Михаил Георгиевич Мальковский
Татьяна Юрьевна Грацианова
Ирина Николаевна Полякова

Прикладное программное обеспечение: системы автоматической обработки текстов

1. Сферы применения систем автоматической обработки текстов

Системы ***автоматической обработки текста*** (т.е. переработки одного вида текста в памяти ЭВМ в другой) по выполняемым функциям (входной и выходной информации) можно классифицировать следующим образом:

Язык входного текста

Язык выходного текста

1

Естественный-1

Естественный-2

2

Искусственный

Естественный

3

Естественный

Искусственный / Естественный

4

Естественный

Естественный + Искусственный

К системам первого типа относятся программы машинного перевода, получающие текст на некотором естественном языке и перерабатывающие его в текст на другом естественном языке. Второй тип - системы генерации (синтеза) текстов по некоторому формальному описанию. Системы третьего типа, наоборот, перерабатывают текст на естественном языке в текст на искусственном (индексирование, извлечение смыслового содержания) или в другой текст на естественном языке (реферирование). К последнему классу отнесем программы, занимающиеся проверкой текста, написанного на естественном языке. Они в результате своей работы либо исправляют входной текст автоматически, либо

формируют некоторый протокол замечаний.

Естественный язык - сложная, многоплановая система, с множеством правил, внутренних связей, имеющая отношение ко всем аспектам деятельности человека. Точность и правильность работы программ определяется глубиной анализа. Достаточно глубокий анализ пока достигается только для определенных узких предметных областей (из-за специфичности подъязыка такой области: в каждой области свои термины, специфические семантические отношения и т.п.).

Для создания систем, работающих со всем естественным языком без потери глубины анализа, в настоящий момент не хватает либо технических возможностей (быстродействия, памяти), либо теоретической базы (например, пока нет даже единой схемы достаточно полного, глубокого и непротиворечивого описания семантики естественного языка). Однако в коммерческих системах, ввиду того, что предназначаются они для большого количества пользователей, разных предметных областей, принята концепция поверхностного анализа, к тому же и производится такой анализ значительно быстрее. Дальнейшее продвижение вперед, использование естественного языка в практических областях невозможно без оснащения этих систем обширными и глубокими (с точки зрения охвата различных явлений языка) описаниями и моделями, созданными лингвистами-профессионалами.

Эта тенденция прогнозируется многими исследователями и прослеживается на примере развития АОТ-систем, уже в наши дни представляющих коммерческий интерес и использующихся при решении следующих прикладных задач:

1. Machine Translation and Translation Aids - машинный перевод;
2. Text Generation - генерация текста;
3. Localization and Internationalization - локализация и интернационализация;
4. Controlled Language - работа на ограниченном языке;
5. Word Processing and Spelling Correction - создание текстовых документов (ввод, редактирование, исправление ошибок)
6. Information Retrieval - информационный поиск и связанные с ним задачи.

Отметим, что это деление несколько условное, и в реальных системах часто встречается объединение функций. Так, для машинного перевода требуется генерация текста, а при исправлении ошибок приходится заниматься поиском вариантов словоформы и т.д.

1.1. Машинный перевод

Исторически ***машинный перевод*** является первой попыткой использования компьютеров для решения невычислительных задач (знаменитый Джорджтаунский эксперимент в США в 1954 г.; работы по машинному переводу в СССР, начавшиеся в 1954 г.). Развитие электронной техники, рост объема памяти и производительности компьютеров создавали иллюзию быстрого решения этой задачи. Идея захватила воображение ученых и администраторов. Практическая цель была простой: загрузить в память компьютера максимально возможный словарь и с его помощью из иноязычных текстов получать текст на родном языке в удобочитаемом виде. Однако первоначальная эйфория по поводу того, что столь трудоемкую работу можно поручить ЭВМ, сменилась разочарованием в связи с абсолютной непригодностью получаемых текстов. Приведем в качестве примера результаты работы одной из современных коммерческих систем перевода. Предложим ей перевести народное английское стихотворение, известное нам в переводе "Робин-Бобин" (текст этот очень простой, московские дети изучают его в начальной школе):

Robin, Robin, what a man!
He eats as much as no one can.
He ate a lot of fish, he ate a lot of meat.
He ate a lot of ice-cream and a sweet.
He ate a lot of porridge and ten eggs

And all the cookies Mother had.
He drank a lot of juice, he ate a cake
Then said: "I have a stomach-ache"

Малиновка, Малиновка, какой человек!
Он ест насколько никто не может.
Он съел много рыб, он съел много мяс.
Он съел много ледяных-сливки и сладкий.
Он съел много каша и десять яйцо
И вся Мать повары имела.
Он пил много соков, он съел торт
Затем сказал: "У меня есть желудок- боль"

Сравним с художественным переводом К.Чуковского:

Робин Бобин Барабек
Скушал сорок человек.
И корову, и быка,
И кривого мясника,
И телегу, и дугу,
И метлу, и кочергу.
Скушал церковь, скушал дом,
И кузницу с кузнецом,
А потом и говорит:
– У меня живот болит!

Следующий пример показывает неустойчивость системы машинного перевода при обработке неоднозначностей. Два предложения по отдельности "*Flyer flies* ." и "*Flyers fly.* " переводятся "*Летчик летает* ." и "*Летчики летают.* ", если же из тех же словосочетаний составить одно предложение "*Flyer flies and flyers fly* " получаем "*Летчик летает и муха летчиков.* ".

Конечно, системы, настроенные на определенную предметную область, дают гораздо более приемлемые результаты. Однако в этом случае системы перевода получаются очень узко ориентированными, и попытка использовать их даже в смежных предметных областях дает совершенно непредсказуемые результаты. Подобные эксперименты даже распространены среди любителей пошутить: инструкция по эксплуатации манипулятора-мыши, переведенная с английского языка на русский системой автоматического перевода, использующей специализированный медицинский словарь, превращается в описание всевозможных издевательств над несчастным маленьким грызуном.

Возникают эти проблемы из-за принципиально разных подходов к переводу человека и машины. Квалифицированный переводчик понимает смысл текста и **пересказывает** его на другом языке словами и стилем, максимально близкими к оригиналу. Для компьютера этот путь выливается в решение двух задач: 1) перевод текста в некоторое внутреннее семантическое представление и 2) генерация по этому представлению текста на другом языке. Поскольку не только не решена сама по себе ни одна из этих задач, а нет даже общепринятой концепции семантического представления текстов, при автоматическом переводе приходится фактически делать "подстрочник", заменяя по отдельности слова одного языка на слова другого и пытаясь после этого придать получившемуся предложению некоторую синтаксическую согласованность. Смысл при этом может быть искажен или безвозвратно утерян.

Более реалистичными являются попытки создать системы ***автоматизированного перевода*** - программы, которые не берут на себя полностью весь перевод, а лишь помогают человеку-переводчику справиться с некоторыми трудностями (Computer Aided Translation). Одним из примеров таких систем является Eurolang Optimizer. Его можно рассматривать как нечто переходное между компьютерным словарем и программой-переводчиком, как некий набор предметно-ориентированных глоссариев, снабженный интерфейсом для удобства переводчика: предлагается несколько вариантов перевода, выделенные разными цветами в зависимости от условий применимости; переводчик может с помощью меню определенным образом настраивать словари для более быстрого и правильного выбора нужного эквивалента.

Подобные программные средства могут помочь в решении проблем, связанных с терминологией и вообще со знаниями переводчика о предметной области: одни и те же слова могут по-разному переводиться в зависимости от того, о каком предмете идет речь.

Автоматически может быть решена проблема согласованности. Понятно, что согласованность важна в рамках одного документа: один и тот же термин, даже если его без потери смысла можно перевести несколькими словосочетаниями, должен переводиться одинаково на протяжении всего документа. Однако еще более важной является согласованность в широком смысле - разработка и применение единой концепции интерпретации одного и того же термина на разных языках (скажем, американский разработчик программного обеспечения может быть недоволен, что термин *dialog box* переводится на итальянский как *finestra* (окно) и как *boite* (коробка, ящик) на французский). Ошибки, возникающие вследствие нарушения согласованности, являются серьезной проблемой, так как, имея только текст-результат перевода, уже невозможно установить, какие термины в оригинале были одинаковыми, а теперь переведены по-разному (в отличие от орфографических ошибок, которые исправить никогда не поздно).

