*Паньгин Андрей Александрович,*
*Поздняков Сергей Николаевич*

# ПОИСКОВЫЕ СИСТЕМЫ И ПРОБЛЕМЫ «ПЕДАГОГИЧЕСКОГО ПОИСКА»

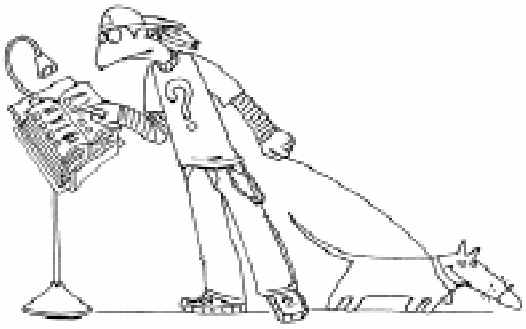
## ВВЕДЕНИЕ

В данной статье дается типология поисковых систем и обсуждаются проблемы создания специализированной системы SciSearch для поддержки работы преподавателей математики и естественнонаучных дисциплин.

## ПОИСК В СЕТИ ИНТЕРНЕТ

По мере развития сети Интернет обостряется парадокс: с увеличением объемов информации вероятность присутствия нужной увеличивается, однако найти ее среди миллионов других документов становится все сложнее и сложнее. Связано это с тем, что наполнение сети громадно, очень разнородно, избыточно, плохо поддается структуризации и управлению и обладает высокой динамикой обновления.

С первых моментов существования Интернета задача нахождения интересующей информации решается при помощи поисковых средств, которые традиционно принято разделять на две категории: каталоги и поисковые машины. Каталоги обычно предусматривают строгое разделение веб-сайтов по рубрикам и пополняются самими владельцами ресурсов, за счет чего оказываются более удобными для направленного поиска в рамках одной тематики, в то время как поисковые машины занимаются пополнением своей базы ресурсов самостоятельно, вследствие чего оказываются более приспособленными к условиям постоянно изменяющегося потока информации в сети, но лишенными возможности фильтровать и структурировать веб-ресурсы. Однако и те, и другие универсальные поисковые средства остаются бессильны в задачах узконаправленного специализированного поиска с повышенными требованиями к качеству информации, что, в частности, характерно для поиска учебных материалов в заданной предметной области преподавателями школ и вузов. В этой ситуации возникает потребность в специальных инструментах, нацеленных на увеличение эффективности поиска при условии, что тематика, характер требований, предъявляемых к искомой информации, и потенциальные пользователи заранее известны.



*...задача нахождения интересующей информации решается при помощи поисковых средств, которые традиционно принято разделять на две категории: каталоги и поисковые машины.*

## ЗАДАЧИ ПОИСКОВОЙ СИСТЕМЫ

Функционирование типичной поисковой системы складывается из комплексного решения нескольких задач с целью составления структурированной базы данных ресурсов Интернет (индекса) и собственно поиска по этой базе данных.

К задачам индексации относятся:

– **обход ресурсов** – обнаружение новых веб-сайтов и документов, а также обновление сведений о ранее посещенных и проиндексированных ресурсах;

– **кластеризация** – объединение в группы близких по содержанию документов;

– **классификация** – определение принадлежности ресурса к заранее определенным категориям.

Задачи поиска включают:

– **формулирование запроса** – преобразование запроса, составленного пользователем, в формат, пригодный для осуществления автоматической выборки из базы данных;

– **фильтрацию документов** – отбор из всего множества документов тех, которые формально удовлетворяют запросу;

– **ранжирование** – определение степени релевантности найденных документов и сортировка по уровню значимости для пользователя.

Для решения любой из задач применяются свои специализированные алгоритмы, а успешная работа поисковой системы в целом зависит от каждого из них. Поскольку система SciSearch была изначально ориентирована на использование баз данных имеющихся поисковых систем, задачи индексации не входили в список исполняемых ею функций, тогда как все задачи поиска являлись для нее актуальными и требовали подходящей реализации в рамках проекта.

