УДК 004.912

СОЗДАНИЕ ЛИНГВИСТИЧЕСКОЙ ОНТОЛОГИИ ОБРАЗОВАТЕЛЬНОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

© 2010 А. В. Абрамов

асс. каф. программного обеспечения и
администрирования информационных систем
e-mail: andbramov@rambler.ru

Курский государственный университет

В статье рассматривается механизм накопления терминологической базы образовательной предметной области для создания лингвистической онтологии.

Ключевые слова: онтология, гипертекст, термин, понятие.

Ранее мы поднимали вопрос о построении лингвистической онтологии с целью автоматизированного построения гипертекста для текстов большого объема. Нами подробно рассмотрена причина, побудившая нас обратиться к лингвистической онтологии как средству создания гипертекстового представления текста [Абрамов 2008]. Напомним некоторые основные моменты.

Гипертекстовая модель представления информации в электронных образовательных изданиях является наиболее используемой в настоящее время. Данная модель предоставляет некоторые преимущества [Григорьев, Гриншкун 2006], одним из которых является возможность использования гипертекста для автоматизированного обучения.

Гипертекстовая модель представляет собой совокупность связей, узлов-объектов и пользовательский интерфейс. Процесс построения гипертекста для автора заключается в самостоятельном создании и изменении узлов, содержания узлов, связей между узлами, форм представления узлов на экране компьютера.

Однако анализ литературы, посвященной гипертекстовым технологиям, показывает, что существуют трудности при создании обширной системы помощи, справочной информации для текстов большого объема. Эти трудности связаны с практической невозможностью расставить ссылки вручную. В связи с этим возникает потребность в автоматическом определении мест в тексте для ссылок, а также в выявлении связей между документами и расстановкой ссылок.

Безусловно, такие инструменты существуют, в том числе и российского производства. Функцию расстановки гиперсвязей включает система HyperMethod петербургской фирмы AI Labs, а также система TextAnalyst.

Однако эти системы работают не очень хорошо, поскольку в них не применяются методы лингвистической обработки текста. Представляется, что применение графематического, морфологического, синтаксического и, безусловно, семантического анализа предположительно повысит качество и точность расстановки ссылок.

Мы рассматривали возможность построения гипертекстового представления материала при помощи лингвистической онтологии [Абрамов 2008]. Исследователи рассматривают различные подходы к определению онтологии [Клещев, Артемьева 2001; FIPA 1998; Gruber 1992], определение онтологии можно выразить математической формулой [Гаврилова, Хорошевский 2000]

$$O = \langle X, R, F \rangle, \qquad (1)$$

где:

X – конечное множество концептов (понятий, терминов) предметной области, которую представляет онтология O;

R – конечное множество отношений между концептами (понятиями, терминами) заданной предметной области;

F – конечное множество функций интерпретации (аксиоматизации). При наложении ограничений $R = \varnothing$ и $F = \varnothing$ онтология O трансформируется в простой словарь

$$O = V = \langle X, \{\ \}, \{\ \} \rangle$$

или вырожденную онтологию. Видоизменим выражение (1) следующим образом

$$O = O' = \langle X, Z, R, \{\ \}, \{\ \} \rangle, \qquad (2)$$

где Z непустое, конечное множество определений терминов из множества X.

Таким образом, лингвистическая онтология – это иерархическая сеть терминов. Каждое понятие связывается отношениями с другими понятиями онтологии.

Воспользуемся результатами работы Доброва, Лукашевича [2007]для определения отношений между терминами в лингвистической онтологии. Первый тип отношений, родовидовое отношение ниже–выше, обладает свойством транзитивности и наследования. Второй тип отношений, отношение часть–целое, используется не только для описания физических частей, но и для других внутренних сущностей понятия, таких как свойства или роли для ситуаций.

Еще один тип отношения – асц2–асц1, называемого несимметричной ассоциацией, связывает два понятия, которые не могут быть связаны выше рассмотренными отношениями, но одно из понятий не может существовать без существования другого. Последний тип отношений, симметричная ассоциация, связывает, например, понятия, которые нельзя «склеить» в одно понятие, хотя они и очень близки по смыслу

Отношения ниже–выше, часть–целое и несимметричная ассоциация являются иерархическими отношениями. Таким образом, на основе свойств иерархичности, транзитивности и наследования для каждого понятия может быть определена совокупность понятий, которые являются для него нижестоящими понятиями.

