Некоторые особенности формирования электронного корпуса текстов с синтаксической разметкой

© А.А. Рогов, Ю.В. Сидоров, А. В. Седов, Г.Б. Гурин, А.А. Котов, М.Ю. Некрасов

Петрозаводский государственный университет
rogov@psu.karelia.ru

Аннотация

В данной статье описывается система, созданная авторами для проведения синтаксической разметки текстов. Система создается и совершенствуется в рамках гранта РГНФ №08-04-12105в (Рук. Рогов А.А.). Эта работа является продолжением разработок по созданию грамматически размеченного корпуса публицистических текстов XIX века [1-3].

1 Выбор синтаксического аннотирования

Существующие немногочисленные корпусы со встроенной синтаксической разметкой опираются либо на общепринятые классификации традиционной («школьной») грамматики (Хельсинкский аннотированный корпус русских текстов ХАНКО; http://www.slav.helsinki.fi/hanco/index.html), либо на доступные узкому кругу специалистов и требующие детального предварительного знакомства классификации, например разметка в терминах деревьев зависимостей и синтаксических отношений, принятых в теории «Смысл-Текст», как в Национальном корпусе русского языка (http://www.ruscorpora.ru). Эти пути аннотирования в целом решают разные задачи и имеют свои достоинства и недостатки. Опора на школьную классификацию существенно облегчает работу с корпусом и расширяет круг потенциальных пользователей до всех, кто получил школьное образование, однако пользователю придется смириться с недостатками традиционного подхода: нечеткостью понятий и, соответственно, разметки, множественностью и некоторой произвольностью синтаксического описания. Такой корпус скорее является удобным источником иллюстраций для преподавателей русского языка, переводчиков, он полезен для редакторов и самого широкого круга заинтересованных лиц. Классификации, принятые в рамках той или иной научной школы, заведомо осложняют процедуру овладения ресурсом, так как требуют тщательного знакомства с принципами разметки и единицами классификации, однако такое аннотирование в большей степени свободно от противоречий традиционного анализа. В создаваемом корпусе в основу синтаксической разметки положена идея структурной схемы в понимании Н. Ю. Шведовой и ее последователей, впервые отчетливо заявленная в «Грамматике современного русского литературного языка» [4], позднее наиболее полно отраженная и развитая в «Русской грамматике» [5]. С одной стороны, это обеспечивает достаточно широкий охват пользователей, так как знакомство с классификацией синтаксических образцов в терминах структурных схем предполагается стандартными вузовскими курсами синтаксиса на филологических факультетах, эти классификации описываются в целом ряде распространенных учебников, с другой стороны, анализ формы предложения позволяет объективировать и упорядочить, насколько это возможно, систему разметки. Создание полного списка структурных схем простого предложения (в корпусе размечаются предикативные клаузы) – отдельная научная проблема, не имеющая пока своего окончательного решения. На данный момент мы можем говорить о том, что в научном обороте существуют как минимум три списка структурных схем – различные как количественно, так и качественно: 1) список схем «Русской грамматики» (1980); 2) список «минимальных схем» В. А. Белошапковой; 3) список схем О. А. Крыловой и Е. Н. Ширяева [6]. Последние на основе достаточно убедительного теоретического обоснования значительно переработали и дополнили исходный список свободных структурных схем «Русской грамматики». Именно эта классификация является на сегодняшний день наиболее полной и точной, и с небольшими изменениями и дополнениями была взята за основу разметки настоящего корпуса. Этот выбор объясняется двумя причинами: во-первых, использование структурных схем для синтаксической разметки в корпусе имеет свою специфику, во-вторых, ситуация изучения вопроса

Труды 11[ая] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2009, Петрозаводск, Россия, 2009.

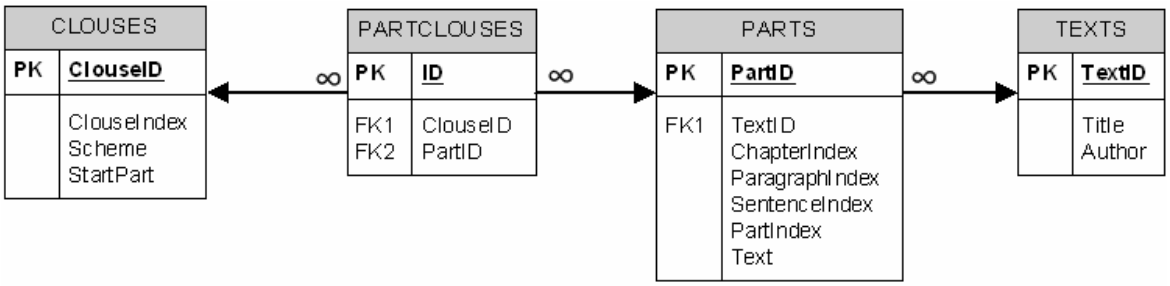

Рис. 1. Структура базы данных предназначенной для хранения синтаксической разметки

такова, что ни один из существующих списков структурных схем нельзя признать окончательно полным. Таким образом, на выходе мы получили наиболее полный и сбалансированный список структурных схем простого предложения, который будет использован для синтаксической разметки текстов.

