ТАРТУСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Ээремаа Кульдев Аугустович

РАЗМЫТАЯ МОДЕЛЬ ДОКУМЕНТАЛЬНОГО ИНФОРМАЦИОННОГО ПОИСКА

05.13.01 – техническая кибернетика и теория информации

Диссертация

представленная на соискание ученой степени кандидата технических наук

Научный руководитель

кандидат физико-математических наук доцент Л. Выханду

Диссертация выполнена при кафедре программирования ТГУ

Тарту – 1981

I. ВВЕДЕНИЕ

I.I. Актуальность автоматизации поиска документов

Количественный рост документальной информации ведет к тому, что традиционные библиографические методы описания и поиска документов оказываются непригодными для эффективного нахождения нужной информации. В фундаментальной работе по информатике [4.6] сказано (стр. 42): "... достаточно быстрая и исчерпывающая информация может быть обеспечена лишь при условии разработки и широкого применения принципиально новых методов и средств, специально предназначенных для переработки документальной информации."

Методы переработки документальной информации, как правило, сориентированы на их применение в некоторой предметной области, например в химии, в патентоведении, в праве и т.д. Это связано со спецификой представления информации в соответствующих областях. В предложенной в настоящей работе модели информационного поиска соблюдается специфика правовых документов, но основные принципы применимы и для составления автоматизированных информационно-поисковых систем в других областях, где исходные документы задаются на естественном языке и существует необходимость учитывания терминологических связей.

Важность разработки методов автоматического анализа правовых документов подчеркивается во многих постановлениях партии и правительства (см. например [2.I, 2.2]). В I973-ем году был соз-

дан центр координирования таких работ - Институт правовой информации. Государственным комитетом по науке и технике в 1979-ом году были поставлены конкретные задания по проектированию и созданию общегосударственной сети центров правовой информации.

Работы по автоматической обработке правовой информации в Тартуском госуниверситете начались в 1964-ом году. Специальным постановлением Совета министров ЭССР нр. 421-к, от 20 июня 1972 лабораторию криминологии ТГУ обязывали провести исследования по автоматическому анализу правовых документов с целью создания документальных информационно-поисковых систем в этой области. В рамки выполнения поставленной задачи входит и настоящая работа.

Среди многочисленных проблем, возникающих при создании информационно-поисковых систем (ИПС), особо актуальными являются адекватное представление содержания документов в информационно-поисковом языке (ИПЯ), автоматизация индексирования и автоматизация применения тезауруса. В настоящей работе ИПЯ конструируется на базе теории размытых множеств и отношений, чем достигается гибкое индексирование и автоматизация применения тезауруса. Во всей работе соблюдается возможность автоматизирования всех информационных процессов.

1.2. Постановка задачи и цель работы

В самом общем виде проблему документального информационного поиска можно поставить следующим образом. Задано некоторое множество документов $\mathcal{D} = \{d_1, d_2, \cdots, d_m\}$, множество знаний $\mathcal{R}$ о предметной области и множество информационных запросов $Q = \{q_1, q_2, \cdots, q_k\}$. Требуется найти функцию $F(q, \mathcal{D}, \mathcal{R})$, сопоставляющую любой запрос $q \in Q$ с множеством релевантных ему документов.

Пусть $\mathfrak{D}$ является множеством правовых документов, характеризуемых специальным содержанием и формой представления. Согласно работам С.С.Москвина [7.1] и С.Н.Юсувова [5.28] правовые документы являются текстами на некотором естественном языке, для которых характерны ясность, лаконичность и точное применение терминологии. Кроме юридической терминологии в них применяется и терминология других, урегулируемых ими предметных областей. Приведенные основные черты юридического документа не характеризуют внутреннюю структуру его построения. Некоторые особенности структуры юридической нормы и возможности ее отражения в формализированном языке были исследованы в ходе планирования эксперимента JURIOS (см. [5.9]), но в созданной ИПС не учтены. Это ввиду того, что, во-первых, отсутствует математическая аппаратура для столь точного анализа юридических текстов и перевода содержания документа на формально-логический язык и, во вторых, - в документальном информационном поиске столь точное отображение информации не требуется (это дело информационно-логических систем, на которые налагаются более высокие требования).

Таким образом, конструируемая правовая ИПС имеет дело с текстами на естественном языке и должна уметь использовать терминологические связи между понятиями разных предметных областей. В дальнейшем допустим, что терминологические связи в основном заданы в виде тезауруса, который входит в множество вспомогательных знаний $\mathfrak{R}$.

Что касается выбора функции F, то это зависит от внутрисистемного представления документа и информационного запроса, а также от формализации понятия релевантности. Важно отметить, что интуитивно при некотором заданном запросе q необязательно существует четкое разделение множества $\mathfrak{D}$ на релевантные и нерелевантные документы. В общем случае $F(q, \mathfrak{D}, \mathfrak{R})$ должна упо-

рядочить $\mathfrak{D}$ в убывающую по степеням релевантности последовательность. Обыкновенно [4.8] степень релевантности (вес) вычисляется в процессе сравнения запроса с поисковым образом документа и является некоторой мерой их близости. При этом Q, $\mathfrak{D}$ и $\mathfrak{R}$ считаются "точными" в том смысле, что данные, записанные в этих множествах, не сопровождаются с оценками существенности - весами. Однако, в действительности при достаточно полном индексировании документов индексы имеют разные весы. Также поисковые признаки в запросе q и отношения, задаваемые в $\mathfrak{R}$, имеют с точки зрения информационного поиска разные степени значимости. Таким образом, не только процесс сравнения запроса с поисковым образом документа, но и всю работу ИПС следует рассматривать "размытой".

Возникает вопрос, как описать работу ИПС, чтобы любые утверждения сопровождались с оценками значимости и полученные оценки, в конечном счете трансформировались в оценки релевантности выдаваемых документов? Какая в таком случае должна быть роль тезауруса, как его применять в ИПС и какого рода сведений он содержит? Какие ограничения надо наложить на размытую ИПС, какие дополнительные возможности возникают и насколько автоматизируема работа размытой ИПС?

Целью настоящей работы и является разработка системного подхода к размытому информационному поиску правовых документов. При этом соблюдается автоматизируемость всех этапов работы ИПС, включая индексирование и применение тезауруса. Вмешательство человека допускается лишь в некоторых режимах работы ИПС, как, например, в случае диалога между пользователем и системой.

В рамках общей проблемы размытого информационного поиска можно выделить следующие наиболее важные подпроблемы:

- определение роли, состава и структуры тезауруса и разработка принципов его автоматического применения в процессах ин-

дексирования и поиска ответа;

- возможность автоматизировать составление тезауруса;

- автоматизация индексирования в размытой ИПС с учетом синонимности, омонимности, словосочетаний и семантических связей между терминами;

- образование, с целью ускорения поиска, тематических классов, имеющихся в ИПС документов.

Указанные проблемы послужили целью исследований в составленной под руководством автора экспериментальной системе поиска правовых документов JURIOS [5.18, 5.37, 5.38]. Настоящая работа обобщает полученный опыт и рассматривает все проблемы в свете принципа размытости информационного поиска. Для моделирования работы ИПС применяется теория размытых множеств и отношений [5.4, 5.39, 5.40], из которой выбирается обладающая специальными свойствами подмножество. До сих пор эта теория применялась для описания некоторых подпроблем информационного поиска, как например для установления свойств размытого тезауруса в [5.35], для описания размытого поиска [5.33]. Размытое индексирование и анализ размытых систем рассматриваются в написанных автором статьях [5.27] и [5.26]. В настоящей работе дается систематическое изложение указанных проблем.

Основной частью описываемой ниже ИПС является множество знаний $\mathcal{R}$ о предметной области. Сюда входят, кроме перечня термов, задаваемые размытыми отношениями тезаурус и таблицы определения синонимов, омонимов и словосочетаний. Во второй главе настоящей работы анализируются проблемы, связанные с ролью тезауруса в автоматическом индексировании, вычислении ответа и в определении дескрипторных классов. Вперед заданными считаются исходные поисковые образы документов, составление которых по текстам документов является содержанием третьей главы.

Четвертая глава посвящается автоматизации составления тезауруса с широкими терминологическими связями. Описывается алгоритм анализа тезауруса системы JURIOS, вариант которого применяется и в Институте правовой информации.

В пятой главе анализируется проблема выделения тематических классов документов. Излагается методика одновременного учета связанности терминов и документов.

I.3. Обзор работ по правовым ИПС

Достаточно полный обзор действующих и проектируемых ИПС приведен в работах [5.I2, 4.7]. Ниже основное внимание выделяется применению весовых коэффициентов, автоматическому индексированию и применению тезауруса в нынешних системах и проектах.

Одной из первых и самых известных систем для поиска правовых документов является созданная в США под руководством Д.Хорти система ASPEN [4.II, 5.6]. По применяемой в системе схеме автоматического индексирования из полных текстов документов отбрасывают несущественные слова, занесенные в специальный список. Оставшиеся слова, приведенные к конкретным грамматическим формам, составляют множество индексов документов. Главным преимуществом этого метода является расширяемость совокупности ключевых слов, недостатком же - трудность автоматического отождествления грамматических форм. Поэтому, данная схема автоматического индексирования (схема Хорти) в основном применяется для языков с простой грамматической структурой, как, например, английский (системы FLITE [4.II], JURIS [5.29, 5.3I]).

Тезаурус в американских системах правовой информации либо вообще не используется, либо используется как метод определения дескрипторных классов на основе синонимности и ассоциативности.

Полезность применения тезауруса в ИПС авторы этих систем не отрицают, и рассматривают включение тезауруса в ИПС как путь дальнейшего развивания системы.

В ИПС западно-европейских стран тезаурус применяется намного чаще и в более широком смысле. Так, в тезаурусах систем IRETI], DOCILIS и SYDON [4.7] наряду с синонимностью и ассоциативностью рассматриваются разные иерархические связи между ключевыми словами. Своеобразное применение тезаурус нашел в бельгийской системе CREDOC [4.7], где в тезаурусе задается перевод ключевого слова на другой язык.

ФРГ стала первой капиталистической страной, где создание правовых ИПС рассматривалось государственной задачей. В 1970 году была создана специальная комиссия по проектированию общегосударственной сети правовых ИПС. Интересно отметить, что в предложенном комиссией проекте [2.3, 2.4] были предусмотрены автоматическое индексирование и применение тезауруса с широкими семантическими связями между терминами.

В социалистических странах автоматизация обработки правовой информации больше всего развита в ГДР. Уже в конце 60-ых годов была создана обширная программа, предусматривающая разработку многоцелевой общегосударственной системы обработки правовой информации (поиск, подготовка к печати, избирательное распространение информации и т.д.). В системе предусмотрено применение тезауруса, но не предвидено автоматическое индексирование. По данным работы [5.12] ручное индексирование займет приблизительно 40% всего объема работ по подготовке документов и, конечно, требует высококвалифицированной рабочей силы.

В Советском Союзе общегосударственная система правовой информации (сеть ИПС) в настоящее время находится в стадии проектирования. Разработано несколько экспериментальных систем, на-

правленных на исследование тех или иных проблем анализа информации (в Институте правовой информации, в АН БССР, в Политехническом институте Кишинева, в Тартуском госуниверситете). Из них только проведенные в Тартуском госуниверситете эксперименты были ориентированы на автоматизирование индексирования и применения тезауруса. Первые эксперименты провели уже в конце 60-ых годов [5.8, 5.17], целью которых было исследование возможностей индексирования текстов на эстонском языке по заданному списку ключевых слов и словосочетаний. В дальнейших экспериментах с ИПС JURIOS [5.18] основное внимание было уделено усовершенствованию индексирования с точки зрения повышения эффективности его реализации на ЭВМ и применения тезауруса. Настоящая работа является теоретическим обобщением опыта, полученного при создании системы JURIOS.

В итоге следует сказать, что автоматическое индексирование пока не обрело общепризнанности и общеприменяемости. Это объясняется тем, что в ностоящее время не существует алгоритма полного анализа текста на естественном языке, переводящего этот текст на некоторый формализованный язык. Крупный специалист Д. Сэлтон пишет по этому поводу: "... существующие алгоритмы анализа текста, которые можно использовать в системах автоматической обработки текстов, не дают результатов, которые оправдывали бы усилия, затраченные на их разработку и внедрение. Полный анализ текста невозможен сейчас, потому что нет еще необходимого для этого понимания структуры языка" [5.20]. Там же указывается, что с другой стороны, применение чисто формальных методов описания содержания текста, дает результаты, сравнимые с результатами ручного индексирования.

Что касается применения тезауруса, то к нынешнему времени не разработана методика его гибкого автоматического применения в

ИПС. Ввиду этого тезаурус применяется в довольно узких целях: либо для определения дескрипторных классов, либо как вспомогательное (ручное) средство при составлении запросов.

I.4. Размытое множество, размытое отношение и операции над ними

Определяем ниже понятия размытого множества и размытого отношения, а также операции над ними [5.4, 5.39, 5.40]. Указываем некоторые свойства операций, являющихся основными в конструируемой в дальнейшем размытой модели ИПС.

Определение I.I. Размытым множеством $A(X)$ на заданном конечном множестве объектов $X=\{x_1, x_2, \cdots, x_n\}$ называется множество пар

$$A(X)=\{x_1 \mid f_A(x_1), x_2 \mid f_A(x_2), \cdots, x_n \mid f_A(x_n)\},$$

где функция принадлежности f_A принимает значения на отрезке $[0,1]$.

Функция f_A характеризует степень принадлежности элементов множества X во множество A. Принято считать, что чем больше значение $f_A(x)$, тем больше степень принадлежности элемента x и наоборот. Если $f_A(x)=1$, то элемент x "без сомнения" входит в A, если же $f_A(x)=0$, то он не входит в A. В дальнейшем, в конкретных примерах элементы с нулевым значением функции принадлежности часто опускаются. Кроме того, вместо термина "функция принадлежности" употребляется и термин "весовая функция", а вместо "степени принадлежности" - "вес".

Пусть заданы размытые множества $A(X)$ и $B(X)$. Равенство и включение этих множеств определяются следующим образом:

$A(X)=B(X)$, если $f_A(x)=f_B(x)\ \forall x\in X$;

$A(X)\subseteq B(X)$, если $f_A(x)\le f_B(x)\ \forall x\in X$.

Объединение, пересечение и разность размытых множеств A и B определяются по формулам:

$C(X)=A(X)\cup B(X)$, где $f_C(x)=\max(f_A(x),f_B(x))$;

$C(X)=A(X)\cap B(X)$, где $f_C(x)=\min(f_A(x),f_B(x))$;

$C(X)=A(X)\setminus B(X)$, где $f_C(x)=\max(0,f_A(x)-f_B(x))$.

В дальнейшем пользуемся еще операцией умножения размытого множества на скаляр $\lambda\in(0,1]$ и понятием мощности, определяемыми следующим образом:

$C(X)=\lambda\cdot A(X)$, где $f_C(x)=\lambda\cdot f_A(x)$;

$$|A(X)|=\sum_{x\in X} f_A(x).$$

Содержание приведенных операций выясняется в дальнейшем из их конкретных применений.

Определение I.2. Размытым (бинарным) отношением $R(X,Y)$ над заданными множествами $X=\{x_1,x_2,\cdots,x_n\}$ и $Y=\{y_1,y_2,\cdots,y_m\}$ называется размытое множество

$$R(X,Y)=\{(x_i,y_j)\,|\,f_R(x_i,y_j):i=1,\cdots,n;\ j=1,\cdots,m\}$$

на множестве $X\times Y$.

Аналогично тому, как были определены равенство, включение, объединение, пересечение и разность размытых множеств, определяются те же понятия и для размытых отношений.

В частном случае, если $X=Y$, т.е. когда размытое отноше-

ние определено на множестве $X \times X$, можно определить свойства симметричности и рефлексивности: размытое отношение $R(X,X)$ называется симметричным, если при каждом i,j имеем $f_R(x_i,x_j) = f_R(x_j,x_i)$ и рефлексивным, если при каждом i имеет место $f_R(x_i,x_i)=1$.

Кроме указанных выше операций при описании работы ИПС применяем еще произведение размытого множества на размытое отношение и композицию размытых отношений. Эти операции определяем таким образом, чтобы задаваемое ими изменение весов в некоторой степени соответствовало нашим интуитивным представлениям об изменении весов индексов и элементов тезауруса в ИПС. Произведение множества на отношение в дальнейшем является операцией расширения некоторого исходного множества по заданным в отношение связям, а композиция - образованием новых связей между элементами. В качестве композиции в литературе чаще всего применяется либо макс-мин-композиция, либо максимальное алгебраическое произведение [5.34]. В ИПС более подходящим оказывается второй вариант, дающий наиболее характерную картину изменения весов индексов.

Определение I.3. Произведением размытого множества $A(X)$ на размытое отношение $R(X,Y)$, где $X = \{x_1, x_2, \cdots, x_n\}$ и $Y = \{y_1, y_2, \cdots, y_m\}$ называется размытое множество

$$B(Y) = A(X) \cdot R(X,Y) = \{y_j \mid f_B(y_j) : j = 1, \cdots, m\},$$

где

$$f_B(y_j) = \max_{1 \le i \le n} (f_A(x_i) \cdot f_R(x_i, y_j)).$$

Определение I.4. Композицией размытых отношений $R_1(X,Y)$ и $R_2(Y,Z)$, где $X = \{x_1, x_2, \cdots, x_n\}$, $Y = \{y_1, y_2, \cdots, y_m\}$ и $Z = \{z_1, z_2, \cdots, z_k\}$ называется размытое отношение

$$R(X,Z) = R_1(X,Y) \circ R_2(Y,Z) =$$

$$= \{(x_i, z_\ell) \mid f_R(x_i, z_\ell) : i = 1, \cdots, n;\ \ell = 1, \cdots, N\},$$

где

$$f_R(x_i, z_\ell) = \max_{1 \le j \le m} \left(f_{R_1}(x_i, y_j) \cdot f_{R_2}(y_j, z_\ell) \right).$$

Определение композиции естественным образом обобщается на произвольное число компонент. В дальнейшем используется еще обозначение

$$R^{(n)}(X,X) = \underbrace{R(X,X) \circ R(X,X) \circ \cdots \circ R(X,X)}_{n \text{ членов}}.$$

Операции произведения и композиции являются основными для описания размытого информационного процесса. В ходе такого описания нам понадобятся некоторые свойства этих операций. При доказательстве свойств, а также в дальнейшем, используем задание размытого отношения $R(X,Y)$ в виде двумерной матрицы $R = (r_{ij})$, где $r_{ij} = f_R(x_i, x_j)$ и $i = 1, \cdots, n;\ j = 1, \cdots, m$.

<u>Свойство I.I.</u> Для любого размытого множества $A(X)$ и размытых отношений $R_1(X,Y) = (r^1_{ij})$ и $R_2(Y,Z) = (r^2_{je})$ имеет место равенство

$$A(X) \cdot R_1(X,Y) \cdot R_2(Y,Z) = A(X) \cdot (R_1(X,Y) \circ R_2(Y,Z)).$$

<u>Доказательство.</u> Для доказательства приведенного равенства следует показать, что для любого $z_\ell \in Z$ имеет место равенство весов:

$$f_{A \cdot R_1 \cdot R_2}(z_\ell) = f_{A \cdot (R_1 \circ R_2)}(z_\ell),$$

где $f_{A \cdot R_1 \cdot R_2}$ является весовой функцией размытого множест-

ва $A \cdot R_1 \cdot R_2$, а $f_{A\cdot(R_1 \circ R_2)}$ - множества $A \cdot (R_1 \circ R_2)$

По определению I.3 получаем

$$f_{A\cdot R_1 \cdot R_2}(z_l) = \max_{1\le j\le m}\left(f_{A\cdot R_1}(y_j)\cdot r^2_{jl}\right) =$$

$$= \max_{1\le j\le m}\left(\max_{1\le i\le n}\left(f_A(x_i)\cdot r^1_{ij}\right)\cdot r^2_{jl}\right) =$$

$$= \max_{1\le i\le n}\left(\max_{1\le j\le m}\left(f_A(x_i)\cdot r^1_{ij}\cdot r^2_{jl}\right)\right)$$

так как ввиду неотрицательности чисел $f_A(x_i)\ r^1_{ij}$ и r^2_{jl} допустимо изменение порядка взятия максимума.

С другой стороны, обозначая $R_1 \circ R_2 = R_{12} = (r^{12}_{il})$ получаем из определений I.3 и I.4

$$f_{A\cdot(R_1\circ R_2)}(z_l) = \max_{1\le i\le n}\left(f_A(x_i)\cdot r^{12}_{il}\right) =$$

$$= \max_{1\le i\le n}\left(f_A(x_i)\cdot \max_{1\le j\le m}\left(r^1_{ij}\cdot r^2_{jl}\right)\right) =$$

$$= \max_{1\le i\le n}\left(\max_{1\le j\le m}\left(f_A(x_i)\cdot r^1_{ij}\cdot r^2_{jl}\right)\right),$$

что и доказывает свойство I.I.

<u>Свойство I.2.</u> Для любого размытого множества $A(X)$ и размытых отношений $R(X,Y)=(r_{ij})$ и $R_1(X,Y)=(r^1_{ij})$ имеет место равенство

$$A(X)\cdot(R(X,Y)\cup R_1(X,Y)) = (A(X)\cdot R(X,Y))\cup(A(X)\cdot R_1(X,Y)).$$

<u>Доказательство.</u> Равенство соответствующих весовых функций доказывается непосредственно:

$$f_{A\cdot(R\cup R_1)}(y_j) = \max_{1\le i\le n}\left(f_A(x_i)\cdot \max(r_{ij}, r^1_{ij})\right) =$$

$$= \max_{1\le i\le n} \left(\max \left(f_A(x_i)\cdot r_{ij}, f_A(x_i)\cdot r'_{ij} \right) \right) =$$

$$= \max \left(\max_{1\le i\le n} \left(f_A(x_i)\cdot r_{ij}, f_A(x_i)\cdot r'_{ij} \right) \right) =$$

$$= f_{(A\cdot R)\cup(A\cdot R_1)}(y_j).$$

<u>Свойство I.3.</u> Композиция $R_1(X,X)\circ R_2(X,X)$ рефлексивных размытых отношений является рефлексивной.

Доказательство этого свойства непосредственно вытекает из определений рефлексивности и композиции.

<u>Свойство I.4.</u> Если $R(X,X)$ рефлексивное размытое отношение, то для любого размытого множества $A(X)$ имеет место

$$A(X)\subseteq A(X)\cdot R(X,X).$$

<u>Доказательство.</u> Так как ввиду рефлексивности отношения $R(X,X)$ при любом $i=1,\dots n$ имеем $r_{ii}=1$, то

$$f_{A\cdot R}(x_i) = \max_{1\le k\le n} \left(f_A(x_k)\cdot r_{ki} \right) =$$

$$= \max \left(f_A(x_1)\cdot r_{1i}, \dots, f_A(x_i)\cdot r_{ii}, \dots, f_A(x_n)\cdot r_{ni} \right) \ge$$

$$\ge f_A(x_i),$$

что и доказывает свойство I.4.

<u>Свойство I.5.</u> Если $R(X,X)$ рефлексивное размытое отношение, то

$$R(X,X)\subseteq R^{(2)}(X,X).$$

<u>Доказательство.</u> Для любого элемента $r^{(2)}_{ij}$ отношения $R^{(2)}(X,X)=R(X,X)\circ R(X,X)$ ввиду рефлексивности имеет место неравенство

$$r^{(2)}_{ij} = \max_{1\le k\le n}(r_{ik}\cdot r_{kj}) \geqslant r_{ij}\cdot r_{jj} = r_{ij}\cdot 1 = r_{ij},$$

что и доказывает свойство I.5.

Свойство I.6. Композиция симметричных отношений симметрична.

Доказательство этого свойства легко получается из определения композиции.

Для формулировки следующего свойства введем понятие пути. Скажем, что последовательность неравных между собой элементов $x_{l_0}, x_{l_1}, \cdots, x_{l_k}$ образует в $R(X,X)$ путь из x_{l_0} в x_{l_k}, если $\mu_R(x_{l_i}, x_{l_{i+1}}) \neq 0$ для $i = 0, 1, \cdots, k-1$. Величину k называем длиной пути. По-другому, между x_{l_0} и x_{l_k} существует в $R(X,X)$ путь, если найдутся неравные между собой индексы $l_0, l_1, \cdots, l_k$ такие, что

$$r_{l_0 l_1}\cdot r_{l_1 l_2}\cdot \cdots \cdot r_{l_{k-1} l_k} \neq 0.$$

Свойство I.7. Если $R(X,X)$ рефлексивное размытое отношение, то найдется значение s такое, что $R^{(s)} = R^{(s+1)} = \cdots$. При этом s является максимальной длиной пути в $R(X,X)$.