*Больше половины поисковых систем в Интернете действуют ..., ставя перед собой лишь цель найти что-нибудь любым способом.*

## МОДЕЛИ ПОИСКА

Одним из ключевых понятий в работе поисковой системы является понятие модели поиска. Больше половины поисковых систем в Интернете действуют вообще без применения каких-либо моделей, ставя перед собой лишь цель найти что-нибудь любым способом. Однако, как только встает вопрос об увеличении объема информации, о повышении качества и скорости поиска, возникает необходимость оперировать математическим аппаратом. В простом случае математическая модель поиска охватывает три составляющие: способ представления документа в системе, способ представления запроса и критерий релевантности – функцию, которая каждой паре документ-запрос сопоставляет некоторое вещественное число – ранг документа, означающий, насколько точно представленный документ соответствует данному запросу.

Перечислим некоторые наиболее популярные модели, использующиеся в настоящее время:

**Дескрипторный поиск** – одна из простейших моделей, в которой документ описывается совокупностью слов или словосочетаний из предметной области (дескрипторов), характеризующих содержание этого документа, причем дескрипторы могут назначаться как вручную экспертами, так и автоматически. Такие системы лучше всего подходят для библиографического поиска или поиска «по каталогу».

**Дублинское ядро** – совокупность метаданных с зафиксированным смыслом. В модели поиска, основанной на дублинском ядре, представлением документа $D$ является множество пар

$$D = \{(N_i, V_i)\},$$

где $N_i$ – имя $i$-го элемента метаданных Дублинского ядра в описании документа; $V_i$ – значение этого элемента метаданных.

Аналогичным образом, запрос представляется в виде

$$Q = \{(N_j, V_j)\},$$

а критерий релевантности будет записываться как $Q \subseteq D$.

**Булевские модели** характеризуются тем, что запрос в них представляется в виде булевского выражения с операторами И, ИЛИ, НЕ. Термами в этих выражениях обычно служат слова, но не обязательно: например, термами могут являться и элементы метаданных, как в Дублинском ядре. В общем случае критерием релевантности служит истинность булевского выражения, заданного в запросе. Несомненным достоинством булевских моделей является их простота, но, вместе с тем, они обладают и рядом недостатков, таких как

– невозможность определения степени релевантности и ранжирования документов;

– сложность использования: далеко не всегда удобно оперировать булевскими операторами при составлении запросов.

**Расширенные булевские модели** как попытка разрешения проблемы ранжирования документов. В этих моделях вводятся обобщения булевских операторов, позволяющие придать повышенный вес документам, в точности удовлетворяющим выражению запроса, и пониженный вес – всем остальным документам.

**Векторные модели** – наиболее распространенные и широко применяемые в наше время. Смысл этих моделей заключается в представлении как документов, так и запросов в виде векторов. Пусть $n$ – количество различных термов во всех документах. Каждому документу сопоставляется $n$-мерный вектор

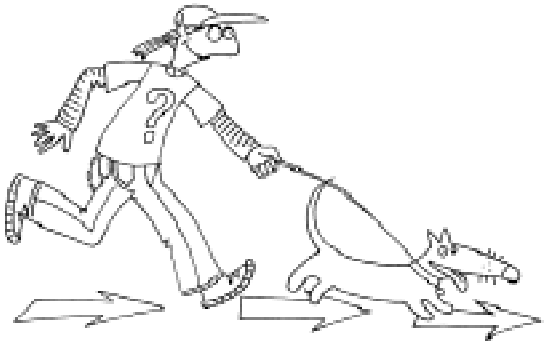
$$d = (w_1, w_2, ..., w_n),$$

где $w_i$ – вес терма $t_i$ в данном документе.