2 Используемые структуры данных

Основой информационной системы синтаксической атрибуции является база данных, в которой хранится информация о синтаксических разборах текстов. Следовательно, первоначальной задачей была разработка структур данных для хранения информации о разборе текста. Основным структурным элементом синтаксической разметки, подвергаемым разбору, является клауза – минимальная предикативная единица, которая может выступать в качестве самостоятельного простого предложения или составной части сложного предложения. В составе сложной синтаксической структуры клауза может быть разбита одной или несколькими другими клаузами. Например, предложение «Вася пошел в бассейн, который открылся на днях, и плавал там до вечера», очевидно, делится на три части «Вася пошел в бассейн», «который открылся на днях», «плавал там до вечера», однако оно состоит из двух клауз: «Вася пошел в бассейн и плавал там до вечера», «который открылся на днях». То есть, как видно, 2 части предложения объединяются в одну клаузу. Для осуществления этого мы решили разделять понятия клаузы и части предложения. Приняв во внимание все написанное, структура текста у нас приобретает следующий вид: текст разбивается на главы, главы – на абзацы, абзацы – на предложения, предложения на клаузы, клаузы состоят из частей. Стоит отметить также, что одна часть предложения может принадлежать нескольким клаузам. Это возможно когда предложение содержит несколько однородных сказуемых, потому что сами схемы, прежде всего, различаются именно по структуре сказуемых, входящих в предложение. Например, предложение «Чудище обло, озорно, огромно, стозевно и лаяй» состоит из 5 клауз, причем слово «чудище» участвует во всех пяти. Предлагается структура данных, для которой это предложение разбивается на 6 частей:

P1=”Чудище”
P2=”обло”
P3=”озорно”
P4=”огромно”
P5=”стозевно”
P6=”лаяй”
Из этих частей составляются 5 клауз:
CL1={P1, P2};
CL2={P1, P3};
CL3={P1, P4};
CL4={P1, P5};
CL5={P1, P6}.

Каждая из 38 синтаксических схем кодируется числом от 1 до 38, поэтому при разборе каждой клаузе ставится в соответствие номер соответствующей схемы. Еще стоит отметить, что для клауз требуется хранение номера части, с которой эта клауза начинается.

На основе разработанной синтаксической разметки текстов были сформированы структуры таблиц и триггеров базы данных. Общая схема представлена на Рис.1. Для ее построения использовалась СУБД Interbase 6.0, и к настоящему времени она содержит 4 таблицы, и занимает объем 50 Мб.

Для каждой части текста определяется ее координаты: номер главы, абзаца, предложения, клаузы и ее содержание. Для каждой клаузы определяется ее номер, разбор этой клаузы и ее начальная часть. Части и клаузы соединяются связью «многие ко многим», так как одна клауза может состоять из нескольких частей, и, вместе с тем, одна часть может входить в состав нескольких клауз.

3 Программа синтаксического разбора

На вход описываемой программы подается текстовый файл формата .txt в кодировке unicode. На его основе создается рабочий проект, содержащий несколько файлов, с информацией о разбиении текста на структурные части, о синтаксическом разборе и прочие служебные файлы. Программа при открытии текста автоматически разбивает текст на структурные компоненты: главы, абзацы и предложения. Признаком новой главы является знак параграфа, расположенный первым на строке, признаком нового абзаца является табуляция, символами конца предложения являются точка, восклицательный и