Доказательство. Существование максимальной длины s очевидно. Для доказательства свойства I.7 следует показать, что $r^{(s)}_{ij} = r^{(m)}_{ij}$ для любого $i,j = 1,2,\cdots,n$ и $m > s$.

Так как R полагается рефлексивным, то по свойству I.3 рефлексивным является и отношение $R^{(s)}$, а тогда по I.5 $R^{(s)} \subseteq R^{(s+1)} \subseteq \cdots$ и следовательно $r^{(s)}_{ij} \le r^{(s+1)}_{ij} \le \cdots$. Таким образом, следует показать, что $r^{(m)}_{ij}$, где $m > s$, не может быть больше $r^{(s)}_{ij}$.

По определению композиции любой элемент $r^{(m)}_{ij}$, где $m = 2, 3, \cdots$ выражается произведением m элементов из R:

$$r_{ij}^{(m)} = \max_{l_1}(r_{il_1}^{(m-1)} \cdot r_{l_1 j}) = \dots =$$

$$= \max_{l_1, l_2, \dots, l_{m-1}} (r_{il_{m-1}} \cdot r_{l_{m-1}l_{m-2}} \cdot \dots \cdot r_{l_1 j}),$$

где $1 \le l_1, l_2, \dots, l_{m-1} \le n$.

Допустим, что $m > \Lambda$ и выбираем некоторый элемент $r_{ij}^{(m)} \neq 0$ Его значение определено некоторой последовательностью индексов $M = (i, l_1, l_2, \dots, l_{m-1}, j)$, при которых достигается максимальное значение произведения $r_{il_1} \cdot r_{l_1 l_2} \cdot \dots \cdot r_{l_{m-1} j}$. Так как максимальной длиной пути в R является Λ, то среди индексов M должны быть равные между собой. Пусть $l_k = l_{k+p}$. В таком случае можно полагать, что $l_k = l_{k+1} = \dots = l_{k+p}$, ибо значение $r_{ij}^{(m)}$ из этого не может меняться. Действительно, ввиду максимальности любой, отличный от M, выбор индексов не может увеличивать значение $r_{ij}^{(m)}$, а с другой стороны ввиду рефлексивности $r_{l_k l_k} = 1$ и того, что $r_{ij} \le 1$ $(i, j = 1, \dots, n)$ значение $r_{ij}^{(m)}$ не может уменьшаться. Следовательно, значение $r_{ij}^{(m)}$ запишется произведением

$$r_{ij}^{(m)} = r_{il_1} \cdot r_{l_1 l_2} \cdot \dots \cdot r_{l_{m-1} j} =$$

$$= r_{il_1} \cdot r_{l_1 l_2} \cdot \dots \cdot r_{l_{m-p-2} j} = r_{ij}^{(m-p-1)}.$$

Если теперь $m - p - 1 \le \Lambda$, то ввиду свойства I.5 $r_{ij}^{(m)} = r_{ij}^{(m-p-1)} \le r^{(\Lambda)}$ и свойство I.7 доказано. Если однако $m - p - 1 > \Lambda$, то описанное выше рассуждение можно повторить, пока не найдется p' такое, что $m - p' \le \Lambda$.

Приведенное выше определение произведения размытого множества на размытое отношение дает в прямом виде неудобный для реализации на ЭВМ алгоритм. Поэтому приведем равносильное предыдущему определение произведения, которое формулируем в виде свойства.

Свойство I.8. Произведение размытого множества $A(X)$ на размытое отношение $R(X,Y)$ выражается формулой

$$B(Y)=A(X)\cdot R(X,Y)=\bigcup_{i=1}^{n}(f_A(x_i)\cdot R_i(Y)) \qquad (I.I)$$

где

$$R_i(Y)=\{y_1|r_{i1},y_2|r_{i2},\cdots,y_m|r_{im}\}$$

определяется i-ой строкой матрицы $R=(r_{ij})$.

Доказательство. По определениям объединения и произведения на скаляр размытых множеств вес элемента $y_j\in Y$ в $B(Y)$ по (I.I) запишется в виде

$$f_B(y_j)=\max(f_A(x_1)\cdot r_{1j},f_A(x_2)\cdot r_{2j},\cdots,f_A(x_n)\cdot r_{nj})=$$

$$=\max_{1\le i\le n}(f_A(x_i)\cdot r_{ij}),$$

которое совпадает с определением I.3.

Значит, при вычислении произведения, учитывая свойство I.8, следует использовать лишь те строки матрицы R, для которых

$f_A(x_i)\neq 0$.

2. МОДЕЛИРОВАНИЕ ПРИМЕНЕНИЯ ТЕЗАУРУСА В ИПС

Целью этой главы является определить роль тезауруса в ИПС и разработать метод его автоматического использования для расширения поисковых образов документов и информационных запросов.

Тезаурус полагается заданным в широком смысле этого слова, содержащим разного рода семантические связи между терминами, с указанием степеней существенности их использования. Формально такой тезаурус описывается множеством размытых отношений $R(X,X)=\{R_1(X,X), R_2(X,X), \cdots, R_m(X,X)\}$ заданных на множестве ключевых слов (индексов) $X=\{x_1, x_2, \cdots, x_n\}$. Каждый из $R_j \in R$ задает некоторый вид связей, а этим и аспект расширения исходного ПОД или запроса.

Исходный поисковый образ документа $\mathcal{S}(X)$ и элементарный запрос $q(X)$ считаются заданными размытыми множествами на том же множестве ключевых слов X. Исходный ПОД получается в результате лексического анализа документа (см. глава 3).

В основном разные виды семантических связей между терминами, задаваемыми в тезаурусе, можно рассматривать независимыми. Однако, при некоторых видах, задаваемых т.н. дескрипторные классы, следует учитывать их особое влияние на другие виды отношений.

В результате получается, что процесс расширения исходного поискового образа с помощью тезауруса описывается простой формулой $\mathcal{S}(X)\cdot\overline{R}(X,X)$, где обобщенный поисковый тезаурус $\overline{R}$ составляется с учетом особенностей связей тезауруса $R(X,X)$ и

весов отдельных связей. Показывается, что применение тезауруса по выбранным принципам в индексировании равносильно применению обратного тезауруса в процессе поиска для расширения запроса.

2.I. Роль тезауруса в ИПС

Тип ИПС определяется способом описания вводимых в систему документов, методом составления таких описаний, способом задания информационных запросов и определением смыслового соответствия между запросом и документом. Проанализируем ниже, как существование тезауруса влияет на выбор этих факторов.

Внутрисистемное описание документа называется поисковым образом документа (ПОД), заданным на информационно-поисковом языке (ИПЯ). Процесс перевода содержания документа в ИПЯ называется индексированием документа. ИПЯ имеет свою лексику и грамматику. Обычно лексика задается в виде множества наиболее существенных понятий предметной области, а грамматика - некоторыми правилами, позволяющими выражать основные грамматические связи между понятиями в документе. В дальнейшем множество лексем ИПЯ обозначаем через

$$X = \{x_1, x_2, \cdots, x_n\}$$

а элементы $x \in X$ называем ключевыми словами.

Грамматика ИПЯ должна содействовать более точному описанию содержания документа. Однако, в настоящее время наши знания о структуре естественного языка еще не позволяют точного, формализованного описания содержания текста [5.20]. С другой стороны, использование "примитивной" грамматики, как показывают описанные в [4.8, 5.I9] эксперименты, не дает ожидаемых результатов. Поэтому в дальнейшем проблему формализации грамматики не рассмат-

риваем и взамен этого постараемся сконструировать систему как можно полным индексированием по терминам с учетом терминологических связей.

В самом простом случае, в ИПЯ без грамматики, поисковый образ δ, документа d, записывается множеством содержащихся в d ключевых слов:

$$\delta \equiv \delta(X) = \{x\},$$

где $x \in X$ и x содержится в d (что обозначается через $x \prec d$).

Возможности задания информационного запроса в основном определяются содержанием и структурой ПОД. При описанном выше ПОД информационный запрос, в конечном итоге, задается множеством ключевых слов, на которые наложены условия содержимости в документе. Такие условия называются критерием смыслового соответствия.

Между ключевыми словами, как понятиями естественного языка, существуют содержательные связи, использование которых существенно с точки зрения более полного информационного поиска. Такие связи принято определять в информационно-поисковом тезаурусе. В настоящей работе (информационно-поисковый) тезаурус понимается как конкретный способ задания множества смысловыражающих элементов (слов, словосочетаний) некоторого языка и определения содержательных отношений между ними. Выбор элементов тезауруса и структуры его представления диктуется способом применения тезауруса в ИПС, хотя имеет место и обратное влияние: "Базовые семантические отношения, задаваемые в тезаурусе, в существенной степени определяют метод индексирования документов и критерий смыслового соответствия при поиске. Поэтому, тезаурус оказывается органической частью информационно-поисковой системы, определяя ее семантические возможности." [5.25].

В общем случае в тезаурусе могут быть рассмотрены разные виды семантических связей между понятиями, такие как, например,

синонимность, род-вид, ассоциативность, часть-целое и т.д. Поэтому представляем тезаурус множеством множеств

$$R(X,X)=\{R_1(X,X), R_2(X,X), \cdots, R_m(X,X)\},$$

где $R_j \in R$ задает некоторый конкретный вид семантических связей и анализирует их роль в процессах информационного поиска.

Согласно Ю.А. Шрейдеру [5.25] и В.Д. Сидоренко [5.16] отношения, входящие в тезаурус, можно делить на синтагматические и парадигматические.

Отношение $R_j \in R$ называется синтагматическим, если при $(x,y) \in R_j$ из того, что ключевые слова x и y входят в один и тот же документ, можно судить о некоторых аспектах этого документа. Такие отношения, позволяющие образовать разные словосочетания и анализировать омонимные значения, будут детально рассмотрены в следующей главе.

Отношение $R_j \in R$ называется парадигматическим, если при $(x,y) \in R_j$ по содержащему понятие x документу d можно судить о документе d', полученном из d заменой в нем x на y.

Для пояснения свойства парадигматичности рассмотрим простой пример. Пусть в некотором документе d речь идет о финансировании учебных заведений и понятие "учебное заведение" является индексом этого документа. Так как разновидностью учебного заведения является университет (отношение род-вид), то d касается и университетов, хотя слово "университет" в нем, может быть, ни разу не упоминается. С точки зрения поиска документ d должен быть ответом как на запрос "учебное заведение", так и на запрос "университет". Для достижения этого, поисковый образ документа d должен содержать наряду с индексом "учебное заведение" и индекс "университет". Отношение "учебное заведение - университет" задается в тезаурусе.

Такого рода расширение некоторого исходного, полученного непосредственным анализом текста, поискового образа δ документа d данными из тезауруса ниже называем применением тезауруса на ПОД и обозначаем $\delta \cdot R$. Однако, даже в таком простом примере было допущено искажение информации. Именно, по дополненному ПОД нельзя узнать, что индекс "университет" не содержится непосредственно в исходном тексте документа, а является добавленным на основе знаний о предметной области. Такое различение существенно, ведь по сравнению с документами, непосредственно посвященными финансированию университетов, упомянутый документ следует считать "второстепенным". Выходом из этого положения может служить применение веса, сопровождающегося слова "университет" в поисковом образе документа d и влияющем на значимость выдаваемого документа. Правила придания веса определяются способом применения тезауруса на исходный ПОД и весами элементов тезауруса.

Рассмотрим более сложный пример, где учитываются несколько видов отношений на некоторых уровнях иерархии.

Допустим, что на множестве ключевых слов $Y = \{y_1$ (медицинское учреждение), y_2(больница), y_3(поликлиника), y_4(врачебная помощь), y_5(скорая медицинская помощь), y_6(больница скорой помощи)$\}$ задан тезаурус $R = \{R_1, R_2\}$, изображенный на рис. 2.1 в виде графа, где R_1 - отношение род-вид и R_2 - ассоциативность.

Пусть термин "медицинское учреждение" (ключевое слово y_1) является единственным термином из Y , применяемым в документе d . Тогда исходный ПОД является множеством $\delta = \{y_1\}$.

Для простоты полагаем, что информационный запрос задается лишь в виде одного ключевого слова и ответом на запрос служит документ, поисковый образ которого содержит это слово. Тогда рассматриваемый документ d, с исходным ПОД $\delta = \{y_1\}$, является ответом на запрос $q_1 = y_1$. Полагая парадигматичность отношений

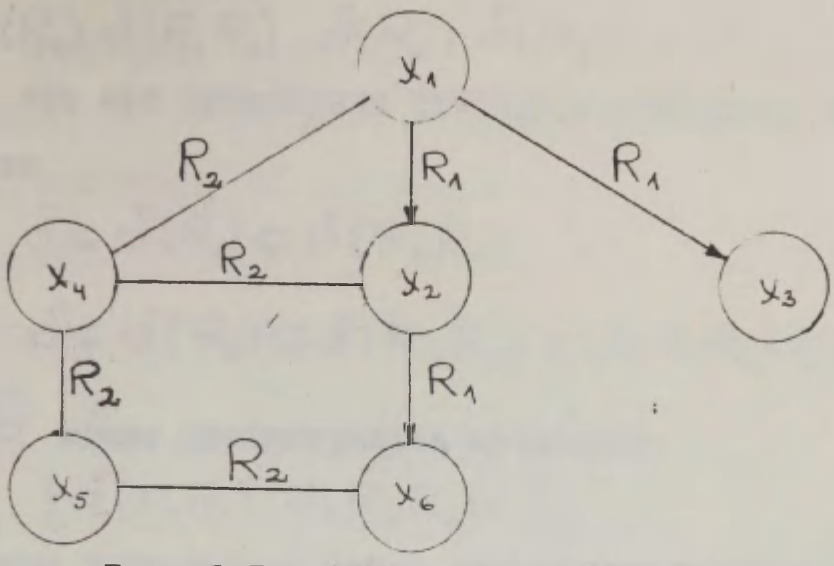

Рис. 2.I. Фрагмент тезауруса.

R_1 и R_2, документ d является ответом и на запросы $q_2 = x_2$, $q_3 = x_3$ и $q_4 = x_4$, так как применяя R_1 и R_2 на δ получаем:

$$\delta(R_1) = \delta \cdot R_1 = \{x_1, x_2, x_3\}$$

и

$$\delta(R_2) = \delta \cdot R_2 = \{x_1, x_4\}$$

Содержательно, после применения тезауруса документ о медицинских учреждениях выдается в ответ и на запросы о больницах, о поликлиниках и о врачебной помощи.

Поисковые образы $\delta(R_1)$ и $\delta(R_2)$ в свою очередь можно расширить с помощью тезауруса, в результате чего получаем поисковые образы

$$\delta(R_1R_1) = \delta(R_1) \cdot R_1 = \{x_1, x_2, x_3, x_6\}$$

и

$$\delta(R_2R_2) = \delta(R_2) \cdot R_2 = \{x_1, x_2, x_4, x_5\}.$$

Теперь документ d реагирует и на запросы $q_5 = x_5$ и $q_6 = x_6$ т.е. является ответом и на запросы о скорой медицинской помощи и о больницах скорой медицинской помощи.

Таким образом, в результате применения тезауруса, получаем множество поисковых образов

$$\bar{\delta}=\{\delta,\delta(R_1),\delta(R_1R_1),\delta(R_2),\delta(R_2R_2),\delta(R_2R_2R_2)\}.$$

Легко видеть, что при применении тезауруса описанным выше способом имеет место

$$\delta\subseteq\delta(R_1)\subseteq\delta(R_1R_1)$$

и

$$\delta\subseteq\delta(R_2)\subseteq\delta(R_2R_2)\subseteq\delta(R_2R_2R_2).$$

Тогда вместо $\bar{\delta}$ можно рассматривать множество

$$\{\delta(R_1R_1),\delta(R_2R_2R_2)\}$$

Недостатком приведенной схемы применения тезауруса является то, что в расширенных ПОД $\delta(R_1R_1)$ и $\delta(R_2R_2R_2)$ индексы считаются равноценными. Не учитывается, что они отличаются друг от друга с точки зрения их получения - своей биографией. Учитывание биографии индексов существенно при трактовке ответа, ведь любой ответ в ИПС осмыслим только с точки зрения логики его получения. Так, например, документ d является ответом на запрос q_1 потому, что ключевое слово x_1 содержится в документе; ответом на q_6 потому, что имеется последовательность утверждений: $x_1 \perp d$, $(x_1,x_2)\in R_1$ и $(x_2,x_6)\in R_1$. Последовательность утверждений можно рассматривать правилом порождения индекса x_6, сопоставляющим документ d на запрос q_6 и характеризующим это соответствие. В общем случае можно полагать, что чем сложнее применяемое правило, в тем меньшей мере соответствует документ запросу. Пользователю ИПС трудно разобраться в сложных правилах вывода, чтобы дать предпочтение тому или другому выданному в ответ документу. Поэтому требуется, чтобы значимость каждого индекса в ПОД оценивалась некоторым числом-весом, зависящим от применяемого правила его порождения. А вес индекса принимается в учет при оценивании значимостей документов в ответе.

Пусть в рассмотренном выше примере задано, что вес индекса

x_1 в δ равняется единице (обозначим это через $\delta = \{x_1 | 1\}$) и применение отношения R_1 уменьшает вес добавляемого индекса на I/2, а отношения R_2 - на I/3. Тогда поисковые образы $\delta(R_1R_1)$ и $\delta(R_2R_2R_2)$ принимают следующий вид:

$$\delta(R_1R_1) = \{x_1|1, x_2|0{,}5, x_3|0{,}5, x_6|0{,}25\}$$

и

$$\delta(R_2R_2R_2) = \{x_1|1, x_2|0{,}11, x_4|0{,}33, x_5|0{,}11, x_6|0{,}03\}.$$

Теперь вес выдаваемого в ответ документа зависит от применяемого варианта поискового образа. Так, например, используя родо-видовые отношения (поисковый образ $\delta(R_1R_1)$), документ будет соответствовать запросу $q_2 = x_2$ (больница) весом 0,5 , а при ассоциативности весом 0,II. В общем случае "рядовому" пользователю трудно будет разобраться в тонкостях отдельных видов связей. Поэтому, весы связей должны быть определены составителями системы таким образом, чтобы они отражали важность каждой из них с точки зрения информационного поиска, а пользователю выдается максимальная оценка. Таким образом, вместо $\delta(R_1R_1)$ и $\delta(R_2R_2R_2)$ можно составить лишь один поисковый образ

$$\delta(R) = \{x_1|1, x_2|0{,}5, x_3|0{,}5, x_4|0{,}33, x_5|0{,}11, x_6|0{,}25\}.$$

Более тонкое применение тезауруса с разрешением выбора элементов отношения рассматриваем в параграфе 2.4, где тезаурус используется для расширения запроса.

Выше был рассмотрен случай, когда один документ соответствует нескольким запросам. Легко видеть, что введенные весы будут характеризовать и соответствие документов из некоторого множества на один запрос.

Приведенные в настоящем параграфе принципы применения тезауруса в ИПС в последующих параграфах излагаются в строгой математической форме, где обсуждаются и вопросы зависимости отношений и

возможности применения тезауруса в процессе поиска.

2.2. Расширение исходного ПОД с помощью тезауруса

Пусть задано множество ключевых слов $X=\{x_1, x_2, \cdots, x_n\}$ где ключевым словом является либо однословное понятие предметной области, либо словосочетание, либо специальное слово, обозначающее значение омонима. На множестве X определяем размытым множеством исходный ПОД и на $X \times X$ - размытый тезаурус.

Определение 2.1. Размытое множество

$$\delta \equiv \delta(X) = \{x_i \mid f(x_i) : i = 1, \cdots, n\}$$

называется исходным ПОД d, если $f(x_i) \neq 0$ тогда и только тогда, когда $x_i \prec d$. Если для некоторого $x \in X$ имеет место $f(x) \neq 0$, то x называется (исходным) индексом документа d.

Если в 2.1 отличный от единицы вес индекса получился в результате применения тезауруса, то тут мы идем немного дальше и допустим, что уже по тексту документа можно судить о значимости индексов. Это вполне естественное расширение, так как ключевые слова имеют в тексте документа различное значение для поиска информации. Проблема составления исходного ПОД будет рассмотрена в главе 3, тут допустим, что существует процедура $P(X, d)$ которая по тексту документа определяет все вхождения ключевых слов и придает им весы.

По определению исходный ПОД содержит и ключевые слова с нулевым весом, однако в конкретных примерах такие слова опускаем.

Определение 2.2. Множество размытых отношений $R(X, X) = \{R_1(X, X), R_2(X, X), \cdots, R_m(X, X)\}$, где

$$R_k(X, X) = \{(x_i, x_j) \mid r_k(x_i, x_j) : i, j = 1, \cdots, n\}$$

и $k=1,\cdots,m$ называем (размытым) тезаурусом.

Содержательно размытое отношение $R_k(X,X)\in R(X,X)$ соответствует некоторому виду связей между ключевыми словами. В общем случае два отношения R_k и R_ℓ могут определить и связь между теми же ключевыми словами. Число таких общих элементов зависит от корректности терминологии и от семантики видов.

Имея в виду дальнейшее применение тезауруса, требуем, чтобы отношения $R_1, R_2, \cdots, R_m$ были бы рефлексивными, т.е. для любого $i=1,\cdots,n$ и $k=1,\cdots,m$ вес $r_k(x_i,x_i)=1$. Это требование не налагает содержательных ограничений на тезаурус и следуя дальнейшему изложению означает, что при расширении ПОД в нем сохраняются индексы по меньшей мере с теми же весами.

Анализ свойств размытого тезауруса приведен Л. Ризингером в [5.35], в настоящей работе основное внимание уделяется автоматизированию применения тезауруса.

<u>Определение 2.3.</u> Размытое множество $\delta^k(R_j)$ называется расширением k-ой степени исходного поискового образа δ , если

$$\delta^k(R_j)=\delta^{k-1}(R_j)\cdot R_j,$$

где $k=1,2,\cdots$ и $\delta^0(R_j)=\delta$. Ключевое слово $x\in X$ называется индексом в $\delta^k(R_j)$, если соответствующий ему вес отличен от нуля.

По приведенному определению, используя отношение $R_j\in R$ можно найти последовательность поисковых образов $\delta^1(R_j), \delta^2(R_j),\cdots$ для которых по свойству I.4 имеет место

$$\delta\subseteq\delta^1(R_j)\subseteq\delta^2(R_j)\subseteq\cdots.$$

Используя свойство I.I, поисковый образ $\delta^k(R_j)$ можно найти по формуле

$$\delta^{n}(R_j) = \delta \cdot (R_j \circ R_j \circ \dots \circ R_j) = \delta \cdot R_j^{(n)}.$$

Из этого следует, что процесс расширения исходного поискового образа с помощью отношения R_j является конечным. Действительно, по свойству I.7 для рефлексивного размытого отношения R_j найдется значение λ_j такое, что $R_j^{(\lambda_j)} = R_j^{(\lambda_j+1)} = \dots$, а этим и $\delta \cdot R_j^{(\lambda_j)} = \delta \cdot R_j^{(\lambda_j+1)} = \dots$. Таким образом, поисковый образ $\delta^{\lambda_j}(R_j)$ является максимальным расширением исходного ПОД с помощью отношения R_j. В дальнейшем $\delta^{\lambda_j}(R_j)$ обозначаем через $\bar{\delta}(R_j)$.

Ввиду того, что $\delta \subseteq \delta^{1}(R_j) \subseteq \dots \subseteq \bar{\delta}(R_j)$, в ИПС можно ограничится лишь максимально расширенным поисковым образом $\bar{\delta}(R_j)$. А в таком случае в ИПС вместо первоначально заданного отношения R_j можно рассматривать его размытое транзитивное замыкание $R_j^{(\lambda_j)}$. В дальнейшем $R_j^{(\lambda_j)}$ обозначаем через $\bar{R}_j$.

Приведенный процесс расширения ПОД по отношению к R_j характеризуется следующими свойствами.