Построение данного вида онтологии (лингвистической онтологии) подразумевает определение множества терминов из обрабатываемого текста и сопоставление их с соответствующими определениями.

Процесс создания лингвистической онтологии будет состоять из следующих этапов:

формирование терминологической базы некоторой предметной области по массиву текстовой информации;

анализ полученной информации человеком-экспертом, с целью «фильтрации» терминов и указания определения данных терминов;

установление человеком-экспертом отношений между набором терминов предметной области.

Каждый из этих этапов, за исключением первого, подразумевает участие человека-эксперта. Рассмотрим далее описание первого этапа построения онтологии. Отметим вначале следующие ограничения и допущения:

1) произвольное содержание и структура текста. Под содержанием понимается принадлежность текста к какой-либо предметной области, под структурой – порядок следования структурных элементов текста (параграф, глава, абзац);

2) размер текста. Мы накладываем определенные ограничения на размер обрабатываемого текстового фрагмента в силу зависимости используемых нами статистических методов от объема исследуемого фрагмента.

Эти допущения и ограничения позволят в некотором смысле универсализировать построение лингвистической онтологии текста O' на естественном языке.

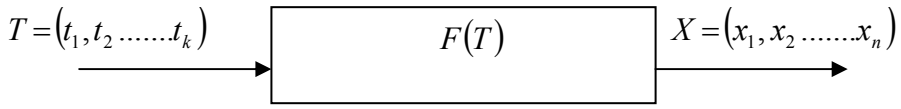
Итак, в рамках первого этапа необходимо из текста выбрать «кандидатов» для включения во множество X. Заведомо мы отметим одно важное допущение: при отборе терминов-«кандидатов» мы основываемся на гипотезе о том, что словарный запас и частота использования слов зависят от темы текста [Дубинский 2001; Ермаков 2000]. Другими словами, мы предполагаем, что ключевые слова текста (т. е. слова, частота повторения которых в тексте выше других) являются основными и предположительно терминами. Мы решили пользоваться статистическими методами поиска ключевых слов в тексте в силу их простоты и ресурсонезависимости.

Следует сказать несколько слов о возможных моделях представления текста, его информационного наполнения. В качестве примера ранее мы рассматривали модель структурного представления текста, предложенную Чугреевым [Абрамов 2008].

Опишем метод поиска ключевых слов (слов-«кандидатов») в тексте. Данный метод будем рассматривать как функцию F.

$$X = F(T),$$

здесь X – искомое множество ключевых слов, T – множество информационных элементов текстового фрагмента.



Элементами множества T могут являться либо отдельные слова, либо предложения текстового фрагмента. В первом случае для предложения «Гражданский служащий, изъявивший желание участвовать в конкурсе, направляет заявление на имя руководителя» множество будет содержать:

t_1 – Гражданский;

t_2 – служащий;

t_3 – руководителя;

……

t_{12} – руководителя.

Задача анализатора из входного потока T подсчитать количество слов, встречающихся наибольшее количество раз, то есть имеющих наибольший вес

Входной поток T представляет собой некоторый массив строк, имеющий конечную длину. Задача выделения слов из входного потока сводится к разбору строк, последовательно поступающих на вход метода разбора. Алгоритм получения очередного слова достаточно прост: на каждом шаге анализируется очередной символ текущей строки; если он не является разделителем, то добавляется к текущей лексеме, в противном случае текущая лексема добавляется в таблицу лексем, после чего ей присваивается пустое значение. Необходимо заметить, что на данном этапе происходит проверка

наличия очередного полученного таким способом слова в стоп-словаре. Если проверка проходит успешно, данная лексема не добавляется в таблицу, а отбрасывается, так как не является необходимой с рассматриваемой точки зрения.

Таким образом обрабатываются все строки входного массива. В результате имеем таблицу лексем, содержащую все слова из входного потока (за исключением стоп-слов). С точки зрения структур данных наша таблица лексем представляет собой хеш-таблицу. Хеширование производится с помощью хеш-функции (функции расстановки). Получая в качестве аргумента некоторое слово, функция выдает в результате некоторое целое число – индекс в таблице лексем, под которым следует хранить это слово. Реализуется это при помощи функции, выдающей по заданной букве ее номер в русском алфавите, и функции, суммирующей коды букв слова [Кубенский 2001].