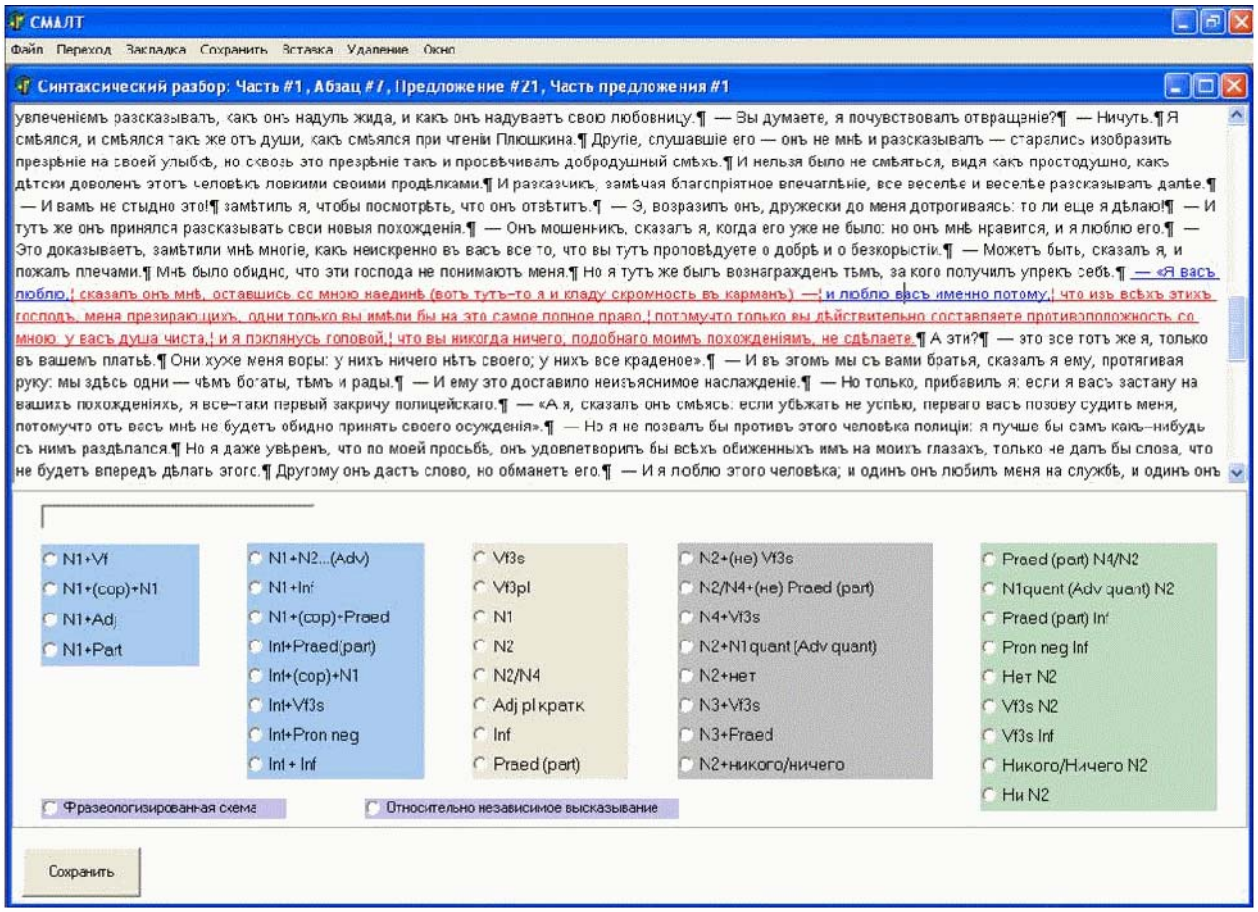

Рис. 2. Интерфейс программы синтаксического разбора

вопросительный знаки и т.д. На этом этапе могут возникать некоторые проблемы с автоматическим разбиением текста:

1. Существуют сложные знаки препинания, служащие концами предложения: «…», «?..», «!..» и пр. При автоматическом разборе каждый из этих знаков будет разделен на составные и получится, что в одном месте заканчиваются сразу 3 предложения. Эта проблема решена путем введения специальных соответствующих шаблонов.
2. Более сложная проблема связана с тем, что точка (как и остальные общепринятые признаки концов предложения) не всегда свидетельствуют о завершении предложения. Зачастую точка является признаком сокращения («…г. Волошин…»). Иногда вышеназванные знаки ставятся внутрь прямой речи, не являясь при этом признаками завершения предложения. Эту проблему можно решить путем анализа следующего слова: если оно начинается с прописной буквы, то можно сделать вывод о том, что рассматриваемый знак препинания не является концом предложения. Но если следующее слово начинается с заглавной буквы, это не гарантирует, что знак являлся концом предложения.
3. Стандартным признаком конца абзаца являются символы перевода строки. Однако перевод строки может использоваться и внутри текста, например в стихотворных отрывках. Выходом для большинства таких случаев будет проверка на то, заканчивалось ли предложение прямо перед переводом строки.

