Проблемы бионики
1999. Вып. 51. С. 34–43
ХТУРЭ
УДК 519.7

Е.М. РОНИН, В.И.РУБЛИНЕЦКИЙ, В.А.ЧИКИНА

ПРОГРАММА СОЗДАНИЯ ЧАСТОТНОГО СЛОВАРЯ СЛОВ И ВЫРАЖЕНИЙ ДЛЯ РУССКОГО ЯЗЫКА

Улучшение качества компьютеров и сканнеров, а также увеличение количества текстов в электронной форме позволяют решать многие задачи прикладной лингвистики полностью на компьютере или преимущественно на компьютере с малым участием человека. В частности, на компьютере удается, с большей или меньшей точностью, считать частоты слов в текстах на естественном языке (в ЕЯ-текстах). Какие полезные прикладные задачи можно при этом решить, описано в работе [2]. Сегодня те исследования, которые велись на протяжении долгих лет огромными научными институтами, могут быть выполнены в течение нескольких часов в небольшой исследовательской лаборатории. Это ни в коей мере не обесценивает труды, ставшие классикой лингвистики [3,5], однако показывает, как далеко ушли вперед методы автоматизированной обработки информации.

В данной работе подробно описан один из подходов к автоматизированному (т.е. не чисто автоматическому, а требующему участия человека) подсчету частот слов. Этот подход можно использовать для составления частотных общих и специальных словарей.

Неискушенному читателю может показаться, что задача подсчета частот слов тривиальна. Поясним, что это не так, на материале русского языка, который рассматривается в данной работе. Слово удобно понимать как множество своих словоформ. Возьмем для примера слово

КОПЬЕ={копье, копья, копью, ..., копьях}.

В ЕЯ-тексте используются словоформы, и трудность состоит в том, что их нужно отождествить с соответствующим словом, так как надо подсчитать частоты слов, а не словоформ. Простой способ подсчета частот состоит в разработке формальных правил преобразования произвольной словоформы в каноническую: например, инфинитив для глагола, именительный падеж единственного числа для существительного и т.п. Если, например, вычисляя частоту слова *КОПЬЕ*, описать основу в виде *коп-*, то возникает опасность отнести ко множеству *КОПЬЕ* словоформу *копать*; если описать основу в виде *копь-*, то появляется опасность включить «чужую» словоформу *копь* и не узнать «свою» словоформу *копий*. А что делать со словоформой *копий* из множества *КОПИЯ*?

Преодолев трудности подсчета частот, мы обнаружим, что слова, встречаемые в научно-технической литературе, можно разбить на несколько

категорий (по убыванию частоты встречаемости): слова-антипризнаки [4] – слова, которые встречаются в любом тексте на данном языке и занимают верхние строки частотной таблицы (предлоги, союзы и некоторые местоимения); вводные слова, не несущие смысловой нагрузки; общеупотребительные слова, встречающиеся в литературных текстах и не имеющие научно-терминологического значения; общенаучные термины, встречающиеся в научно-технических текстах разнообразной направленности, и, наконец, узкоспециальные термины, характерные для данной конкретной предметной области. Из этого множества, полученного из анализа специальных текстов, составляются глоссарии для специальных словарей.

При создании алгоритмов, предназначенных для обработки информации ЕЯ-текстов, необходимо рассматривать несколько основных граней данной задачи. Наиболее важными являются следующие подзадачи:

1) Выбор способа хранения словарной и грамматической информации. Этот этап особенно важен для систем, работа которых базируется на поиске информации в словаре. Выбор способа включает в себя разработку модели морфологического деления слова (в современных системах применяются следующие модели деления: ОСНОВА-ОКОНЧАНИЕ, ОСНОВА-[СУФФИКС]-ОКОНЧАНИЕ, реже [ПРЕФИКС]-ОСНОВА-[СУФФИКС]-ОКОНЧАНИЕ) [1]; выбор и создание системы хранения информации на жестком диске и в оперативной памяти (выбор средств реализации хранения грамматической информации очень широк – ассоциативные базы данных (БД), иерархические, реляционные, а также способы хранения информации в оперативной памяти – различные формы деревьев и списков) [1].