Для разрешения коллизий используется линейный список. Каждый элемент данного списка содержит два поля: строковое, в котором хранится лексема, и целочисленное, содержащее количество повторений данной лексемы во входном потоке. Следующий этап выделения возможных терминов заключается в поиске морфологически родственных слов в полученной таблице и замене их так называемым главным словом. Для этого необходимы дополнительный проход по хеш-таблице, а также дополнительная структура данных – морфологический словарь, древовидная сильноветвящаяся структура, каждая вершина которой представляет собой массив, содержащий элементы, имеющие три поля: поле флагов, уникальный номер и указатель на вершину следующего уровня.

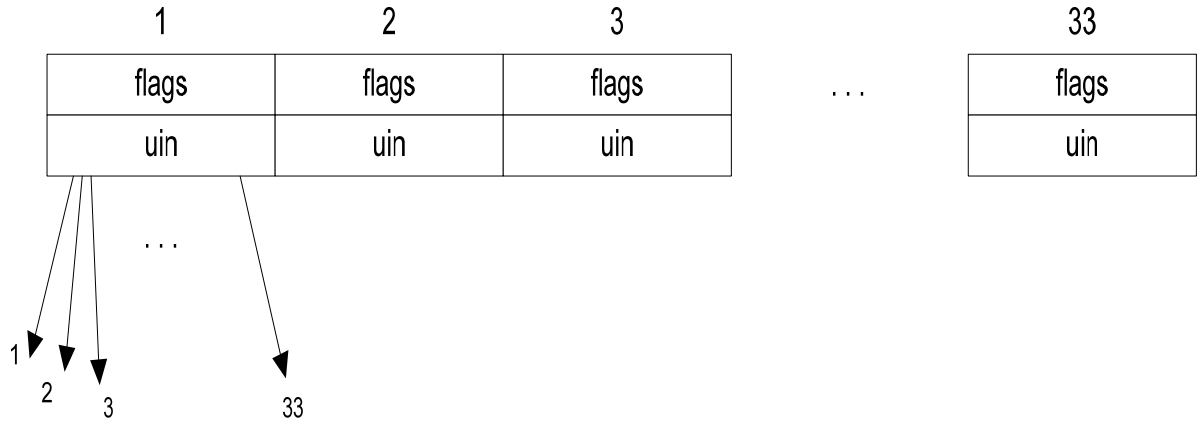


Рис. 1. Структура вершины словаря

Структура вершины представлена на рисунке 1. Каждый элемент пронумерован, и ему соответствует буква русского алфавита. Следует заметить, что данный подход позволяет не хранить букву непосредственно в памяти. Таким образом, слово представляет собой путь в дереве от корня до вершины, имеющий признак конца слова. Рассмотрим структуры поля флагов. Данное поле содержит в себе несколько признаков: признак окончания слова, признак основы, признак главного слова в группе родственных слов. Таким образом, поиск морфологически родственных слов заключается в рекурсивном спуске по дереву и нахождению элемента вершины, имеющего признак основы. После этого осуществляется левосторонний обход дерева, корнем которого является найденная вершина, и все найденные слова добавляются в список родственных слов с пометкой слова, являющегося главным. Во избежание ошибок, в случаях вложения основ друг в друга, для каждого элемента вершины введено дополнительное целочисленное поле (uin), разделяющее группы родственных слов, у которых одна основа

включается в другую. Поиск морфологически родственных слов схематично изображен на рисунке 2.