На этапе синтаксического разбора, который начинается сразу же за этапом разбиения текста, пользователь, если разбиение текста на предложения содержит ошибки и неправильно расставленные концы предложений/абзацев, может исправить вручную и имеет возможность редактировать разметку текста и разбивать предложения на части. Если клауза состоит из нескольких частей, пользователь может объединять их. В интерфейсе программы пользователю выводится пять или менее абзацев (текущий абзац, два предыдущих и два последующих), в которых возможна только покомпонентная навигация, то есть переходы только по клаузам. Всегда отдельно выделяется текущая клауза, которая разбирается в данный момент. Пользователь выбирает разбор (синтаксическую конструкцию) клаузы при помощи радиокнопок (т.н. «radio button»), путем выбора одной из схем. При сохранении разбора программа автоматически переходит на следующую клаузу. При помощи меню или горячих комбинаций клавиш пользователь может переходить на соседние клаузы/предложения/абзацы/главы. Кроме того, по двойному щелчку на участке текста, пользователь может переходить на выбранную клаузу. По

просьбам специалистов, работающих с программой, в ней реализованы различные варианты поиска: поиск следующей или предыдущей клаузы, кроме того, точный переход при выборе главы, абзаца, предложения и клаузы. Отметим, что в программе есть функция закладки: пользователь может поставить закладку на ту клаузу, которую разбирает в данный момент, выйти из программы, и при дальнейшей работе просто перейти по закладке на ту же часть текста. Фрагмент работы программы представлен на Рис.2.

Для работы программы синтаксической разметки требуется ОС Windows, объем оперативной памяти: больше 128 Мб, место на жестком диске: 10 Мб для программы, в среднем 5 Мб на хранение каждого рабочего проекта (в зависимости от объема текста). Было произведено наполнение базы данных текстами суммарным объемом более 36000 клауз из текстов, принадлежащие Ф.М. Достоевскому и его современникам, из журналов «Время» и «Эпоха», тексты В.И.Даля и ряд различных публицистических текстов XIX века. На данный момент размечено порядка 60 текстов.

4 Оптимизация работы программы

С целью оптимизации работы программы синтаксической атрибуции было решено провести ряд исследований, направленный на ускорение процедуры атрибуции текстов. Естественным вариантом ускорения работы видится автоматизация начального выбора синтаксической атрибуции клауз. Для ускорения процесса разметки проведен анализ различных вариантов его автоматизации на основании различных статистических методов. Критерием оптимальности является наименьшее время работы пользователя с программой при атрибуции текста. Для автоматической атрибуции было решено исследовать следующие эмпирические подходы:

1. Простейший статистический метод. Синтаксическая схема по умолчанию выбирается как наиболее часто используемая конструкция.
2. Анализ разбора клаузы, предшествующей текущей.
3. Анализ разбора двух предшествующих клауз.

Заметим, что все эти исследования можно провести, основываясь только на информации из базы данных уже имеющихся разборов. Действительно, взяв за основу клаузу с ее координатами в тексте, можно получить ее действительный разбор и разбор, прогнозируемый по каждому из вышеописанных подходов. Тут же надо оговориться, что внедрение слишком сложных интеллектуальных методов замедляет работу программы, и то преимущество во времени, которое может быть получено за счет автоматического выбора клауз, может быть нивелировано замедлением функционирования самой программы.

Существуют и другие подходы к оптимизации работы с программой. Можно, например, расставлять структурные схемы на панель выбора в соответствии с частотой их использования: самые часто используемые схемы поместить слева и, по мере уменьшения использования, располагать другие схемы все правее. Однако, как подтвердили специалисты работающие с программой, побочный эффект данного метода намного хуже, так как они привыкли к определенному расположению кнопок с выбором схем, как правило нажимают их уже не глядя, и постоянная перестановка этих кнопок сбивает специалистов и ощутимо замедляет время работы с программой. Это так называемый “эффект qwerty”.

Статистический метод

На основе разобранных текстов мы выбираем наиболее часто встречаемую схему, то есть схему, при помощи которой разбирается наибольшее количество клауз.

Результаты исследования:

Проведя анализ, мы выделили 7 наиболее употребляемых структурных схем. Они приведены в следующей таблице:

Схема	Количество	Частота
$N_1 + V_f$	21195	58,5 %
N_1 + Adj	3021	8,3 %
N_1 + (cop) + N_1	2344	6,5 %
N_1 + Part	1474	4 %
$Praed_{(part)}$ Inf	1160	3,2 %
N_1	1087	3 %
$N_1 + N_2 \ldots_{(Adv)}$	1038	2,9 %

Табл. 1 Статистика частотности употребления структурных схем

Общее количество проанализированных клауз равно 36224. Как видно из результатов, при помощи схемы $N_1 + V_f$ разбирается больше половины клауз текста. При таком подавляющем преимуществе можно говорить о целесообразности выбора этой схемы как схемы по умолчанию.