2) После создания модели хранения информации логично выбрать метод ее обработки. В настоящее время обычно применяется следующий метод работы с грамматической информацией: ЗАПРОС НА ЧТЕНИЕ→ВЫБОРКА ИНФОРМАЦИИ С ЖЕСТКОГО ДИСКА→ ФОРМИРОВАНИЕ СТРУКТУРЫ ДАННЫХ В ПАМЯТИ→РАБОТА С ИНФОРМАЦИЕЙ →СБРОС РЕЗУЛЬТАТА НА ЖЕСТКИЙ ДИСК. Однако в связи с увеличением объемов оперативной памяти и вычислительной мощности компьютерного оборудования, а также благодаря применению в основных операционных системах технологии виртуальной памяти на сегодняшний день более перспективной является следующий метод обработки информации: ЗАПРОС НА СЧИТЫВАНИЕ В ОПЕРАТИВНУЮ ПАМЯТЬ ИНФОРМАЦИИ→СЕАНС РАБОТЫ С ПРОГРАММОЙ→ЗАПИСЬ ИЗМЕНЕНИЙ.

3) В результате выбора метода обработки грамматической информации открывается возможность выбора метода анализа поступающей ЕЯ-информации. Для практического применения зачастую используют методы, не требующие анализа контекста. Это словарный метод - каждое

слово, которое система способна обработать, находится в БД программы, где ему приписана соответствующая грамматическая и семантическая информация. Словарный метод позволяет с высокой точностью идентифицировать словоформу, однако требует большого объема оперативной памяти и повышенных затрат труда при заполнении БД.

4) Есть еще один метод, применимый к языкам со сложным словоизменением, – статистический, когда на основе обработки большого массива текстовой информации строится статистическая модель применения определенных слов и словосочетаний. В случае применения этого метода программа настраивается на текст определенной области знаний. Статистический метод отличается высокой ресурсоёмкостью. Указанные методы имеют свои достоинства и недостатки и обычно используются комбинированно [2].

5) Выбор способа пополнения словаря – чрезвычайно важный этап в разработке алгоритма. Три базовых способа могут быть применены по отдельности либо комплексно: пополнение вручную, когда пользователь сам вводит новую информацию; автоматизированное пополнение – когда программа предлагает пользователю определенные рекомендации по вводу информации или исправляет ошибки ввода; и, наконец, автоматическое пополнение (обычно с постредакцией) – когда программа должна сама выделить из определенного источника информацию и поместить ее в информационную базу.

6) Следующим фактором, который необходимо учесть при разработке программы, является способность алгоритма к сбору и последующему использованию словарно-грамматической информации. С этой целью обычно применяются методы анализа текста в несколько проходов с последовательным уточнением информации.

В качестве среды реализации программы был выбран компилятор языка Object Pascal - Borland Delphi версии 3.02. Этот компилятор выбран в связи с тем, что он содержит в своем составе гибкие и мощные средства работы с БД, создает 32-битные приложения, работающие в среде Windows и не требующие дополнительных библиотек исполнения. Все указанные факторы наряду с тем, что язык Object Pascal предоставляет программисту широкие возможности в реализации алгоритмов, послужили причиной выбора именно этого программного продукта. Для хранения информации планируется применить реляционную БД типа Paradox. В связи с тем, что язык выполнения запросов ANSI SQL предоставляет широкие возможности выполнения различных сложных операций над отношениями, необходимо воспользоваться им для работы с БД. В результате анализа проблемной области и перечня задач, которые необходимо решить, был сделан вывод о необходимости применения библиотеки компонентов RX Library. RX Library - это библиотека

компонент и функций для Borland Delphi и Borland C++ Builder, программный продукт, распространяемый бесплатно.

Русский язык является одним из самых сложных в плане моделирования его словоизменительного механизма – изменение основы внутри парадигмы и наличие множества омонимов [3] затрудняют создание аналитической модели словоизменения.

Обычно для моделирования словоизменительного механизма слова применяют следующую схему [1]:

ОСНОВА1[ОСНОВА2 (ПОЗИЦИИ)] - НАБОР_ОКОНЧАНИЙ

Здесь ОСНОВА1 – основа слова (префикс+корень+суффикс); ОСНОВА2 – основа слова в случае ее изменения (например: Корен_ь/Корн_я); ПОЗИЦИИ – список номеров в парадигме, где ОСНОВА1 подменяется ОСНОВА2; НАБОР_ОКОНЧАНИЙ – пронумерованный список окончаний.

Пример такого представления показан в табл. 1.

Таблица 1. Пример табличного представления парадигмы словоизменения

ОСНОВА1	ОСНОВА2	ПОЗИЦИИ	НОМЕР_ОКОНЧАНИЯ	ОКОНЧАНИЕ	Комментарий
КОРЕН	КОРН		1	Ь	Именит. Ед.
		X	2	Я	Род. Ед.
		X	3	Ю	Дат. Ед.
			4	Ь	Винит. Ед.
		X	5	ЕМ	Творит. Ед.
		X	6	Е	Предл. Ед.
		X	7	И	Именит. Мн.
		X	8	ЕЙ	Род. Мн.
		X	9	ЯМ	Дат. Мн.
		X	10	И	Винит. Мн.
		X	11	ЯМИ	Творит. Мн.
		X	12	ЯХ	Предл. Мн.

