т.д.).

С нашей точки зрения, по отношению к задаче лингвистического поиска первый класс систем обладает следующими достоинствами:

1. Обычно нет существенных ограничений на размер массива.
2. Обычно обрабатывается большое количество форматов документов входного массива.
3. Обычно система безопасна и устойчива при работе в параллельном режиме в Интернете.

С другой стороны, есть общие недостатки:

1. Очень трудно, если вообще возможно, расширять язык запросов, добавляя туда поиск по морфологическим и синтаксическим признакам.
2. Результатом поиска является целый документ, тогда как лингвиста обычно интересует одно предложение.
3. Крайне высокая цена на данные системы.

К достоинствам академических систем можно отнести следующее:

1. Изначальная направленность на лингвиста: поиск по предложениям или синтаксическим группам.
2. Часто предоставляется полный исходный код программы.
3. Низкая цена или даже отсутствие цены.

Недостатки же, по нашему мнению, следующие:

1. Ограничение на функциональность: невозможность добавить новый файл в корпус без полного переиндексирования или ограничение на размер или формат файлов.
2. Обычно система работает либо на Linux, либо на Windows.
3. Отсутствие документации и часто неотлаженность программы.

Учитывая все эти соображения, мы взялись за разработку собственной системы поиска.

Лингвистические процессоры

DDC использует во время индексации и поиска следующие лингвистические процессоры системы Диалинг(www.aot.ru):

1. Графематический процессор;
2. Морфологический процессор;
3. Поверхностно-синтаксический процессор(Shallow syntax).

Графематический процессор делит входной текст (html или plain формат) на слова, предложения и абзацы. Морфологический процессор для каждого слова создает набор морфологических интерпретаций, где морфологическая интерпретация - это пара <P,G>, где P – часть речи, а G – набор граммем. Сейчас в системе Диалинг есть три морфологических словаря – русский, английский и немецкий, соответственно входной корпус может быть английским, русским или немецким. Базой для немецкой морфологии послужила система Morphy(http://www-psycho.uni-paderborn.de/lezius/). Поверхностно-синтаксический процессор строит для предложения проективный набор клауз(простых предложений) и проективный набор синтаксических групп внутри этих клауз. Группы и клаузы определяются двумя параметрами: координатами в предложении и типом, где тип – это некоторая строковая константа.

DDC может получать морфологические интерпретации из морфологического модуля или из синтаксиса. Поскольку синтаксис снимает часть омонимии (примерно в половине случаев), поиск по морфологическим интерпретациям, взятым из синтаксиса, становится намного полезней для лингвиста.

Индексация

Один корпус для DDC системы состоит из трех частей:

1. файл перечня всех входных текстов корпуса;

2. файл опций индексирования и поиска;

3. входные тексты, каждый из которых лежит в отдельном файле.

Упрощая, можно сказать, что существуют два типа индексов, которые надо построить:

1. Индексы для предложений и абзацев, по которым можно по номеру слова в массиве получить границы предложения, которое это слово содержит.

2. Индексы для слов, или более обобщенно, индексируемых элементов, с помощью которых можно перейти от слова ко всем его вхождениям в корпусе.

Индекс первого типа строится довольно быстро и легко, поскольку он имеет небольшой размер относительно индекса второго типа.

Индексы второго типа существенным образом зависят от типа индексируемых элементов. Текущая версия программы способна обрабатывать следующие типы индексируемых элементов:

1. Строка (входная словоформа, лемма);
2. Морфологическая интерпретация;
3. Синтаксическая группа или клауза.
4. Номер входа в некоторый тезаурус.

Один индекс второго типа состоит из упорядоченного набора уникальных индексируемых элементов, причем от каждого элемента идет ссылка на перечень всех вхождений данного элемента в корпусе. Например:

МАМА -> 1, 199, 1001, 99999...

МАМЕ -> 111, 991, 2101.

МАМУ -> 11, 99, 1101

Одно вхождение элемента – это четырехбайтовое число, которое является номером этого элемента во входном корпусе, считая с самого начала корпуса. Отсюда уже следует, что один корпус для DDC не может содержать более 2^{32} слов. Это ограничение, будучи совершенно неприемлемым для информационно-поисковых систем, не является, по нашему мнению, существенным для лингвистически ориентированного поиска.

Программа индексации работает в ограниченной памяти, это означает, что она временами сохраняет данные на диск, освобождая таким образом оперативную память.