В последнее время также появляются автоматизированные системы "доперевода" или "перевода изменений". Их возникновение связано с тем, что большинство технических текстов (описания, инструкции) не являются целиком новыми (как и явления, продукты, механизмы и т.п., ими описываемые), а содержат в себе лишь некоторые изменения, связанные, например, с усовершенствованием конструкции. Система "доперевода" извлекает из памяти знакомые предложения, а новые куски предлагает переводчику. Заметим, что такой человеко-машинный способ генерации новых текстов также помогает согласованности в стиле и терминологии при переходе от одной версии к другой.

Развитием систем подобного вида можно считать канадскую (Канада - двуязычная страна, постоянно сталкивающаяся с проблемой перевода на государственном уровне) систему генерации прогнозов погоды Forecast Generator (FOG). Можно считать, что в ней перевод полностью заменен генерацией текстов. В памяти системы хранится 20 миллионов слов и словосочетаний, связанных с прогнозами погоды, что позволяет генерировать как английский, так и французский вариант непосредственно из базы данных. Конечно, успешная работа этой системы в значительной мере объясняется ограниченной природой текстов: сообщения о погоде являются классическим примером подъязыка. Ограниченность словаря, грамматики и семантики дает возможность достичь отличных результатов сравнительно простыми методами.

1.2. Генерация текста

С необходимостью генерации хотя бы простейших фраз разработчики практических систем столкнулись еще на заре их создания. Даже в столь примитивно организованной (в плане дружественности пользовательского интерфейса) среде, как DOS, при попытке сгенерировать стандартное сообщение о количестве скопированных файлов мы сталкиваемся с проблемой построения фразы: в зависимости от этого количества необходимо использовать разные слова (в английской версии *file* в случае одного файла и *files* , если больше; в

русской - и того хуже: могут встретиться варианты *файл* , *файла* и *файлов* , причем правила, в каком случае какой из них использовать, достаточно сложны).

По степени сложности и выразительности существующие методы генерации сообщений принято подразделять на 4 класса (часто используются комбинации методов). Рассмотрим их на примере генерации сообщений о копировании файлов.

1) ***Canned-based methods***

Неизменяющийся шаблон - просто печать строки символов без каких-либо изменений.

Для генерации сообщений создаются таблицы шаблонов, которые будут выдаваться в зависимости от ситуации. В нашем варианте при копировании одного файла будет напечатана первая строка таблицы:

1 file copied,

а в случае, например, трех - третья:

3 files copied

2) ***Template-based methods***

Изменяющийся шаблон - бесконтекстная вставка слов в образец-строку (именно этот метод используется в MS-DOS):

Шаблон: вЂ№ЧисловЂє file(s) copied

может быть использован для генерации сообщений:

0 file(s) copied,

1 file(s) copied,

2 file(s) copied

3) *Phrase-based methods*

Контекстная вставка.

В зависимости от вида сообщения (контекста) шаблон может быть несколько изменен. Скажем, система может распознавать, с каким окончанием писать слово *file* в зависимости от их количества.

Шаблон: вЂ№ЧисловЂє вЂ№ОпределениевЂє вЂ№file/files при =1, вЂє1вЂє

вЂ№Глагол: время - прош.вЂє

может использоваться для генерации сообщений:

1 file copied,

2 marked files copied,

2 marked files deleted

4) *Feature-based methods*

Синтез сообщения на основе набора свойств (грамматических признаков).

Это наиболее сложный метод, он требует привлечения обширных лингвистических знаний, но, в то же время, он и наиболее привлекателен. Предложение определяется набором характеристик составляющих его слов (например, наличие/отсутствие отрицания, настоящее/прошедшее время) и правилами их сочетаемости.

Шаблон: вЂ№ЧисловЂє вЂ№ОпределениевЂє вЂ№file/files при =1, вЂє1вЂє

вЂ№Глагол: время - любоевЂє

позволяет генерировать сообщения:

1 file should be copied,

1 file was copied,

2 marked files were copied

Понятно, что генерация логически связных, целостных текстов является гораздо более сложной задачей: к правилам построения предложений добавляются правила их сочетаемости, правила развития сюжета, соблюдения стиля и т.п. Ввиду невозможности их полной формализации задачу генерации полноценных художественных текстов можно считать на настоящий момент неразрешимой. Однако для некоторых специализированных технических текстов эти правила строго оговорены некоторыми стандартами, немногочисленны и поэтому поддаются формализации. Примером таких текстов могут служить различные инструкции, техническая документация, тем более задача ее

автоматической генерации давно назрела.

На Западе уже давно разработка документации превратилась в особую подотрасль разработки любых достаточно сложных систем (в том числе программного обеспечения). Сопроводительная техническая документация весьма разнообразна: руководство пользователя, руководство для менеджера (администратора) системы, руководство по монтажу (инсталляции) и первичному запуску, руководство по эксплуатации, руководство по интегрированию системы с другими устройствами (программами), проектные материалы и т.д. Однако часто пользователь не получает своевременно и в полном объеме необходимый ему материал, соответствующий используемой им версии системы. Это можно объяснить двумя причинами. Во-первых (субъективная причина), подготовка документации - это дополнительная работа, требующая дополнительного времени и дополнительных навыков (разработчику трудно изложить требуемое на понятном рядовому пользователю языке, остальным же надо сначала детально изучить систему). Во-вторых (объективная причина), документация устаревает по ходу модернизации системы.

Поиски решения этих проблем привели в свое время к появлению новой профессии "технического писателя". Однако понятно, что привлечение дополнительных работников ведет к удорожанию продукта. Поэтому в последние годы появились практические системы, осуществляющие помощь в разработке документации, вплоть до ее автоматической генерации. Форма и содержание документации часто выбирается не столько из соображений удобства и полезности для пользователя, сколько из соображений простоты ее создания.

Документация, как правило, содержит графическую и текстовую части. Графическую часть проще сформировать, однако без текстовой не обойтись: в ней описывается семантика продукта (назначение, технические данные, ограничения, детализация работы в разных режимах). Очевидно, что качественная система должна генерировать текст, правильный с точки зрения грамматики и синтаксиса естественного языка. Поскольку предметная область точно определена, а техническая документация составляется по определенным строго заданным правилам, степень формализации в постановке данной задачи существенно выше, чем в задаче машинного перевода, что позволяет надеяться на более высокие результаты.

1.3. Локализация и интернационализация

Для того чтобы иметь успех на международном рынке, программные продукты должны быть локализованы, т.е. приспособлены к культурным и языковым нормам потенциальных покупателей.

Для многих программных приложений локализация может быть сравнительно простой, когда основная программа (алгоритм) изменяется незначительно. Конечно, опции меню, сообщения об ошибках, экранные подсказки и другие текстовые строки, вставленные в программу, должны переводиться, но это не создает особых проблем, если при разработке приложения была предусмотрена возможность локализации. Для решения этой задачи программный код и текст должны быть разделены. По установленному стандарту текстовые строки оформляются в отдельном файле, вызываемом из программы. Таким способом текстовые строки можно переводить, не затрагивая исходный код.

Подобные принципы облегчения локализации возможны не для всех приложений. Системы, в которых естественный язык используется не только для формирования сообщений на экране, но и является предметом деятельности самой системы (например, программы-автокорректоры), поддаются локализации с большим трудом. Здесь могут потребоваться большие специализированные словари и полная переработка алгоритмов. Часто эта задача настолько сложна, что разработчик ею заниматься не может, и проблема локализации приложений является заботой пользователя-носителя языка.

В идеале для нашего многоязычного мира программные средства должны быть интернациональными; пользователь, купив версию программы для некоторого языка, не должен покупать другую версию для другого. Назрела необходимость иметь программные

средства, позволяющие автоматически настраивать приложение на заданный язык. Пока мы довольно далеки от этой цели, но работы в этой области ведутся с большой интенсивностью, особенно в Европе, где в связи с образованием Европейского Союза возникает необходимость вести дела и документацию на всех официальных и некотором количестве неофициальных языков.

1.4. Работа на ограниченном языке

Одним из способов разрешения проблем, связанных с обработкой естественного языка, является упрощение и некоторая формализация самих текстов: использование ограниченного языка (подмножества языка). Под ограниченным понимается упрощенный язык, использующий ограниченный словарь, грамматику, строго определенные несложные синтаксические конструкции. Обычно в нем запрещаются длинные предложения, длинные цепочки существительных (типа "*решение проблемы разработки систем перевода на базе представления текста в виде последовательности предложений* ..."), не используются пассивные и негативные конструкции, вводятся строгие правила использования терминов. Тексты должны соответствовать одному из стандартных стилей или даже быть составлены по определенному шаблону, принятому в данной предметной области для документов подобного рода.