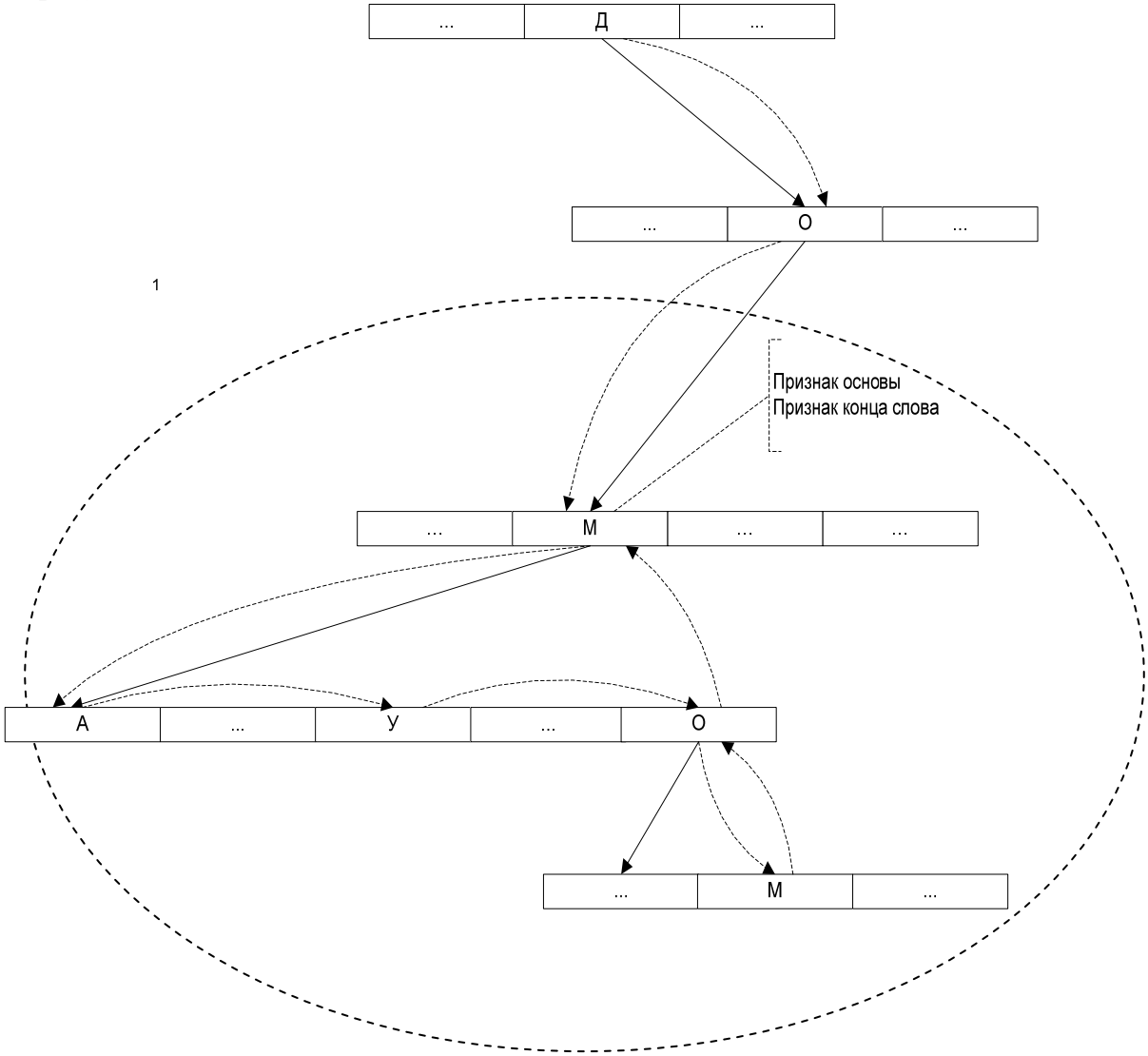