К сожалению, такое представление парадигмы словоизменения имеет определенные недостатки – нет возможности обрабатывать слова, имеющие больше двух основ (например, общеупотребительные глаголы *быть, ходить*). В таких случаях обычно прибегают к ряду определенных искусственных приемов (деление парадигмы на несколько частей, разбиение по грамматическому признаку). Список окончаний во многих случаях жестко задан и нет возможности вводить и хранить многовариантные слова (как, например: на *шелк_у*, о *шелк_е* – два окончания предложного падежа единственного числа).

Для снятия этих ограничений есть возможность применить усовершенствованную схему хранения словарной информации:

СПИСОК_ОСНОВ – КОД_СООТВЕТСТВИЯ - НАБОР_ОКОНЧАНИЙ

СПИСОК_ОСНОВ – это все основы, существующие в разных словоформах этого слова; КОД_СООТВЕТСТВИЯ – число, указывающее, какая из основ используется для этого окончания; НАБОР_ОКОНЧАНИЙ – пронумерованный список окончаний. Пример такого представления показан в табл. 2.

Таблица 2. Пример усовершенствованного табличного представления парадигмы

СПИСОК_ОСНОВ	КОД_СООТВЕТСТВИЯ	НОМЕР_ОКОНЧАНИЯ	ОКОНЧАНИЕ	Комментарий
КОРЕН	1	1	Ь	Именит. Ед.
	2	2	Я	Род. Ед.
	2	3	Ю	Дат. Ед.
	1	4	Ь	Винит. Ед.
	2	5	ЕМ	Творит. Ед.
	2	6	Е	Предл. Ед.
КОРН	2	7	И	Именит. Мн.
	2	8	ЕЙ	Род. Мн.
	2	9	ЯМ	Дат. Мн.
	2	10	И	Винит. Мн.
	2	11	ЯМИ	Творит. Мн.
	2	12	ЯХ	Предл. Мн.

Представленная схема применяется в модели хранения данных, приводимой в этой работе.

Алгоритм дополнения слов в базу данных

(Исходные данные:

порог вхождения – число, кодирующее минимальный уровень сходства парадигмы словоизменения с собранными данными,

осмысленный текст на русском языке,

база данных со списком парадигм языка и слов.

Результаты:

слова в базе данных):

1. Построить список предполагаемых основ слов и наборов их окончаний.

1.1. Проверить, нет ли обрабатываемого слова в словаре БД. Если есть, то перейти к пункту 2. Иначе - к 1.2.

1.2. Попытаться отделить наиболее длинное окончание из списка допустимых окончаний, если удачно – перейти к пункту 1.4. Иначе перейти к 1.3.

1.3. Взять более короткое окончание и перейти к 1.1. Если список окончаний пуст, завершить построение списка предполагаемых основ для данного слова и перейти к следующему слову. Если текст просмотрен, перейти к пункту 3.

1.4. Если в списке нет отделенной основы, то 1.5. Иначе 1.6.

1.5. Внести в список предполагаемых основ часть слова, оставшуюся после выделения окончания. Связать с ней это окончание. Присвоить частоту 1. Перейти к 1.1.

1.6. Добавить в список окончаний для данной основы, инкрементировать частотный счетчик основы. Перейти к 1.1.

2. Провести эвристический анализ грамматических признаков.

2.1. Поиск обрабатываемого слова в списке слов-признаков. Если найдено – 2.4.

2.2. Поиск обрабатываемого слова в списке местоимений. Если найдено – 2.4, иначе 2.3.

2.3. Определение связки прилагательное-существительное. В случае удачи – 2.4, иначе завершение без информативного результата.

2.4. Передача информации в пункт 3 для обработки.

3. Поиск и определение наиболее подходящей парадигмы словоизменения.

3.1 Подсчет соответствий для нового слова в окончаниях для всех парадигм в БД.

3.2. Коррекция согласно пункту 2.

3.3. Если слово удовлетворяет по количеству корректных окончаний парадигме N (превышен порог вхождения), то 3.4, иначе переход к 3.1.

3.4. Дополнение слова в БД, переход к 3.1.