Ниже будут приведены ресурсные параметры процесса индексирования[1]:

Название корпуса	Язык	Число слов	Размер корпуса	Время	Опер. память
DWDS- corpus1	немецкий	11 млн.	85 МБ	9 минут	40 МБ
DWDS- corpus2	немецкий	30 млн.	160 МБ	20 минут.	60 МБ
Moshkov-subset1	русский	15 млн	100 МБ	13 минут	60 МБ
Moshkov-subset2	русский	54 млн.	350 МБ	55 минут	80 МБ

Скорость индексирования зависит от опций индексирования и, прежде всего, от языка корпуса. Для всех вышеперечисленных тестовых массивов строился только индекс словоформ и индекс морфологических интерпретаций.

Размер полученного индекса для тестовых массивов примерно в 1,5 раза больше самого массива. В общем случае размер индекса зависит от настроек, в частности, можно задать параметр ArchiveOccurrences, который позволит сократить размер индексов на 35 процентов, скорость обработки запросов с архивированным индексом уменьшится на 20 процентов.

Максимальный размер проиндексированного корпуса на сегодняшний день составляет 300 млн. слов (немецкий язык).

Язык запросов

Текущая версия языка запросов DDC поддерживает следующие конструкции:

Тип запроса	Назначение	Пример	Результат
*слово**	описание слова	до*	все предложения, в которых есть слово, имеющее префикс "до"
**слово*	описание слова	*до	все предложения, в которых есть слово, которое заканчивается на постфикс "до"
[M] (где, M – морфологическая интерпретация)	описание слова	[С ед,тв]	все существительные в единственном числе и творительном падеже
@слово	описание слова	@дом	все предложения, в которых есть словоформа "дом" (точное соответствие)
"X1 X2 … XN"	последова-тельность слов	"мой новый дом"	все предложения, в которых есть "мой новый дом"
		"дом [Г]"	все предложения, в которых есть "дом", за которым сразу идет какой-нибудь глагол
Q1 && Q2	конъюнкция описаний слов	дом && [С ед]	все предложения, в которых есть "дом" и существительное в единственном числе
Q1 || Q2	дизъюнкция описаний слов	[Г 2л] || [С мн]	все предложения, в которых есть глагол во втором лице или существительное во множественном числе
near(Q1;Q2;n)	два слова	NEAR (дом; [С]; 2)	все предложения, в которых

[1] Все расчеты выполнены на P4 1,5 GHz, 256 MB ОЗУ, Linux.

	рядом друг с другом 0<= n <= 10		есть "дом" и какое-нибудь существительное, и между ними стоит не больше двух слов.
"X1 #D1 X2 #D2 .. XN"	последова-тельность слов с максимальными дистанциями	"мой #1 дом"	все предложения, в которых есть "мой", за которым следует "дом", и между ним не больше одного слова
_ГРУППА	синтаксическая группа или клауза	_ПРЯМ_ДОП	все предложения, в которых есть глагольная группа с прямым дополнением.

Вообще говоря, запрос DDC может преследовать две разные цели. Во-первых, пользователь может просить систему выдать ему число предложений, удовлетворяющих данному запросу, это т.н. статистические запросы. Во-вторых, пользователь может хотеть получить только примеры использования данной конструкции. Мы называем такие запросы запросами контекстов. Время обработки сложных статистических запросов должно линейно зависеть от размера массива. Что значит "сложный" - зависит от конкретной поисковой системы. Конечно, выдача числа вхождений данного слова в корпусе обычно происходит за константное время, поскольку эта информация включается в индекс. Но, например, для получения числа предложений, в которые должно входить несколько слов из запроса, уже требуется получить пересечение наборов вхождений, которое должно быть выполнено за линейное от размера корпуса время. Запросы же контекстов обычно работают за константное время, поскольку пользователь всегда требует какое-то ограниченное число примеров (20 или 30). Таким образом, запросы контекстов обычно работают на небольшой части индекса (обычно – это начальная часть), а статистические запросы – на всем индексе, поэтому статистические запросы обрабатываются медленней.

Различие между статистическими запросами и запросами контекстов используется некоторыми поисковыми системами. Например, Google в случае, когда число Интернет-страниц по данному запросу превышает некоторый порог, выдает уже приблизительное число найденных страниц, что, мы полагаем, позволяет сильно убыстрить поиск.