Аналогично запрос представляется в виде

$$q = (v_1, v_2, ..., v_n),$$

а релевантность оценивается как корреляция между векторами $d$ и $q$. Например, корреляция может быть вычислена через скалярное произведение этих векторов. Веса термов могут определяться разными способами, к примеру, как отношение частоты использования данного терма в документе к общему числу термов в данном документе (TF). Более сложные модели учитывают также и обратную встречаемость терма (IDF), то есть насколько редко терм встречается в остальных документах. Зачастую обозначение TF*IDF используется как синоним векторной модели.



*Смысл этих моделей заключается в представлении как документов, так и запросов в виде векторов.*

**Вероятностные модели** основываются на принципе ранжирования документов по убыванию вероятности их релевантности запросу. При этом предполагается наличие набора априорно релевантных документов. Так, вероятность соответствия очередного документа запросу находится из соотношения встречаемости термов документа

*Какими бы ни были методы поиска, невозможно полностью избежать ошибок ...*

в релевантном наборе и в остальной части коллекции (вероятность Байеса).

## КРИТЕРИИ ЭФФЕКТИВНОСТИ

Какими бы ни были методы поиска, невозможно полностью избежать ошибок в результатах, предоставляемых в конечном итоге пользователю. Ошибки разделяются на два типа:

– ошибки первого рода, означающие, что релевантный документ не был найден;

– ошибки второго рода, возникающие в случае отображения среди результатов документа, не соответствующего запросу.

Чтобы научиться судить об эффективности поиска, нужны количественные критерии, которые помогли бы оценить число ошибок первого и второго рода. К таким критериям относятся:

– точность (*precision*) – доля обнаруженных релевантных документов среди результатов поисковой системы;

– полнота (*recall*) – доля найденного релевантного материала среди всех релевантных документов коллекции;

– доля найденной нерелевантной информации (*junk*).

Если $C$ – вся коллекция документов, $A$ – ответ поисковой системы, а $B$ – множество истинно релевантных документов, то эти критерии формально определяются так (рисунок 1):

$$precision = \frac{|A \cap B|}{|A|},$$

$$recall = \frac{|A \cap B|}{|B|},$$

$$junk = \frac{|C \setminus (A \cup B)|}{|C \setminus A|}.$$

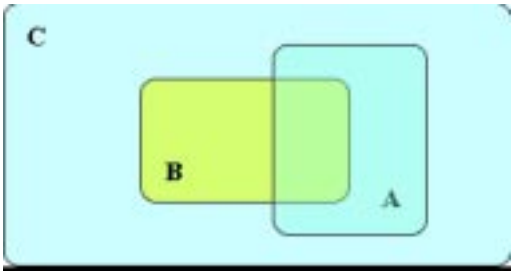



**Рисунок 1.**

Очевидно существует альтернатива между двумя типами ошибок: снижение числа ошибок первого рода приводит к росту количества ошибок второго рода и наоборот. Иными словами, увеличение полноты поиска приводит к уменьшению точности, а повышение точности сказывается на полноте поиска. Если при ошибке второго рода в худшем случае пользователь потеряет время на просмотр ненужного документа, то ошибки первого рода могут привести к более неприятным последствиям и, казалось бы, являются менее желательными. И все же, полностью избавиться от ошибок первого рода нельзя, поскольку в этом случае ценность поиска сводилась бы к нулю за счет большой доли нерелевантных результатов.

Система SciSearch разрешает конфликт между полнотой и точностью, исходя из общего размера ответа поисковой машины, который являлся бы разумным для большинства пользователей. Точнее, эффективными считаются те результаты поиска, которые предоставляют от 5 до 50 ссылок. В частности, это означает, что если результаты содержат более 50 ссылок, то точность поиска можно повысить.