Эвристический анализ слов

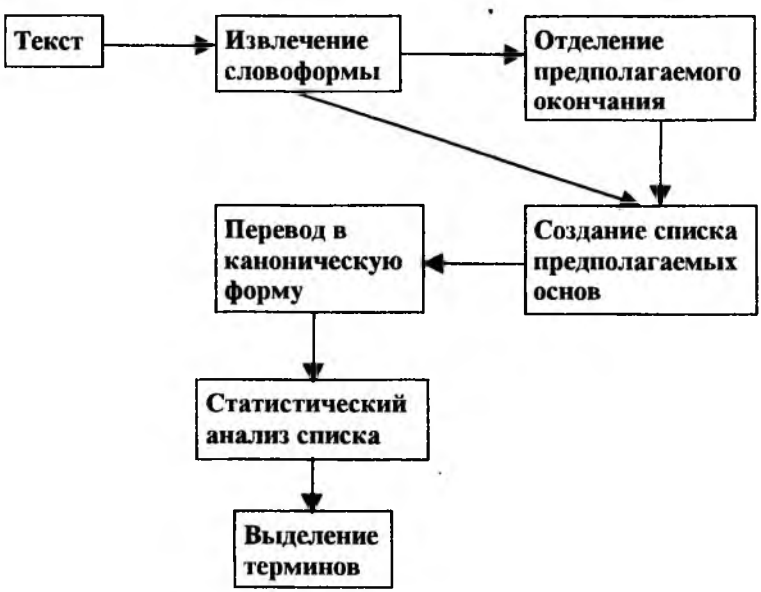
К сожалению, даже в системах, пользующихся словарными методами, из-за развитой омонимии русского языка нет возможности четко определить грамматические признаки и на их основе отнести слово к определенному частотному типу. В системах, применяющих аналитические и статистические методы, эта проблема стоит еще острее. Для уточнения полученной предположительной грамматической информации применяются эвристические алгоритмы.

1) Связка ПРЕДЛОГ–ПРИЛАГАТЕЛЬНОЕ или ПРЕДЛОГ–СУЩЕСТВИТЕЛЬНОЕ (Примеры: *в* доме, *на* зеленом дереве). Опознавая неизменяемый предлог, можно определить примерные грамматические признаки. Однако здесь важно учитывать, что некоторые предлоги могут соответствовать нескольким падежам (*в доме* – предложный падеж или *в дом* - винительный) .

2) Связка МЕСТОИМЕНИЕ–ПРИЛАГАТЕЛЬНОЕ или МЕСТОИМЕНИЕ––СУЩЕСТВИТЕЛЬНОЕ (Пример: *этот* дом, *те* деревья). В связи с тем, что в русском языке сравнительно мало местоимений, а их вес в частотной таблице велик, есть смысл хранить все словоформы местоимений в БД словаря. Этим, кроме того, обеспечивается возможность поиска связок местоимений и имен.

3) Для связки ПРИЛАГАТЕЛЬНОЕ–СУЩЕСТВИТЕЛЬНОЕ есть возможность выделить грамматические признаки по характерным окончаниям прилагательных и присвоить эти признаки существительным.

Алгоритм построения списка терминов
(Исходные данные:
текст на русском языке,
база данных со списком парадигм слов языка.
Результаты:
Файл со словами-терминами):
Схема работы алгоритма представлена на рисунке.



Последовательность обработки текста в режиме поиска терминов

1. Фаза 1.

1.1.Применить для построения списка слов текста алгоритм дополнения слов в БД (см. выше).

2. Фаза 2.

2.1. Взять новое слово из списка, полученного в 1, проверить, нет ли его в списке слов-антипризнаков. Если слово не содержится в списке, перейти к 2.2, иначе снова 2.1.

2.2. Если слово опознано алгоритмом из 1 – заменить на каноническую форму, поместить в список. Иначе поместить в список в данной форме. Перейти к 2.3.

2.3. Дополнить список следования (для поиска словосочетаний). Вернуться к 2.1, если весь текст еще не обработан. Иначе перейти к 3.

3. Фаза 3.

3.1. Взять слово из списка 2. Перейти к 3.2.

3.2 .Проверить, не превышает ли частота данных слов заданную, если так - 3.3, иначе 3.1.

3.3. Проверить список следования. Если следующее слово также проходит частотную проверку, добавить словосочетание в файл результата, иначе добавить слово. Перейти к 3.4.

3.4. Проверить, не обработан ли весь текст. Если да, то завершение, иначе 3.1.

Структуры данных

В связи с тем, что алгоритм предназначен для хранения словарно-грамматической информации в виде реляционной БД, приведем схемы отношений БД (табл. 3-8).