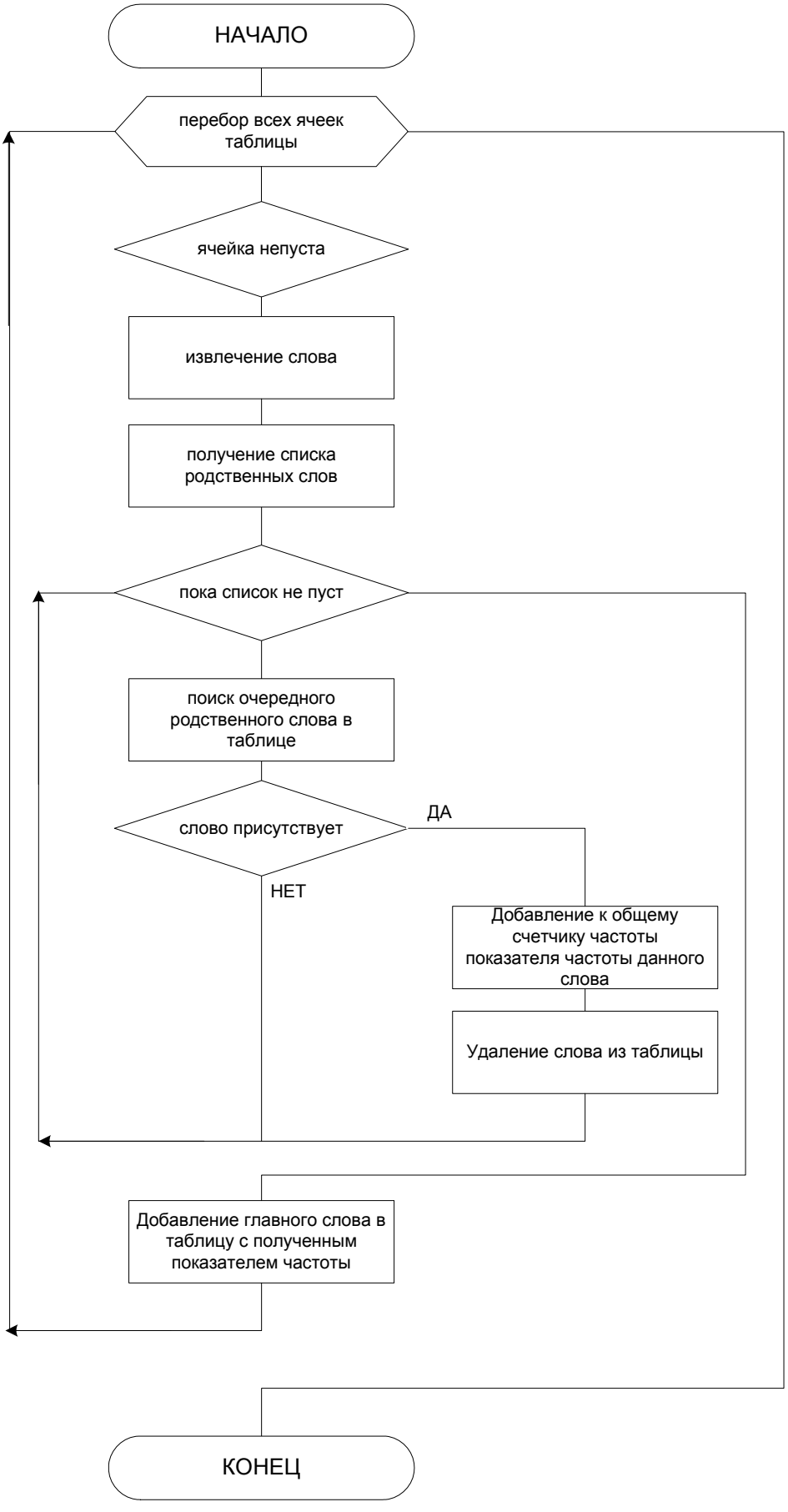