## МЕТОДЫ ТЕМАТИЧЕСКОГО АНАЛИЗА

Поскольку речь идет не о простой поисковой системе, а об интеллектуальной, что в нашем случае означает способность извлекать в некотором виде смысл из текстовой информации в Интернете, то нельзя не упомянуть и про методы тематического анализа текстовой информации. В общем случае тематический анализ заключается в определении степени принадлежности документа к одной или нескольким тематикам. Совокупность имеющихся на сегодняшний день методов такого анализа можно разбить на две большие группы: **лингвистический анализ** и **статистический анализ**.

Лингвистический анализ состоит из четырех взаимодополняющих компонент:

**Лексический анализ**, заключающийся в разборе текстовой информации на отдельные абзацы, предложения, сло-

ва. Это наиболее простой алгоритмически вид анализа.

**Морфологический анализ**, который сводится к распознаванию частей речи каждого слова. Обычно он применяется для приведения слова к его канонической форме.

**Синтаксический анализ**, заключающийся в автоматическом выделении элементов предложения: подлежащего, сказуемого и т.п.

Наконец, **семантический анализ**, целью которого является оценка смыслового содержания текстовой информации. Этот шаг хуже всего поддается формализации. В настоящее время отсутствуют сложившиеся подходы к реализации семантического анализа, что обусловлено исключительной сложностью проблемы.

Суть статистического анализа заключается, как правило, в подсчете частотных характеристик слов для конкретных целей, например, для вычисления весовых коэффициентов ключевых слов. Остановимся на одном из наиболее популярных и эффективных статистических методов, получившем название латентно-семантического анализа.

**Латентно-семантический анализ** (LSA) основывается на сингулярном разложении прямоугольной матрицы, связывающей документы и термы. Элементами этой матрицы являются, к примеру, весовые коэффициенты термов, вычисленные так же, как и в векторной модели. Доказано, что вещественную матрицу $X$ размерности $n*m$ можно разложить в произведение

$$X = USV^T,$$

где $U$ – ортогональная матрица $n*n$, $V$ – ортогональная матрица $m*m$, а $S$ – диагональная матрица $n*m$, элементы $s_{ij}$ которой равны нулю, если $i \neq j$. В этом случае диагональные элементы $(s_{ii})$ называются сингулярными числами матрицы. Известно, что, если оставить лишь первые $k$ по величине сингулярных чисел, приравняв остальные нулю, то мы получим ближайшую аппроксимацию ранга $k$ матрицы $X$:

*Лексический анализ, заключающийся в разборе текстовой информации на отдельные абзацы, предложения, слова.*

$$X \approx X_k = U_k S_k V_k^T$$

Данное свойство позволяет рассматривать вместо исходной матрицы огромной размерности ее приближение небольшого ранга ($k$ = 50...150), причем оставшиеся $k$ сингулярных чисел будут соответствовать «скрытым смыслам» исходного документа. Как еще одно следствие, латентно-семантический анализ позволяет находить даже те тематически близкие документы, которые не содержат слов исходного запроса. Увеличение ранга $k$ приводит к снижению эффективности алгоритма, приближая его к векторным методам, а чрезмерное уменьшение $k$ не позволяет различать нюансы схожих между собой документов.

Одной из проблем тематического анализа информации из глобальной сети Интернет является отсутствие общепринятых стандартов в этой области. Серьезная попытка устранить этот недостаток была предпринята всемирной организацией *W3C*, в результате чего Интернет-сообществу были предложены две новые технологии: *RDF (Resource Definition Framework)* – схема описания ресурсов и *OWL (Web Ontology Language)* – язык описания онтологий. Предполагается, что язык онтологий не только поможет пользователю охарактеризовать содержимое веб-ресурса, но и позволит компьютерам извлекать эту информацию и обмениваться ею между собой автоматически, без участия человека. На сегодняшний день *RDF* и *OWL* имеют статус рекомендации *W3C*. Система *SciSearch*, хотя и призвана эксплуатировать мощь *OWL*, отнюдь

не опирается на его использование, поскольку в настоящее время технология еще не получила повсеместного распространения, и подавляющее большинство Интернет-ресурсов по-прежнему обходятся без нее.