Таблица 3. Схема отношения базы данных СВЯЗИ

Номер поля	Атрибут	Размерность	Примечание
1	Счетчик COUNT	4-байтовое число	Используется как основной ключ
2	Ссылка на отношение ОСНОВЫ PTR2BASEWRDS	4-байтовое число	
3	Код-счетчик слов ID	4-байтовое число	
4	Ссылка на отношение ПАРАДИГМЫ PTR2PARADIGMA	2-байтовое число	
5	Вектор-шаблон покрытия парадигмы № 1 POLYPTR1	4-байтовое число	
	Вектор-шаблон покрытия парадигмы № 2 POLYPTR2	4-байтовое число	
6	Код части речи PART	2-байтовое число	
7	Признак автоматического пополнения AUTOADD	Логическая величина	
8	Код типа слова TYPEWORD	2-байтовое число	

Отношение СВЯЗИ содержит грамматическую информацию о слове (к какой части речи относится, тип слова и ряд служебных характеристик), ссылки на список основ и парадигму словоизменения.

Таблица 4. Схема отношения базы данных ОСНОВЫ

Номер поля	Атрибут	Размерность	Примечание
1	Номер основы COUNT	4-байтовое число	Используется как основной ключ
2	Основа слова BASEWORD	32 символа	Нет повторений

Отношение ОСНОВЫ представляет собой список основ слов, находящихся в БД, с уникальными индексными номерами.

Таблица 5. Схема отношения базы данных ПАРАДИГМЫ

Номер поля	Атрибут	Размерность	Примечание
1	Счетчик COUNT	4-байтовое число	Используется как основной ключ
2	Номер парадигмы	2-байтовое число	
3	Код грамматических признаков CYPHER	4-байтовое число	
4	Порядок следования внутри парадигмы ORDER	2-байтовое число	
5	Ссылка на отношение ОКОНЧАНИЯ PTR2END	4-байтовое число	

Отношение ПАРАДИГМЫ содержит комплекс парадигм словоизменения русского языка. В табл.6 в закодированном виде хранятся грамматические признаки словоформы.

Таблица 6. Схема отношения базы данных ОКОНЧАНИЯ

Номер поля	Атрибут	Размерность	Примечание
1	Номер окончания COUNT	2-байтовое число	Используется как основной ключ
2	Окончание ENDING	10 символов	Нет повторений

Отношение ОКОНЧАНИЯ представляет собой пронумерованный список окончаний, обрабатываемых алгоритмом.

Таблица 7. Схема отношения базы данных СЛОВА-ПРИЗНАКИ

Номер поля	Атрибут	Размерность	Примечание
1	Слово-признак WORD_SIGN	25 символов	Используется как основной ключ
2	Признак последействия BEFORE	Логическая величина	
3	Ссылка на отношение КОД ПРИЗНАКОВ PTR2DECYPHER	2-байтовое число	

Отношение СЛОВА-ПРИЗНАКИ содержит список неизменяющихся слов, использующихся для эвристического анализа принадлежности к определенной парадигме.

Таблица 8. Схема отношения базы данных КОД ПРИЗНАКОВ

Номер поля	Атрибут	Размерность	Примечание
1	Счетчик COUNT	4-байтовое число	Используется как основной ключ
2	Номер толкования TARGET	2-байтовое число	
3	Код части речи PART	2-байтовое число	
4	Код толкования CYPHER	4-байтовое число	

Отношение КОД ПРИЗНАКОВ содержит расшифровку кодов признаков, а также информацию для эвристического алгоритма определения слов.

Список литературы: 1. *Бондаренко М.Ф., Осыка А.Ф.* Автоматическая обработка информации на естественном языке: Учеб. пособие. К.: УМК ВО, 1991. 144 с. 2. *Бондаренко М.Ф., Рублинецкий В.И., Чикина В.А.* О прикладных задачах машинной лингвистики, решаемых подсчетом частот слов и выражений // Проблемы бионики. 1999. Вып. 50. С. 12-17. 3. *Зализняк А.А.* Грамматический словарь русского языка: Словоизменение. М.: Рус. яз., 1980. 880 с. 4. *Лаптева М.В., Вайнер В.Г.* Метод автоматизированного составления глоссария/ Экономико-экологическое моделирование. Учеб. пособие/ Под ред. В.Г. Вайнера. Харьков: «Бизнес-информ». 1997. С. 334-346. 5. *Частотный* словарь русского языка / Под ред. Л.Н. Засориной. М.: Русский язык.1977. 936с.

Поступила в редколлегию 05.06.99