I^{0} Вес исходного индекса при расширении ПОД не может уменьшаться и возрастает лишь тогда, когда найдется ключевое слово $y \in \delta$ такое, что

$$f(y) \cdot \bar{r}_j(y, x) > f(x).$$

2^{0} В ПОД прибавляется новый индекс $y \in X$ тогда и только тогда, когда найдется $x \in \delta$ такое, что $f(x) \neq 0$ и $\bar{r}_j(x, y) \neq 0$.

3^{0} Вес прибавленного индекса зависит от веса вызывающего его индекса, от веса связи и от определения произведения размытого множества на размытое отношение.

Выше расширение исходного ПОД было определено одним видом связей, заданным в тезаурусе. Применяя все виды отношений $R_1, R_2, \dots, R_m$ получаем максимальные расширения ПОД: $\bar{\delta}(R_1), \bar{\delta}(R_2), \dots, \bar{\delta}(R_m)$. Составление общего, многоаспектного

поискового образа документа возможно лишь в том случае, если весы в тезаурусе выбраны сравнивая полезности их применения в информационном поиске и весы элементов разных видов отношений с этой точки зрения сравнимы между собой.

Определение 2.4. Обобщенным поисковым образом документа называется размытое множество

$$\bar{\partial}(R) = \lambda_1 \cdot \bar{\partial}(R_1) \cup \lambda_2 \bar{\partial}(R_2) \cup \cdots \cup \lambda_m \cdot \bar{\partial}(R_m),$$

где $\lambda_1, \lambda_2, \cdots, \lambda_m \in (0, 1]$.

Коэффициент λ_j определяет существенность применения максимально расширенного поискового образа $\bar{\partial}(R_j)$ при поиске по сравнению с другими расширениями. Легко проверить, что $\lambda_j \cdot \bar{\partial}(R_j) = \partial(\lambda_j \cdot \bar{R}_j)$, а тогда по свойству I.2 обобщенный поисковый образ записывается в виде:

$$\bar{\partial}(R) = \partial \cdot (\lambda_1 \cdot \bar{R}_1 \cup \lambda_2 \cdot \bar{R}_2 \cup \cdots \cup \lambda_m \cdot \bar{R}_m).$$

Сумму $\bar{R} = \lambda_1 \cdot R_1 \cup \lambda_2 \cdot R_2 \cup \cdots \cup \lambda_m \cdot R_m$ называем обобщенным поисковым тезаурусом, а тогда $\bar{\partial}(R)$ записывается произведением

$$\bar{\partial}(R) = \partial \cdot \bar{R}. \qquad (2.1)$$

Иллюстрируем процесс расширения исходного ПОД на простом примере.

Пусть на множестве ключевых слов $Y = \{ y_1$ (пенсия), y_2 (пенсия по старости), y_3 (пенсия по инвалидности), y_4 (пенсионер), y_5 (пенсионный документ), y_6 (пенсионная книжка) $\}$ задан тезаурус $R = \{R_1, R_2\}$ в изображенном на рис. 2.2 виде, где R_1 - родо-видовое отношение, а R_2 - ассоциативность. Учитывая приведенный в Y порядок ключевых слов, отношения $\bar{R}_1$ и $\bar{R}_2$ можно написать матрицами:

$$\bar{R}_1 = \begin{pmatrix} I & 0,8 & 0,8 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{pmatrix}$$

и

$$\bar{R}_2 = \begin{pmatrix} I & 0,54 & 0,54 & 0,9 & 0,8I & 0,73 \\ 0,54 & I & 0,36 & 0,6 & 0,54 & 0,49 \\ 0,54 & 0,36 & I & 0,6 & 0,54 & 0,49 \\ 0,9 & 0,6 & 0,6 & I & 0,9 & 0,8I \\ 0,8I & 0,54 & 0,54 & 0,9 & I & 0,9 \\ 0,73 & 0,49 & 0,49 & 0,8I & 0,9 & I \end{pmatrix}$$

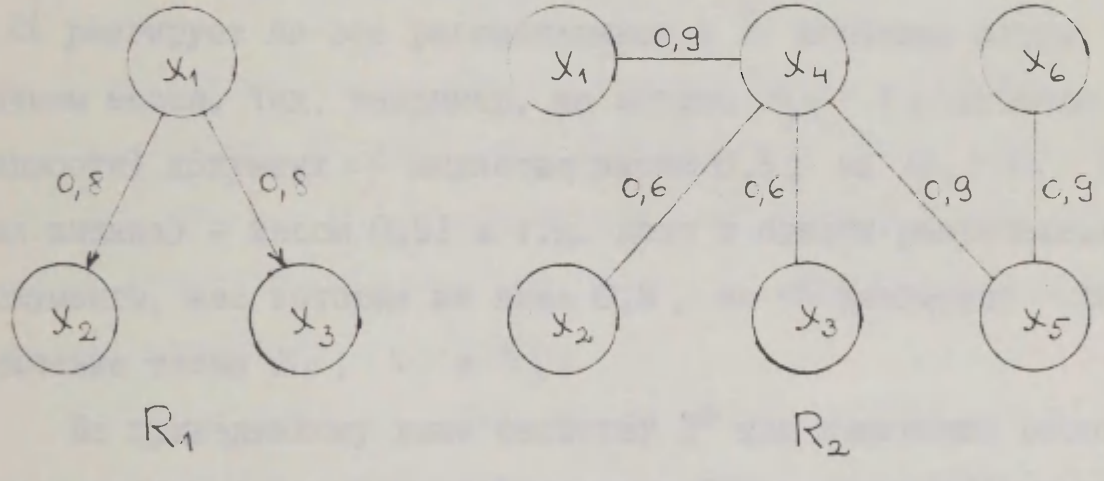

Рис. 2.2. Фрагмент тезауруса; цифры указывают на соответствующие весы (рефлексивность на рисунке не указывается).

Полагая, что ассоциативность является на 0,7 раз важнее чем родо-видовое отношение, т.е. $\lambda_1 = 1$, $\lambda_2 = 0,7$, найдем обобщенный поисковый тезаурус

$$\bar{R} = \bar{R}_1 \cup_{0,7} \cdot \bar{R}_2 = \begin{pmatrix} I & 0,8 & 0,8 & 0,63 & 0,57 & 0,5I \\ 0,38 & I & 0,25 & 0,42 & 0,38 & 0,34 \\ 0,38 & 0,25 & I & 0,42 & 0,38 & 0,34 \\ 0,63 & 0,42 & 0,42 & I & 0,63 & 0,57 \\ 0,57 & 0,38 & 0,38 & 0,63 & I & 0,63 \\ 0,5I & 0,34 & 0,34 & 0,57 & 0,63 & I \end{pmatrix}$$

Пусть теперь для некоторого документа d найден исходный ПОД в виде размытого множества $\mathcal{J} = \{x_1|1,0\ ,\ x_5|0,7\}$, т.е. в документе касается проблема назначения (выплачивания и т.д.) пенсий и в некоторой степени (весом 0,7) и пенсионные документы. Тогда, применяя информационно-поисковый тезаурус $\bar{R}$, получаем обобщенный ПОД:

$$\bar{\mathcal{J}}(R) = \{x_1|1,0\ ,\ x_2|0,8\ ,\ x_3|0,8\ ,\ x_4|0,63\ ,\ x_5|0,7\ ,\ x_6|0,51\}$$

и d реагирует на все рассмотренные в X ключевые слова, но с различием весов. Так, например, на запрос $q_3 = x_3$ (пенсия по инвалидности) документ d выдается весом 0,8, на $q_6 = x_6$ (пенсионная книжка) - весом 0,5I и т.д. Если в ответе разрешаются лишь те документы, вес которых не ниже 0,8, то d реагирует только на ключевые слова x_1, x_2 и x_3.

По приведенному выше свойству 3° ход изменения весов индексов в процессе расширения исходного ПОД определяется правилом произведения размытого множества на размытое отношение. Предложенное в этой работе определение не является единственно возможным. Можно найти целый ряд других вариантов, преимущество тому или другому способу дает практика. Например, вес элемента $y_j \in Y$ в определении I.3 можно дать формулой

$$f_B(y_j) = \min\left(1, \sum_{i=1}^{n} (f_A(x_i) \cdot f_R(x_i, x_j))\right),$$

в которой учитывается не только самая сильная связь, но и остальные связи.

Что касается практической реализации расширения ПОД на ЭВМ, то свойство I.8 дает для этого приемлемый способ. Именно, при расширении δ из базы данных следует вызывать только те строки матрицы $\overline{R}$, соответствующие которым ключевые слова являются индексами в δ.

До сих пор мы молча полагали, что заданные в тезаурусе виды отношений $R_1, R_2, \cdots, R_m$ являются независимыми с точки зрения их применения. Однако, в общем случае независимость не имеет места. Причиненные из этого изменения в схеме применения тезауруса рассматриваются в следующем параграфе.

2.3. Составление дескрипторных классов

В предыдущем параграфе было рассмотрено независимое применение видов отношений тезауруса. Оказывается, что в некоторых случаях, при определенном семантическом содержании вида отношения, следует учитывать взаимную зависимость отношений и применять иную схему использования тезауруса.

Приведем пример. Пусть на множестве ключевых слов $\mathcal{X} = \{ x_1$(лечебное учреждение), x_2(курорт), x_3(поликлиника), x_4(республиканская больница), x_5(военный госпиталь), x_6(санатория), x_7(детская санатория), x_8(туберкулезная санатория), x_9(военная больница), x_{10}(республиканская клиника), x_{11}(медицинское учреждение)$\}$ заданы отношение род-вид (R_1) и синонимность (R_0) показанными на рис. 2.3 связями. Для простоты весы связей считаем равными единице и опускаем.

Пусть некоторый документ содержит только ключевое слово $x_1 \in \mathcal{X}$, т.е. $\delta = \{x_1\}$. Расширяя δ по приведенной в 2.2 схе-

ме, получаем:

$$\bar{\delta}(R_0) = \{x_1, x_{11}\} \; ; \quad \bar{\delta}(R_1) = \{x_1, x_2, x_3, x_4, x_5\}$$

и обобщенный ПОД

$$\bar{\delta}(R) = \{x_1, x_2, x_3, x_4, x_5, x_{11}\}.$$

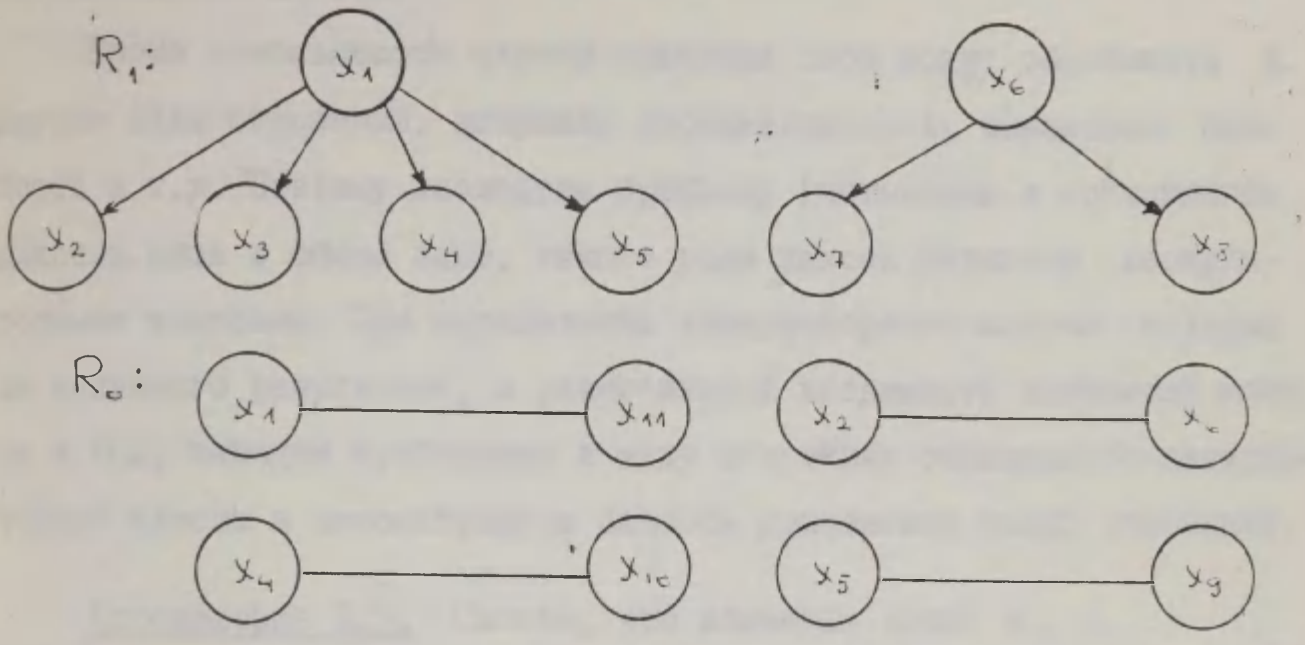

Рис. 2.3. Рассматриваемые отношения R_0 и R_1.

Явно, что полученный ПОД $\bar{\delta}(R)$ является недостаточным, так как анализируемый документ вообще не реагирует на запросы $q_6 = x_6$(санатория), $q_{10} = x_{10}$(республиканская клиника) и так далее, хотя юридический акт, касающийся лечебных учреждений, является действующим и во всех его подразделениях. Причиной полученного противоречия является несоблюдение того факта, что отношение R_0 имеет особое место среди других отношений.

Отношение R_0 в данном случае создает классы синонимных слов: $\{x_1, x_{11}\}$, $\{x_2, x_6\}$, $\{x_4, x_{10}\}$ и $\{x_5, x_9\}$. Если считать элементы одного класса взаимно заменяемыми, то появление ключевого слова в некотором варианте ПОД должно повлечь за собой появление всех синонимных ему слов или же представителя соответствующего класса. Согласно сказанному, полученный выше $\bar{\delta}(R)$

следует расширить ключевыми словами x_6, x_9 и x_{10}, т.е. на $\bar{\delta}(R)$ применить отношение R_0. Однако, если x_6 входит в поисковый образ, можно ли тогда оставить в сторону связи $(x_6, x_7) \in R_1$ и $(x_6, x_8) \in R_1$? По содержанию в данном примере нет, а в общем случае это зависит от того, как определить применение отношения, составляющего классы.

Кроме синонимности классы ключевых слов могут образовать и другие виды отношений, например ассоциативность, смысловая близость и т.д. Поэтому исследуем проблему учитывания и составления классов слов в общем виде, такого рода классы называем дескрипторными классами. При определении дескрипторного класса исходим из желаемого результата, о равносильной входимости элементов класса в ПОД, выводим требования к виду отношения образующего дескрипторные классы и рассматриваем способы применения таких отношений.

Определение 2.5. Скажем, что ключевые слова $x_1, x_2, \cdots, x_t$ образуют на множестве поисковых образов $D = \{\delta_1, \delta_2, \cdots, \delta_n\}$ дескрипторный класс, если для любого $\delta_j \in D$ имеет место

$$f_j(x_1) = f_j(x_2) = \cdots = f_j(x_t).$$

Приведенное определение дескрипторного класса базируется на совместной равносильной встречаемости некоторой совокупности ключевых слов в множестве поисковых образов документов. Если в качестве множества D взять исходные ПОД, то требование равносильной встречаемости является очень строгим и практически невыполняемым. Однако, как правило, совместная встречаемость получается в результате применения определенного, удовлетворяющего некоторым требованиям вида отношения тезауруса. Для установления таких требований докажем сначала следующую теорему.

Теорема 2.1. Если $R_0(X, X) = (r_{ij})$, где $i, j = 1, \cdots, n$ и $X = \{x_1, x_2, \cdots, x_n\}$, рефлексивное размытое отношение, то клю-

чевые слова $x_i, x_j \in X$, при которых $r_{ij} = r_{ji} = 1$, принадлежат на множестве поисковых образов $\bar{\delta}(R_0) = \{ \delta_\ell \cdot \bar{R}_0 : \ell = 1, \cdots, \kappa \}$ к одному дескрипторному классу, где $D = \{ \delta_1, \delta_2, \cdots, \delta_\kappa \}$ множество исходных ПОД и $\bar{R}_0 = (\bar{r}_{ij})$ - транзитивное замыкание отношения R_0.

Доказательство. Покажем сначала, что при любом $t = 1, \cdots, n$ имеет место $\bar{r}_{ti} = \bar{r}_{tj}$ если $r_{ij} = r_{ji} = 1$.

Допустим противное, что при заданных i и j найдется индекс t_1 такой, что $\bar{r}_{t_1 j} > \bar{r}_{t_1 i}$ (или же $\bar{r}_{t_1 i} > \bar{r}_{t_1 j}$). Вычисляем матрицу $R_0^{(s+1)} = (\bar{R}_0 \circ R) = (r_{ij}^{(s+1)})$, где $i, j = 1, \cdots, n$, для которой по определению транзитивного замыкания должно иметь место равенство $R_0^{(s+1)} = \bar{R}_0$. Однако, при заданных индексах i и j , при которых $r_{ij} = r_{ji} = 1$ сделанные предпосылки выведут к неравенствам

$$r_{t_1 i}^{(s+1)} = \max_{1 \le k \le n} (\bar{r}_{t_1 k} \cdot r_{ki}) \geqslant \bar{r}_{t_1 j} > \bar{r}_{t_1 i}$$

и

$$r_{t_1 j}^{(s+1)} = \max_{1 \le k \le n} (\bar{r}_{t_1 k} \cdot r_{kj}) \geqslant \bar{r}_{t_1 i} > \bar{r}_{t_1 j} ,$$

являющимся противоречиями. Отсюда следует, что $\bar{r}_{ti} = \bar{r}_{tj}$ при любом $t = 1, \cdots, n$ для i и j , удовлетворяющим предпосылкам теоремы. Но тогда весы ключевых слов x_i и x_j в $\delta_\ell \cdot \bar{R}_0$ равны между собой, ведь

$$f_{\delta_\ell \cdot \bar{R}_0}(x_i) = \max_{1 \le t \le n} (f_\ell(x_t) \cdot \bar{r}_{ti})$$

и

$$f_{\delta_\ell \cdot \bar{R}_0}(x_j) = \max_{1 \le t \le n} (f_\ell(x_t) \cdot \bar{r}_{tj}) =$$

$$= \max_{1 \le t \le n} (f_\ell(x_t) \cdot \bar{r}_{ti}).$$

А этим теорема 2.I доказана.

Из теоремы следует, что для установления принадлежности ключевых слов в один и тот же дескрипторный класс в определяющем классы виде отношения тезауруса соответствующие весы необходимо приравнивать единице.

Выше, на приведенном примере, было показано, что если в тезаурусе некоторый вид отношения (R_o) является определяющим дескрипторные классы, то при применении всех других видов необходимо учитывать их связь с R_o. Другими словами, надо уточнить, как понимаются остальные виды отношений тезауруса: либо как отношения, задающие связи между дескрипторными классами, либо - между ключевыми словами. Какой трактовкой пользоваться в ИПС, зависит в первую очередь от семантического содержания связей. Если, например, дескрипторный класс рассматривается множеством абсолютно синонимных слов ([4.6], стр. 404), то естественно рассматривать связи между дескрипторными классами. Если же в дескрипторные классы объединены отождествляемые условно в процессе индексирования слова (ассоциативные, близкие по смыслу), то каждое слово имеет свою структуру, ничем не связанную с его принадлежностью к некоторому дескрипторному классу и задаваемые в тезаурусе связи следует трактовать как связи между ключевыми словами.

В обоих случаях полагаем, что элементами тезауруса являются множества бинарных связей между ключевыми словами, а разные трактовки дают разные схемы применения тезауруса.

Иллюстрируем обе возможности трактовки отношений тезауруса на примере. Рассмотрим из приведенного на рис. 2.3 тезауруса только связи между ключевыми словами Y_1, Y_2, Y_6, Y_7, Y_8, Y_{11} и допустим, что R_o задает дескрипторные классы. На рис. 2.4 показаны два варианта применения тезауруса: случай а) - связи рассматриваются между дескрипторными классами и б) - между ключевыми словами.

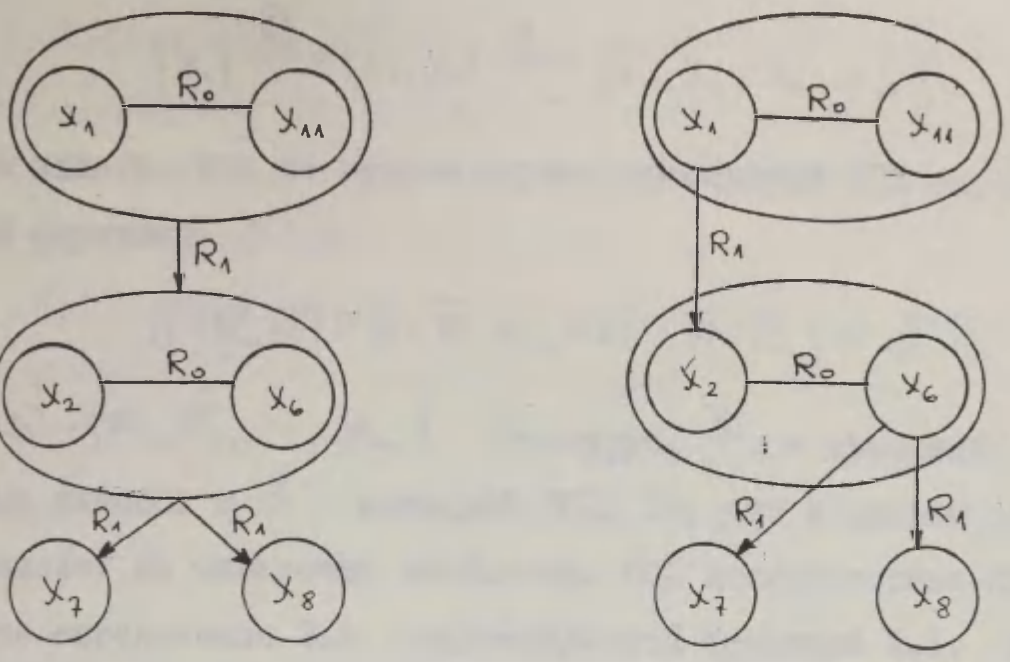

Рис. 2.4. Варианты толкования отношения "дескрипторный класс" относительно других видов отношений.

Пусть теперь исходный ПОД содержит лишь индекс x_1, т.е. $\delta = \{x_1\}$ (весы опускаем). Максимально расширенный поисковый образ $\bar{\delta}(R)$ в первом случае задается множеством

$$\bar{\delta}(R_0 R_1) = \{x_1, x_2, x_6, x_7, x_8, x_{11}\},$$

так как x_1 представляет группу $\{x_1, x_{11}\}$, эта группа по R_1 связана с группой $\{x_2, x_6\}$, вызывающей в свою очередь x_7 и x_8. Отметим, что в данном случае расширение исходного ПОД идет будто бы только по R_1, однако на каждом шагу вместо ключевого слова может появиться определенный отношением R_0 класс.

Во втором случае исходный ПОД расширяется с помощью отношения R_1, а на результат применяется отношение R_0. Таким образом, основную роль играют связи между ключевыми словами, лишь под конец учитывается их принадлежность к дескрипторным классам. В приведенном выше примере получаем последовательность вывода обобщенного поискового образа в следующем виде:

$$\{x_1\} \xrightarrow{\bar{R}_1} \{x_1, x_2\} \xrightarrow{\bar{R}_0} \{x_1, x_2, x_6, x_{11}\}.$$

Легко видеть, что во втором случае обобщенный ПОД определяется общей формулой

$$\bar{\delta}(R_0 R) = \delta \cdot \bar{R} \cdot \bar{R}_0 = \delta \cdot (\bar{R} \circ \bar{R}_0) = \delta \cdot \tilde{R} \qquad (2.2)$$

где $R = \{R_0, R_1, \cdots, R_m\}$ тезаурус, R_0 - отношение дескрипторных классов и δ - исходный ПОД. То, что в данном случае R_0 составляет на множестве обобщенных ПОД дескрипторные классы в смысле определения 2.5, подтверждается теоремой 2.I. Действительно, пусть задано множество исходных ПОД $\{\delta_1, \delta_2, \cdots, \delta_n\}$. Тогда обобщенные поисковые образы $\{\bar{\delta}_t(R_0, R) : t = 1, \cdots, n\}$ согласно формуле (2.2) записываются в виде

$$\bar{\delta}_t(R_0 R) = \delta_t \cdot \bar{R} \cdot \bar{R}_0 = \bar{\delta}_t(R) \cdot \bar{R}_0$$

и так как R_0 полагается удовлетворяющим условиям теоремы 2.I, то на указанном множестве обобщенных ПОД определены дескрипторные классы.