## ЗАКОНЫ ЗИПФА

Многие алгоритмы автоматического анализа текста базируются на законах Зипфа. Зипф обнаружил, что все тексты, составленные человеком, устроены в некотором смысле по единым правилам. А именно, основываясь на постулате, что слова с большим количеством букв встречаются в тексте реже коротких слов, Зипф вывел два универсальных закона:

**Первый закон «ранг-частота».**

Посчитаем количество повторений (частоту) всех слов произвольного документа. Отсортировав частоты по убыванию, выпишем значения по порядку, при этом, когда несколько слов имеют одну и ту же частоту, значение записывается лишь один раз. Рангом частоты назовем порядковый номер соответствующего значения в списке. Таким образом, наиболее часто встречающиеся слова будут иметь ранг 1, следующие за ними – ранг 2 и т. д. Закономерность, обнаруженная Зипфом, состоит в том, что если частоту вхождения слова умножить на ранг этой частоты, то полученное значение будет приблизительно константой. Иначе говоря, если самое частое слово встречается в тексте 100 раз, то второе по частоте слово будет входить около 50 раз, но навряд ли 90 или 10.

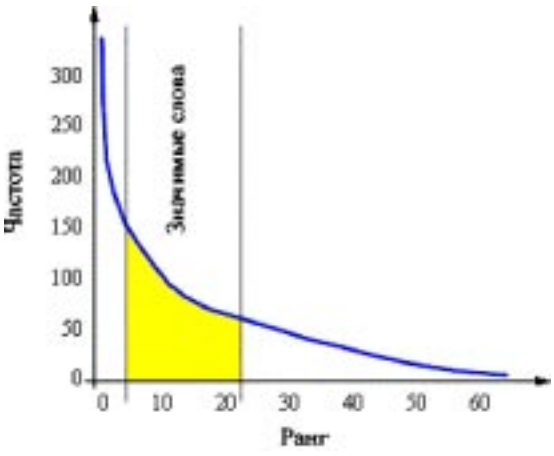



**Рисунок 2.**

**Второй закон «количество-частота».**

Первый закон никак не учитывал количество разных слов с одинаковой частотой, но, оказывается, частота и количество слов, входящих в текст с этой частотой, тоже связаны между собой. Если построить график зависимости количества слов от частоты, то получившаяся кривая будет сохранять свой вид для любых текстов.

Как же все-таки законы Зипфа помогают выявить слова, отражающие смысл текста? Построим график зависимости ранга и частоты по первому закону. Исследования показывают, что наиболее значимые слова будут лежать в средней части графика (см. рисунок 2). Это наблюдение легко объяснить: ведь наиболее частые слова, как правило, являются общими для всех текстов и не несут смысловой нагрузки. Сюда же входят и стоп-слова, то есть те, которые употребляются не для передачи информации, а для связи остальных частей предложения: предлоги, союзы, местоимения и т. п. Слова, встречающиеся слишком редко, тоже не играют ключевой роли в формировании смысла.

Законы Зипфа играют большую роль в задачах тематического анализа информации, поскольку при небольших вычислительных затратах позволяют с большой достоверностью выделить из текста наиболее значимые по смыслу слова. Именно это свойство использует система *SciSearch* для формирования оптимальных запросов к поисковым машинам в задачах поиска документа по образцу. Таким образом, стратегия составления запросов описывается следующей последовательностью действий:

1) выбор текста-источника с описанием проблемы;

2) удаление из текста стоп-слов;

3) вычисление частоты вхождения каждого термина без учета морфологии слов;

4) определение среднего диапазона частот;

5) выбор порядка 10 слов из найденного диапазона;

6) составление запроса к поисковой системе из выбранных слов с помощью логической связки ИЛИ.