Эти правила не являются современным изобретением: именно их обычно применяют при написании технической документации. Достаточно "древним" примером ограниченного языка является "Бэйсик Инглиш", введенный англичанами для общения с туземным населением в колониях. Неожиданно он оказался полезен и для общения самих туземцев друг с другом: колонизация ввела в их быт множество предметов и понятий, просто не имеющих названий в их родных языках. Забавно, что через много лет при "колонизации" Европы и всего мира англоязычными техническими средствами используются практически те же методы. Например, все специалисты в области компьютерной техники пользуются английскими терминами (*файл , принтер* и т.д.), не пытаясь подыскать эквивалент на родном языке, и мы по-русски говорим *word для windows* , а не *слово для окон* .

Применение ограниченного языка делает документ более понятным, удобным для восприятия, он становится легче для переводчиков, поскольку дает меньше возможностей для неоднозначного толкования: такой документ легче составить автору, не являющемуся носителем языка документа. Правительства, особенно в Европе, начинают вводить стандарты на подготовку документации, нормы, по которым требуется использование ограниченных языков, особенно в международной торговле. В связи с этим возникает потребность автоматизации проверки соответствия текста правилам ограниченного языка; появляется задача создания систем, осуществляющих перевод с естественного языка на ограниченный.

Boeing, Caterpillar и несколько других компаний призвали вести всю документацию только на ограниченном языке. Ими разработана система Boeing Simplified English Checker для проверки соответствия текстов различным промышленным стандартам и государственным нормам. На ее базе создается программа Clearcheck, не только контролирующая правильность текста на ограниченном языке, но и исправляющая ошибки.

Некоторые разработчики прогнозируют создание систем с использованием ограниченных языков, в которых полный и корректный перевод документации будет производиться без вмешательства человека.

1.5. Создание текстовых документов (ввод, редактирование, исправление ошибок)

Нет необходимости говорить о многообразии систем для подготовки текстовых документов: текстовых редакторов, издательских систем и т.п. Они прочно вошли в нашу

жизнь, без них не может обойтись ни один пользователь и ни одна область деятельности. Более того, создание текстовых документов - одна из основных сфер применения персональных компьютеров. Использование текстовых редакторов обусловлено не только тем, что они облегчают работу, но и тем, что в последнее время во многих сферах деятельности введены стандарты на подготовку текстов, основанные на применении определенных редакторов.

В отличие от машинного перевода разработка систем редактирования текстов еще на заре своего развития, в 60-е годы, считалась коммерчески перспективной прикладной областью. В настоящее время рынок перенасыщен подобными системами; среди их создателей существует жесткая конкуренция, поэтому при введении одним из поставщиков каких-либо новых возможностей (например, проверка стиля) остальные вынуждены вводить в свои системы нечто подобное. Одним из первых массовых нововведений стало включение в состав текстового редактора программ проверки правописания и внесения необходимых исправлений - ***автокорректоров*** . Чтобы придать своему продукту новые коммерчески перспективные свойства, создатели вынуждены все больше использовать лингвистические знания, применять методы морфологического и синтаксического анализа. На очереди - создание систем, выполняющих функции научного редактора, т.е. осуществляющих литературную и научную правку текстов, способных производить сложное автоматизированное редактирование текстов на естественном языке.

Проверка текста в таких системах может вестись в режиме "off-line" - когда формируется протокол замечаний по тексту, либо в режиме "on-line" - когда исправление ошибок ведется по мере их обнаружения (возможно, после получения соответствующего подтверждения от пользователя). При обнаружении ошибки система может предложить вариант ее исправления (при наличии нескольких вариантов - их упорядоченный список). Замечания по тексту также могут носить различный характер. Они могут быть локальными (указывается фрагмент текста с ошибкой) и глобальными (выдается диагностическое сообщение, касающееся всего текста, например: "данный текст труден для восприятия"). В третьей главе мы рассмотрим подробнее проблемы создания систем подобного рода.

1.6. Поиск информации

Не вызывает сомнений необходимость автоматизации поиска заданных текстовых фрагментов в текстах на естественном языке.

Однако часто даже при поиске информации другого рода (например, аудио- и видео-) работа на самом деле ведется с описаниями на естественном языке (например, для организации поиска фотографий необходимо снабдить каждую из них набором словесных характеристик типа "портрет, профиль, полный рост, женщина", "пейзаж, лес, осень" и т.п.).

В последних разработках классических систем поиска текста основное внимание уделяется дополнению их разнообразными средствами текстовой обработки, что приводит к расширению возможностей и облегчению работы для пользователя-непрофессионала.

Применение компьютеров не только ускоряет создание и обработку документов, но и чрезвычайно стимулирует рост их количества и объема. Очень многие пользователи регулярно сталкиваются с необходимостью быстро просматривать большой объем документов и выбирать из них действительно нужные. Эта задача возникает при работе с текстовыми базами данных, с электронной почтой, при поиске в Интернете. Сократить количество просматриваемых документов могут помочь системы ***категоризации*** . Большой поток входных документов эти системы распределяют по небольшому количеству классов. При категоризации могут учитываться как чисто внешние показатели документов (объем, расширение имени соответствующего файла и т.п.), так и их содержательные характеристики (название, фамилия автора, ключевые слова), которые могут позволить отнести текст к той или иной тематической рубрике. В последнем случае мы имеем дело с ***рубрицированием*** текстов.

Часто бывает, что в крупных организациях, особенно государственных, правила делопроизводства предписывают сопровождать каждый документ кратким описанием или набором ключевых слов. Во всех указанных случаях была бы весьма полезна возможность автоматически составлять сжатые описания содержания документов - рефераты.

К сожалению, автоматические методы не настолько совершенны, чтобы создать полноценный реферат путем генерации предложений текста. Однако уже сейчас возможно ***автоматическое реферирование*** - составление более или менее информативных и связных рефератов заданного объема (квазирефератов**)** - путем выбора информативных предложений из исходного текста, а также выделение достаточно представительного списка ключевых слов.

В качестве ключевых слов система может выбирать слова, наиболее часто встречающиеся в тексте (и являющиеся при этом информативными, т.е. не предлоги, союзы и проч.), либо использовать для отбора какие-либо синтактико-семантические признаки (из фрагмента: *"Определение. Интегралом ... называется ..."* можно заключить, что *интеграл* - ключевое слово).

При реферировании из текста отбираются предложения, в наибольшей степени характеризующие его содержание. Таковыми могут считаться, например, предложения, содержащие ключевые слова (чем больше, тем лучше), либо отобранные по некоторым особым признакам. Размер реферата (коэффициент сжатия) или количество ключевых слов задается пользователем. Результатом работы такой системы может являться некоторый новый текстовый документ (реферат или набор ключевых слов) или же данный документ, в котором ключевые слова или наиболее информативные предложения выделены по тексту.

В главе 4 мы рассмотрим проблемы информационного поиска подробнее.

2. Лингвистическое обеспечение систем автоматической обработки текстов

Один из главных путей развития функциональных возможностей прикладных АОТ-систем и повышения качества их работы - создание и внедрение более полных и точных моделей естественных языков, более совершенных алгоритмов анализа и синтеза текста. В данной главе мы рассмотрим некоторые проблемы построения, формализации и компьютерной реализации моделей естественного языка на примере русской морфологии (словоизменения).

2.1. Лингвистические банки данных

Под ***лингвистическими банками данных*** (ЛБД) понимаются представленные в электронной форме языковые источники (корпусы текстов) и лингвистические описания. Отметим, что в наше время, в ситуации, когда надежность работы систем оптического распознавания близка (на хороших по качеству печатных текстах) к 100%, в электронную форму легко переводимы и традиционные источники информации о языке. Поэтому можно считать, что в ЛБД можно перевести любые полиграфические источники: тексты на том или ином естественном языке, словари, справочники, книги по лингвистике. Спектр ЛБД достаточно широк: это как необработанные ("сырые") корпусы текстов, так и тексты с некоторыми добавлениями, например грамматическими характеристиками слов, стилистическими пометами (разговорное, специальное и т.п.), или описаниями синтаксической структуры предложений. Сюда также входят разнообразные компьютерные словари: частотные, грамматические, словоформ, тезаурусы, словари словосочетаний и моделей управления, своды грамматических правил и т.п.

Различаться может и назначение лингвистических банков данных. Часть ЛБД предназначена для автоматизации деятельности лингвистов и разработчиков прикладных систем, часть - для непосредственного использования в системах обработки текста и речи:

автокорректорах, системах распознавания текста и речи, информационно-поисковых системах.