Анализ разбора предыдущей схемы

Суть метода в следующем: мы пытаемся предположить, что существует связь между разборами двух рядом стоящих схем. Тогда мы предлагаем определять начальный разбор текущей клаузы на основе разбора предыдущей. Для этого мы анализируем для всех структурных схем разборы клауз, следующих за клаузами, разобранными этими схемами, и выбираем самый распространенный из них. В процессе атрибуции, при выборе структурной схемы для текущей клаузы, следующая клауза будет атрибутироваться автоматически.

Формально:

Есть структурная схема St;

Получаем набор клауз из текстов, атрибутированных ею $Cl = (cl_1, .. cl_n)$;

Получаем набор клауз, следующих за Cl : $Clp = (clp_1, .. clp_n)$;

Группируем эти клаузы в зависимости от их разбора: $Gr_{ij} = \{gr_{ij}\}$, i – номер группы, j – номер клаузы в группе;

Для каждой группы записываем соответствующую ей структурную схему: Stpi;

Считаем количество клауз в каждой группе: $N_i = \{n_1, n_2, .., n_k\}$;

Ищем номер наибольшей группы $i^* = \text{argmax}(n_1, n_2, .., n_k)$;

Тогда предполагаемый разбор клаузы, следующей за клаузой с разбором St, будет St_i^*.

Видно, что описанный метод является динамическим. То есть определять предполагаемый разбор клаузы необходимо непосредственно в процессе атрибуции, после того как разобрана предыдущая клауза, что приводит к замедлению разбора.

Можно рассматривать разборы не одной, а двух предыдущих клауз. Подробнее описывать этот метод не стоит по одной простой причине: как показало исследование этот и предыдущий методы абсолютно неэффективны. Учитывая, насколько одна структурная схема встречается чаще, чем все остальные, очевидно, что для любой схемы, следующей будет выбираться именно схема $N_1 + V_f$. То есть при применении методов анализа одной или двух предыдущих клауз, мы будем получать тот же результат, как и при применении статистического метода, только куда более алгоритмически сложно и ресурсоемко.

Из всего написанного можно сделать вывод, что наиболее эффективным будет применение статистического метода, выраженного в том, чтобы каждой схеме ставить начальным разбором структуру $N_1 + V_f$. Этот метод будет давать верное предсказание примерно в 58,5 % случаев.

5 Представление результатов

Полученный в результате выполнения проекта размеченный корпус представляется конечным пользователям при помощи Web-ресурса, его адрес http://smalt.karelia.ru. На нём представлена возможность формирования собственного подкорпуса из предложенных проанализированных текстов и получения статистических данных, например, таких как частота встречаемости синтаксической конструкции в выбранном подкорпусе. На Интернет ресурсе предоставлена возможность нахождения требуемой синтаксической конструкции в выбранном подкорпусе текстов, для чего пользователь вводит при помощи дополнительного окна список искомых конструкций, выбирает тексты, по которым будет производиться поиск, и нажимает на кнопку «найти». В результате открывается html страница, содержащая набор предложений, разбитый на группы по 10 предложений. Переход между группами осуществляется при помощи кнопок навигации, а также при помощи ввода номера группы и непосредственный переход к ней. По желанию, пользователь может ввести слово, которое должно содержаться в данной синтаксической конструкции. Имеется возможность получения контекста данного предложения, а также переход к целиком произведению, в котором содержалось данное предложение.

Заметим, что работа по созданию системы синтаксической разметки является продолжением работы, в рамках которой ранее была создана система грамматической разметки. Эта разметка проводилась по более традиционной схеме: каждому слову ставился в соответствие ряд его грамматических параметров, как-то, часть речи, число, род и пр. Потребовалась синхронизация информации из обоих проектов в одну базу данных. Как следствие, на web-ресурсе стал возможен поиск не только отдельно по синтаксическим или по грамматическим параметрам, но также и смешанный поиск по обеим разметкам. В связи с этим встала проблема модернизации базы данных для оптимизации смешанного поиска. Отметим, что при синхронизации грамматической и синтаксической разметок возникал ряд сложностей. При грамматическом разборе каждому слову тоже ставились в соответствие координаты в тексте (глава, абзац, предложение, слово). Однако, так как разметки проводились разными людьми в разное время, структурное разбиение текста на части могло быть различным. Как следствие, приходилось проводить сканирование всех текстов для выравнивания разбиения текста на части. Для организации смешанного поиска было решено сделать отдельную таблицу, дублирующую ряд информации из других таблиц, но оптимизированную именно под поиск одновременно слов с выбранными грамматическими параметрами, находящихся в клаузах с выбранной синтаксической структурой. Каждая запись этой таблицы содержит слово из словаря, его грамматические параметры и ряд полей, содержащих информацию о том, встречается ли это слово в заданной синтаксической схеме.