Для обоих типов запросов построение результирующего множества осуществляется обходом в глубину дерева синтаксического разбора запроса. Это означает, что, например, для запроса (A || B) && C сначала вычисляется объединение (A && B), а потом уже главное пересечение. Принципиальная последовательность, а не параллельность вычислений пересечений и объединений приводит иногда к очень неэффективной работе алгоритма. Если в формуле (A || B) && C мощность объединения (A || B) очень велика, а мощность C, наоборот, низка, вычисление сначала полного объединения (A || B) не является наиболее эффективной стратегией. Чтобы до некоторой степени преодолеть такого рода неэффективность, весь корпус текстов поделен на внутренние подкорпуса. Вычисление любой формулы сначала происходит отдельно на каждом подкорпусе, а потом все результаты объединяются. С помощью разделения целого корпуса на подкорпуса мы добиваемся того, что запросы контекстов (нестатистические) начинают работать за время, зависящее от длины одного подкорпуса, а не от длины всего корпуса. Это происходит, потому что порция контекстов, которую требует пользователь, обычно может быть найдена в одном подкорпусе.

Разделение на подкорпуса также позволяет осуществлять поиск в константной оперативной памяти. Это означает, что программа всегда может ограничивать использование оперативной памяти в пределах 100 Мб. Однако, например, для системы Windows формально небольшое использование оперативной памяти не является главным критерием. Для Windows главным является размер самого индекса. Если размер индекса существенно превышает размер оперативной памяти, тогда статистические запросы с большими результирующими множествами начинают выполняться в два раза медленней.

Ниже приводится время(в секундах) выполнения запросов:

Запрос	Тип запроса	Moshkov1 Русск. 15 млн	Moshkov2 Русск. 54 млн.
Мама	(нестат.)	0.05	0.05
Мама	(стат.)	0.007	0.015
ба*	(нестат.)	0,06	0.07
ба*	(стат.)	0.1	0.3
“ [П] [С] [Г] “	(нестат.)	1,1	1,1
“ [П] [С] [Г]“[2]	(стат.)	2,5	14

[2] Последовательность из трех слов, первое слово – существительное, второе - прилагательное, третье – глагол.

В данной таблице показано время исполнения запроса для статистических и нестатистических запросов. Например, статистический запрос "ба*" для русского массива в 15 млн. слов обрабатывается за 0,1 секунду.

Программная функциональность

Система DDC написана на C++. Компилируется под GCC и Microsoft C++. Система работает в двух вариантах:

1. однопользовательский режим;
2. распределенный режим.

В однопользовательском режиме доступны следующие программы:

1. ConcordIndex – программа индексации корпуса.
2. ConcordSimple – программа выполнения одного запроса, заданного в командной строке.
3. ConcordAdd – программа, которая сливает два проиндексированных корпуса в один, объединяя индексы.
4. Concordance – программа с графическим интерфейсом (только Windows), которая позволяет интерактивно индексировать и задавать запросы.

В распределенном режиме доступны следующие программы:

1. ConcordDaemon – демон под Unix, способный по TCP/IP отвечать на запросы по массиву.
2. Search – CGI-программа, которая получает запросы от HTML-формы и передает их ConcordDaemon.

Одна из опций распределенного режима заключается в том, что разные демоны могут быть запущены на разных машинах. Каждый демон работает со своим корпусом, и существует еще центральный демон, который опрашивает всех остальных демонов и объединяет результаты. В распределенной схеме проблема подсчета скорости обработки запроса зависит от числа компьютеров

Практические применения

Ниже мы опишем несколько практических применений DDC:

1. Прежде всего, система может быть использована как справочная система по русскому языку. Мы проиндексировали корпус русской литературы и обеспечили возможность поиска по этому корпусу из Интернета (http://www.aot.ru/search.html). Набрав запрос, пользователь может получить примеры употребления заданной конструкции в корпусе.
2. Система может быть использована для лингвистических проектов, использующих статистические методы анализа текста. В таком случае лингвистам нужно получить частоты некоторых заданных языковых конструкций, чтобы потом сравнивать эти частоты между собой. Такие методы, например, развиваются в проекте DWDS[3], в котором сейчас участвует автор.
3. Система может быть использована как простая поисковая машина, например, с помощью нее был организован поиск по www.aot.ru

Благодарности

Автор благодарит Берлинскую Академию Наук (Berlin-Brandenburgische Akademie der Wissenschaften) за поддержку этого проекта. Автор благодарен также Андрею Путрину за развитие графической оболочки DDC и участникам проекта www.aot.ru за предоставленные лингвистические модули. Автору очень приятно отметить, что система DDC распространяется с лицензией LGPL, любой может использовать ее бесплатно, скачав исходники с сайта www.aot.ru.

[3] DWDS - Das digitale Woerterbuch der deutschen Sprache des 20. Jahrhunderts, www.dwds.de