Отметим, что в качестве пользователя ЛБД может выступать как человек (исследователь-лингвист или разработчик программного продукта), так и тот или иной модуль компьютерной системы обработки текстов. В двух этих случаях требования к организации лингвистических банков данных и к степени эксплицитности, строгости и формальности представленных в них описаний естественного языка разнятся весьма существенно.

Ситуация здесь несимметричная. Пользователь-человек часто может извлечь интересующую его информацию из ЛБД, встроенного в компьютерную систему обработки текстов. Однако компьютерная система обычно не может извлечь нужную для ее работы информацию непосредственно из ЛБД, ориентированного на человека. Особенно остра эта проблема для флективных языков, в частности, для русского языка.

Так, во всех распространенных русскоязычных словарях (толковых, орфографических, словарях синонимов и антонимов и др.) входом в словарную статью служит так называемая начальная форма слова. Поскольку словари ориентированы на пользователя-человека, по умолчанию предполагается, что он знает правила русского словоизменения (склонения и спряжения) и может распознать в тексте любую форму интересующего его слова, т.е., восстановив начальную форму, добраться до соответствующей словарной статьи. Предполагается также, что он может решить и обратную задачу - употребить слово из словаря в требуемой грамматической форме.

При использовании словарей в составе компьютерных систем обработки текстов ситуация иная. Самоочевидные для человека грамматические свойства слова, определяющие особенности его склонения/спряжения, должны быть тем или иным способом явно представлены в компьютерном словаре и в программах морфологического анализа и синтеза, позволяющих определять грамматические признаки словоформ текста и генерировать слова в требуемой форме.

Как распределить знания о чрезвычайно сложных и запутанных правилах русского словоизменения между словарями и программными компонентами?

Здесь возможны два решения:

в словаре описываются только словоизменительные признаки слов (тип и частные особенности склонения/спряжения), а работа по анализу и синтезу словоформ “поручается” программам морфологического компонента компьютерных систем;

в словаре приводятся все формы слов, каждой из которых сопоставлены все необходимые признаки (в частности, грамматические: число, падеж, лицо, время, наклонение и др.).В целом, задача построения и сопровождения лингвистически полного, обоснованного и покрывающего представительное подмножество выбранного естественного языка ЛБД, особенно в случае пользователя-программы, очень сложна. Ее решение требует привлечения квалифицированных специалистов в области лингвистики и инженерии знаний, создания необходимой инфраструктуры, серьезной финансовой и организационной поддержки (часто - на государственном уровне).

2.2. Библиотека программ "Русская морфология"

2.2.1. Словарь Зализняка

Одним из широкодоступных (и активно используемых) русскоязычных ЛБД является электронный вариант фундаментального «Грамматического словаря русского языка» А.А.Зализняка. Текст словаря был перенесен на машинные носители в начале 80-х годов. С тех пор словари всех русскоязычных коммерческих автокорректоров (в том числе, ОРФО, Word), словари практически всех экспериментальных и коммерческих систем машинного

перевода и других систем автоматической обработки текстов строятся на основе словаря Зализняка.

Полиграфический вариант словаря Зализняка состоит из двух частей: "Грамматические сведения" (около 120 страниц) и собственно "Словарь" (около 740 страниц). В первой части представлена разработанная автором словаря с необычайной тщательностью оригинальная модель русского словоизменения (склонения и спряжения). Во второй - приведено около 100 тысяч слов, которым приписаны грамматические индексы, характеризующие тип их словоизменения и схему ударения. Слова упорядочены по концам, что естественно и удобно для грамматического словаря, поскольку слова со сходным грамматическим поведением (одинаковыми суффиксами и окончаниями) располагаются компактными группами.

Словарная статья в словаре Зализняка состоит из заголовка (начальная форма слова) и словарной (грамматической) информации. Для некоторых слов даются также дополнительные сведения, необходимые для различения вариантов. Статьи с заголовками *лев, стричь* и *прихожая* выглядят так:

лев мо 1*b (животное)
лев м 1a (денежная единица)
стричь нсв 8b (-г-)
прихожая ж (п 4a)

По первому элементу словарной информации определяется грамматический класс (***спрягаемое*** слово, слово ***субстантивного, адъективного*** или ***местоименного*** склонения - эти термины будут разъяснены в следующем разделе), для слов субстантивного склонения также одушевленность и род, для спрягаемых слов - вид. Если, например, этот элемент "п", то слово относится к словам адъективного склонения; "ж" - к словам субстантивного склонения, женского рода, неодушевленным; "мо" - к словам субстантивного склонения, мужского рода, одушевленным; "нсв" - к спрягаемым словам (глаголам) несовершенного вида.

Если второй элемент - не цифра, то это означает, что слово изменяется по необычной модели (существительное *прихожая* изменяется по модели слов адъективного склонения). Остальные элементы словарной статьи либо уточняют тип склонения/спряжения, либо свидетельствуют о наличии в слове чередований (символ *), об отсутствии у слова некоторых форм или о других частных особенностях словоизменения. Буквенный индекс после цифры (или после символа *) характеризует схему ударения во всех формах описываемого слова; эта информация полезна при автоматизированной генерации фонетического словаря словоформ русского языка.

Отметим, что исходный (полиграфический) вариант словаря Зализняка был ориентирован на пользователя-человека. Основной сценарий использования словаря предусматривал возможность просклонять/проспрягать любое слово из "Словаря" на основе его грамматического описания и правил, приведенных в "Грамматических сведениях". Эти операции, вообще говоря, требовали выполнения некоторых трудноформализуемых действий, определенной языковой компетенции: поиск уместных грамматических таблиц, определение типа чередования, рассуждения по аналогии. Поэтому непосредственное использование словаря Зализняка (даже в электронном виде) в составе компьютерных систем обработки текста/речи затруднительно.

Разработчики компьютерных словарей, базирующихся на словаре Зализняка, выбирают обычно один из трех путей:

– генерация на основе словаря Зализняка словаря русских словоформ;

– использование электронного "Словаря" в исходной форме и разработка (достаточно сложных) алгоритмов, моделирующих работу с "Грамматическими сведениями";

– создание на основе словаря Зализняка формальной модели словоизменения и необходимое переструктурирование словарной части (явное введение в словарную статью некоторой информации из "Грамматических сведений"), позволяющее существенно упростить алгоритмы.

После подобных преобразований компьютерный словарь может использоваться для решения двух практически важных задач:

задача морфологического анализа - определения начальной формы слова по произвольной словоформе (и, возможно, грамматических признаков словоформы);

задача синтеза - построения всех форм (или указанной формы) слова по начальной форме.Одна из первых формальных моделей русского словоизменения на базе словаря Зализняка (третий из указанных выше путей) была разработана еще в середине 80-х годов на кафедре алгоритмических языков факультета ВМК МГУ под руководством М.Г.Мальковского. Модель была реализована на лиспоподобном языке программирования Плэнер (ЭВМ БЭСМ-6, а позже - МВК «Эльбрус-2» и IBM-совместимые ПК). При этом широко использовались динамические структуры, мощные средства обработки списков и сопоставления образца с выражением. В плэнерских структурах данных явно указывались все морфологические свойства для каждого слова, включая чередования в основе слова. Поэтому плэнерское представление достаточно легко воспринималось человеком, явно отражало морфологические особенности описываемых в компьютерном словаре слов.

Однако язык Плэнер является интерпретируемым, а следовательно, довольно медленно работающим, что затрудняет его применение в системах, к которым предъявляются высокие требования по быстродействию. Обработка сложной структуры списков требует существенных затрат машинного времени, даже при реализации алгоритма их обработки на компилируемых языках, ориентированных на написание эффективных программ (C, C++). Поэтому было принято решение о переходе к другой структуре словаря и соответствующей модификации алгоритмов анализа и синтеза.

Плэнерские структуры, описывающие морфологические особенности всех различных классов слов, были пронумерованы. Затем словам/основам и флексиям были сопоставлены соответствующие номера классов. При чередовании в основе и при наличии у слова супплетивных - образованных от другой основы - форм (*хорош-ий - лучше*) были организованы дополнительные входы в словарные статьи.

Новое представление словаря трудно воспринимаемо для человека. Однако унификация и упрощение структур данных позволили создать условия для значительного увеличения скорости обработки.