Рис 2. Поиск родственных слов

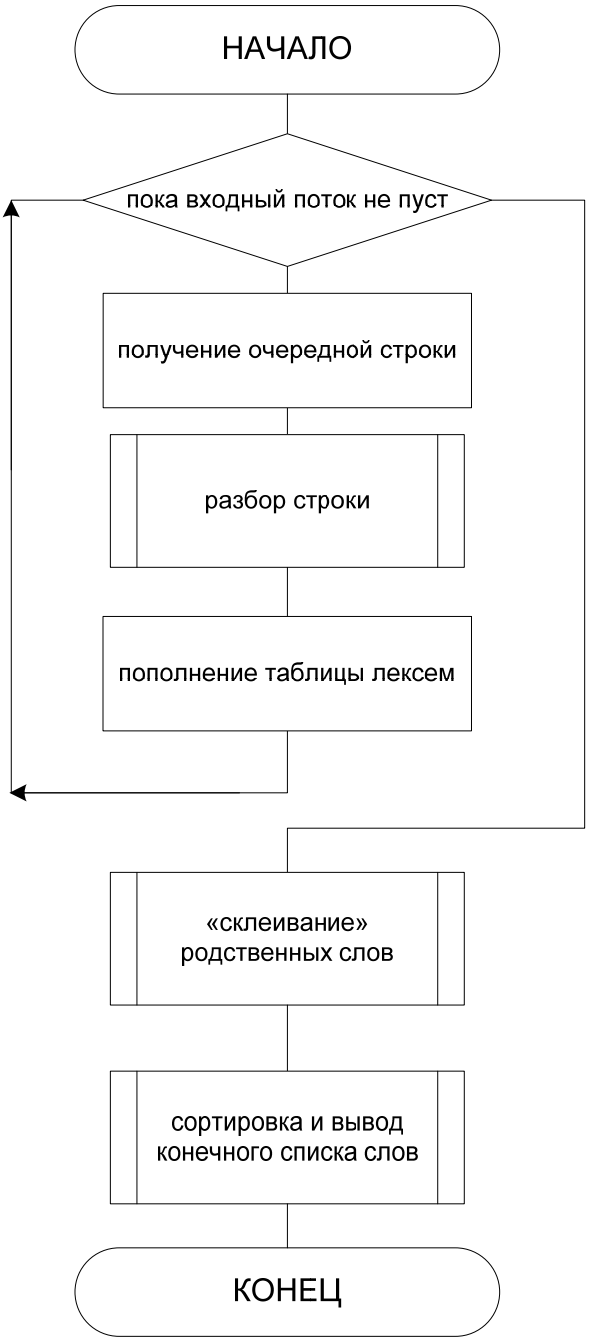
Как уже упоминалось, после разбора текста необходим дополнительный проход по таблице лексем для поиска и замены групп родственных слов соответствующим главным словом путем сложения частоты их появления в тексте. Алгоритм данного «склеивания» представлен на блок-схеме 1.