Итак, законы Зипфа позволяют выработать стратегию для формирования первоначального запроса к поисковой системе, в котором с неплохим приближением будет заключена формулировка потребности пользователя. Однако, как показывает практика, крайне редко удается обойтись единственным запросом для получения наиболее адекватных результатов. В зависимости от количества ресурсов, предложенных поисковой машиной, и степени их релевантности, возникает необходимость модифицировать исходный запрос тем или иным образом. Как правило, такие модификации сводятся к применению одного из нижеперечисленных правил или их совокупности:

1) уточнение запроса путем добавления новых терминов – в случае, если поисковая машина вернула неоправданно большое количество ссылок, либо среди результатов присутствует множество документов из другой тематики;

2) ослабление запроса посредством исключения терминов, если не нашлось документов, в точности соответствующих запросу;

3) расширение запроса.

## ТЕЗАУРУС КАК СРЕДСТВО РАСШИРЕНИЯ ПОИСКОВЫХ ЗАПРОСОВ

Каким же образом следует выбирать термины для модификации запроса? Для ответа на этот вопрос на помощь приходит концепция тезауруса.

Проще всего тезаурус представить в виде графа, в котором вершинами являются термины, а ребрами – отношения между терминами. Отношения могут быть как общими (не зависящими от тематики поиска), например, «синоним», «часть целого», «представитель», так и специализированными. В частности, в области математики специализированными отношениями могут быть «выводится из», «доказывается через» и т. п.

*...уточнение запроса путем добавления новых терминов ... если поисковая машина вернула неоправданно большое количество ссылок,*

Определенные неудобства доставляет неоднозначность отображения предметной области на графовую структуру тезауруса, из чего ясно, что составление тезауруса – ответственная работа, которая должна выполняться специалистами в данной предметной области.

На рисунке 3 приведен фрагмент тезауруса из области «математика».

Для расширения запроса при помощи тезауруса выбираются термины, связанные (не обязательно напрямую) с терминами запроса. Типы отношений между понятиями тезауруса позволяют выбрать подходящую логическую связку. Отношения мо-

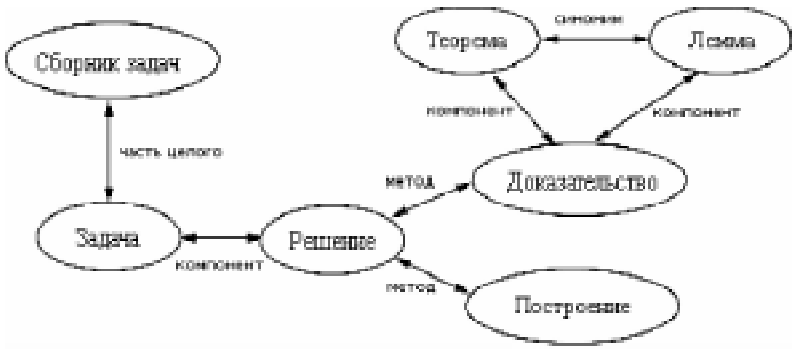



**Рисунок 3.**

гут интерпретироваться и другими способами, если такая интерпретация поддерживается поисковой системой, например, как максимальное расстояние в словах между связанными понятиями. Проиллюстрируем сказанное на примере запроса «задачи с решениями», который на основе предложенного выше фрагмента тезауруса может быть модифицирован следующим образом:

(«сборник задач» | задача) & (решение | доказательство | построение)

Заметим, что для описания тезауруса естественным образом подходит язык OWL, что дает возможность осуществлять его автоматическую обработку стандартными средствами, поддерживающими этот формат.

## ПЕДАГОГИЧЕСКАЯ СОСТАВЛЯЮЩАЯ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА

Не будем забывать, что разрабатываемая поисковая система SciSearch изначально ориентирована на вполне определенный круг пользователей и ставит перед собой задачу находить и анализировать в первую очередь учебные материалы. Хотя эти условия и ограничивают область применения системы, но, вместе с тем, накладывают дополнительные требования на ее функциональность. Поскольку материалы, размещаемые в сети Интернет, как правило, не подвергаются проверке, существует опасность предоставления некачественной информации, не соответствующей нормам, принятым в сфере образования, либо не согласующейся с взглядами на предмет конкретного преподавателя. Заставляя пользователя самостоятельно судить о качестве всех найденных материалов, мы уменьшаем ценность системы с точки зрения педагога.