2.2.2. Формальная модель русского словоизменения

В ***Формальной модели русского словоизменения*** (ФМРС) множество слов русского языка разбивается на два основных класса - ***неизменяемые*** (Н-слова) и ***изменяемые*** , т.е. склоняемые или спрягаемые (И-слова). Совокупность форм И-слова (словоформ) образует его ***парадигму*** . В каждой словоформе можно выделить ***основу*** и окончание, или ***флексию*** (возможно, пустую, которую мы обозначим: -∅), соответствующую конкретной форме И-слова; за флексией может следовать ***постфикс*** , например, возвратная частица *ся* /*сь* .

С основой И-слова, Н-словом, флексией и словоформой связывается описание значения соответствующего объекта, включающее описание его грамматических характеристик; лексических связей (синонимы, производные слова); семантического значения (ассоциированные с объектом понятия). Грамматические характеристики определяют сочетаемость основ и флексий и синтаксические признаки объектов всех четырех типов.

К грамматическим характеристикам морфологического уровня относятся:

морфологический *(словоизменительный)* ***класс*** - М-класс, ***парадигматический класс*** - П-класс, ***чередование , исключение*** . Синтаксическим показателем является ***синтаксический класс*** (С-класс). Если М-класс определяет, как изменяется слово (склоняется, спрягается), то С-класс характеризует его синтаксическое поведение (сочетаемость с другими словами) Как словоизменительные, так и синтаксические признаки

определяются набором значений грамматических переменных.

Грамматическая переменная (ГП) - переменная одного из следующих типов: одушевленность, род, число, падеж, вид, лицо, залог, возвратность, время, наклонение, степень - принимает закодированное целым числом значение из некоторого множества допустимых. Значение ГП "род", например, кодируется так: мужской - 1, женский - 2, средний - 3. Если значение неопределенно, указывается список возможных значений или число 0 (которое, по соглашению, обозначает любое допустимое значение ГП).

Совокупность ГП, по которым изменяется И-слово (свободных ГП), определяет его парадигму, а спектр значений этих переменных - число элементов парадигмы. Множество И-слов с общим набором ГП, общим набором свободных ГП и общим спектром значений переменных образует М-класс. Основе (и словоформе) сопоставлен упорядоченный набор (вектор) значений соответствующих ГП. Так, например, с основой *лев-* слова *лев* (денежная единица) связан такой вектор (7 8 2 1 0 0)- слово 7-го М-класса, 8-го П-класса, неодушевленное (2), мужского рода (1), значения ГП "число" и "падеж" не определены (0 и 0). Для словоформы *левами* вектор будет иметь вид (7 2 1 2 5), здесь добавились значения ГП "число" (2 - множественное) и "падеж" (5 - творительный).

Понятие М-класса является уточнением традиционного понятия "часть речи": 7-й класс образован в основном существительными, 8-й - прилагательными, 9-й - глаголами.

В ФМРС рассматриваются три класса склоняемых И-слов: местоименные (М-класс номер 5), субстантивные (класс номер 7), адъективные (класс номер 8) и один класс спрягаемых (класс номер 9). Представители 5-го и 8-го М-классов изменяются по родам, числам и падежам, 7-го - по числам и падежам, 9-го - по лицам, родам, числам и временам. Отсутствие у И-слова одной или нескольких форм (например, форм единственного числа у слова *ножницы* , формы родительного падежа множественного числа у слова *мгла*) не препятствует отнесению его к соответствующему М-классу.

Подмножество М-класса, представители которого при совпадающих значениях свободных ГП имеют одинаковые флексии, образует парадигматический класс. В ФМРС рассматриваются 24 П-класса для слов субстантивного склонения, 8 - для слов адъективного склонения, 2 - для слов местоименного склонения, 9 - для спрягаемых слов. К 1-му П-классу субстантивных И-слов относятся, например, существительные *завод* и *артист* (флексии: -∅, *-а* , *-у* , -∅ или *-а* , *-ом* , *-е* - для шести традиционных падежей единственного числа; *-ы* , *-ов* , *-ам* , *-ы* или *-ов* , *-ами* , *-ах* - для множественного); к 11-му П-классу - *карта* и *корова* ; к 21-му - *болото* . К 1-му П-классу местоименных И-слов относятся: притяжательное прилагательное *отцов* , существительное *кабельтов* (не изменяется по родам), ко 2-му П-классу - местоимение *мой* , прилагательное *лисий* , порядковое числительное *третий* .

Хотя П-классы задают более детальную классификацию сочетаемости основ с флексиями чем традиционные типы склонения и спряжения, они недостаточны для описания многих частных особенностей русского словоизменения. Эти особенности можно было бы учесть с помощью еще более дробной классификации, однако, во избежание чрезмерного увеличения числа П-классов, в ФМРС используются другие методы.

Как исключения описываются случаи сочетания основы с "нестандартной" для данного П-класса и данной формы флексией: *-а* в форме именительного падежа множественного числа существительных вместо характерной для 1-го П-класса флексии *-ы* (*глаза* , но *заводы*), пустая флексия вместо флексии *-ов* в родительном падеже множественного числа (*глаз* , но *заводов*). Исключением считается и наличие у некоторых существительных 2-го родительного (партитивного) и 2-го предложного (локативного) падежей: *кусок сахару* , *в шкафу* , но *из сахара* , *о шкафе* . Всего в ФМРС учитываются 26 исключений такого вида.

К особенностям словоизменения относятся и чередования в основе. В ФМРС учтено 55 чередований, например: *ова - у* (*рис-**ова** -ть - рис-**у** -ю*), *та - щ* (*клеве-**та** -ть - клеве-**щ** -у*), *е -* вЂ№пустовЂє (*царев-**е** -н - царев-н-а*). Для И-слов с чередованиями достаточно рассматривать только один "стандартный" вариант основы, указывая тип и контекст

чередования в описании значения основы. Так, для стандартного варианта основы *царевн-* указывается, что при пустой флексии перед последней буквой основы вставляется буква *е* .

Относительно редкие чередования (встречающиеся у 1-3 слов) в ФМРС учитываются по-иному: парадигмы таких слов задаются несколькими основами и Н-словами, образующими "семейство" слова (основы *зай-* , *зайд-* и *заш-* и И-слово *зайти* для глагола *зайти*). Семейства вводятся и для слов с супплетивными формами (*хороший - лучше*) или уникальными наборами флексий (некоторые числительные, личные местоимения).

В синтаксический класс объединяются слова и конструкции с общим набором ГП и общими синтаксическими функциями. Каждому представителю некоторого С-класса сопоставлен (как и в случае М-классов) вектор значений характерных ГП. Для большинства И-слов номер С-класса и соответствующий набор ГП совпадают с номером и набором ГП М-класса. Так, многие существительные - С-класс номер 7 - относятся и к 7-му М-классу. Однако некоторые слова изменяются по "необычной" модели: существительные *прохожий* , *гончая* склоняются как представители 8-го М-класса, для существительного *кабельтов* характерно местоименное склонение. В подобных ситуациях в описании значения основы указывается и синтаксический класс, а иногда значения "дополнительных" ГП (например, вида и залога для причастий - С-класс номер 18, склоняющихся по модели 8-го М-класса).

2.2.3. Основные программы

Морфологический анализ знакомых слов. Программа МОРФ1

Программа МОРФ1 строит все возможные разбиения входной словоформы на основу и флексию и ищет соответствующие части в словаре (первоначально МОРФ1 пытается найти в словаре совпадающее со словоформой Н-слово, а затем последовательно рассматривает словоформу как основу с пустой флексией, основу с флексиями длиной 3, 2 и 1) или неизменяемое слово.

Проверку правильности разбиения - сочетаемости основы и флексии - осуществляет вспомогательная программа, она же устанавливает значения ГП, определяемые флексией. Когда МОРФ1, отщепив флексию, не может найти полученную основу в словаре, происходит обращение к подпрограмме, применяющей к основе правила чередования. Если и после применения правил чередования найти основу в словаре не удалось, слово признается незнакомым и формируется обращение к программе морфологического анализа незнакомых слов МОРФ2 - список вариантов трактовки словоформы (грамматически корректные разбиения на основу и флексию, неизменяемое слово).

Результат работы МОРФ1 (для знакомого слова) - список вариантов анализа, каждый из которых содержит: грамматические признаки словоформы и ссылку на словарную статью, описывающую семантическое значение слова.

Примеры:

стекла → (7 2 3 1 2) - существительное (неодуш.,ср.род) *стекло*

в форме: ед.число, родит.падеж

(7 2 3 2 (1 4)) - существительное (неодуш.,ср.род) *стекло*

в форме: мн.число, именит. или винит.падеж

(9 1 1 3 2 1 1) - глагол *стечь*

в форме: прош.вр., женск.род, ед.число

Упрощенный вариант программы МОРФ1 - программа МОРФ3 - решает так называемую задачу ***лемматизации*** : определяет только начальную форму слова, не формируя список грамматических характеристик словоформы.