В первом случае на каждом шагу расширения исходного ПОД отношениями $R_0, R_j \in R$, поочередно следует применить оба отношения. Таким образом

$$\delta^{n+1}(R_0 R_j) = \delta^n(R_0 R_j) \cdot R_j \cdot R_0 =$$

$$= \delta^n(R_0 R_j) \cdot (R_j \circ R_0).$$

Обозначая $R_j \circ R_0 = R_j^\circ$ и через $\bar{R}_j^\circ$ соответствующее транзитивное замыкание, то обобщенный поисковый образ $\bar{\delta}(R_0 R)$ запишется в виде суммы

$$\bar{\delta}(R_0 R) = \bigcup_{j=1}^{m} \lambda_j \cdot \delta \cdot R_j^\circ =$$

$$= \delta \cdot (\lambda_1 \cdot \bar{R}_1^\circ \cup \lambda_2 \bar{R}_2^\circ \cup \cdots \cup \lambda_m \bar{R}_m^\circ) = \delta \cdot \tilde{\bar{R}} \qquad (2.3)$$

Легко доказать, что и в этом случае на множестве поисковых образов образуются дескрипторные классы в смысле определения 2.5.

Таким образом, при автоматическом расширении исходных поисковых образов документов с помощью тезауруса $R = \{R_0, R_1, R_2, \dots, R_m\}$ прежде всего, в зависимости от схемы применения составляется обобщенный поисковый тезаурус $\bar{R}$, $\tilde{R}$ или $\tilde{\bar{R}}$ в смысле формул (2.I), (2.2) или (2.3), а сам процесс расширения осуществляется произведением, которое в дальнейшем запишем в виде

$$\bar{\mathcal{S}} = \mathcal{S} \cdot \bar{R}. \qquad (2.4)$$

Какой вариант обобщенного поискового тезауруса выбирать, зависит от содержательных соображений, от семантического содержания тезауруса и от идеологии построения ИПС.

Приведенное определение дескрипторного класса соответствует традиционному применению этого понятия. Однако, в общем случае вместо равносильного вхождения, т.е. равенства весов, можно требовать просто совместного вхождения индексов (весы соответствующих ключевых слов не равняются нулю, но могут быть неравными). Этим получается иное представление о дескрипторных классов. Приведенная выше схема применения остается в силе и в этом случае, лишь в теореме 2.I вместо $r_{ij} = r_{ji}$ следует требовать, чтобы если $r_{ij} \neq 0$, то и $r_{ji} \neq 0$.

2.4. Моделирование процесса применения тезауруса в информационном запросе

Если применение тезауруса при индексировании позволяет расширять ПОД по всем заданным в тезаурусе направлениям, т.е. по всем видам отношений, то применение тезауруса в процессе выработки ответа дает пользователю ИПС возможность самому выбирать

интересующие его направления расширения и определить на его взгляд наиболее существенные отношения. В последующем исследуются формы представления тезауруса и информационного запроса, требуя, чтобы любому расширению запроса соответствовало некоторое расширение ПОД в описанном выше смысле.

Пусть $\mathcal{D}=\{d_1, d_2, \cdots, d_m\}$ является множеством документов и каждому документу $d_j \in \mathcal{D}$ на оонове ключевых слов $X=\{x_1, x_2, \cdots, x_n\}$ сопоставлен в соответствие исходный ПОД

$$\delta_j(X)=\{x_1 | f_{1j}(x_1), x_2 | f_{1j}(x_2), \cdots, x_n | f_{1j}(x_n)\}$$

и расширенный тезаурусом $\bar{R}(X,X)=(\bar{r}_{ij})$ поисковый образ

$$\bar{\delta}_j(X)=\{x_1 | \bar{f}_{1j}(x_1), x_2 | \bar{f}_{1j}(x_2), \cdots, x_n | \bar{f}_{1j}(x_n)\}$$

Исходный поисковый образ документа $d_j \in \mathcal{D}$ можно рассматривать как размытое отношение $\delta(d_j, X)$, определенное на множестве $\{d_j\} \times X$:

$$\delta(d_j, X)=\{(d_j, x_i) | f(d_j, x_i)\},$$

где $i=1, \cdots, n$ и $f(d_j, x_i)=f_{1j}(x_i)$. Множество исходных поисковых образов документов можно записать в виде размытого отношения

$$S_0(\mathcal{D}, X)=\{(d_j, x_i) | s_0(d_j, x_i)\},$$

где $i=1, \cdots, n$; $j=1, \cdots, m$ и $s_0(d_j, x_i)=f_{1j}(x_i)$.

Аналогично, множество расширенных ПОД можно задавать размытым отношением

$$\bar{S}(\mathcal{D}, X)=\{(d_j, x_i) | \bar{s}(d_j, x_i)\}$$

где $\bar{s}(d_j, x_i)=\bar{f}_{1j}(x_i)$.

Размытые отношения $S_0=(s^0_{ji})$ и $\bar{S}=(\bar{s}_{ji})$ являются

теми исходными данными о документах, которые применяются для вычисления ответа на информационный запрос. Применяя отношение S_0 тезаурус в ИПС либо совсем не учитывается, либо должен быть учтен при формировании запроса.

В описании процесса применения тезауруса для расширения информационного запроса введем понятие обратного размытого отношения. С формальной точки зрения обратное некоторому отношению отношение получается транспонированием его матрицы. Обычно обратному отношению можно придавать и содержательный смысл, например обратным к отношению род-вид является отношение вид-род, отношению индекс-документ соответствует документ-индекс и т.д.

Определение 2.6. Обратным для размытого отношения

$$R(X,Y)=\{(x_i,y_j)\mid r(x_i,y_j)\},$$

где $i=1,\dots,n$ и $j=1,\dots,m$ называется размытое отношение

$$R^*(Y,X)=\{(y_j,x_i)\mid r^*(y_j,x_i)\},$$

где $r^*(y_j,x_i)=r(x_i,y_j)$.

Ниже символ "$*$" обозначает как обратное отношение, так и операцию его вычисления:

$$(R(X,Y))^*=R^*(Y,X).$$

С другой стороны, в отношении $\bar{S}$ обобщенный информационно-поисковый тезаурус применен в стадии индексирования документов и в анализе запроса следует учитывать лишь те информационные связи, которые по какой-то причине не включены в тезаурус. Понятно, что наряду с S_0 и $\bar{S}$ можно рассматривать и другие размытые отношения $S_1, S_2, \dots$ полученные в результате применения некоторых под-

частей тезауруса $\bar{R}(X,X)$, однако ниже ограничимся только отношениями S_0 и $\bar{S}$, как своего рода крайними.

Оказывается, что между размытыми отношениями S_0 и $\bar{S}$ имеет место следующая связь:

$$\bar{S}(\mathcal{D},X) = S_0(\mathcal{D},X) \circ \bar{R}(X,X). \qquad (2.5)$$

Действительно, по определению композиции и по принципу применения тезауруса на исходный ПОД, элемент $\bar{s}_{ji}$ выражается в виде

$$\bar{s}_{ji} = \max_{1 \le k \le n} (s^0_{jk} \cdot \bar{r}_{ki}) = \max_{1 \le k \le n} (f_j(x_k) \cdot \bar{r}_{ki}) = \bar{f}_j(x_i),$$

что соответствует определению отношения $\bar{S}$.

Для описания процесса выработки ответа используем следующее свойство операции обратного отношения, которое сформулируем в виде теоремы.

<u>Теорема 2.2.</u> Если на множествах $X = \{x_1, x_2, \cdots, x_n\}$, $Y = \{y_1, y_2, \cdots, y_m\}$ и $Z = \{z_1, z_2, \cdots, z_t\}$ определены размытые отношения $R_1(X,Y) = (r^1_{ij})$ и $R_2(Y,Z) = (r^2_{jk})$, где $i = 1, \cdots, n$; $j = 1, \cdots, m$ и $k = 1, \cdots, t$ то

$$(R_1(X,Y) \circ R_2(Y,Z))^* = R_2^*(Z,Y) \circ R_1^*(Y,X),$$

где R_1^* и R_2^* являются обратными к R_1 и R_2 отношениями.

<u>Доказательство.</u> Обозначим

$$\bar{R}(X,Z) = R_1(X,Y) \circ R_2(Y,Z) = (r_{ik})$$

и

$$R^*(Z,X) = R_2^*(Z,Y) \circ R_1^*(Y,X) = (g_{ki})$$

Тогда для доказательства теоремы следует показать, что $r_{ik} = g_{ki}$. Действительно, используя определения композиции размытых множеств и обратного отношения, получаем .

$$\varrho_{ki} = \max_{1 \le j \le m} (\varrho^2_{kj} \cdot \varrho^1_{ji}) = \max_{1 \le j \le m} (r^2_{jk} \cdot r^1_{ij}) =$$

$$= \max_{1 \le j \le m} (r^1_{ij} \cdot r^2_{jk}) = r_{ik},$$

где ϱ^2_{kj} и ϱ^1_{ji} являются соответственно элементами отношений R^*_2 и R^*_1.

Полученное равенство и доказывает теорему.

Информационный запрос задает условия выделения из совокупности документов некоторого подмножества, элементы которого удовлетворяют определенным в запросе критериям. Ниже допустим, что информационный запрос можно формулировать, комбинируя по определенным правилам элементарные запросы и ответ на заданный запрос получается проведением действий над ответами на элементарные запросы. Такое допущение является естественным, так как в самом простом случае элементарным запросом может быть рассмотрено ключевое слово, а ответом на него - множество содержащих это слово документов. Благодаря отсутствию в элементарном запросе сложных условий выделения документов, с ним хорошо связывать автоматическое применение тезауруса.

Как индексированию, так и составлению запросов свойственна размытость. Пользователь ИПС, как правило, не способен выражать свою информационную потребность в строгих категориях. Поэтому, и в информационном запросе естественно использовать размытые множества, позволяющие оценить значимость каждого применяемого ключевого слова. Вес выделяемого в ответ документа зависит от весов слов в запросе и весов соответствующих индексов.

Считаем элементарным запросом любое размытое множество $q(X)$ определенное на множестве ключевых слов X и определяем ответ на него следующим образом.

<u>Определение 2.7.</u> Ответом на элементарный запрос $q(X)$ на-

зывается размытое множество

$$A(\mathfrak{D}) = q(X) \cdot S(X, \mathfrak{D}),$$

где $S(X,\mathfrak{D})$ некоторое размытое отношение между ключевыми словами и документами, называемое базой ответа.

Таким образом в ответ $A(\mathfrak{D})$ входят те и только те документы, которые связаны по отношению к $S(X,\mathfrak{D})$ с некоторым ключевым словом x, входящим в $q(X)$, т.е. $f_q(x) \neq 0$. Вес документа d в множестве $A(\mathfrak{D})$ зависит от веса выделяющего его индекса и от веса соответствующей связи и вычисляется по правилу произведения. Упорядочив ответ по убыванию весов документов, получаем т.н. эшелонированный ответ. Выбором весов в $q(X)$ пользователь может придать предпочтение тому или иному индексу, и тем самым влиять на весы в $A(\mathfrak{D})$.

В определении 2.7 база вычисления ответа $S(X,\mathfrak{D})$ не уточняется. В общем случае базой можно выбирать любое отношение между индексами и документами; выбором базы определяется содержание ответа. Выше были рассмотрены два отношения $S_0(\mathfrak{D},X)$ и $\bar{S}(\mathfrak{D},X)$ связывающие документы и ключевые слова по входимости. Чтобы при выработке ответа был учтен и тезаурус, выбираем базой вычисления ответа отношение $\bar{S}(\mathfrak{D},X)$, вернее обратное ему отношение $\bar{S}^*(X,\mathfrak{D})$. Тогда ответ на запрос $q(X)$ задается в виде

$$A(\mathfrak{D}) = q(X) \cdot \bar{S}^*(X, \mathfrak{D}).$$

Используя формулу (2.5), теорему 2.2 и свойство I.I, ответ может быть выражен на базе исходных ПОД:

$$\begin{aligned} A(\mathfrak{D}) &= q(X) \cdot \bar{S}^*(X, \mathfrak{D}) = \\ &= q(X) \cdot (\bar{R}^*(X,X) \circ S_0^*(X, \mathfrak{D})) = \\ &= q(X) \cdot \bar{R}^*(X,X) \cdot S_0^*(X, \mathfrak{D}). \end{aligned} \qquad (2.6)$$

Полученная формула (2.6) дает принцип применения тезауруса в процессе выработки ответа, соответствующий описанному в параграфе 2.2 его применению для расширения ПОД.

Понятно, что вместо обобщенного поискового тезауруса $\bar{R}(X,X)$ можно выбирать его любой подтезаурус. Таким образом получается, что в процессе информационного поиска применяются только указанные пользователем связи. Такая возможность и дает основу выделить ИПС "с открытым пользованием тезауруса". Ведь по формуле (2.6), если применяется весь тезаурус, с точки зрения пользователя безразлично, применяется ли он либо при индексировании либо при расширении запроса. Такая система удовлетворяет "среднего клиента", желающего получить всестороннюю информацию о интересующем его вопросе, с указанием степеней соответствия. Однако, в специальных исследованиях, например, в исследованиях о согласованности юридических документов, требуется выбор документов с учетом определенных связей между терминами.

Комбинируя элементарные запросы, можно составить более сложные запросы. Вид информационного запроса в целом зависит от конкретной реализации ИПС. Нет существенной разницы между заданиями запросов в размытой и обыкновенной системах.

Определение 2.8.

I. Произведение элементарного запроса на некоторое отношение является элементарным запросом.

2. Элементарный запрос является запросом.

3. Если $q_1(X)$ и $q_2(X)$ являются запросами с ответами $A_1(\mathfrak{D})$ и $A_2(\mathfrak{D})$ на базе $S(\mathfrak{D},X)$, то $(q_1(X) \circ q_2(X))$ является запросом с ответом $A_1(\mathfrak{D}) \circ A_2(\mathfrak{D})$ на базе $S(\mathfrak{D},X)$, где $\circ \in \{\cup, \cap, \setminus\}$.

Ввиду того, что все упомянутые в определении множества и от-

ношения являются размытыми, то определенный таким образом запрос следует рассматривать как размытый запрос. В размытом запросе применение тезауруса связано с расширением элементарного запроса. Все указанные в запросе операции $\cup, \cap$ и $\setminus$, перечень которых в общем случае можно расширять, рассматриваются операциями между ответами на элементарные запросы.

Имея в виду значение знака $\cup$, ответом $A(\mathfrak{D})$ на запрос $q_1(X) \cup q_2(X)$ является множество тех документов, которые содержатся по меньшей мере в $A_1(\mathfrak{D})$ или в $A_2(\mathfrak{D})$. Весом документа d в ответе является максимальный из его весов в $A_1(\mathfrak{D})$ и $A_2(\mathfrak{D})$.

Операция $\cap$ найдет общую часть из соответствующих запросам $q_1(X)$ и $q_2(X)$ множеств документов. Весом выдаваемого документа является минимальный из его весов в $A_1(\mathfrak{D})$ и $A_2(\mathfrak{D})$.

Операция $\setminus$ соответствует применяемой в неразмытых системах операции отрицания: из множества ответных документов исключаются те, которые содержат отрицаемые ключевые слова. Однако, в размытой ИПС согласно определению вычитания, получается более гибкий подход: ответом на $q_1(X) \setminus q_2(X)$ считается множество $A(\mathfrak{D}) = A_1(\mathfrak{D}) \setminus A_2(\mathfrak{D})$, полученное из A_1 уменьшением весов его элементов на соответствие весы из A_2. В предельном случае, если весы документов в A_2 равняются единице, получается обыкновенная трактовка. Выбором весов в q_2 можно повлиять на величину уменьшения весов.

Таким образом, мы получили схему применения тезауруса в процессе выработки ответа, соответствующую принципам применения тезауруса при индексировании. Оба варианта могут быть реализованы в одной ИПС.

3. АВТОМАТИЗАЦИЯ ИНДЕКСИРОВАНИЯ ЛЕКСИКИ

В этой главе рассматриваются проблемы автоматизации лексического анализа документа, т.е. автоматического составления исходного ПОД. Анализируются как теоретические, так и технические аспекты этих проблем. Лексический анализ разделяется на три последовательно выполняемых этапа: выделение термов (однословных понятий), учет словосочетаний и различение значений омонимных слов. Оказывается, что словосочетания и омонимность можно определить размытыми отношениями, а их автоматическое учитывание приводится к операциям между размытыми множествами и отношениями. Это требует составления двух специальных отношений, входящих, как и тезаурус, в состав задаваемых ИПС множеств знаний о предметной области. В случаях, когда по заданным формальным критериям невозможно точно различать значения омонима, документ индексируется по нескольким значениям с присваиванием соответствующих весов.

Принято считать (напр. [4.4]), что обменным документом между ИПС, ввиду трудоемкости индексирования, служит ПОД, чтобы избегать повторного индексирования. Такое утверждение правильно в системах с ручным индексированием, где полные тексты документов в ИПС и не сохраняются. Однако, это противоречит ориентированию ИПС на конкретный тип потребителя, имеющего свой запас индексов, свои специфические знания о предметной области и свои (может быть ограниченные) технические средства. Дублирование индексиро-

вания допустимо и в системах с автоматическим индексированием лишь в случае существования достаточно быстрых алгоритмов индексирования. Главным образом это касается непосредственной работы с исходным текстом документа, т.е. процесса выделения термов. Ниже описывается соответствующая процедура, анализируются возможности организации и сжатия словаря термов в памяти ЭВМ, исходя из специфики индексирования. Показывается, что учитывая с одной стороны статистические закономерности языка и с другой - специфику индексирования, объем словаря термов можно значительно сократить, а этим уменьшать количество обменяемой между накопителями ЭВМ информации.

Описанная методика применялась в системе JURIOS, откуда взяты и приведенные конкретные данные.

3.I. Сущность индексирования лексики

Процедура $P(d, X)$ введенное в 2.2 сопоставляет документ d с его исходным поисковым образом $\delta(X)$ определенным на множестве ключевых слов $X = \{x_1, x_2, \cdots, x_n\}$. Для составления такой процедуры следует решить целый ряд семантических проблем, как отождествление грамматических форм слова, распознавание вхождения словосочетаний и различение значений омонимов. Процесс составления $\delta(X)$ рассматривается как последовательность $\delta(Z) \to \delta(Y) \to \delta(X)$, где $\delta(Z)$ - ПОД, определенное на множестве термов $Z = \{z_1, z_2, \cdots, z_e\}$ и $\delta(Y)$ - на множестве понятий $Y = \{y_1, y_2, \cdots, y_m\}$. Под термом $z \in Z$ понимаем корень слова, соответствующего некоторому индексу или входящего в состав словосочетания. В словарь Y наряду с корнями однословных понятий входят и корни словосочетаний, в словаре X различают и значения омонимных слов.

Все три разновидности поискового образа $\delta(z)$, $\delta(y)$ и $\delta(x)$ рассматриваются размытыми множествами. Из них только $\delta(z)$ получается непосредственным анализом документа. Вес индекса $z_k \prec d$ определяется по некоторым частотным характеристикам; например, можно положить

$$f_z(z_k) = \left(\frac{g(z_k)}{\sum_{j=1}^{\ell} g(z_j)} \right)^{\frac{1}{g(z_k)}},$$

где $g(z_j)$ – частота терма z_j в документе d. Дополнительно можно учитывать и место расположения терма в документе, например, считать более важными термы заглавия [5.1]. Выбор наиболее подходящей функции для определения веса требует дальнейших экспериментов и в настоящей работе не рассматривается. Однако отметим, что критерием выбора функции служит сходство ручного и автоматического индексирования, проведенных по тем же основам.

Переход от поискового образа $\delta(z)$ к ПОД $\delta(y)$ определяется отношением $R(z, y)$, задающим связь словосочетаний с термами. Ниже полагается, что словосочетание $y \in Y$ тогда и только тогда входит в документ d, когда этот документ содержит все составляющие понятия y термы. Вес $f_y(y)$ определяется соответствующей операцией.

Для различения значений омонимного слова с каждым его значением связывается множество характерных для него слов. При этом каждое такое слово сопровождается весом (вероятностью), характеризующим степень его совместной встречаемости с указанным значением. Таким образом определяется размытое отношение $R(y, x)$, применяемое для перехода от $\delta(y)$ к $\delta(x)$.

Так как в настоящей работе словарь ключевых слов полага-

ется заданным, то описываемая система является системой с фиксированной лексикой. Преимуществом таких систем перед системами со свободной лексикой является простота грамматического анализа, что особенно важно для языков со сложной грамматикой (эстонский, русский, литовский и т.д.). По сути дела, сравнивая лишь корни слов, анализ грамматических форм заключается просто в их игнорировании. Такая "примитивная" методика, однако, может дать достаточно хорошие результаты и применяется она в нескольких системах [5.22].

3.2. Распознавание термов в документе

Пусть задано множество термов $Z = \{z_1, z_2, \cdots, z_e\}$ и требуется найти множество содержащихся в документе d термов $\delta(Z)$. Так как слова документа задаются в некоторой грамматической форме и термы определены корнями слов, то вместе равенства следует рассматривать содержимость терма в слове. При этом необходимо учитывать и взаимные содержательные отношения между термами словаря Z, а также свойственные рассматриваемому естественному языку методы составления слов.

Например, если содержимость терма a в слове b $(a \prec b)$, где $a = a_1 a_2 \cdots a_k$ и $b = \beta_1 \beta_2 \cdots \beta_n$, определить как существование индекса $1 \le i \le n$ такого, что $a_1 = \beta_i, a_2 = \beta_{i+1}, \cdots, a_k = \beta_{i+k-1}$, то одно и то же слово текста может порождать несколько индексов. Действительно, если термами выбраны слова $\{$ труд, рудник, сотрудник, ... $\}$, то в слове текста "сотрудника" содержатся все три указанные термы, хотя по смыслу должно иметь место лишь: сотрудник $\prec$ сотрудника.

Полученное в приведенном примере противоречие устраняется правилом: если в некотором слове содержится несколько термов, то индексом считается только самый длинный из них. Легко видеть,

что такой подход требует достаточно полного словаря термов, например, исключение терма "сотрудник" приведет к неправильному индексированию. Выявление всех таких критериев определения содержимости термов в слове (документе) не является целью настоящей работы. Существенно лишь отметить, что от определения входимости прямым образом зависит организация словаря термов и организация поиска в словаре. Ниже приведено определение входимости терма в документе, примененное в системе JURIOS для индексирования текстов на эстонском языке и исходя из этого разрабатываются методы организации словаря и поиска в нем.

Определение 3.I.

I. Скажем, что слово $a = \alpha_1 \alpha_2 \cdots \alpha_k$ содержится в слове $b = \beta_1 \beta_2 \cdots \beta_m$ и обозначаем $a \prec b$, если $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2, \cdots$ и $\alpha_k = \beta_k$.

2. Скажем, что терм $z \in Z = \{z_1, z_2, \cdots, z_c\}$ содержится в документе d ($z \prec d$), если в документе найдется такое слово a, что $z \prec a$ и не найдется отличного от z терма $z' \in Z$ такого, что $z' \prec a$ и $z \prec z'$.

3.2.I. Организация словаря термов.

Методы организации словарей в памяти ЭВМ оцениваются с точки зрения расходования памяти и скоростей выполнения операций поиска, устранения и добавления слов [4.3, 4.5, 4.I0]. Анализируя процесс использования словаря термов в ИПС, заметим, что основной операцией является поиск, добавление и устранение окажутся крайне редкими операциями. Из этого следует, что организация словаря термов в некоторой базе данных должна быть сориентирована на минимальный расход памяти и на быстрый поиск.

Отметим, что поиск на содержимость терма в слове текста документа является нестандартной для баз данных работой, ведь во

всех методах предусматривается поиск на совпадение, т.е. проверяемое слово (ключ) должно совпадать с некоторым словом словаря. Согласно определению 3.I из слова текста отбрасыванием с конца букв составляется множество возможных термов, которые по убыванию длины должны быть проверены на их наличие в словаре. Для ускорения такого процесса воспользуемся двумя правилами:

I° начало слова текста определяет подмножество возможных термов;

2° длину проверяемой части слова определяет сравниваемый терм.