*...SciSearch ... ставит перед собой задачу находить и анализировать в первую очередь учебные материалы.*

Представим, например, содержание какого-нибудь виртуального учебного пособия, построенного на различных ресурсах сети Интернет. Обычно текст такого пособия разбивается на отдельные блоки – лекции или уроки. При этом порядок изложения материала вовсе не обязан совпадать с порядком, в котором он был написан, более того, отдельные главы могут быть созданы разными авторами или взяты из различных источников. Нередко содержание курса изменяется со временем, подвергается ревизии и т.п. Все эти причины приводят к неизбежному появлению различного рода логических ошибок. Вот примеры некоторых из них:

– разные блоки используют недостаточно согласованный между собой тезаурус;

– в одном из уроков вводится некоторое понятие, а в другом уроке оно вводится повторно;

– новое понятие *A* выводится из понятия B в одной лекции, в то время как в другой лекции понятие *B* основывается на *A*.

Человеку трудно обнаружить подобного рода ошибки, не прибегая к полному прочтению материала, поэтому предлагается переложить отчасти эту функцию на компьютер.

Каждому уроку $L_i$ сопоставим набор $W(L_i)$ используемых в нем терминов. $W(L_i)$ разбивается на два подмножества: $W(L_i)^-$ – множество терминов, введенных ранее, до использования в уроке $L_i$, и $W(L_i)^+$ – множество понятий, определяемых в данном уроке. Ясно, что

$$W(L_i) = W(L_i)^- \cup W(L_i)^+ \text{ и}$$

$$W(L_i)^- \cap W(L_i)^+ = \varnothing.$$

Далее, назовем урок $L_i$ предшествующим уроку $L_j$, если $W(L_i)^+ \cap W(L_j)^- \neq \varnothing$.

Связав уроки отношением предшествования, мы получим ориентированный граф. Если в нем обнаружатся циклы, то отсюда будет следовать, что в тексте учебника присутствуют логические ошибки.

Вообще говоря, автоматический анализ структуры учебного материала является отдельной интересной задачей, решение которой открывает принципиально новые возможности для педагогов. Известно, например, что один и тот же учебный курс может быть построен разными способами. Скажем, в рамках одной методики понятие *A* базируется на понятии *B*, а в другой – понятие *B* вводится через *A*. Естественно, при поиске материалов к заданному курсу преподавателя будут интересовать, как правило, лишь те материалы, которые не противоречат используемой им методике преподавания.

Один из способов «научить» компьютер отличать учебные пособия разной структуры заключается в построении ориентированных графов, отражающих иерархические зависимости между понятиями (определениями, теоремами и т. п.) с дальнейшим поиском общего подмножества двух или более графов. Поскольку построить такой граф – задача трудоемкая, предлагается решить ее приближенно, основываясь на частичном порядке терминов, который

*...автоматический анализ структуры учебного материала является отдельной интересной задачей...*

можно определить, например, из предметного указателя в конце учебника, где каждому термину сопоставлен номер страницы, на которой данный термин употребляется первый раз.

К другим задачам, связанным со структурой учебного материала, относятся задачи классификации похожих по структуре учебников, подбор материалов к учебнику заданной иерархии, соединения нескольких учебных пособий в одно и т. д.

*Работа выполняется при подержке Российского гуманитарного научного фонда: грант 05-06-06271а.*

***Паньгин Андрей Александрович,***
***аспирант математико-механического факультета Санкт-Петербургского университета,***

***Поздняков Сергей Николаевич,***
***профессор кафедры ВМ-2 СПбГТЭУ (ЛЭТИ).***