Примеры:

стеки → стек, стечь

стекла → стекло, стечь

стеками → стек

Морфологический анализ незнакомых слов. Программа МОРФ2

На вход программы поступает сформированный МОРФ1 список вариантов трактовки словоформы.

Пример (словоформа *квазибиологом*)**:**

квазибиологом+∅ (ср. *космодром/управдом*)

квазибиолог+ом (ср. *биолог+ом*)

квазибиологом (ср. *бегом*)

При обработке незнакомого слова МОРФ2 учитывает флексию и строение основы. В большинстве случаев исследование флексии не позволяет однозначно установить не только П-класс, род слов субстантивного склонения, вид спрягаемых слов, но даже М-класс анализируемого слова, так как, например, флексия *-а* встречается у слов всех четырех рассматриваемых М-классов (*класс-а , красив-а , дядин-а , ворош-а*). Для уточнения грамматических признаков незнакомых слов МОРФ2 учитывает следующие составляющие (диагностические сегменты) основы: префикс, суффикс или некоторую цепочку букв в конце основы, последнюю букву основы.

По префиксу можно обнаружить некоторые Н-слова и установить вид некоторых глаголов. Анализ суффикса помогает установить М-класс, П-класс, род (а иногда и одушевленность) слова субстантивного склонения, вид глагола или даже все нужные (описываемые в словарной статье) грамматические признаки слова. По последней букве основы легко уточняется П-класс, а иногда и М-класс слова. Программа МОРФ2 работает с таблицами, содержащими 28 префиксов и 67 суффиксов. Анализ незнакомого слова МОРФ2 начинает с варианта расщепления с максимальной длиной флексии.

Если анализируется не отдельно взятое слово, а слово в составе предложения, появляется возможность учета контекста (синтаксических связей данного слова с соседними). Информация о контексте передается программам морфологического анализа от объемлющих их программ синтаксического анализа с помощью предсказаний - списка ожидаемых грамматических признаков обрабатываемого слова. Так, при анализе незнакомого слова *Верхневартовск* в контексте *приехала из далекого Верхневартовска* ожидаемые характеристики последнего слова фрагмента таковы: неодушевленное существительное в форме единственного числа, родительного падежа.

В таких ситуациях результат работы МОРФ2 сопоставляется с предсказаниями, и, в случае соответствия, запоминается. Если же предсказание не подтвердилось, начинает обрабатываться другой вариант разбиения словоформы. Если ожидаемый результат не получен, либо слово признается неизменяемым, либо в нем ищутся и исправляются ошибки.

Для каждого отобранного варианта формируются результаты анализа словоформы (и вариант/варианты новой словарной статьи).

Пример (словоформа *квазибиологом*)**:**

(7 0 1 1 (1 4)) - существительное (одуш. или неодуш., ср.род)

квазибиологом в форме: ед.число, именит. или винит.падеж

(7 1 1 1 5) - существительное (одуш.,муж.род)

квазибиолог в форме: ед.число, творит.падеж

(11) - неизменяемое слово (возможно, наречие)

Заполнение словаря по грамматическим описаниям слов. Программа СЛОВ1

Основная сервисная программа автоматической генерации словарных статей - программа СЛОВ1. В ходе ее разработки были составлены таблицы соответствия словарной

информации из словаря Зализняка и словарной информации ФМРС. Отметим, что программа СЛОВ1 автоматизирует трудоемкую, требующую хорошего знания ФМРС работу по составлению словарных статей. Действия, выполняемые программой, зачастую весьма нетривиальны из-за различий морфологической модели словаря Зализняка, и ФМРС. На вход программы поступает словарная статья, взятая из словаря Зализняка или (если такого слова там нет) сформированная экспертом.

Программа автоматически определяет: 1) основу записываемого в словарь системы слова; 2) номера М-класса, П-класса, С-класса; 3) наличие чередований и их контекст; 4) наличие других частных особенностей словоизменения. При работе с программой СЛОВ1 словарные статьи кодируются по определенным стандартным правилам, в частности, заменяются символы, отсутствующие на клавиатуре (например, цифра в кружке заменяется на цифру в круглых скобках).

По первому элементу словарной информации из словаря Зализняка в большинстве случаев определяется номер М-класса, у слов субстантивного склонения также одушевленность и род, у спрягаемых слов - вид. Если, например, этот элемент "п", то слово относится к 8-му М-классу; "ж" - к 7-му М- классу, женскому роду, неодушевленное; "мо" - к 7-му М-классу, мужскому роду, одушевленное; "нсв" - к 9-му М-классу, несовершенному виду.

После определения М-класса происходит переход на соответствующую ветвь алгоритма, где по второму элементу - цифре - определяется номер П-класса. Если второй элемент - не цифра (это означает, что слово изменяется по необычной модели), то СЛОВ1 фиксирует несовпадение номера С-класса с номером М-класса (т.е. наличие соответствующего исключения) и формирует необходимый фрагмент словарной статьи.

Остальные элементы исходной словарной статьи либо уточняют номер П-класса, либо свидетельствуют о наличии в слове чередований, исключений или об отсутствии у слова некоторых форм. Например, символ "П2" означает, что у слова есть второй предложный падеж (локатив), символ "*" является признаком чередования. Для определения конкретного номера чередования СЛОВ1 анализирует строение начальной формы слова. Так, при обработке первого варианта слова *лев* номер чередования (4 - чередование: *ь - е*) определяется по буквам *ле* , стоящим перед последней согласной основы (буква *в* в данном случае неинформативна). Стандартный вариант основы (*льв-*) определяется по номерам П-класса и чередования.

Результатом работы программы СЛОВ1 является словарная статья или список таких словарных статей - в случае, когда слово из словаря Зализняка представляется в ФМРС семейством Н-слов и/или основ И-слов (для спрягаемых слов, например, программа строит словарную статью, описывающую личные формы глагола и деепричастия, и несколько статей для причастий).

Заполнение словаря по тексту. Программа СЛОВ2

Программа СЛОВ1 используется в ситуации, когда список слов, предназначенных для включения в компьютерный словарь, составлен заранее. Другая технологическая схема предполагает автоматизацию не только этого, но и предыдущего этапа - этапа выявления незнакомых слов по характерным текстам.

Отдельные программы различаются:

– глубиной лингвистического анализа текста (пословный анализ, частичный синтаксический анализ, полный синтаксический анализ, синтактико-семантический анализ);

– "степенью самостоятельности" программ формирования словаря (работа без обращения за помощью к человеку, работа в диалоге с пользователем/администратором и под его контролем)

При пакетной обработке текстов на печать выдается так называемый "протокол

формирования словаря", в который могут вставляться вопросы, адресуемые администратору. Рассмотрим фрагмент протокола диалога администратора-лаборанта с программой пословного анализа текста (будем считать, что слова: *колба, стержень, стекло, стечь* - отсутствуют в словаре):

```
* РАБОТАЕТ ПРОГРАММА ФОРМИРОВАНИЯ СЛОВАРЯ ПО ТЕКСТУ *
УКАЖИТЕ, ПОЖАЛУЙСТА, ОТКУДА БУДЕТ ВВОДИТЬСЯ ТЕКСТ
(К - С КЛАВИАТУРЫ, вЂ№ИМЯ ФАЙЛАвЂє - ИЗ ВНЕШНЕГО ФАЙЛА)
К
ВВОДИТЕ ТЕКСТ. ПРИЗНАК КОНЦА - //.
ВНУТРИ КОЛБЫ РАСПОЛАГАЕТСЯ СТЕРЖЕНЬ, ВЫПОЛНЕННЫЙ ИЗ
СТЕКЛА. ЧЕРЕЗ СТЕРЖЕНЬ ... //
ВЫБЕРИТЕ РЕЖИМ РАБОТЫ, УКАЗАВ НОМЕР РЕЖИМА:
1 - ЗАПИСЬ СЛОВАРНЫХ СТАТЕЙ В БУФЕР БЕЗ ВЫДАЧИ ИНФОРМАЦИИ
2 - ЗАПИСЬ СЛОВАРНЫХ СТАТЕЙ В БУФЕР С ВЫДАЧЕЙ ИНФОРМАЦИИ
3 - ФОРМИРОВАНИЕ СЛОВАРНЫХ СТАТЕЙ С ПОМОЩЬЮ ПОЛЬЗОВАТЕЛЯ
3
ПЕРВАЯ ФРАЗА: НЕЗНАКОМЫЕ СЛОВА: КОЛБЫ, СТЕРЖЕНЬ, СТЕКЛА
СЛОВО 'КОЛБЫ' - СУЩЕСТВИТЕЛЬНОЕ? (Д/Н)
Д
К КАКОМУ РОДУ ОТНОСИТСЯ СУЩЕСТВИТЕЛЬНОЕ 'КОЛБЫ'? (М/Ж/С)
Ж
ЯВЛЯЕТСЯ ЛИ ОНО ОДУШЕВЛЕННЫМ? (Д/Н)
Н
К КАКОМУ РОДУ ОТНОСИТСЯ СУЩЕСТВИТЕЛЬНОЕ 'СТЕРЖЕНЬ'? (М/Ж/С)
М
ЯВЛЯЕТСЯ ЛИ ОНО ОДУШЕВЛЕННЫМ? (Д/Н)
Н
СЛОВО 'СТЕКЛА' - ГЛАГОЛЬНАЯ ФОРМА? (Д/Н)
Н
К КАКОМУ РОДУ ОТНОСИТСЯ СУЩЕСТВИТЕЛЬНОЕ 'СТЕКЛА'? (М/Ж/С)
С
СФОРМИРОВАНЫ СЛОВАРНЫЕ СТАТЬИ С ОСНОВАМИ:
КОЛБ-, СТЕРЖН-, СТЕКЛ-.
ВТОРАЯ ФРАЗА: ...
...
```