6 Оптимизация базы данных

Возникла задача разработки такой структуры базы данных, чтобы минимизировать время выполнения запроса пользователя. Рассмотрим возможные критерии оптимальности предоставления и хранения данных. Среди них можно выделить среднюю скорость предоставления информации, объём хранимой информации, полноту информации, представительность.

Оптимизация по времени поиска. Рассмотрим вопрос разработки оптимальной структуры БД, предназначенной непосредственно для поиска. Для этого необходимо учесть структурные особенности поиска (поиск вхождений слова, грамматической или синтаксической конструкции и выдача результата согласной некой мере близости). Рассмотрим первый критерий оптимальности:

минимизация среднего времени поиска. На сервер периодически поступают запросы, необходимо максимально уменьшить среднее время поиска необходимой информации. Обозначим, через n число обращений, t_i - время обработки запроса, тогда среднее время обработки всех запросов равно:

$$T_{ср} = \frac{\sum_{i=1}^{n} t_i}{n}$$

На нынешний момент в базе существует два вида разборов: грамматический и синтаксический. На их базе производятся различные виды поисков (по слову, по грамматическим признакам, по синтаксическим признакам, комбинированный поиск). Для каждого вида можно подобрать свою структуру. Пользователю в результате поиска предоставляется следующая информация:

- Получение одного слова, с грамматическими признаками, а также по возможности наборов синтаксических конструкций, в которые оно входит;
- Получение предложения целиком и характеристики каждого слова, в составе данного предложения.

Остановимся более подробно на определении среднего времени поиска. Поскольку слово может быть как в современной орфографии, так и в орфографии языка XIX века, поэтому приходится производить поиск в базе по двум полям (словоформа и современное написание). Это выполнимо за линейное время. Необходимо иметь доступ к предложению, содержащему данное слово, тогда таблица для поиска должна содержать ссылки каждого слова на предложения, где оно может быть найдено. Обозначим за t_{11} - время поиска слова, а за t_{12} - время поиска всех слов из того же предложения по таблице, состоящей из слов, их современных написании и «координат» слов в тексте. Тогда время обработки одного запроса t_i будет равно: $t_i = t_{11} + t_{12}$.

В случае необходимости получения грамматических и синтаксических параметров для каждого слова можно хранить характеристики каждого слова в той же таблице, добавив соответствующие поля:

а) Добавив на каждый грамматический признак соответствующий столбец, и один столбец на синтаксический параметр. При этом время поиска данных в БД изменится. Тогда время обработки одного запроса t_i будет равно времени поиска слова t_{21} + времени поиска всех предложений содержащих слово t_{22} + время, потраченное на поиск значения определённого параметра в соответствующей таблице t_{23}. Общее время поиска составит $t_i = t_{21} + t_{22} + t_{23}$;

б) Добавив 15 столбцов (по максимальному числу различных грамматических параметров + вид синтаксической конструкции). Тогда время обработки одного запроса t_i будет равно времени поиска слова t_{31} + время поиска всех предложений содержащих слово t_{32} + время на поиск признака в таблице t_{33} и расшифровку значения определённого параметра в соответствующей таблице $t_{рек.расш.}$. Общее время поиска: $t_i = t_{31} + t_{32} + t_{33} + t_{рек.расш}$.

Можно осуществлять хранение в одной таблице самого слова, начальной формы и грамматических параметров, а принадлежность к определённому предложению перенести в отдельную таблицу. Тогда время обработки одного запроса t_i будет складываться из времени потраченного на поиск слова t_{41} время на поиск предложений во второй таблице t_{42} плюс время на поиск слов оставшихся слов в первой таблице t_{43} плюс расшифровка параметров ($t_i = t_{41} + t_{42} + t_{43} + t_{рек.расш.}$). При данной структуре таблиц мы перестаём хранить избыточную информацию (если одно слово употребляется очень часто, то оно не копируется, а используется одна ссылка.) Преимущество - меньше записей в таблице - меньше время поиска.

В результате получаем несколько видов таблиц для различных видов поиска. Обозначим за a_{ij} - среднее время j-го вида поиска при использовании i-го вида таблиц. Тогда введём $\Sigma_i = \beta_1\alpha_{i1} + \beta_2\alpha_{i2} + \ldots + \beta_{10}\alpha_{i10}$ - функция, характеризующая среднее время, которое тратится на поиск пользователем необходимой информации. Коэффициенты β_j - определяют предпочтение определённого вида поиска. Изначально все коэффициенты β_j приравнивались к $k = \frac{1}{число\ видов\ поиска}$ (использование различных типов запросов является равновероятным). В результате выбирается структура данных обеспечивающая наименьшее время.