По первому правилу множество термов разлагается на подмножества, каждый из которых определяется первой буквой или же первыми буквами. Тут возникает проблема составления как можно маленьких подгрупп, не перерасходуя при этом памяти. Начинаем со второго аспекта, т.е. от требования минимальности памяти, а проблему о длины подгрупп рассматриваем в 3.2.2. Полагаем: слово (терм) состоит из последовательности битов, т.е. рассматриваем слово в кодированном виде.

Итак, пусть задан словарь термов $Z = \{z_1, z_2, \cdots, z_e\}$ и пусть средней длиной терма является σ битов. Тогда для последовательной записи словаря Z, без каких либо разделителей между термами и без учета байтной структуры памяти, потребуется $M = e \cdot \sigma$ битов. Полученное число M принимаем в дальнейшем за основу для оценивания экономии памяти.

Фиксируем некоторое число χ, задающее длину начал термов в битах и разобъем каждый терм $z \in Z$ на две части: $z = (z^1, z^2)$ где z^1 - начало терма и z^2 - оставшаяся часть (конец терма). После такого разбиения словарь Z распадает на множество начальных частей $Z^1 = \{z^1_\kappa\}$, где $\kappa = 1, \cdots, e_1$; $1 \le e_1 \le e$ и на множество окончаний $Z^2 = \{Z^2_\kappa\}$, где Z^2 является множеством оконча-

ний, соответствующим началу z^1_k и $k = 1, \cdots, l_1$. Понятно, что число элементов в Z_1 не превышает 2^x, т.е. $l_1 \le 2^x$, а элемент z^1_k, как число, удовлетворяет неравенству $0 \le z^1_k \le 2^x$. С каждым $z^1_k \in Z^1$ связываем ссылку на соответствующее ему множество Z^2_k. Обозначим множество ссылок через $A = \{a_0, a_1, \cdots, a_n\}$, где $n = 2^x$ и положим, что a_t является ссылкой на Z^2_k, если такое существует и $t = z^1_k$. Тогда z^1_k можно рассматривать индексом элементов множества A и нет смысла сохранить его значение в памяти.

Таким образом, словарь Z заменяется двумя множествами: A и Z^2. Множество A содержит 2^x элементов с фиксированной длиной $-entier(\log_2 \frac{1}{l})$, вместо которого в дальнейшем применяем грубо $\log_2 l$. Множество $Z^2 = Z^2_1 \cup Z^2_2 \cup \cdots \cup Z^2_{l_1}$ содержит l элементов средней длины $\delta - x$ битов.

Для вычисления объема памяти организованного таким образом словаря допустим, что элементы в Z^2_k записаны подряд, без каких-либо ограничителей. Тогда объем памяти M_1 вычисляется по формуле

$$M_1 = entier(2^x \cdot \log_2 l + (\delta - x) \cdot l).$$

Отсюда легко найти значение переменной x, при котором M_1 является минимальным. Простыми вычислениями можно показать, что упомянутый минимум достигается при

$$x = \log_2 l - \log_2(\ln 2 \cdot \log_2 l).$$

По полученному выражению видно, что искомое значение x зависит только от величины словаря Z и не зависит от средней длины термов. Например, если Z содержит 16 384 термов, то желательно положить $x = 10$ битов. В таком случае, полагая $\delta = 32$, получаем оценки $M = 524\,288$ битов и $M_1 = 374\,798$ битов, т.е. достигается экономия памяти на $\sim 29\%$. Но, учитывая байтовую структуру памяти ЭВМ, вместо $x = 10$ более разумно выбирать

$\chi = 8$, достигая при этом $\sim 24\%$ экономии.

Отметим, что описанная организация словаря термов соответствует индекс-последовательному файлу, где индекс выбран по специальным соображениям. Выбирая слово текста документа, по его началу непосредственно определяется множество проверяемых окончаний - Z^2_κ. Число элементов в каждом множестве Z^2_κ $(\kappa = 1, \cdots, \ell_1)$ зависит от распределения начальных букв термов. При равномерном распределении в словаре, содержащем 16 384 терма, при $\chi = 8$, в каждом Z^2_κ ожидается 64 окончания. Но, если множество Z^2_κ достаточно мало, в нем можно провести последовательный поиск, при котором учитывается приведенное выше правило 2^0.

Применяя стандартный код для кодирования словаря термов, равномерное распределение начал не имеет места. Кроме того, с возрастанием средней длины терма достигаемая экономия уменьшается ($\sigma = 32$ соответствует лишь четырехбуквенным словам в стандартном коде). Описанный выше метод дает существенную пользу только при его совместном применении с перекодированием слов соответственным образом.

3.2.2. Кодирование словаря термов.

Пусть задано множество термов $Z = \{z_1, z_2, \cdots, z_c\}$ и пусть $S = \{s_1, s_2, \cdots, s_m\}$ обозначает множество встречающихся в документе слов. Без ограничения общности можно полагать, что для любого $z \in Z$ найдется $s \in S$ такое, что $z \prec s$, так как в противном случае нет смысла выбирать z в термы. Условно словарь S можно рассматривать в виде $S = Z \cup S_1$, где S_1 состоит из слов, не соответствующих никакому терму, а остальные слова $s \notin S_1$ заменяются соответствующими термами.

Слово $s \in S$ представимо последовательностью букв некоторого альфавита T, т.е. $s = t_1 t_2 \cdots t_n$, где $t_i \in T$.

($i = 1, \cdots, r$) и T содержит $|T| = u$ букв. Кроме того допустим, что любое слово можно дополнить справа специальными символами (пробелами), не подлежащими кодированию. Значит, длину любого слова $s \in S$ считаем равным r.

Требуется найти функцию кодирования K, удовлетворяющую условиям

$$K(z) \neq K(s) \qquad (3.1)$$

при любых $z \in Z$, $s \in S$, где $s \neq z$ и

$$|K(Z)| = \sum_{j=1}^{c} |K(z_j)| = \min,$$

где $|K(z_j)|$ - длина кода слова z_j и $|K(Z)|$ - длина закодированного словаря Z.

Под применением функции кодирования K на слово s понимаем конкатенацию

$$K(s) = K(t_1) K(t_2) \cdots K(t_r). \qquad (3.2)$$

Если положить, что для любой буквы $t_i \in T$ ($i = 1, \cdots, u$) вычислена вероятность ее появления в Z, то функция K определяется схемой Хаффмана [4.12]. Вероятности появления букв в Z можно рассматривать и в зависимости от позиции в слове, т.е. задавать матрицей $P = (p_{ij})$, где $i = 1, \cdots, u$; $j = 1, \cdots, r$ и p_{ij} вероятность того, что слово имеет по меньшей мере j букв и в позиции j встречается буква t_i. Тогда функцию K следует рассматривать последовательностью функций $K = (K_1, K_2, \cdots, K_r)$, а вместо (3.2) применить формулу

$$K(s) = K_1(t_1) K_2(t_2) \cdots K_r(t_r). \qquad (3.3)$$

Так как в (3.3) вероятности появления букв учитываются более

детально, то ожидается и более сильное сжатие словаря Z. Для вычислений функций $K_1, K_2, \ldots, K_n$ опять применима схема Хаффмана. Математическим ожиданием длины терма $z \in Z$ является величина

$$E(|K(z)|) = \sum_{j=1}^{n} \sum_{i=1}^{u} (p_{ij} \cdot |K_j(t_i)|),$$

где $|K_j(t_i)|$ – длина кода i-ой буквы в позиции j.

Описанная выше методика применялась в системе JURIOS, где средней длиной терма из I5 букв получилась 3I бит. Таким образом, словарь из ~I6 000 слов (а I5 букв) составит лишь ~496 000 битов, если не учитывать структуру памяти и разделителей между словами. Кроме того, имея в виду описанный в части 3.2.I метод организации словаря, требуемый объем можно сократить еще примерно на одну четверть. В реальных условиях, учитывая структуру памяти, разделителей между словами, а также некоторую дополнительную информацию, сохраняемую в словаре, объем необходимой памяти, конечно, возрастает. Поэтому возникает вопрос, возможно ли более сильное сжатие словаря и как это осуществить, соблюдая требования индексирования?

Отметим, что до сих пор мы пользовались только частотой появления букв в словаре термов и не учитывали закономерности следования букв в словах. Опираясь на соответствующие статистические исследования текстов на эстонском языке [5.30], а также на данные, приведенные в [5.36] об английском, немецком и испанском языках, такие закономерности могут дать существенную пользу. Однако, выбирая кодируемыми элементами более крупные единицы чем буквы (слоги, диграммы и т.д.), возникают трудности с определением содержимости термов в словах.

В работах [5.2, 5.I0, 5.2I] описывается метод, сущность ко-

торого заключается в том, что, исходя из конкретного словаря, в каждой позиции слов составляют соответствующие одному и тому же коду группы букв таким образом, чтобы сохранилось минимальное расстояние между кодами слов. Основным недостатком этого метода, препятствующим его применению для сжатия словаря термов, является то, что не гарантируется расстояние между термами и нетермами (т.е. словами из S_1).

Идею группировки букв можно использовать и в сочетании с данными о вероятностях распределения букв в разных позициях слов. Поставим задачей найти функцию кодирования K такую, что $|K(Z)| \le N$, где N - величина выделенного для записи словаря Z участка памяти и вероятность $\pi(K(s) = K(z))$ при $s \in S$ и $z \in Z$ был бы минимальным. Если кроме матрицы P полагать известной аналогичную матрицу $\bar{P} = (\bar{p}_{ij})$ для текстов документов, то вероятность π вычисляется по формуле

$$\pi(K(s) = K(z)) = \prod_{j=1}^{n} \sum_{i=1}^{u} (p_{ij} \cdot \bar{p}_{ij}) = \prod_{j=1}^{n} \pi_j . \qquad (3.4)$$

Если вычисленная по схеме Хаффмана функция кодирования в виде (3.3) удовлетворяет условию $|K(Z)| \le N$, то π характеризует вероятность совпадения выбранных случайным образом терма со словом документа и проблема решена. В противном случае воспользуемся предпосылкой, что в словаре Z , в некоторой позиции слов существуют буквы, неразличение которых не приведет к массовому совпадению термов между собой и с нетермами. С такими буквами сопоставим один и тот же код, этим уменьшаем число кодируемых знаков, число различных кодов и в конечном итоге среднюю длину кода. Для выяснения критерия объединения букв допустим, что в позиции j объединены в одну группу буквы t_ℓ и t_κ и считаем представителем группы букву t_ℓ . Тогда таблицы P и $\bar{P}$ превра-

щаются в новые таблицы P' и $\bar{P}'$, в которых $p'_{\kappa j} = \bar{p}'_{\kappa j} = 0$, $p'_{\ell j} = p_{\ell j} + p_{\kappa j}$ и $\bar{p}'_{\ell j} = \bar{p}_{\ell j} + \bar{p}_{\kappa j}$, а остальные значения остаются неизменными.

Далее, по таблице P' можно найти новую функцию кодирования K' и по (3.4) оценку π'. Требуем, чтобы сделанная группировка букв наименьшим образом повлияла на вероятность совпадения кодов, т.е. $\pi' - \pi = \min$. Простой расчет показывает, что индексы j, κ и ℓ следует выбирать из условия

$$p_{\ell j} \cdot \bar{p}_{\kappa j} + p_{\kappa j} \cdot \bar{p}_{\ell j} = \min$$

для любых $j = 1, \cdots, u$; $\kappa, \ell = 1, \cdots, r$; $\kappa \neq \ell$ и естественно $p_{\ell j} \cdot p_{\kappa j} \neq 0$.

Легко показать, что при найденной функции K' средняя длина кода уменьшается: $E(|K'(z)|) < E(|K(z)|)$. Описанный процесс соединения букв в группы повторяется, пока $|K(z)| \leq N$.

В результате такого процесса вычисления функции кодирования каждой позиции j слов словаря термов соответствует свой альфавит T_j, полученный определенной группировкой букв исходного алфавита T. Так как каждой группе (группу может составить и одна буква), соответствует один и тот же код, то "способность различения" кодирующей функции в разных позициях различное и тем больше, чем больше групп в рассматриваемой позиции. Учитывая, что в описанных выше таблицах вероятностей P и $\bar{P}$ ввиду условных вероятностей имеет место

$$\sum_{i=1}^{u} p_{i1} \geq \sum_{i=1}^{u} p_{i2} \geq \cdots \geq \sum_{i=1}^{u} p_{ir}$$

и

$$\sum_{i=1}^{u} \bar{p}_{i1} \geq \sum_{i=1}^{u} \bar{p}_{i2} \geq \cdots \geq \sum_{i=1}^{u} \bar{p}_{ir},$$

то в последних позициях ожидается меньшее число групп чем в первых. Отсюда вытекает, что разработанная описанным выше образом кодирующая функция лучше различает слова с несовпадающими началами.

Конечно, предложенное сжатие словаря термов можно делать в ограниченных пределах, когда допущенные ошибки индексирования компенсируются логикой системы. Для оценки эффективности описанной методики сжатия в системе JURIOS был проведен эксперимент, в ходе которого требовали сжать максимально I5-буквенные термы словаря из I6 000 слов на 20-битовые коды (напомним, что код Хаффмана дал длиной терма 3I бит). В результате такого, достаточно сильного сжатия в позициях I-6 оказалось 9-II групп, в позициях 7-I0 - 5-8 групп и в позициях II-I5 - 2-4 групп. Проверка разработанной функции кодирования на различимость термов между собой дала лишь в II% случаях совпадение кодов. Это указывает на то, что в некоторой степени такой метод сжатия допустим.

3.3. Индексирование понятиями

Вторым этапом индексирования лексики является переход от поискового образа $\delta(Z)$, заданного на множестве термов Z, на поисковый образ $\delta(Y)$, где $Y = \{y_1, y_2, \cdots, y_m\}$ является множеством понятий. Такой переход неминуем в ИПС с автоматическим применением тезауруса, так как тезаурус невозможно задавать на множестве термов. Понятие может быть выражено либо одним словом, либо словосочетанием. Полагаем, что составляющие понятие слова выбраны в термы и $y_j \in Y$ можно записать в виде последовательности $y_j = z^j_1 \sqcup z^j_2 \sqcup \cdots \sqcup z^j_k$. Если в Y, как правило, соблюдается определенная последовательность термов, то в тексте документа, в зависимости от построения предложения, эти

термы могут встречаться в любом порядке и чередоваться с другими словами. Поэтому, при описании процесса индексирования понятиями, любое понятие $y_j \in Y$ целесообразно рассматривать лишь как множество входящих в его состав термов: $y_j = \{z^d_1, z^d_2, \cdots, z^d_k\}$. Кроме того, понятие y_j входит в документ d тогда и только тогда, когда все составляющие его термы входят в одно и то же предложение документа.

Из сказанного выше следует, что при составлении словаря термов надо учитывать и составные части понятий, а в индексировании термами в поисковом образе $\delta(Z)$ надо сохранить группировку термов по предложениям. Поэтому $\delta(Z)$ рассматриваем в виде множества

$$\delta(Z) = \{\delta^1(Z), \delta^2(Z), \cdots, \delta^\Lambda(Z)\},$$

где

$$\delta^t(Z) = \{z_1 | f^t_z(z_1), z_2 | f^t_z(z_2), \cdots, z_e | f^t_z(z_e)\}$$

является поисковым образом предложения и Λ задает число предложений в документе d.

Для перехода от $\delta(Z)$ к $\delta(Y)$ требуется сообщить ИПС составные термы каждого понятия. Это выражается в виде отношения

$$R(Z,Y) = \{(z_k, y_j) | r_{zy}(z_k, y_j)\},$$

где $k = 1, \cdots, \ell$; $j = 1, \cdots, m$ и

$$r_{zy}(z_k, y_j) = \begin{cases} 1, & \text{если } z_k \in y_j \\ 0, & \text{если } z_k \notin y_j. \end{cases}$$

Убедимся теперь, что размытое множество

$$\delta^t(Y) = \{y_j | f^t_y(y_j)\},$$

где $j = 1, \cdots, m$ и

$$f_y^t(y_j) = \min_{1 \le k \le \ell} (\max(f_z^t(z_k), 1 - r_{zy}(z_k, y_j)))$$

является искомым поисковым образом предложения документа d на основе понятий. Для этого надо показать, что $f_y^t(y_j) \neq 0$ тогда и только тогда, когда $f_z^t(z^d) \neq 0$ для всех $z^d \in y_j$.

Пусть $y_j = \{z_1^d, z_2^d, \dots, z_k^d\}$. Тогда по определению отношения $R(Z, Y)$ для любого $i = 1, \dots, k$ имеет место $r_{zy}(z_i^d, y_j) = 1$. Если при этом все z_i^d $(i = 1, \dots, k)$ входят в $\delta^t(Z)$, т.е. $f_z^t(z_i^d) \neq 0$ то

$$f_y^t(y_j) = \min_{1 \le k \le \ell} f_z^t(z_k^d) \neq 0.$$

Если однако некоторый $z_u^d \in y_j$ не входит в $\delta^t(Z)$, т.е. $f_z^t(z_u^d) = 0$, то

$$\max(f_z^t(z_u^d), 1 - r_{zy}(z_u^d, y_j)) = 0$$

и тем самым $f_y^t(y_j) = 0$.

Значит, определенное выше множество $\delta^t(Y)$ действительно можно считать поисковым образом соответствующего предложения. Кроме того отметим, что весом понятия в $\delta^t(Y)$ выбирается наименьший из весов составляющих его термов.

В общем случае множества Z и Y могут пересекаться: $Z \cap Y \neq \emptyset$, т.е. однословные термы могут быть понятиями. Для того, чтобы они сохранились в $\delta^t(Y)$, соответствующие им весы в $R(Z, Y)$ следует приравнивать единице. Переход от термов на понятия, выраженные словосочетаниями, является конкретизацией индексирования. Если наряду с таким понятием индексом документа преднамереваются выбирать и некоторые составляющие его термы, то связь между понятием и термами естественно задавать в тезаурусе и учитывать описанным в главе 2 образом. Так, например, если сло-

ва "пенсия" и "инвалидность" являются термами, совместное вхождение которых в одно предложение вызывает понятие "пенсия по инвалидности", то судя по содержанию, индексом можно выбирать и "пенсия", но вряд ли "инвалидность". Однако, и "пенсия" является уже порожденным от "пенсия по инвалидности" индексом, вес которого следует вычислить, учитывая вес соответствующей связи в тезаурусе.

Поисковый образ $S(Y)$ определяется суммой поисковых образов предложений $S^1(Y) \cup S^2(Y) \cup \cdots \cup S^n(Y)$, однако структура предложений потребуется еще и при учитывании омонимности.

3.4. Индексирование омонимов

Последним, третьим этапом индексирования лексики является различение значений омонимов. Описываем ниже возможности формализации этого процесса в размытой ИПС.

С каждым омонимом легко связывать множество его возможных значений. Например, с омонимом "выписка" сопоставим множества {выписка I, выписка 2, выписка 3}, означающие соответственно выписку человека, выписку газет и выписку из документа. В ручном индексировании индексатор выбирает подходящее значение омонима по содержанию документа. В автоматическом индексировании распространено определение значения омонима по некоторым словам, наиболее часто встречающимся в тексте вместе с рассматриваемым значением [5.22]. Множество связанных со значением омонима понятий, называем его возможным контекстом. Так, например, контекстом слова "выписка 2" может служит множество { газета, журнал, материал, склад и т.д.} . Возможный контекст каждого значения определяется либо статистическими исследованиями, либо составителями тезауруса. Понятно, что слова в возможном контексте неко-

торого значения могут иметь различный вес, характеризующий степень его связанности с указанным значением. Таким образом, возможный контекст уместно задавать размытым множеством.

Ниже не различаем омонимные и неомонимные слова: значением каждого неомонимного слова считаем само это слово, которое и является своим контекстом. Таким образом, все понятия рассматриваются как омонимные и обрабатываются по общей схеме. Процесс различения значений омонимов описывается двухэтапным, где с одной стороны поисковый образ предложения документа дополняется всевозможными значениями входящих в нее понятий, а с другой - лишь понятиями, вызванными контекстом. Общая часть этих двух разновидностей поискового образа предложения и дает искомый поисковый образ.

Пусть $d^t(y)$ обозначает поисковый образ предложения документа. Для любого понятия $y_j \in Y$ определено множество его всевозможных значений на множестве ключевых слов $X = \{x_1, x_2, \dots, x_n\}$, т.е. $y_j = \{x^d_1, x^d_2, \dots, x^d_k\}$. Указанное соответствие можно записать отношением на множестве $Y \times X$. Хотя это отношение в изложенной выше трактовке является неразмытым, имея в виду дальнейшее использование, его целесообразно определить размытым.

<u>Определение 3.2.</u> Размытое отношение

$$R'(Y, X) = \{(y, x) \mid f'_{yx}(y, x)\}$$

где $x \in X$ и $y \in Y$, называется отношением перечисления значений понятий, если $f'_{yx}(y, x) = 1$ лишь тогда, когда x является значением понятия y, а в остальных случаях $f'_{yx}(y, x) = 0$

Аналогично, связь между значениями омонимов и их возможным контекстом задается размытым отношением на множестве $Y \times X$.

Определение 3.3. Размытое отношение

$$R''(y,x)=\{(y,x)\,|\,\ell''_{yx}(y,x)\},$$

где $x\in X$ и $y\in Y$, называется отношением определения ключевых слов, если $\ell''_{yx}(y,x)\neq 0$ тогда, и только тогда, когда y входит в контекст слова x.

Что касается неомонимных понятий, определяемых равенством $y=x$, то для них естественно положить $\ell'_{yx}(y,x)=1$ и $\ell''_{yx}(y,x)=1$ означающие, что неомонимное понятие является своим значением, контекст которого состоит из множества $\{y\}$.

Считаем, что ключевое слово $x\in X$, являющееся значением понятия $y\in Y$ индексирует документ, если наряду с понятием в одном и том же предложении документа содержится и некоторое слово из контекста x. Применяя описанные выше отношения R' и R'', поисковый образ предложения документа запишется в виде

$$\delta^t(x)=\delta^t(y)\cdot R'(y,x)\cap\delta^t(y)\cdot R''(y,x). \qquad (3.5)$$

Легко проверить, что формула (3.5) действительно соответствует приведенному в начале настоящего абзаца принципу.

По поисковым образам предложений можно найти поисковый образ документа. В самом простом случае ПОД является суммой поисковых образов предложений, т.е.

$$\delta(x)=\bigcup_{t=1}^{\Delta}\delta^t(x),$$

однако можно использовать и более сложные функции, учитывающие частоту появления ключевых слов.

Приведем пример использования изложенной методики для различения значений омонимов.

Пусть заданы множество понятий $Y=\{y_1,y_2,\dots,y_5\}$ и множество ключевых слов $X=\{x_1,x_2,\dots,x_7\}$. Допустим, что

на множестве $Y \times X$ задано отношение $R'(Y,X)$ в виде таблицы

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
y_1	I	I	I	0	0	0	0
y_2	0	0	0	I	0	0	0
y_3	0	0	0	0	I	0	0
y_4	0	0	0	0	0	I	0
y_5	0	0	0	0	0	0	I

Отсюда видно, что единственным омонимом является понятие y_1, принимающее одно из значений x_1, x_2 или x_3. Понятия y_2, y_3, y_4 и y_5 не являются омонимами и им соответствуют ключевые слова x_4, x_5, x_6 и x_7.

Далее допустим, что появление понятия y_3 всегда указывает на значение x_1, понятие y_4 с равной вероятностью появляется вместе со значениями x_1 и x_2, а y_5 фиксирует значение x_3. Тогда отношение $R''(Y,X)$ можно записать в виде таблицы

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
y_1	0	0	0	0	0	0	0
y_2	0	0	0	I	0	0	0
y_3	I	0	0	0	I	0	0
y_4	0,5	0,5	0	0	0	I	0
y_5	0	0	I	0	0	0	I

Пусть теперь для некоторого предложения документа на основе множества понятий Y найден поисковый образ

$$S^t(Y) = \{y_1|0,9\ ,\ y_2|0,6\ ,\ y_3|0,8\ ,\ y_4|0,8\}.$$

Тогда его поисковым образом на множестве ключевых слов X по

формуле (3.4) является размытое множество

$$\delta^t(X) = \{x_1|0{,}8\ ,\ x_2|0{,}4\ ,\ x_4|0{,}6\ ,\ x_5|0{,}8\ ,\ x_6|0{,}8\}.$$

В данном случае формальными методами невозможно было определить точное значение омонима y_1, так как в $\delta^t(Y)$ одновременно использовали понятия y_3 и y_4, указывающие одновременно на ключевые слова x_1 и x_2. Поэтому, в $\delta^t(X)$ появились и x_1 и x_2, хотя с существенно различными весами.