Морфологический синтез форм слова. Программа ФОРМ1

По словарной статье (знакомого слова) и набору значений ГП строится соответствующая словоформа.

Примеры:

ЛЕВ (животное), творит.падеж, ед.число (7 0 0 1 5) → ЛЬВОМ

ЛЕВ (ден.единица), творит.падеж, ед.число (7 0 0 1 5) → ЛЕВОМ

Морфологический синтез парадигмы. Программа ФОРМ2

По словарной статье (знакомого слова) строится массив всех форм этого слова. Порядок элементов массива определяется номером М-класса.

Примеры:

синтез всех форм знакомого существительного КАССИРША

КАССИРША КАССИРШИ - им.падеж, ед. и мн.число
КАССИРШИ КАССИРШ - род.падеж, ед. и мн.число
КАССИРШЕ КАССИРШАМ - дат.падеж, ед. и мн.число
КАССИРШУ КАССИРШ - вин.падеж, ед. и мн.число
КАССИРШЕЙ КАССИРШАМИ - твор.падеж, ед. и мн.число
КАССИРШЕ КАССИРШАХ - предл.падеж,ед. и мн.число
синтез всех форм знакомого глагола ВОРОШИТЬ
ВОРОШИТЬ - начальная форма
ВОРОШИ ВОРОШИТЕ - формы повелит. наклонения
ВОРОШУ (БУДУ ВОРОШИТЬ) - 1 лицо,ед.ч,наст.и буд.вр.
ВОРОШИШЬ (БУДЕШЬ ВОРОШИТЬ) - 2 лицо,ед.ч,наст.и буд.вр.
ВОРОШИТ (БУДЕТ ВОРОШИТЬ) - 3 лицо,ед.ч,наст.и буд.вр.
ВОРОШИМ (БУДЕМ ВОРОШИТЬ) - 1 лицо,мн.ч,наст.и буд.вр.
ВОРОШИТЕ (БУДЕТЕ ВОРОШИТЬ) - 2 лицо,мн.ч,наст.и буд.вр.
ВОРОШАТ (БУДУТ ВОРОШИТЬ) - 3 лицо,мн.ч,наст.и буд.вр.
ВОРОШИЛ ВОРОШИЛА ВОРОШИЛО ВОРОШИЛИ - формы прош.времени
ВОРОША ВОРОШИВ - деепричастия

Рассмотрим примеры, показывающие возможность комбинирования отдельных программ библиотеки "Русская морфология". Пусть написана управляющая программа, получающая на входе некоторую словоформу, обращающаяся к программе МОРФ1 (и - если слова нет в словаре - к МОРФ2) и генерирующая все формы (программа ФОРМ2) для каждого варианта анализа. Среди этих форм обязательно должна быть входная словоформа.

Примеры:

обработка незнакомого слова ХРЮША
ВАРИАНТ 1
склонение по образцу слова НОЖ/БОГАЧ
* значение ГП "одушевленность" неизвестно *
ХРЮШ ХРЮШИ
ХРЮША ХРЮШЕЙ
ХРЮШУ ХРЮШАМ
ХРЮША / ХРЮШ ХРЮШЕЙ / ХРЮШИ
ХРЮШОМ ХРЮШАМИ
ХРЮШЕ ХРЮШАХ
ВАРИАНТ 2
склонение по образцу слова МАРШ
* значение ГП "одушевленность" неизвестно *
ХРЮШ ХРЮШИ
ХРЮША ХРЮШЕЙ
ХРЮШУ ХРЮШАМ
ХРЮША / ХРЮШ ХРЮШЕЙ / ХРЮШИ
ХРЮШЕМ ХРЮШАМИ
ХРЮШЕ ХРЮШАХ
ВАРИАНТ 3
склонение по образцу слова ТУЧА/КАССИРША
* значение ГП "одушевленность" неизвестно *
ХРЮША ХРЮШИ
ХРЮШИ ХРЮШ
ХРЮШЕ ХРЮШАМ
ХРЮШУ ХРЮШ / ХРЮШИ
ХРЮШЕЙ ХРЮШАМИ
ХРЮШЕ ХРЮШАХ
ВАРИАНТ 4

склонение по образцу слова СВЕЖИЙ
ПОХРЮШЕЕ ХРЮШЕЕ
ХРЮШ ХРЮША ХРЮШЕ ХРЮШИ
ХРЮШИЙ ХРЮШАЯ ХРЮШЕЕ ХРЮШИЕ
ХРЮШЕГО ХРЮШЕЙ ХРЮШЕГО ХРЮШИХ
ХРЮШЕМУ ЖРЮШЕЙ ХРЮШЕМУ ХРЮШИМ
ХРЮШЕГО amp; ХРЮШИЙ ХРЮШУЮ ХРЮШЕЕ ХРЮШИХ amp; ХРЮШИЕ
ХРЮШИМ ХРЮШЕЙ ХРЮШИМ ХРЮШИМИ
ХРЮШЕМ ХРЮШЕЙ ХРЮШЕМ ХРЮШИХ
ВАРИАНТ 5
спряжение по образцу слова ТОЧИТЬ/СЛЫШАТЬ
ХРЮШИТЬ
ХРЮШИ ХРЮШИТЕ
ХРЮШУ (БУДУ ХРЮШИТЬ)
ХРЮШИШЬ (БУДЕШЬ ХРЮШИТЬ)
ХРЮШИТ (БУДЕТ ХРЮШИТЬ)
ХРЮШИМ (БУДЕМ ХРЮШИТЬ)
ХРЮШИТЕ (БУДЕТЕ ХРЮШИТЬ)
ХРЮШАТ (БУДУТ ХРЮШИТЬ)
ХРЮШИЛ ХРЮШИЛА ХРЮШИЛО ХРЮШИЛИ
ХРЮША ХРЮШИВ
ВАРИАНТ 6
неизменяемое слово типа АНТРАША
ХРЮША

Заметим, что если бы слово *хрюша* анализировалось с предсказаниями, результат был бы более точен. Так, при предсказании «существительное женского рода» был бы выдан только третий вариант, при предсказании «форма глагола» - только пятый.

обработка незнакомого слова КРОВАТЬ
ВАРИАНТ 1
спряжение по образцу слова ПИРОВАТЬ
* значение ГП "вид" неизвестно *
(выбран несовершенный вид)
КРОВАТЬ
КРУЙ КРУЙТЕ
КРУЮ (БУДУ КРОВАТЬ)
КРУЕШЬ (БУДЕШЬ КРОВАТЬ)
КРУЕТ (БУДЕТ КРОВАТЬ)
КРУЕМ (БУДЕМ КРОВАТЬ)
КРУЕТЕ (БУДЕТЕ КРОВАТЬ)
КРУЮТ (БУДУТ КРОВАТЬ)
КРОВАЛ КРОВАЛА КРОВАЛО КРОВАЛИ
КРУЯ КРОВАВ
ВАРИАНТ 2
склонение по образцу слова ПЕЧАТЬ
* значение ГП "одушевленность" неизвестно *
КРОВАТЬ КРОВАТИ
КРОВАТИ КРОВАТЕЙ
КРОВАТИ КРОВАТЯМ
КРОВАТЬ КРОВАТЕЙ / КРОВАТИ
КРОВАТЬЮ КРОВАТЯМИ
КРОВАТИ КРОВАТЯХ

ВАРИАНТ 3
неизменяемое слово типа ДЕСКАТЬ
КРОВАТЬ
Start to type here

3. Исправление ошибок в русскоязычных текстах

3.1. Проблема речевых ошибок

Использование естественного языка в качестве средства общения (***речевая деятельность*** человека) неизбежно сопровождается теми или иными нарушениями языковых правил. Такие нарушения - вне зависимости от того, обусловлены они неполнотой знаний человека о языке или же случайными сенсомоторными "сбоями" (описки, опечатки, оговорки) - мы будем называть ***речевыми ошибками*** .