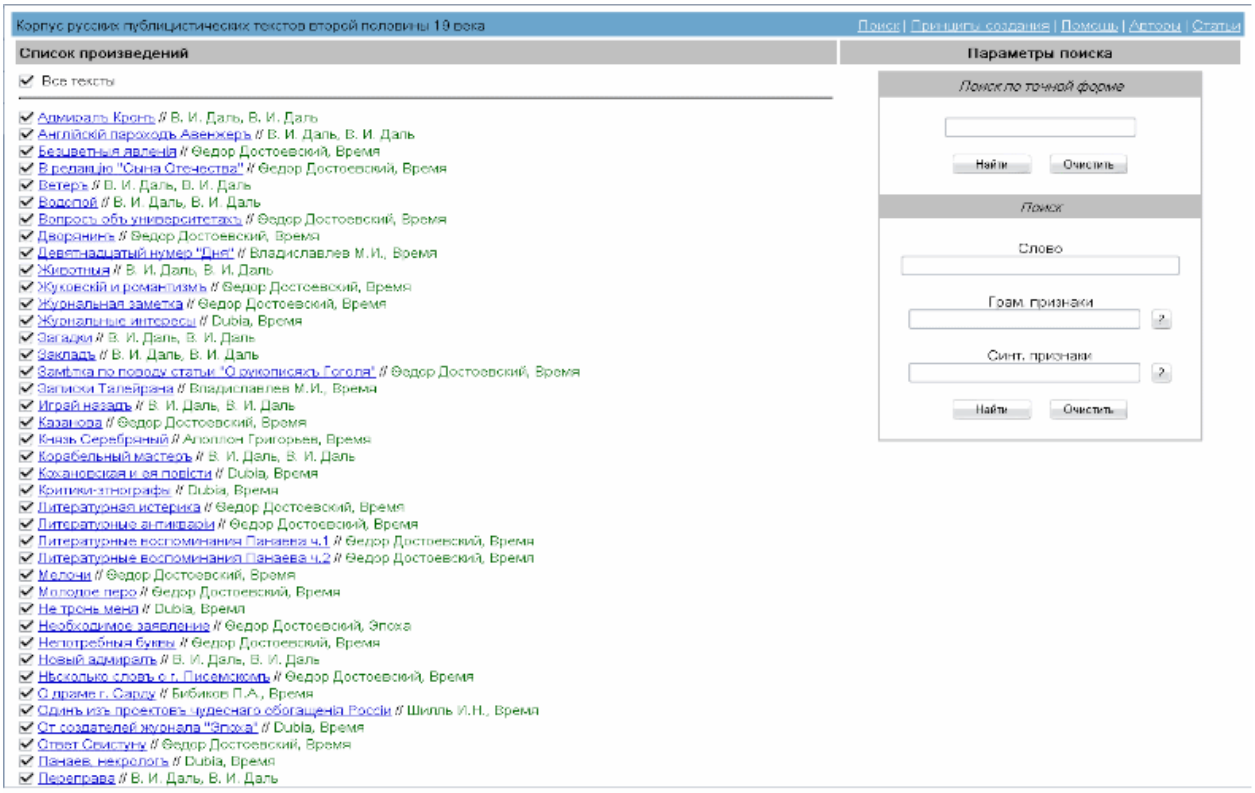

Рис. 3. Главная форма выбора параметров размеченного корпуса текстов.

Причина, по которой рассматривается среднее время поиска, состоит в следующем: предполагается, что изначально пользователя (специалиста) интересуют все слова одинаково. В дальнейшем, при сборе информации относительно "излюбленных" слов можно модернизировать БД с учетом статистики запросов по конкретным словам. То есть требуется построение и упорядочивание данных таким образом, чтобы наиболее часто искомые слова находились вначале.

Оптимизация по объёму. Время – не единственная характеристика для оптимизации БД. Важным параметром БД является её объём. Можно максимально сократить объём базы данных и при этом получить очень сложный поиск, и наоборот, создав очень быстрый поиск, мы сталкиваемся с проблемой не эффективного использования памяти. Поэтому для исследуемых баз данных рассматривали характеристику $Q = \Sigma_i * M$ - соотношение скорости получения информации к объёму БД. И на основании данной величины производился выбор подходящей структуры.