4. СОСТАВЛЕНИЕ ИНФОРМАЦИОННО-ПОИСКОВОГО ТЕЗАУРУСА

Тезаурус в ИПС применяется для очень конкретных целей, но его нельзя рассматривать только как рабочий инструмент информационного поиска. Как средство представления знаний в некоторой области, он носит и нормативно-терминологическую функцию [5.24]. Фиксированные в тезаурусе связи между понятиями определяют в значительной степени их содержание и порядок использования в рассматриваемой предметной области. Поэтому, составление тезауруса с широкими семантическими связями между терминами, по сути дела, является самостоятельной проблемой. В настоящей работе не затрачиваются проблемы выбора видов связей и методики ручного составления исходного варианта тезауруса [5.7], а основное внимание уделяется формальному анализу правильности тезауруса на ЭВМ. В таком анализе ЭВМ имеет две задачи:

I) систематизированием терминов определенным образом, учитывая фиксированные связи, делать взаимные связи между терминами более наглядными и тем самым облегчить обнаружение ошибок и недостатков;

2) проверкой соответствия множества заданных связей некоторой структуре, обнаружить структурные конфликты, указывающие на несостоятельность некоторой связи или неоднозначность терминологии.

В этой главе описывается методика автоматического анализа тезауруса и приводятся соответствующие алгоритмы, примененные

для проверки составленного тезауруса. При разработке алгоритмов в первую очередь учитывали два обстоятельства:

I) фиксированное первоначально множество связей между терминами может содержать противоречивые элементы, искажающие пологаемую структуру вида отношения;

2) учитывая объем тезауруса, в алгоритме анализа должны быть предвидены ввод информации по порциям, сборка мусора и методы ускорения процесса поиска.

Отметим, что хотя поставленная задача анализа тезауруса похожа на топологическую сортировку [4.2], но ввиду приведенных обстоятельств она требует иного подхода. В описываемом ниже алгоритме в сторону оставлены проблемы ввода-вывода информации, как зависящие от базы данных, и рассматриваются только проблемы сортировки.

В конце главы приводятся конкретные данные о составленном тезаурусе. Эти данные с одной стороны характеризуют объемы разных видов связей, а с другой - число допущенных человеком при составлении тезауруса ошибок. Так как анализу подвергали объявленный человеком готовый тезаурус, то полученные оценки на число ошибок показывают, насколько анализ на ЭВМ помогает составителям тезауруса.

Учитывая характеристики составленного тезауруса, можно утверждать следующее:

I) учитывая объемы групп связанных между собой терминов, автоматическое расширение ПОД (или запроса) может иметь место только при хорошо разработанной системе весов, противостоящей чрезмерному росту существенных индексов;

2) учитывая различную содержательную роль отдельных видов отношений при информационном поиске, их совместное использование в ИПС предполагает существование математической аппаратуры,

способной сохранить различие между ними; это позволяет предложенная методика использования весов;

3) предложенная методика автоматического анализа тезауруса способствует выявлению ошибок в групных тезаурусах и этим ускоряет процесс их составления.

4.I. Автоматизация составления тезауруса

Ввиду трудоемкости составление тезаурусов стараются автоматизировать. Чаще всего для этого применяется метод дистрибутивно-статистического анализа [5.5, 5.II, 5.23]. В этом методе вычисляется частота совместного вхождения терминов исследуемой области в отрезок текста определенной длины. Таким образом, в тезаурусе рассматривается лишь один вид связей между терминами - совместная встречаемость. Соблюдая изложенное в настоящей работе, это соответствует определению понятий по термам (см. 3.3). Это подтверждается и данной в [5.II] характеристикой: "... метод дистрибутивно-статистического анализа позволяет автоматически получить вводимые в информационно-поисковый тезаурус словосочетания, парадигматические связи двух уровней, полезные в работе ИПС, и систематические перечни ключевых слов отрасли.".

В настоящей работе тезаурус понимается несколько шире - как множество семантических связей между терминами некоторой отрасли. Такого рода связи не зависят от совместного пользования терминов, скорее наоборот. Составление такого тезауруса в настоящее время не автоматизируемо, так как невозможно формализовать процесс определения связей. Однако, ЭВМ может оказать существенную помощь при создании семантических тезаурусов, и это путем составления терминологических словарей и систематизированием и логическим контролем фиксированных связей.

Допустим, что имеется составленный человеком тезаурус

$$R(X,X)=\{R_1(X,X), R_2(X,X), \cdots, R_m(X,X)\},$$

где $R_k(X,X)=\{(x,y): x,y\in X\}$ называется видом отношения (отношением) тезауруса. В данном случае весы элементов отношений опускаем, так как их целесообразно определить на логически проверенном тезаурусе. Каждый вид отношения R_k характеризуется некоторыми, определенными семантикой вида, свойствами. На основе этих свойств на каждом отношении определена структура, которая должна соответствовать применению терминов в реальности.

Составленный человеком тезаурус, как правило, содержит ошибки, обусловленные недостатком человеческих способностей представления всей структуры связей в целом. Без сомнения человек способен соответствующей проверкой устранить допущенные ошибки, но вопрос в том, как ему помочь в этом деле?

Пусть отношение R_k задается в виде пар $(x,y)\in R_k$. Тогда допущенными человеком ошибками могут быть лишь отсутствие некоторых пар и/или фиксирование $(x,y)\in R_{k-1}$, хотя между x и y не имеет места связь, определенная видом R_k. В обоих случаях получается искаженная структура, представляющая связи между терминами исследуемой предметной области неточно. Понятно, что ЭВМ не имеет достаточно данных, чтобы найти все отсутствующие или лишние связи. Задача ЭВМ - организация отношения R_k таким образом, чтобы человеку легко было представить структуру связей и указание на те связи, которые явно нарушают определенную свойствами R_k логическую структуру.

Если, по мере надобности, вместо некоторого отношения R_k допускать рассмотрение обратного ему отношения R_k^* (например, вместо вид-род использовать род-вид), то каждое отношение тезауруса является либо эквивалентностью, либо иерархией. Тогда в ре-

зультате анализа некоторого отношения мы должны получить либо множество классов эквивалентности, либо - лес. При этом, $R_к$ заменяется на $R^*_к$ только в процессе машинного анализа. Первоначально в тезаурусе могут быть зафиксированы оба отношения: в таком случае имеется дополнительная возможность проверки правильности тезауруса. Именно, результат анализа отношения $R_к$ должен совпадать с результатом анализа отношения $(R^*_к)^*$, а также структурой отношения $R_к \cup (R^*_к)^*$.

Если известно, что отношение $R_к$ задает иерархию, то легко обнаружить элементы $\{(x,y),(z,y)\} \in R_к$ нарушающие структуру дерева или множество элементов $\{(x,x_1),(x_1,x_2),\dots,(x_n,x)\} \in R_к$ задающие циклы. Однако, формальными методами невозможно установить, какие элементы являются ложными, поэтому от алгоритма анализа требуется лишь отметить такие элементы. Устранение ошибок, если они не обусловлены несогласованностью установившейся терминологии, является задачей человека.

4.2. Алгоритм анализа тезауруса

Пусть для некоторого бинарного отношения $R_к(X,X) \in R(X,X)$ известно, что оно является иерархией. Требуется построить определенный им лес, вершины деревьев которого соответствуют ключевым словам. Ввиду ошибок составления $R_к$, а также возможной несогласованности терминологии, структура дерева может быть нарушена. Алгоритм анализа должен обнаружить все места нарушения структуры и устранить такой структурный конфликт вводом фиктивной вершины. Для построения структур алгоритм анализа должен:

I) найти все вхождения одного и того же ключевого слова $x \in X$, заменить их одним словом и изменить соответствующим образом ссылки;

2) проверить, не подчиняется ли одно и то же ключевое слово разным словам, если да, то конструированием фиктивной вершины устранить структурный конфликт;

3) проверить, существует ли в каждой группе связанных ключевых слов корень дерева.

Таким образом, основными операциями в алгоритме анализа являются поиск по ключевым словам, которые могут входить в состав анализируемой информации неопределенное число раз, поиск по ссылкам на одноименные слова и движение по ссылкам. Особенно "неудобными", ввиду неуникальности ключей, являются операции поиска. Воспользуясь тем, что в алгоритме анализа поиск в основном предназначен для выявления "равноценных" элементов (одноименных слов, ссылок на одни и те же слова), в предложенном ниже алгоритме поиск заменяется сортировкой по соответствующим признакам, в результате которого образуются требуемые группы.

Пусть $T=\{t_1, t_2, \cdots, t_n\}$ является множеством вершин, где каждая вершина (соответствует некоторому ключевому слову) $t \in T$ является кортежом

$$t=(NAME(t), NR(t), UP(t), DOWN(t), LINK(t)),$$

где $NAME(t)$ - имя вершины, $NR(t)$ - его порядковый номер и $UP(t), DOWN(t), LINK(t)$ - ссылки соответственно на предок, сын и брат.

Каждой вершине соответствует некоторое ключевое слово, которое является его именем. При этом одно и то же ключевое слово может именовать несколько вершин. Каждую вершину рассматриваем как запись фиксированной длины. Тогда место расположения n -ой вершины в последовательности $t_1, t_2, \cdots, t_n, \cdots, t_n$ является функцией от n и длины записи. Тогда по порядковому номеру легко найти место расположения соответствующей вершины, и поэтому

значениями ссылок можно использовать номера вершин.

По заданному отношению R_k для любого элемента $(x_i, x_j) \in R_k$ в памяти ЭВМ составляют две расположенные подряд вершины

$$t_r = (x_i, r, 0, r+1, r)$$

и

$$t_{r+1} = (x_j, r+1, r, 0, r+1),$$

где r – порядковый номер составления вершины. В такой паре t_r называется предком вершины t_{r+1}, а t_{r+1} – сыном t_r.

Вершины $t_{j_1}, t_{j_2}, \dots, t_{j_k}$ называются братьями, если $NAME(UP(t_{j_1})) = NAME(UP(t_{j_2})) = \dots = NAME(UP(t_{j_k}))$. В ходе анализа тезауруса братья объединяют ссылками $LINK$ в кольцо: $LINK(t_{j_1}) = j_2$, $LINK(t_{j_2}) = j_3, \dots, LINK(t_{j_k}) = j_1$. Общий предок группы братьев указывает ссылкой $DOWN$ на любую из них.

Любая вершина имеет имя и номер. Если у вершины $t \in T$ нет предка, т.е. она является корнем дерева, то $UP(t) = 0$, если отсутствует сын, то $DOWN(t) = 0$ и если нет брата, то $LINK(t) = NR(t)$.

В ходе анализа из группы одноименных вершин сохраняется лишь один экземпляр, называемый оригиналом. Остальные экземпляры называются дублетными вершинами, отмечаются специальным признаком и обеспечиваются ссылкой на свой оригинал. Признаком дублетной вершины используется $NAME(t) = 0$. По сути дела нулевой является только некоторая часть поля $NAME$ и на остальной части задается ссылка на соответствующий оригинал $(ORIG)$. Так как дублетные вершины в конце работы алгоритма стираются, то все связанные с ней ссылки "передают" оригиналу. Оригиналом называют и вершину, имя которой является уникальным во всем рассматриваемом множестве вершин.

Если отношение $R_к$ действительно задает иерархию, то каждый оригинал имеет лишь одного предка и те образуются кольца по ссылкам DOWN и UP. Однако, так как иерархичность является лишь предполагаемым свойством отношения $R_к$ и, кроме того, в $R_к$ могут содержаться ошибки, то приведенные выше утверждения могут быть нарушены. В таком случае автоматически создается дополнительная, т.н. фиктивная вершина, связываемая со всей структурой таким образом, чтобы устранился структурный конфликт. Именем фиктивной вершины выбирается имя некоторой участвующей в конфликте вершины, дополненное специальным символом. По этому символу в распечатке легко найти конфликтные места и имя соответствующей вершины.

Роль ссылки NR, в описанном ниже алгоритме заключается в том, чтобы сделать вершины в некоторой степени независимыми от их места расположения, ведь по NR всегда можно восстановить исходный порядок, а также найти значения соответствующих ссылок.

Алгоритм анализа тезауруса разделяется на следующие этапы.

A. Составление групп одноименных вершин и отметка дублетных.

B. Корректура ссылок предка: ссылки UP, показывающие на дублеты, заменяются ссылками на соответствующие оригиналы.

C. Исследование ссылок UP и DOWN дублетных вершин. Вершины, подчиняющиеся дублетным вершинам, подчиняют соответствующим оригиналам. Если предок дублета не совпадает с предком его оригинала, то создают фиктивную вершину.

D. Проверка ссылок DOWN. Если DOWN показывает на дублет, то она заменяется ссылкой на соответствующий оригинал.

E. Составление групп братьев. Вершины одной группы соединяют ссылками LINK.

F. Сборка мусора (выбрасывание дублетов, уплотнение информации и соответствующая корректировка ссылок).

G. Проверка на существование колец по ссылкам предка. По мере надобности создаются фиктивные вершины.

Для более детального описания алгоритма анализа тезауруса применяем следующие процедуры:

$$up(t) = \begin{cases} UP(t), & \text{если } NAME(UP(t)) \neq 0 \\ ORIG(UP(t)), & \text{если } NAME(UP(t)) = 0, \end{cases}$$

$$down(t) = \begin{cases} DOWN(t), & \text{если } NAME(DOWN(t)) \neq 0 \\ ORIG(DOWN(t)), & \text{если } NAME(DOWN(t)) = 0, \end{cases}$$

$name(t)$ - создание имени фиктивной вершины по $NAME(t)$,
$sort(UP), sort(NR), sort(NAME)$ - сортировка множества вершин по ссылкам предка, номера и имени.

Теперь алгоритм анализа отношения иерархии записывается приведенными ниже шагами, где обозначения шагов соответствуют описанным выше этапам.

A1. $sort(NAME)$: упорядочить вершины $T = \{t_1, t_2, \dots, t_n\}$ по имени в лексикографическом порядке.

A2. $i := 1$: начало установления дублетов.

A3. $j := i + 1$.

A4. Если $j > n$, то перейти к B1: все дублеты найдены.

A5. Если $NAME(t_i) \neq NAME(t_j)$, то перейти к A7.

A6. $NAME(t_j) := 0$; $ORIG(t_j) := NR(t_i)$; $j := j + 1$; вернуться к A4.

A7. $i := j$; вернуться к A3.

B1. $sort(NR)$: восстановить исходный порядок вершин.

B2. $i := 1$: начинается редактирование ссылок предка.

B3. $UP(t_i) := up(t_i)$.

B4. $i := i + 1$; если $i \leq n$ то вернуться к B3.

C1. $j := 1$: цикл анализа связей дублетов.

C2. Если $NAME(t_j) \neq 0$, то перейти к C15.

C3. $i := ORIG(t_j)$: т.е. вершина t_i является оригиналом дублета t_j .

C4. Если $DOWN(t_j) = 0$, то перейти к C10: вершина не имеет сыновей.

C5. $DOWN(t_i) := DOWN(t_j)$: не существенно, на какой сын вершина t_i указывает, поэтому эта операция допустима и тогда, когда $DOWN(t_i) \neq 0$.

C6. $r := s := DOWN(t_j)$: начинается передача сыновей дублетов их оригиналам.

C7. $UP(t_r) := i$.

C8. $r := LINK(t_r)$; если $r \neq s$, то вернуться к C7.

C9. $DOWN(t_j) := 0$.

C10. Если $UP(t_j) = 0$, то перейти к C15.

C11. Если $UP(t_i) \neq 0$, то перейти к C13.

C12. $UP(t_i) := UP(t_j)$; перейти к C15.

C13. Если $UP(t_i) = UP(t_j)$, то перейти к C15.

C14. $NAME(t_j) := name(t_i)$: оригинал и дублет имели разные предки, образуется фиктивная вершина.

C15. $j := j + 1$, если $j \leq n$, то вернуться к C2.

D1. $i := 1$: начало проверки сыновей на дубельность.

D2. $DOWN(t_i) := down(t_i)$.

D3. $i := i + 1$; если $i \leq n$, то вернуться к D2.

E1. $sort(UP)$: после сортировки образуются группы братьев.

E2. $i := 1$.

E3. Если $NAME(t_i) = 0$, то $i := i + 1$ и повторить E3.

E4. $LINK(t_i) := i$: найдена первая вершина из группы братьев.

E5. $j := i + 1$.

E6. Если $j>n$, то перейти к F1.

E7. Если $NAME(t_j)=0$, то перейти к E10.

E8. Если $UP(t_i)\neq UP(t_j)$, то перейти к E11: т.е. рассмотрение группы окончено.

E9. $LINK(t_{j-1}):=LINK(t_j)$; $LINK(t_j):=i$.

E10. $j:=j+1$; перейти к E6.

E11. $i:=j$; перейти к E4.

F1. sort(NR): восстанавливается исходный порядок вершин.

F2. $i:=j:=i$: начинается выбрасывание дублетов и уплотнение информации; вершины получают новую нумерацию.

F3. Если $NAME(t_i)=0$, то перейти к F6.

F4. $NR(t_i):=j$: новый номер вершины.

F5. $j:=j+1$.

F6. $i:=i+1$, если $i\leq n$, то вернуться к F3.

F7. $i:=1$: начинается корректура ссылок по новой нумерации.

F8. Если $NAME(t_i)=0$, то $NR(t_i):=n$; перейти к F10.

F9. $UP(t_i):=NR(UP(t_i))$; $DOWN(t_i):=NR(DOWN(t_i))$; $LINK(t_i):=NR(LINK(t_i))$.

F10. $i:=i+1$, если $i\leq n$, то вернуться к F8.

F11. $n:=j$: число вершин уменьшилось.

F12. sort(NR): вершины упорядочивают по новым номерам, дублеты передвигаются в конец списка, т.е. в свободную часть памяти.

G1. $m:=n$; $i:=1$: начинается устранение циклов по ссылкам UP.

G2. $j:=UP(t_i)$.

G3. Если $j=0$, то перейти к G7.

G4. $j:=UP(t_j)$.

G5. Если $j\neq i$, то вернуться к G4.

G6. $m:=m+1$; $NAME(t_m):=name(t_i)$;

$NAME(t_m) := m$; $UP(t_m) := 0$; $LINK(t_m) := m$;
$DOWN(t_m) := i$; $UP(t_j) := m$: фиктивная вершина будет предком некоторой вершины цикла.

G7. $i := i + 1$; вернуться к G2.

G8. $n := m$: анализ окончен, показывает число оставшихся вершин.

Приведенный алгоритм предназначен для анализа отношения иерархии. Однако, им можно пользоваться и для анализа отношений эквивалентности, если при вводе пары $(x, y) \in R_K$ представить в лексикографическом порядке. Это допустимо, так как ввиду симметрии из $(x, y) \in R_K$ следует правильность утверждения $(y, x) \in R_K$. Конечно, из алгоритма можно исключить составление фиктивных вершин или же просто пропустить их при распечатке.

4.3. Некоторые результаты анализа тезауруса юридической терминологии

Так как составленный тезаурус юридической терминологии по своему объему и рассматриваемым в нем видам отношений является уникальным и, по данным составителей, первым такого рода тезаурусом в СССР, то он представляет интерес с многих точек зрения. Но в данной работе ограничиваемся лишь некоторыми числовыми характеристиками, выявленными в ходе машинного анализа. Особо выдвигаем данные об обнаруженных ошибках, чтобы ответить на вопрос: содействует ли машинный анализ обнаружению ошибок или же является составленный человеком тезаурус безошибочным с точки зрения формальных ошибок?

Процесс составления тезауруса является рекурсивным в том

смысле, что исходный вариант тезауруса передается для анализа ЭВМ, затем корригируется и снова анализируется до тех пор, пока не обнаружатся ошибки в смысле структурных конфликтов. Для того, чтобы схарактеризовать точность составленного человеком тезауруса, ниже приведены количества найденных ошибок на первом цикле анализа. В последующих циклах количественный анализ найденных ошибок не проводился. Проведенный эксперимент не является "чистым" в том смысле, что составители тезауруса знали о последующем машинном анализе, и, может быть, не проделали окончательные ручные проверки установления корректности тезауруса.

Составленный тезаурус юридических терминов содержит приблизительно 16 000 ключевых слов. В нем рассматриваются семь основных видов отношений: синонимность, род-вид, неравенство, целое-часть, функциональная зависимость, существенная юридическая связь и ассоциативность. (Обоснование выбора видов отношений излагается группой составителей в статье [5.7].) Кроме того, для создания возможности дополнительной проверки потребовали и фиксирование обратных к отношениям род-вид и целое-часть отношений вид-род и часть-целое.

Рассматриваем результаты анализа тезауруса по видам отношений.

Отношение синонимность, являющееся эквивалентностью, понимается в смысле отождествляемости ключевых слов в процессе информационного поиска. Сюда относятся в первую очередь синонимные по содержанию слова (например, лекарство - медикамент), а также разновидности корней ключевых слов. В составленном тезаурусе отношением синонимности было связано 5366 ключевых слов, образующих 2197 классов эквивалентности. В основном каждый класс содержит лишь пару ключевых слов, однако максимальный из них имеет 10 слов. В этом случае анализ отношения никаких структурных ошибок не выявил.

Иерархия, задаваемая отношением род-вид (аналогично вид-род), понимается по общеакцептируемым критериям. Машинный анализ был проведен отдельно по обоим видам отношения. Оказалось, что родо--видовое отношение является самым объемным в тезаурусе юридической терминологии, содержащем 6868 ключевых слов, которые разбиты на 1150 деревьев. Максимальная глубина дерева достигала 5 уровней, а максимальное число вершин в дереве - 95. В результате автоматического анализа было поставлено под подозрение 5% из фиксированных связей, которые подвергались дополнительной ручной проверке.

В идеальном случае определенная отношением род-вид структура должна совпадать со структурой отношения вид-род. Однако, в первоначальном варианте такое совпадение не имело места. Уже число используемых ключевых слов в отношении вид-род (6733) отличалось от соответствующего числа в родо-видовом отношении. Полное совпадение построенных деревьев установилось лишь при 10% деревьев. Это указывает, что оба вида отношения были составлены независимо друг от друга и они требуют тщательного пересмотра. Кроме того столь сильное различие указывает на неполноту обоих видов отношения.

Отношение целое-часть имеет место между терминами, один из которых составляет часть другого, например, криминалистика - трассология. Как и полагалось, в юридической терминологии, противоположно технической, этот вид отношения является малочисленным. Он содержал лишь 210 терминов (отношение часть-целое содержало 204 термина), составляющих 92 (соответственно 88) дерева. Максимальная глубина дерева 3 (3) уровня и максимальное число вершин - 9 (8). При анализе подозрительными считались 2% всех элементов отношения; при сравнении с обратным отношением 73% составленных деревьев совпадали. Такой, относительно хороший, ре-

зультат, сравнительно с отношением род-вид, в первую очередь объясняется небольшим объемом рассматриваемого отношения.

Функциональное отношение в составленном тезаурусе охватывает виды связей деятель - действие, действие - результат, причина - результат и обратные им. Оказывалось, что этот вид отношения охватывает 635 терминов, образующих 304 деревьев с максимальной величиной в 10 слов и глубиной - 3 уровня. В результате анализа 5% из зафиксированных связей считались некорректными. Основной причиной некорректности было несоблюдение направления связи.

Фиксированное в тезаурусе отношение отрицания понимается в виде пар антонимных слов, как, например, равенство - неравенство, важный - неважный и т.д. Число элементов в этом виде отношения невелико, лишь 271 слово, образующее 129 групп.

Существенная юридическая связь является разновидностью ассоциативной связи. Она была зафиксирована в тех случаях, когда ассоциативная связь имеет юридически существенное содержание, как, например, завещатель - наследство - наследник. При анализе этот вид отношения рассматривался эквивалентностью. В результате анализа было установлено существование 1682 групп, связывающих 3740 ключевых слов.

Ассоциативной связью были связаны 6473 термина. Рассмотрение ассоциативной связи как образующей классы эквивалентности, не дало удовлетворительных результатов, так как в одном классе оказались довольно далекостоящие термины. Чтобы сохранить схему связей между терминами, ассоциацию анализировали как иерархию. В результате этого было построено 938 графов с числом вершин в графе от двух до 246. Конечно, построенные алгоритмом анализа фиктивные вершины в данном случае не указывают на ошибки, а на существование многосторонних связей.