После завершения данного этапа таблица лексем содержит уже главные слова всех групп родственных слов, которые встретились во входном потоке. Необходимо заметить, что при наличии в тексте слов, не присутствующих в морфологическом словаре, формируется выходной список данных слов. На его основе в дальнейшем дополняется морфологический словарь, что производится уже отдельной операцией.

В итоге алгоритм выбора и подсчета слов входного потока можно представить в виде обобщенной блок-схемы 2.

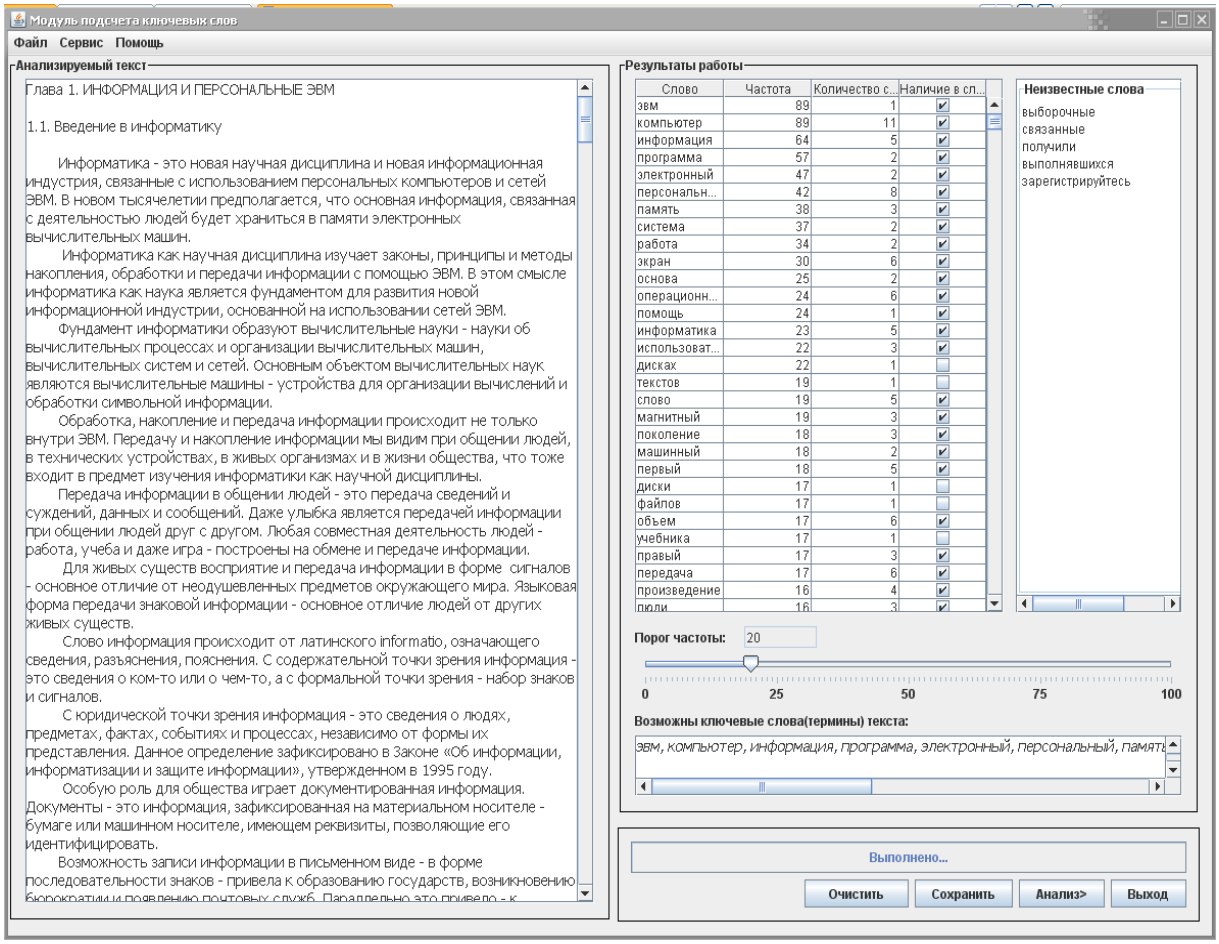

НАЧАЛО
перебор всех ячеек таблицы
ячейка непуста
извлечение слова
получение списка родственных слов
пока список не пуст
поиск очередного родственного слова в таблице
слово присутствует
ДА
НЕТ
Добавление к общему счетчику частоты показателя частоты данного слова
Удаление слова из таблицы
Добавление главного слова в таблицу с полученным показателем частоты
КОНЕЦ

Блок-схема 1


НАЧАЛО
пока входный поток не пуст
получение очередной строки
разбор строки
пополнение таблицы лексем
«склеивание»
родственных слов
сортировка и вывод
конечного списка слов
КОНЕЦ

Блок-схема 2

Обработаем данным способом первую главу учебника по информатике В.А. Каймина.



На выходе системы получаем множество X' слов-«кандидатов» вида

$$X' = \{x, V\},$$

где

x – слово-«кандидат»; V – вес «кандидата».

Например, для слова-кандидата «ЭВМ» вес будет равен 85. Задача эксперта на данном этапе – провести анализ полученной информации с целью «фильтрации» терминов и указания их определения. Так, понятию ЭВМ будет соответствовать определение: «ЭВМ – это электронно-вычислительная машина». Проанализировав полученную информацию, эксперт ограничивает круг терминов. Далее указываются отношения между выбранными терминами и заканчивается формирование терминологической базы образовательной предметной области.

Библиографический список

Абрамов А.В. Онтологический подход к систематизации контента // Материалы II Международной научно-практической конференции «Информационные технологии в образовании (ИТО-Черноземье – 2008)». Ч. 1. Курск, 8–11 декабря 2008 г. Курск: Курск. гос. ун-т, 2008. С. 87–90.

Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000.

Григорьев С.Г., Гриншкун В.В. Образовательные электронные издания и ресурсы. М., 2006.

Добров Б.В., Лукашевич Н.В. Лингвистическая онтология по естественным наукам и технологиям для приложений в сфере информационного поиска. 2007 // URL: http://fccl.ksu.ru/issue_spec/docs/oent-kgu.doc (дата обращения: 10.04.2010).

Дубинский А.Г. Некоторые вопросы применения векторной модели представления документов в информационном поиске // Управляющие системы и машины. 2001. №4. С. 77–83.

Ермаков А.Е. Полнотекстовый поиск: проблемы и их решение // Мир ПК. 2000. №5.

Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Ч. 1. Существующие подходы к определению понятия «онтология» // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2001. № 2. С. 20–27.

Клещев А.С., Артемьева И.Л. Математические модели онтологии предметной области. Ч. 2. Компоненты модели // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2001. № 3. С. 19–28.

Кубенский А.А. Создание и обработка структур данных в примерах на Java. Серия «Мастер». СПб.: БХВ-Петербург, 2001. 336 с.

FIPA, 1998. Ontology Service. FIPA 98 Specification. Part 12.

Gruber T.R. Ontolingua: A Mechanism to Support Portable Ontologies. Technical Report KSL-91-66, Stanford University, Knowledge Systems Laboratory, 1992.