Вывод информации на экран пользователя. Обработанная информация будет предоставлена конечному пользователю при помощи веб-интерфейса. Необходимо ограничить число результатов предоставляемых пользователю одновременно. На наш взгляд, наиболее наглядным является предоставление информации на один экран, без скроллинга. Отсюда появляется задача определения среднего числа строк, занимаемых каждым из результатов поиска. На основании данного параметра можно будет определить необходимое число результатов, выдаваемое при поиске. Наиболее популярным разрешением экрана на данный момент является 1024x768 точек на дюйм.

Пусть A - число строк, помещающихся на экране, $\overline{c}$ - среднее число строк после одного запроса, тогда величина $S = \frac{A}{\overline{c}}$ - число результатов, которые должны быть представлены пользователю. Величина A -считается исходя из разрешения экрана, а $\overline{c}$ - число символов результатов запросов поделённое на число запросов и на количество символов, помещающихся в одной строке, округлённое вверх.

Полученные результаты. В результате проведённых исследований была выбрана база данных, содержащая слово, современное написание и зашифрованные значения параметров в одной таблице. Время поиска искомого слова по такой базе составил в среднем 0.602 секунды, поиск всех слов по заданным грамматическим и синтаксическим параметрам занял около 19 секунд.

Разработанный интерфейс предоставляет возможность начать поиск с главной страницы. При этом пользователю предоставляется возможность в отдельных окнах выбрать искомые параметры и ввести искомое слово. Вид экранной формы представлен на Рис. 3. При желании можно выбрать подкорпус текстов, в котором производить поиск. После чего предоставляется страница поиска, где результаты представляются блоками по 10 элементов в каждом.

7 Заключение

Полученный в результате выполнения проекта синтаксически размеченный корпус может быть использован при научных изысканиях в области истории, грамматики, лексикографии, а также при изучении соответствующих курсов студентами филологических специальностей. Кроме того, он может быть востребован специалистами по литературе XIX века.

Заметим, что создаваемая информационная система универсальна по отношению к языку текста и типу разметки. В дальнейшем пользователь сможет самостоятельно определять элементы текста и формировать список атрибутов для этих элементов. Для введенных атрибутов можно будет указать структурные связи. Предусмотрена возможность создания собственных правил для парсера текста при помощи определенного метаязыка (язык регулярных выражений, набор правил), поиск вхождений текстов, частей текстов. С использованием введенных атрибутов размеченный текст можно будет представить в виде графа.

Литература

[1] Рогов А. А., Гурин Г.Б., Котов А.А., Сидоров Ю.В. Морфологически размеченный корпус по русской публицистике второй половины XIX века // Проблемы компьютерной лингвистики: сборник научных трудов. Вып. 3. Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2008. С. 209-219.

[2] Рогов А.А., Гурин Г.Б., Котов А.А. Некоторые особенности грамматически размеченного корпуса по русской публицистике второй половины XIX века. / Труды международной конференции «Корпусная лингвистика - 2008». – СПб.: С.–Петербургский гос. университет, факультет филологии и искусств, 2008. С. 326-333.

[3] Рогов А.А., Гурин Г.Б., Котов А.А., Сидоров Ю.В., Суровцева Т.Г. Программный комплекс «СМАЛТ». // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 10 Всероссийской научной конференции «RCDL-2008» (Дубна, Россия 7-11 октября 2008г.). – Дубна: ОИЯИ, 2008. С. 155-160.

[4] Грамматика современного русского литературного языка. М.: Наука. 1970.

[5] Русская грамматика. М.: Наука. 1980. Т. 2.

[6] Современный русский язык: Фонетика. Лексикология. Словообразование. Морфология. Синтаксис / Под общ. ред. Л. А. Новикова. Спб.: Лань. 2003. С. 631-644.

Some features of formation of digital corpus of texts with syntactic markup

Rogov A.A., Sidorov Yu.V., Sedov A.V., Gurin G.B., Kotov A.A., Nekrasov M.Yu.

In paper we describe system of syntactical analysis. For marking we use 39 structural schemes. The main marking unit is the clause – simple sentence.

For marking was created application in Delphi. In it user can move from clause to clause and attribute them. We also analyze some paths to improve effectiveness of program.

The structure of dictionary database tables was developed. Query execution speed was constantly analyzed during all period of work. The results of the analysis were taken into account while improving the structure of tables.

For online access to the database the web-resource was created. Modules of the information system were realization on PHP 4. For providing the maintenance of the pre-revolutionary Russian alphabet symbols all texts and wordforms are stored in the coding Unicode. The type Palatino Linotype is used for representation.

At present, database is composed of works, which belong to F.M. Dostoevskij and his contemporaries from the magazines “Vremja”, “Epoha”, “Svetoch”, “Sovremennik”, “Molva”, “Biblioteka dlja chtenija”, “Zarja”, “Grazdanin”. It also contains texts by Dahl and some other publicistic texts.