Сравнивая объемы видов отношений между собой, можно сказать,

что основную часть юридического тезауруса составляют родо-видовое отношение и ассоциативность вместе с юридически существенной связью. Учитывая тот факт, что образующиеся множества связанных между собой терминов могут содержать до несколько сот ключевых слов, то их применение в автоматическом индексировании без учета весов может привести к чрезмерному росту числа индексов документа. Число добавляемых индексов должно регулироваться механизмом присваивания весов и индексы с достаточно малым значением веса можно отбрасывать.

Что касается автоматического анализа тезауруса, то существование такого алгоритма позволяет фиксировать тезаурус по отдельным связям, из которых в результате анализа вырисовывается целостная картина.

5. СОСТАВЛЕНИЕ ТЕМАТИЧЕСКИХ ПОДМНОЖЕСТВ ДОКУМЕНТОВ И КЛЮЧЕВЫХ СЛОВ

Ввиду большого количества вводимых в ИПС документов и большого числа ключевых слов, существенным для повышения эффективности работы системы является группировка документов и ключевых слов на тематические классы [4.9]. В общем случае создаваемые классы могут пересекаться. Естественно требовать, чтобы число и объемы создаваемых классов определялись внутренними связями между объектами, и не являлись задаваемыми вперед параметрами. В основном по этому, а также по некоторым другим причинам (невозможность одновременного рассмотрения объектов и признаков, объем вычислительной работы и т.д.) применение традиционных методов классификации [4.I] затруднено.

В этой главе излагается новый метод выделения классов документов и ключевых слов [5.26]. Размытое отношение между документами и индексами, получаемое в результате индексирования, рассматривается определяющим монотонную систему $S = (W, g_W)$, где W – множество элементов системы (документы и/или ключевые слова) и g_W – весовая функция, присваивающая каждому элементу вес. Множество W и функция g_W выбираются в соответствии с целью исследований системы; функция g_W должна отражать внутреннюю зависимость между выбранными элементами. Ищется такая подсистема $S' = (W', g_{W'})$, называемая ядром, в которой минимальный вес элементов достигает максимального значения. Опреде-

ленное таким образом ядро, при соответствующих W и g_W, является классом наиболее связанных документов и/или ключевых слов. Алгоритм выделения ядра применим повторно, в результате этого получается множество классов. При этом повторное применение алгоритма можно определить так, что получаются возможно пересекающиеся классы, где уровень перекрытия, а также общее число классов не определено заранее.

В конце главы показывается, что найденные при некоторых функциях g_W ядра являются кластерами в смысле Линга [5.32]. Таким образом, изложенный метод является более общим, но он совместим с традиционными методами.

5.I. Определение ядра монотонной системы

Прежде чем рассматривать разделение документов и ключевых слов на тематические подмножества, коротко приведем определения монотонной системы и его ядра. Также опишем алгоритм вычисления наибольшего ядра. Более подробное изложение теории монотонных систем можно найти в работах И. Муллата и Л. Выханду [5.3, 5.I3, 5.I4, 5.I5].

Пусть задано некоторое конечное множество элементов $W = \{w_1, w_2, \dots, w_n\}$, на которых определена весовая функция $g = g_W$. Системой S называется пара $S = (W, g)$, а значение $g_W(w)$ называется весом элемента $w \in W$. Допустим, что для любого подмножества $W' \subseteq W$ определено сужение $g_{W'}$ весовой функции . Тогда $S' = (W', g_{W'})$ считается подсистемой системы S. Так как весовая функция зафиксирована для рассматриваемой системы S, то любая подсистема S' определена множеством ее элементов W'.

Определение 5.I. Система $S=(W,g_W)$ называется монотонной, если для любых двух ее подсистем $S_1=(W_1,g_{W_1})$ и $S_2=(W_2,g_{W_2})$ при $w\in W_2$ и $W_2\subset W_1$ имеет место $g_{W_2}(w)\geqslant g_{W_1}(w)\geqslant g_W(w)$ или же $g_{W_2}(w)\leqslant g_{W_1}(w)\leqslant g_W(w)$. В первом случае обозначаем систему через S^+, во втором – через S^-.

Легко доказывается, что подсистема монотонной системы является монотонной в том же направлении. Среди всевозможных подсистем монотонной системы особый интерес представляют такие, на которых весовая функция g принимает экстремальные значения в определенном ниже смысле. Такие подсистемы называются ядрами системы. Ядро системы можно считать ее самой существенной частью.

Определение 5.2. Ядром системы $S^+=(W,g_W)$ называется ее подсистема H^+, при которой функция $F(H)=\max\limits_{w\in W'} g_{W'}(w)$ определенная на подсистемах $H=(W',g_{W'})$ достигает минимума. Аналогично, ядром системы $S^-=(W,g_W)$ называется ее подсистема H^-, при которой функция $F(H)=\min\limits_{w\in W'} g_{W'}(w)$ достигает максимума.

В настоящей работе в дальнейшем рассматриваются только системы $S^-\equiv S=(W,g)$ и соответственно ядра $H^-\equiv H=(W,g)$. В случаях, когда ссобо подчеркивается минимальный вес элементов ядра $\lambda=\min\limits_{w\in W'} g_{W'}(w)$, ядро обозначается через H^λ.

Ядро системы не обязательно определено однозначно: функция g может достигать экстремальных значений на нескольких подсистемах. В статье [5.I3] доказывается, что если $H_1=(W_1,g_{W_1})$ и $H_2=(W_2,g_{W_2})$ ядра данной системы, то ядром оказывается и подсистема $H=(W',g_{W'})$, где $W'=W_1\cup W_2$. Объединение всех ядер системы называется наибольшим ядром.

Алгоритм вычисления наибольшего ядра [5.3] заключается в на-

хождении такого численного значения $u \in [L, M]$, где

$$L = \min_{w \in W} g_W(w), \quad M = \max_{w \in W} g_W(w)$$

при котором специальная процедура СЛОЙ выделяет наибольшее ядро. Работа процедуры СЛОЙ (u, W') при $W' \subseteq W$ заключается в последовательном применении вспомогательной процедуры слой (u, W^i) и описывается следующим образом:

$$\text{СЛОЙ}\ (u, W') = \text{слой}\ (u, W^n),$$

где

$$W^i = \text{слой}\ (u, W^{i-1}) = \{ w : w \in W^{i-1}, g_{W^{i-1}}(w) > u \},$$

$i = 1, \dots, n$; $W^0 = W'$ и значение n определяется условием $W^n = W^{n+1}$.

Алгоритм вычисления наибольшего ядра описывается теперь следующими шагами:

I. $L := \min_{w \in W} g_W(w)$; $M := \max_{w \in W} g_W(w)$.

2. $u := \varphi(L, M)$; где φ некоторая зафиксированная функция вычисления значения $u \in [L, M]$.

3. $W' :=$ СЛОЙ (u, W); если $W' = \emptyset$, то положить $M := u$ и вернуться к шагу 2.

4. $u := \min_{w \in W'} g_{W'}(w)$.

5. $W'' :=$ СЛОЙ (u, W); если $W'' \neq \emptyset$, то положить $L := u$ и вернуться к шагу 2, в противном случае наибольшее ядро $H^u = (W', g)$ найдено.

Примеры вычисления наибольших ядер приводим в дальнейшем.

В настоящей работе выделение "подъядер" наибольшего ядра рассматривается только в частном случае, при определенных теоремой 5.I условиях.

Теорема 5.1. Если в наибольшем ядре $H^{\Delta}=(W', g)$ системы $S=(W, g)$ существует подсистема $H_1=(W_1, g)$, где $W_1 \subset W'$ такая, что для любого элемента $w \in W_1$ имеет место $g_{W_1}(w)=g_{W'}(w)$, то H_1 является ядром системы S.

Доказательство. Для доказательства теоремы следует показать, что H_1 является одной из тех подсистем, на которых $\Delta_1 = \min_{w \in W_1} g_{W_1}(w)$ достигает максимальное значение Δ. Понятно, что Δ_1 не может быть больше Δ, так как в таком случае H^{Δ} не являлось бы ядром. Покажем, что Δ_1 не может быть меньше Δ. Действительно, если $\Delta_1 < \Delta$ и значение Δ_1 достигается при элементе $\bar{w}$, то по предпосылкам теоремы $\Delta_1 = g_{W_1}(\bar{w}) = g_{W'}(\bar{w}) < \Delta$ т.е. Δ уже не является минимальным весом. Значит $\Delta = \Delta_1$ и подсистема H_1 является ядром.

5.2. Нахождение тематических классов документов

Пусть множество документов $\mathfrak{D}=\{d_1, d_2, \cdots, d_m\}$ и множество ключевых слов $\mathfrak{X}=\{x_1, x_2, \cdots, x_n\}$ в результате индексирования связаны размытым отношением

$$S(\mathfrak{D}, \mathfrak{X})=\{(d_j, x_i) \mid f(d_j, x_i)\},$$

где $i=1, \cdots, n$ и $j=1, \cdots, m$ а $f(d_j, x_i)$ - вес ключевого слова x_i в документе d_j. Тогда каждый документ $d_j \in \mathfrak{D}$ характеризуется его поисковым образом

$$d_j=\{x_i \mid f(d_j, x_i) : i=1, \cdots, n\},$$

а каждое ключевое слово $x_i \in \mathfrak{X}$ характеризуется его вхождением в документы множества

$$x_i=\{d_j \mid f(d_j, x_i) : j=1, \cdots, m\}.$$

На основе размытых множеств δ_j и γ_i можно установить связи между ключевыми словами, документами и между собой. На основе этих связей ниже определяем тематические классы терминов и документов.

Поставим сначала задачей найти совокупность наиболее четко выраженные тематические классы документов, где в один тематический класс входят документы общей тематики. Допустим, что тема документа определяется совокупностью примененных в этом документе ключевых слов с учетом их весов. Таким образом, ПОД, заданный в виде размытого множества, является тематическим описанием документа. Тогда общность тематики двух документов $d_j, d_k \in \mathcal{D}$ характеризуется совпадением входящих в эти документы ключевых слов и можно найти по формуле

$$\varphi(d_j, d_k) = |\delta_j \cap \delta_k| . \qquad (5.1)$$

В частном случае, когда документы описываются неразмытыми множествами, формула (5.1) дает число общих обоим документам ключевых слов.

Поставим каждому документу $d_j \in \mathcal{D}$ в соответствие число

$$g_{\mathcal{D}}(d_j) = \sum_{\substack{k=1 \\ k \neq j}}^{m} \varphi(d_j, d_k), \qquad (5.2)$$

характеризующее его связанность со всеми остальными документами из $\mathcal{D}$.

На основе функции $g_{\mathcal{D}}$ на множестве документов можно конструировать систему $S = (\mathcal{D}, g_{\mathcal{D}})$. Легко проверить, что конструированная таким образом система S является монотонной. Имея в виду содержание весовой функции g, наибольшее ядро системы S задает множества тематически наиболее сильно связанных документов и этим оказывается искомым множеством.

Рассмотрим пример.

Пусть совокупность документов $\mathfrak{D} = \{d_1, d_2, \dots, d_8\}$ описывается ключевыми словами $\mathfrak{X} = \{x_1, x_2, \dots, x_8\}$ показанным в таблице 5.I образом. Тогда по формуле (5.I) можно установить

$\mathfrak{D}$ \ $\mathfrak{X}$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
d_1	0,8	0,9						
d_2	0,4		I,0					
d_3			0,2	I,0				
d_4			0,8	0,2				
d_5		0,3			0,5	I,0		
d_6					0,3	0,2	I,0	
d_7					0,5			I,0
d_8							0,2	0,2

Таблица 5.I. Выбранная для примера связь между документами и ключевыми словами.

связи между документами (см. рис. 5.I), а по (5.2) найти вес каждого из них. Вычисляя наибольшее ядро составленной системы (см.

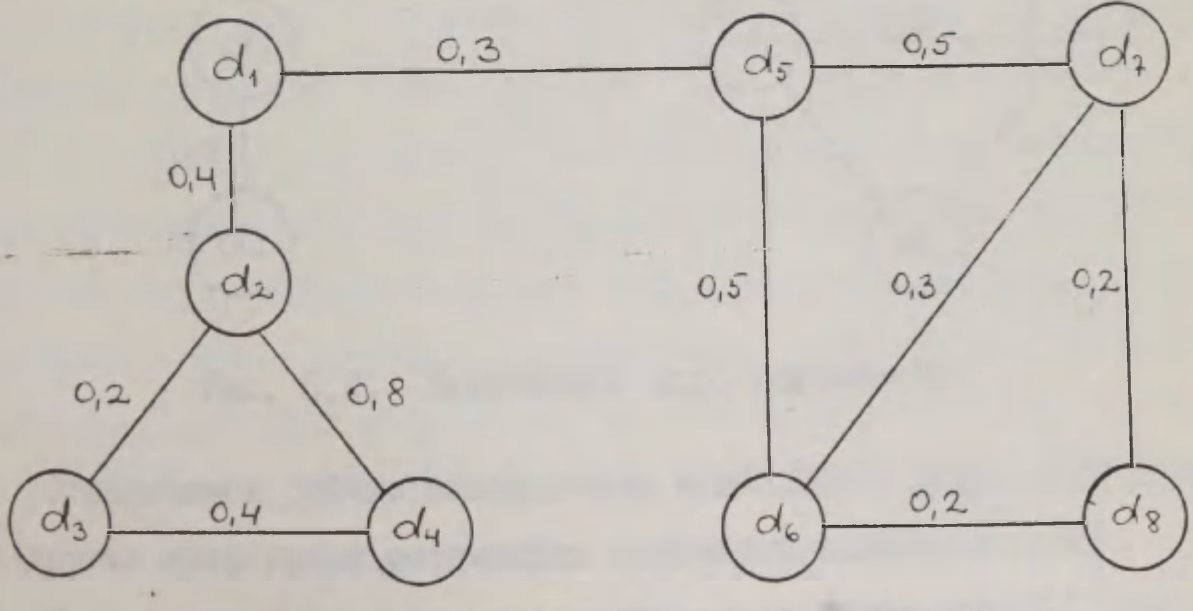

Рис. 5.I. Граф связей между документами.

табл. 5.2), получаем множество $\{d_2, d_4, d_5, d_6, d_7\}$ связи между элементами которого приведены на рис. 5.2. Судя по ри-

u \ g	$g_\mathfrak{D}(d_1)$	$g_\mathfrak{D}(d_2)$	$g_\mathfrak{D}(d_3)$	$g_\mathfrak{D}(d_4)$	$g_\mathfrak{D}(d_5)$	$g_\mathfrak{D}(d_6)$	$g_\mathfrak{D}(d_7)$	$g_\mathfrak{D}(d_8)$	Примечания
	0,7	I,4	0,6	I,2	I,3	I,0	I,0	0,4	$L=0{,}4$; $M=1{,}4$; $u:=(L+M)/2$
0,9	–	0,8	–	0,8	I,0	0,8	0,8	–	СЛОЙ $(0{,}9, \mathfrak{D})$
		–		–	0	–	–		$\mathfrak{D}'=\emptyset$; $M:=0{,}9$
0,65	0,7	I,2	–	0,8	I,3	0,8	0,8	–	$\mathfrak{D}'\neq\emptyset$; $u:=\min g_{\mathfrak{D}'}(d_i)$
0,7	–	0,8		0,8	I,0	0,8	0,8		СЛОЙ $(0{,}7, \mathfrak{D})\neq\emptyset$; $L:=0{,}65$
0,77	–	0,8	–	0,8	I,0	0,8	0,8	–	СЛОЙ $(0{,}77, \mathfrak{D})\neq\emptyset$; $u:=\min$
0,8		–		–	0	–	–		$\mathfrak{D}'$ – ядро

Таблица 5.2. Ход вычисления наибольшего ядра.

сунку, наибольшее ядро распадается на две несвязанные между собой части $W_1 = \{d_2, d_4\}$ и $W_2 = \{d_5, d_6, d_7\}$. Легко проверить, что выполнены предпосылки теоремы 5.I и $H_1 = (W_1, g)$ и $H_2 = (W_2, g)$ являются ядрами. Минимальным весом элемента ядра окажется 0,8.

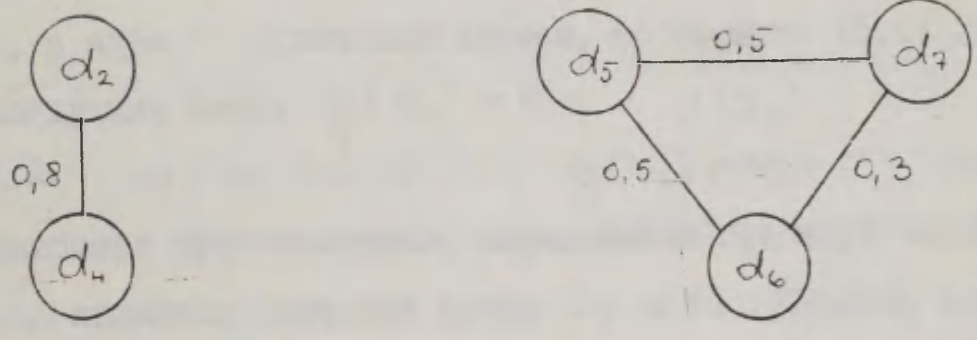

Рис. 5.2. Наибольшее ядро документов.

Ограничимся сейчас нахождением наибольшего ядра, а вычисление других ядер среди оставшихся элементов исследуем позже.

По сделанным предпосылкам каждое ядро можно считать темати-

ческим классом документов, характеризуемым применяемыми там ключевыми словами. В данном случае тематика ядра $\{d_2, d_4\}$ выражается словами x_1, x_3 и x_4, а ядра $\{d_5, d_6, d_7\}$ словами x_2, x_5, x_6, x_7 и x_8. В общем случае тематики образованных таким образом тематических классов могут пересекаться. Кроме того, среди ключевых слов, описывающих некоторый тематический класс, могут оказаться "более важные" и "менее важные" слова. Естественно считать ключевое слово в тематическом классе тем важнее, в чем больших документах оно встречается. Тогда аналогично формулам (5.I) и (5.2) можно определить формулы

$$\varphi(x_k, x_i) = |X_k \cap X_i| \qquad (5.3)$$

и

$$g_X(x_i) = \sum_{\substack{k=1 \\ k \neq i}}^{n} \varphi(x_k, x_i), \qquad (5.4)$$

показывающие соответственно силу связи x_i с x_k и связанность слова x_i со всеми другими словами множества X.

Тогда по формулам (5.3) и (5.4) легко найти значимости отдельных ключевых слов в ядрах документов. Например, в приведенном выше примере, в ядре H_2 ключевым словам, по формуле (5.4) сопоставляются следующие весы: $g(x_2) = 0,6$, $g(x_5) = 1,5$, $g(x_6) = 1,2$, $g(x_7) = 0,4$ и $g(x_8) = 0,7$. Отсюда видно, что наиболее существенными, выражающими основную тематику множества W_2 являются ключевые слова x_5 и x_6. Однако, возникает вопрос, как установить границу между существенными и несущественными словами? Кроме того, простое отбрасывание менее важных ключевых слов, с целью нахождения основной тематики ядра, влияет на структуру самого ядра и рассматриваемая совокупность документов может уже не оказаться ядром в смысле определения 5.2.

Во избежание такой ситуации и для учитывания весов ключевых слов в ходе вычисления ядра документов, построим систему $S=(W,G)$, элементами которой являются как документы, так и ключевые слова $(W=\mathfrak{D}\cup\mathfrak{X})$, причем вес элемента определяется либо формулой (5.2) либо (5.4):

$$G_W(w)=\begin{cases} g_{\mathfrak{D}}(w), & \text{если } w\in\mathfrak{D} \\ g_{\mathfrak{X}}(w), & \text{если } w\in\mathfrak{X}. \end{cases}$$

Понятно, что конструированная система $S=(W,G)$ распадается на две зависящие друг от друга части. К структуре документов, изображенной на рис. 5.I прибавляется еще и структура ключевых слов (рис. 5.3). Используя данные таблицы 5.I убедимся, что наибольшим ядром этой системы оказывается множество $\{d_5, d_6, d_7, x_5, x_6\}$.

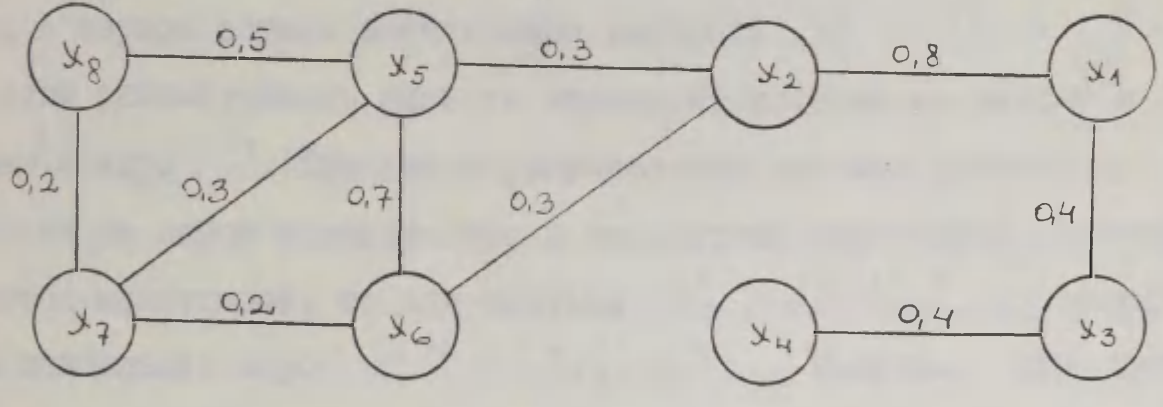

Рис. 5.3. Граф связей между ключевыми словами таблицы 5.I.

Полученное ядро (с минимальным весом 0,7) можно рассматривать как состоящее из двух частей: множества документов $\{d_5, d_6, d_7\}$ и множества характерных для этих документов ключевых слов $\{x_5, x_6\}$. Подчеркиваем, что в данном случае ни совокупность документов $\{d_5, d_6, d_7\}$ ни совокупность слов $\{x_5, x_6\}$ отдельно взятыми не являются ядрами. Ядро системы $S=\{W,G\}$ следует трактовать как наиболее четко выраженный тематический класс документов $\{d_5, d_6, d_7\}$ с основной тематикой $\{x_5, x_6\}$.

Таким образом, в изложенном выше методе одновременно учитываются связи между документами, между ключевыми словами, а также их взаимная зависимость. По характеру рассмотренных связей составленные классы можно трактовать ассоциативными классами, где ассоциативная связь определена совместным использованием ключевых слов в документах.

Отметим, что классы ключевых слов получаются и путем анализа системы $S=(X, g_X)$.

Допустим теперь, что для системы $S=(W, g)$ найдено наибольшее ядро $H^{\Delta_1}=(W_1, g)$. В результате этого система $S=(W, g)$ распадает на две части: ядро H^{Δ_1} и оставшаяся подсистема $S'=(W', g')$. Подсистему S' можно определить либо наложив ограничения на выбор элементов множества W', либо наложив ограничения на рассматриваемые связи между элементами, т.е. на g'. В первом случае естественно выбирать $W'=W\setminus W_1$, $g'=g$ и этим рассматривать лишь те элементы, которые не входят в полученное ядро H^{Δ_1}. Так как по определению весовая функция g определена на любом подмножестве и подсистема монотонной системы является монотонной, то для системы $S'=(W\setminus W_1, g)$ можно найти наибольшее ядро $H^{\Delta_2}=(W_2, g)$. Понятно, что процесс вычисления ядер можно продолжить и полученные множества W_1, $W_2, \dots, W_e$ не имеют общих элементов. Применяя этот процесс для нахождения классов документов, получаем непересекающиеся классы.

Однако, в реальных условиях вряд ли можно удовлетвориться непересекающимися классами документов и ключевых слов. Поэтому исследуем упомянутый выше второй вариант составления подсистемы S'. Положим,

$$g'(w_i)=\sum_{k} g(w_i, w_k), \qquad (5.5)$$

где сумма взята по всем индексам k, если $k\neq i$ и если либо

w_i либо w_k не входит в W_1. Этим в составленной подсистеме мы не рассматриваем связи между элементами найденного ядра H^{Δ_1}, но учитываем их связи с элементами, не входящими в ядро. В результате для системы $S'=(W, g')$ найдем ядро $H^{\Delta_2}=(W_2, g')$, где может иметь место $W_1 \cap W_2 \neq \emptyset$. Получается разложение элементов W на возможно пересекающиеся классы.

В результате повторного применения процесса выделения ядер, в обоих случаях получается последовательность $H^{\Delta_1}, H^{\Delta_2}, \ldots, H^{\Delta_\ell}$, которой соответствуют множества $W_1, W_2, \ldots, W_\ell$ и весы $\Delta_1 > \Delta_2 > \cdots > \Delta_\ell$. Если зафиксировать некоторое значение Δ, считаемое уровнем классификации, то процесс естественно оборвать на таком шаге k, для которого $\Delta_k \geqslant \Delta > \Delta_{k+1}$.

Обращаясь снова к рассмотренному выше примеру, найдем для системы $S=(\mathfrak{D} \cup \mathfrak{X}, G)$. "второе по важности" ядро, допуская пересекание классов, т.е. найдем ядро ее подсистемы $S'=(\mathfrak{D} \cup \mathfrak{X}, G')$, где при определении G' имеется в виду формула (5.5). Им оказывается множество элементов $\{d_1, d_2, d_3, d_4, x_1, x_2, x_3, x_4\}$ с минимальным весом 0,4. Удовлетворяясь полученным уровнем классификации, множество оставшихся элементов $\{d_5, x_7, x_8\}$ можно рассматривать как класс наименее существенных элементов.

В приведенном выше разложении документов и ключевых слов на подклассы мы исходили лишь из взаимных связей между элементами и упустили характеристики самих элементов. Ввиду этого в рассмотренном примере получается, что основная тематика $\{x_5, x_6\}$ класса $\{d_5, d_6, d_7\}$ не содержит наиболее существенное, судя по таблице 5.I, ключевое слово x_7 документа d_6. Для устранения указанного недостатка, допустим, что существенность применения ключевого слова для индексирования некоторого множества документов характеризуется его входимостью в эти документы. Также каждому до-

кументу ставим в соответствие число, показывающее, в какой степени X или некоторое его подмножество индексирует этот документ. В самом простом случае приведенные характеристики можно включить в функцию G, опустив в формуле (5.2) требование $k \neq j$ и в (5.4) требование $k \neq i$. Легко проверить, что в таком случае на приведенном примере образуются подклассы $\{d_1, d_2, d_3, d_4, x_1, x_2, x_3, x_4\}$, $\{d_5, d_6, d_7, x_5, x_6, x_7, x_8\}$ и $\{d_8, x_7, x_8\}$, в которых указанный недостаток не имеет места.

Следует подчеркнуть, что функцию G можно определить многими способами, выдвигая тот или иной аспект связей между ключевыми словами и документами. Единственным требованием при этом является монотонность полученной системы.

5.3. Связь ядра с кластерами

Сравниваем приведенный в этой главе метод классификации с другими методами. Покажем, что при соответствующем определении монотонной системы ядро системы является K-кластером.

Пусть заданы конечное множество объектов $W = \{w_1, w_2, \dots, w_n\}$ и матрица различий $R = (r(w_i, w_j)) = (r_{ij})$, где $i, j = 1, \dots, n$. Определяем вес каждого элемента $w_i \in W$ функцией матрицы R следующим образом: $g_W(w_i) = F(w_i, R)$. Если при этом соблюдается требование, что для любых $W' \subset W'' \subseteq W$ с матрицами различий R' и R'' при всех $w \in W'$ имеет место

$$g_{W'}(w) = F(w, R') \leq F(w, R'') = g_{W''}(w),$$

то пара $S = (W, F)$ определяет монотонную систему.

Содержательное значение наибольшего ядра системы S определяется семантикой функции F. Например, если положить

$$g^1_W(w_i) = \sum_{j=1}^{n} r_{ij},$$

то элементами ядра системы являются те объекты, при которых минимальное суммарное различие от других объектов достигает своего максимального значения. Таким образом, ядром является множество наиболее удаленных объектов.

С другой стороны, если при фиксированном r поставить

$$g^2_W(w_i) = \sum_{j=1}^{n} r^*_{ij}, \qquad (5.6)$$

где

$$r^*_{ij} = \begin{cases} 1, \text{ если при } i \neq j \quad r_{ij} \leq r \\ 0, \text{ если } r_{ij} > r \quad \text{или} \quad i = j, \end{cases} \qquad (5.7)$$

то вес объекта w в системе $S = (W, g^2)$ означает число "подобных ему", различия от которых не превышают заданного предела. В этом случае некоторое ядро $H^k_i = (W_i, g^2)$ дает подмножество объектов W_i, имеющих в W_i по меньшей мере k подобных. При этом по определению весовой функции k является целым числом и по определению ядра принимает на W_i максимальное значение.

По своей внутренней структуре ядро системы $S = (W, g^2)$ напоминает k-кластер, построенный на зафиксированном уровне различия r с максимально возможным числом связей k. Чтобы показать, что ядро системы $S = (W, g^2)$ действительно является k-кластером, напомним сначала определение понятия кластера [5.32].

Пусть задано множество объектов $W = \{w_1, w_2, \cdots, w_n\}$ и матрица различий $R = (r(w_i, w_j))$, где $i, j = 1, \cdots, n$. Тогда подмножество $W' \subseteq W$ называется k-кластером при заданном значении r, если:

I^0 для любых $w_i, w_j \in W'$ существует цепь $w_i = w_{i_1}, w_{i_2}, \cdots$

$\dots, w_{i_m} = w_j$ такая, что $r(w_{i_\ell}, w_{i_{\ell+1}}) \le r$, где $\ell = 1, \dots, m-1$.

2^0 для любого $w \in W'$ найдется по меньшей мере k-элементное подмножество $W^w \subset W'$ $(w \notin W^w)$ такое, что $r(w, w') \le r$ при $w' \in W^w$;

3^0 подмножество W' является максимальным в том смысле, если не найдется множества $W'' \supset W'$ такого, что условия I^0 и 2^0 выполнялись бы на W''.

Приведенное определение легко переформулировать для матрицы $R^* = (r^*_{ij})$, полученной из матрицы различий R с помощью формулы (5.7). Действительно, в первом условии следует лишь писать равносильное неравенству $r(w_{i_\ell}, w_{i_{\ell+1}}) \le r$ равенство $r^*(w_{i_\ell}, w_{i_{\ell+1}}) = 1$, а во втором $r^*(w, w') = 1$ вместо $r^*(w, w') \le r$. Таким образом, за основу выделения кластеров и наибольшего ядра принимаются одни и те же исходные данные.

Понятно, что в общем случае наибольшее ядро не является k-кластером, так как ничем не гарантируется выполненность условия I^0. Допустим сначала, что в частном случае наибольшее ядро $H^k = (W', g^2)$ является связным множеством в смысле условия I^0 определения кластера. Тогда по определению ядра для любого элемента $w \in W'$ имеет место

$$g^2_{W'}(w) = \sum_{w' \in W'} r^*(w, w') \ge k \qquad (5.8)$$

т.е. найдется по меньшей мере k-элементное множество W^w такое, что $r^*(w, w') = 1$ при $w' \notin W^w$. Тем самым условие 2^0 в определении кластера выполнено. Выполненность условия 3^0 обеспечивается тем, что W' является наибольшим множеством, для которого выполняется неравенство (5.8).

Значит, если множество W' является связным множеством, то

оно является κ -кластером. Кроме того, по определению ядра не существует подсистемы $H^{\kappa'}=(W'', g^2)$ такой, что $\kappa'>\kappa$ и этим связное наибольшее ядро является кластером при максимально возможной связанности элементов.

Пусть теперь для наибольшего ядра $H^{\kappa}=(W', g^2)$ условие I^0 не выполняется. Разобьем множество элементов W' на подмножества $W'=W_1'\cup W_2'\cup\cdots\cup W_m'$ таким образом, что $w, w'\in W_i'$ если между ними существует определенная условием I^0 цепь. Понятно, что при таком разбиении $W_i'\cap W_j'=\emptyset$, если $i\neq j$. Оказывается, что в этом случае определенная множеством W_i' система $S_i=(W_i', g^2)$ является ядром системы $S=(W, g^2)$. Действительно, так как для любого $w\in W_i'$

$$g^2_W(w)=\sum_{i=1}^{m}\Big(\sum_{w'\in W_i'} r^*(w,w')\Big)=\sum_{w'\in W_i'}{}' r^*(w,w'),$$

то на множествах W_i' $(i=1,\cdots,m)$ выполнены предпосылки теоремы I, т.е. $H_i^{\kappa}=(W_i', g^2)$ является ядром, а учитывая конструкцию W_i' - связным ядром. Для связного ядра, как было показано выше, выполняется условие 2^0, выполненность же условия 3^0 обеспечивается тем, что для любого элемента $w\in W_i'$ не найдется $w'\in W'\setminus W_i'$ такого, что $r^*(w,w')=1$.

Таким образом, разбиение наибольшего ядра $H^{\kappa}=(W', g^2)$ системы $S=(W, g^2)$, где функция g^2 определена формулой (5.6) на ядра, соответствует выделению κ -кластеров множества W при заданных r и R. При этом κ является максимальным числом подобных элементов, при котором образуются кластеры.

Для выделения кластеров на более низком уровне $\kappa_1<\kappa$ следует рассматривать подсистему $S_1=(W_1, g^2)$, где W_1 содержит элементы, подобные по меньшей мере одному из $W\setminus W'$. По определению кластера, кластеры на уровне κ_1 получаются в резуль-

тате объединения множеств $W'UW''$, где $H^{k_1}=(W'', g^2)$ является наибольшим ядром системы S_1.

В итоге можно сказать, что подходящим выбором весовой функции метод выделения ядер можно превратить в метод определения кластеров. Но в методе определения ядер имеются более гибкие возможности определения связей между исследуемыми элементами, и этим он больше подходит для анализа размытых ИПС.

6. РЕЗУЛЬТАТЫ И ВЫВОДЫ

I. Описывается новый тип ИПС - размытая ИПС, основными преимуществами которой являются

- использование на единых основах системы весов и ее связывание с оценками на выдаваемые документы;

- возможность автоматического применения независящих от конкретных документов данных о рассматриваемой предметной области, представляющих разные семантические связи между терминами.

2. Конструированная модель ИПС ориентирована на полную автоматизацию работ в документальной ИПС, включая такие, до сих пор мало автоматизированные процессы, как индексирование и применение тезауруса.

3. С информационно-поисковой системой связывается множество априорных знаний о предметной области, которые система использует в ходе своей работы. На основе этих знаний автоматически решается целый ряд семантических проблем, как образование словосочетаний, учет омонимности, применение индексов, не входящих прямым образом в текст документа и т.д. Соблюдая общую идеологию системы, знания о предметной области сопровождаются оценками, влияющими на их применение.

4. Детально анализируется процесс автоматизации индексирования лексики, как самой трудоемкой операции в индексировании документов. Излагается методика организации словаря термов и проведения индексирования, лучшим образом соответствующая определе-

нию вхождения терма в текст документа и требованиям скорости индексирования. Описанный метод реализован в системе

5. Составлен тезаурус юридической терминологии с широкими семантическими связями между терминами. Приводится алгоритм логического контроля тезаурусов, примененный при составлении упомянутого тезауруса.

6. Излагается новая методика анализа информационно-поисковых систем с целью выявления внутренней структуры множества документов и терминов. Преимуществами предложенного метода являются возможность совместного рассмотрения связей между разного рода объектами и возможность выбора объектов и связей в соответствии с целью исследований. При этом, в отличие от классических методов классификации, не требуется задания вперед ни числа классов, ни уровня различимости. Показывается, что в частном случае этот метод совместим с методами выделения кластеров.

По изложенному в настоящей работе можно сделать следующие выводы.

I. Рассмотрение документального информационного поиска размытым процессом позволяет достаточно гибко представить внутрисистемные связи и придать ИПС часть той гибкости, которой пользуется человек при ручном поиске документов. В то же время сохраняется автоматизируемость всех этапов работы ИПС.

2. Хотя настоящая работа является обобщением скопленного автором опыта составления ИПС в области права и во всем изложении имелась в виду правовая ИПС, конструированная модель является более общей. Принципы построения размытой ИПС применимы во всех предметных областях, где исходные документы представляются текстами на естественном языке и не содержат существенных с точки зрения поиска схем, формул, таблиц и т.д.

3. Предложенная схема информационного поиска является обобщением традиционной схемы, построенной на базе теории множеств, так как соответствующим выбором весовых функций она превращается в традиционную схему.

4. Изложенная модель информационного поиска основывается на операциях над размытыми множествами и отношениями. Определив примененные в настоящей работе операция иным способом, получаются различные варианты размытой ИПС, в которых соблюдается ориентация на некоторую предметную область.

5. Тезаурус следует рассматривать органической частью информационного поиска, которым система пользуется либо автоматически, либо полуавтоматически под контролем пользователя. Составление тезаурусов с широкими терминологическими связями в настоящее время не автоматизируемо, однако автоматический контроль его логичности и систематизирование зафиксированных связей в большой степени помогают составителям.

СПИСОК ЛИТЕРАТУРЫ

2. Официально-документальные материалы

2.I. Указ Государственного комитета науки и техники Совета министров СССР нр. I95 от 30 мая I973 года.

2.2. Указ Коммунистической партии СССР и Совета министров СССР нр. I025 от 23 дек. I970 года.

2.3. Das juristische Informationssystem - Analyse, Planung, Vorschläge. Bericht der Projektsgruppe. - Bundesministerium der Justiz, Karlsruhe, 1972.

2.4. Erster Zwischenbericht über die Arbeit der Projektsgruppe "Juristisches Informationssystem" an dem Bundesministerium der Justiz vom 1. Februar 1971. - Beilage № 5171 zum Bundesanzeiger vom 31. März 1971.

4. Книги

4.I. Айвазян С.А., Бежаева З.И., Староверов О.В. Классификация многомерных наблюдений. - М., I974.

4.2. Кнут Д. Искусство программирования на ЭВМ, Т.I. - Основные алгоритмы. - М.: Мир, I976.

4.3. Кнут Д. Искусство программирования на ЭВМ, Т.3. - Сортировка и поиск. - М.: Мир, I978.

4.4. Кулик А.Н. Информационные сети и языковая совместимость дескрипторных ИПС.- М.: Советское Радио, I977.

4.5. Курбаков К.И. Кодирование и поиск информации в автомати-

ческом словаре. – М.: Советское Радио, I968.

4.6. Михайлов А.И., Черный А.И., Гиляревский Р.С. Основы информатики. – М.: Наука, I968.

4.7. Прянишников Е.А. Автоматизированные системы правовой информации капиталистических стран. – Обзоры по электронной технике. – М.: Электроника, I978.

4.8. Сэлтон Д. Автоматическая обработка, хранение и поиск информации. – М.: Советское Радио, I973.

4.9. Сэлтон Д. Динамические библиотечно-информационные системы. – М.: Мир, I979.

4.I0. Aho, A.V., Hopcroft, J.E., Ullman, J.D. The Design and Analysis of Computer Algorithms. – London-Amsterdam.

4.II. Fritjof, H. Elektronische Datenverarbeitung im Recht. – Berlin: J. Schweitzer Verlag, 1970.

4.I2. Picard, C.F. Theorie der Fragebogen. – Berlin: Academie--Verlag, 1973.

5. Статьи

5.I. Базарнова С.В. Об информативности и терминологичности заглавий журнальных статей по литературоведению и литературной критике. – Теория и методика словарных информационно--поисковых языков по общественным наукам. – М.: Наука, I975, с. 253–256.

5.2. Борщев В.Б., Рохлин Ф.Э. Алгоритм свертывания матрицы с сохранением различия ее строк. – Научно-техническая информация, I963, № II.

5.3. Выханду Л.К. Монотонные методы анализа данных. – Труды ТПИ, I979, № 468, с. I5–26.

5.4. Заде Л.А. Основы нового подхода к анализу сложных систем и принятия решения. - Сб. Математика сегодня. - М.: Знание, I974, сер. 7.

5.5. Иванова Н.С. К вопросу об автоматическом построении тезауруса. - Научно-техническая информация, I969, сер.2, № 6, с. I7-I9.

5.6. Керимов Д.А., Покровский И.Ф. Опыт использования средств кибернетики для автоматизации информационной службы в области права. - Вестник Ленинградского университета, I964, вып.I, № 5, с. I2I-I24.

5.7. Кулль И.Г., Сильдмяэ И.Я., Хелемяэ А.К., Ыим Х.Я. О разработке тезауруса юридических терминов для информационно-поисковой системы. - Сб. Правовая кибернетика. - М.: Наука, I973, с. 54-62.

5.8. Кулль И.Г., Сильдмяэ И.Я., Ээремаа К.А., Нигол Р.П. О применении вычислительных машин в поиске юридической информации. - Ученые записки Тартуского госуниверситета, I967, вып. I99, с. 289-292.

5.9. Кулль И.Г., Ээремаа К.А., Сильдмяэ И.Я. Об одной возможности формализации правовых норм. - Ученые записки Тартуского госуниверситета, I97I, вып. 272, с. I82-I99.

5.I0. Курбаков К.И., Смирнов Р.В. Поиск информации в словаре, основанной на методе сжатия кодов. - Научно-техническая информация, I963, № 2.

5.II. Маршакова В.И. Построение информационно-поискового тезауруса методом дистрибутивно-статистического анализа. - Научно-техническая информация, I977, сер. 2, № 5, с. II-I5.

5.I2. Москвин С.С., Прянишников Е.А., Романов Р.М. Автоматизированные системы правовой информации европейских социалистических стран. - Обзоры по электронной технике, I978, сер. 9, № 4.

5.13. Муллат И.Э. Экстремальные подсистемы монотонных систем I. – Автоматика и телемеханика, 1976, № 5, с. 130–139.

5.14. Муллат И.Э. Экстремальные подсистемы монотонных систем II. – Автоматика и телемеханика, 1976, № 8, с. 169–178.

5.15. Муллат И.Э. Экстремальные подсистемы монотонных систем III. – Автоматика и телемеханика, 1977, № 1, с. 109–119.

5.16. Сидоренко В.Д. Семантическая структура тезауруса: современное состояние и направления ее совершенствования. – Научно-техническая информация, 1976, сер. 2, № 9, с. 3–12.

5.17. Сильдмяэ И.Я., Кулль И.Г., Нигол Р.П., Ээремаа К.А. Информационно-поисковая система для законодательного материала. – Правоведение, 1970, № 4, с. 162–165.

5.18. Сильдмяэ И.Я., Ыйм Х.Я., Ээремаа К.А. Автоматизированная система правовой информации. – Ученые записки Тартуского госуниверситета, 1978, вып. 444, с. 90–103.

5.19. Соколов В.А. Исследование потерь информации и информационного шума в дескрипторных ИПС. – Научно-техническая информация, 1965, № 12.

5.20. Сэлтон Д. Автоматический анализ текстов и поиск документов. – Кибернетический сборник, новая серия, 1975, № 12, с. 150–183.

5.21. Фрид Л.М. Сокращение перебора при минимизации словарных кодов. – Научно-техническая информация, 1969, сер.2, № 9, с. 24–30.

5.22. Чернявский В.С., Лахути Д.Г., Федоров Е.Б. К вопросу об автоматическом построении алгоритмов индексирования для системы "Пусто–Непусто". – Научно-техническая информация, 1969, сер. 2, № 10, с. 24–28.

5.23. Шайкевич А.Я. Дистрибутивно-статистический анализ в семантике. – Принципы и методы семантических исследований. – М.:

Наука, 1976.

5.24. Шемакин Ю.И., Кулик А.Н. О совместимости информационно-поисковых систем и месте политехнического тезауруса в единой информационной сети. - Научно-техническая информация, 1970, сер. 2, № 8, с. 30-34.

5.25. Шрейдер Ю.А. Тезаурусы в информатике и теоретической семантике. - Научно-техническая информация, 1971, сер. 2, № 3, с. 21-24.

5.26. Ээремаа К.А. Выделение наиболее существенных классов данных. - Труды ВЦ ТГУ, 1979, № 43, с. 74-90.

5.27. Ээремаа К.А. Модель ИПС с тезаурусом на базе теории размытых множеств. - Ученые записки Тартуского госуниверситета. Труды по искусственному интеллекту II: Семантика и представление знаний, 1979, с. 145-155.

5.28. Юсупов С.Н. К вопросу об учете особенностей правовой информации при индексировании. - Научно-техническая информация, 1969, сер. 2, № 9, с. 13-16.

5.29. Emard, J.P., Staenberg, J.B. An Overview of Computerized Legal Information Systems. - Law and Computer Technology, 1977, v. 10, N° 1, 91-105.

5.30. Kaasik, Ü., Laugaste, E., Ääremaa, K. Tähtede ja silpide sagedusest eestikeelsetes tekstides. - Keel ja Kirjandus, 1975, N° 1, 21-29.

5.31. Kondos, G.S. JURIS - a Progress Report. - Law and Computer Technology, 1974, v. 7, N° 1, 11-16.

5.32. Ling, R.F. On the Theory and Construction of k-clusters. - The Computer Journal, 1972, v.15, N° 4, 326-332.

5.33. Radecki, T. Fuzzy Set Theoretical Approach to Document Retrieval. - Information Processing and Management, 1979, v.15, 244-259.

5.34. Ragade, R.K., Gupta, M.M. Fuzzy Set Theory: Introduction. - Fuzzy Automata and Decision Process. - North-Holland - - New-York, 1977.

5.35. Riesinger, L. On Fuzzy Thesauri. - Proceedings in Computational Statistics, 1974, 119-127.

5.36. Seppänen, J. Symbolic Association by Word Rotation in Nested Odered Binary Tree Structure. - Helsinki University of Technology, Computation Center, 1976, N° 20.

5.37. Sildmjae, I., Aaremaa, K. Automatizovaný systém vyhledáváni normativnich informaci JURIPS. - Prawny Obzor, Bratislava, 1978, N° 61, 28-37.

5.38. Sildmäe, I., Ääremaa, K. Le système de recherche de l'information juridique JURIPS. - L'Informatique Juridique, 1976, N° 15.

5.39. Zadeh, L.A. Fuzzy Sets. - Information and Control, 1965, v. 8, N° 3, 338-353.

5.40. Zadeh, L.A. Similarity Relations and Fuzzy Orderings. - Information Sciences, 1971, v. 3, N° 1-2, 177-200.

7. Авторефераты

7.I. Москвин С.С. Теоретические проблемы системы правовой информации в СССР. - Автореферат докторской диссертации, М., 1977.

Содержание

I. ВВЕДЕНИЕ 2
I.I. Актуальность автоматизации поиска документов 2
I.2. Постановка задачи и цель работы 3
I.3. Обзор работ по правовым ИПС 7
I.4. Размытое множество, размытое отношение и операции над ними . . . . . IO

2. МОДЕЛИРОВАНИЕ ПРИМЕНЕНИЯ ТЕЗАУРУСА В ИПС I9
2.I. Роль тезауруса в ИПС 20
2.2. Расширение исходного ПОД с помощью тезауруса 27
2.3. Составление дескрипторных классов 33
2.4. Моделирование процесса применения тезауруса в информационном запросе 40

3. АВТОМАТИЗАЦИЯ ИНДЕКСИРОВАНИЯ ЛЕКСИКИ 48
3.I. Сущность индексирования лексики 49
3.2. Распознавание термов в документе 5I
3.2.I. Организация словаря термов 52
3.2.2. Кодирование словаря термов 55
3.3. Индексирование понятиями 60
3.4. Индексирование омонимов 63

4. СОСТАВЛЕНИЕ ИНФОРМАЦИОННО-ПОИСКОВОГО ТЕЗАУРУСА 68
4.I. Автоматизация составления тезауруса 70
4.2. Алгоритм анализа тезауруса 72
4.3. Некоторые результаты анализа тезауруса юридической терминологии 79

5. СОСТАВЛЕНИЕ ТЕМАТИЧЕСКИХ ПОДМНОЖЕСТВ ДОКУМЕНТОВ И КЛЮЧЕВЫХ СЛОВ 84
5.I. Определение ядра монотонной системы 85
5.2. Нахождение тематических классов документов 88
5.3. Связь ядра с кластерами 96

6. РЕЗУЛЬТАТЫ И ВЫВОДЫ . . . . . IOI

СПИСОК ЛИТЕРАТУРЫ IO4