В идеале обработка речевой ошибки предполагает соотнесение ошибочной речевой единицы с полным описанием языка и с контекстом рассматриваемого коммуникативного процесса. Лингвист (или другой специалист), занимающийся исследованием каких-либо теоретических аспектов проблемы речевых ошибок, например, их классификацией, и располагающий источниками, в которых содержится исчерпывающее описание единиц и правил того или иного естественного языка (словари, своды правил), находится в ситуации, достаточно близкой к такому идеалу.

В случае же повседневной речевой практики - непосредственного (диалог) или опосредованного (чтение текста) речевого взаимодействия рядовых носителей языка - ситуация иная. Лингвистические знания рядового носителя языка неполны, воспользоваться справочной литературой он может далеко не всегда, а сам факт ошибки никаким явным образом в анализируемом тексте не указан.

Обнаружить речевую ошибку в этой ситуации непросто. Действительно, для получателя сообщения (реципиента) внешним признаком речевой ошибки служит появление в тексте какой-либо незнакомой ему речевой единицы. Однако такая "подозреваемая" речевая единица может оказаться и правильной конструкцией или формой (например, просторечным вариантом или термином), не знакомой реципиенту.

С другой стороны, абсолютно правильная на первый взгляд единица может быть ошибкой, обнаружить которую удается лишь на "высших" этапах анализа. Так, в предложении "*Пуск ракеты осуществляется нажатием крас**к** ой кнопки"* все слова известны, синтаксические связи правильны; опечатка обнаруживается только на семантическом/ смысловом уровне.

Если одним из участников общения является компьютерная система, положение становится еще более сложным. И лингвистические знания, и интеллектуальные способности (в том числе - в плане работы с языком) такого "собеседника" пока весьма скромны. Однако, как мы уже знаем из материала 1-й главы, достаточно широко и успешно применяются системы обнаружения и исправления ошибок.

Отметим еще одно обстоятельство. Как бы ни разнились характер использования и назначение АОТ-систем (системы машинного перевода, автоматического реферирования или индексирования, работающие в пакетном режиме; системы обеспечения диалога с машиной на естественном языке), оснащение их средствами обнаружения и исправления речевых ошибок повышает устойчивость и эффективность функционирования таких систем, облегчает (в случае диалоговых систем) процесс общения человека с ЭВМ.

Классификация речевых ошибок

Первый критерий классификации речевых ошибок, в соответствии с которым ошибки подразделяются на мотивированные и случайные, связан с понятием индивидуальной языковой модели. ***Индивидуальная языковая модель*** (ИЯМ) - это то подмножество языковых единиц и правил, которое усвоил и использует в своей речевой практике конкретный носитель некоторого естественного языка. Субъективное преломление языка (как знаковой системы социального уровня) в процессе его усвоения приводит к тому, что в ИЯМ не попадают (или попадают в искаженном варианте) некоторые языковые единицы и правила языка.

Поэтому в речи конкретных носителей языка начинают проявляться некоторые индивидуальные особенности, либо вступающие в противоречие с языковыми нормами, либо нет.

В первом случае мы имеем дело с мотивированными речевыми ошибками - точнее, с ошибками, мотивированными особенностями ИЯМ конкретного носителя языка (автора анализируемого АОТ-системой текста). К ошибкам такого рода относятся, например, ошибки в словоизменении (*контейнер**á*** - в форме именительного падежа множественного числа), орфографические ошибки в основах (*еде ница*), некоторые пунктуационные ошибки, смешение слов-паронимов (*представить - предоставить*), нарушение лексической сочетаемости (*делать горе*), искажение фразеологизмов (*не так страшен черт, как его малютки*).

Ошибки, обусловленные внешними по отношению к ИЯМ факторами: сбой речевого аппарата человека, несвоевременное переключение регистра клавиатуры, нажатие соседней клавиши, сбой на линии связи с ЭВМ - мы будем называть случайными.

Как правило, мотивированные речевые ошибки регулярно повторяются в речи носителя языка, а случайные ошибки могут как повторяться (например, при западании клавиши), так и не повторяться.

Отметим, что иногда отличить случайную ошибку от мотивированной сложно. Так, употребление слова *представить* вместо *предоставить* в контексте *представлено право* может быть или результатом случайной ошибки (пропуск буквы), или результатом мотивированной ошибки (смешения паронимов).

Мотивированные речевые ошибки могут различаться степенью серьезности (грамматичности). Помимо серьезных, абсолютно недопустимых грамматических ошибок - типа орфографических ошибок в основах или смешения слов - рассматриваются и ошибки, в результате которых появляются "полуграмматичные" формы (*контейнерá , сидевши*), которые имеют в словарях стилистические пометы: просторечное, устарелое, разговорное, областное и др.

Следующий критерий классификации ошибок (мотивированных и случайных) связан с языковыми уровнями, нормы (правила) которых оказываются нарушенными в результате речевых ошибок. В соответствии с этим критерием речевые ошибки можно классифицировать следующим образом:

1) орфографические ошибки: пропуск одной буквы, замена одной буквы, перестановка двух рядом стоящих букв, одна лишняя буква (отдельно может рассматриваться случай удвоения буквы), замена буквы русского алфавита буквой латиницы и др.;

2) морфологические (словоизменительный уровень) ошибки: ошибки в окончаниях (флексиях) при склонении и спряжении слов (рассматриваются различные подклассы таких ошибок), употребление отсутствующих в языке форм слов, несоблюдение правил чередования в основе, употребление незнакомых АОТ-системе вариантов слов, испытывающих колебания в роде, одушевленности;

3) синтаксические ошибки: ошибки в моделях управления слов-предикатов, пунктуационные ошибки, нарушение нормативного порядка слов (в том числе - в устойчивых словосочетаниях), вставка пробела внутрь слова, пропуск пробела (отдельно могут рассматриваются случаи слитного и раздельного написания частиц *не* и *ни*);

4) лексико-семантические ошибки: употребление слов в ненормативном значении, нарушение лексической сочетаемости, семантические противоречия.

Диагностика речевых ошибок

Методы обнаружения и исправления орфографических и морфологических ошибок в текстах широкой тематики базируются на представлении о тексте как о цепочке независимо появляющихся словоформ. Известно три основных метода обнаружения орфографических ошибок - статистический, полиграммный и словарный.

При статистическом методе словоформы, обнаруживаемые в тексте, упорядочиваются согласно частоте их встречаемости. Искаженные слова оказываются среди малоупотребительных слов в конце списка.

При полиграммном методе все встречающиеся в тексте двух- или трёхбуквенные сочетания (полиграммы) проверяются по таблицам, содержащим информацию об их допустимости в русском языке. Если в словоформе имеются недопустимые полиграммы, то она считается неправильной.

При словарном методе все входящие в текст словоформы проверяются по компьютерному словарю. Если словарь такую форму допускает, она считается правильной, а иначе либо сразу признаётся ошибочной, либо предъявляется человеку.

В настоящее время первые два метода практически не используются, т.к. уже есть хорошие компьютерные словари, достаточно большие по объёму и с эффективным доступом.

Диагностика же и исправление синтаксических, пунктуационных и лексико-семантических ошибок предполагает взгляд на текст как на последовательность связанных единиц, комбинирование которых имеет свои закономерности. Подходы к автоматизации выявления и коррекции этих ошибок можно разбить на две группы: синтаксически-ориентированные подходы и подходы, основанные на концептуальных фреймах. Последние больше пригодны для систем, работающих в строго ограниченных предметных областях. Для текстов широкой тематики предназначены синтаксически ориентированные подходы. Сначала поступившее на вход предложение обрабатывается средствами грамматики, рассчитанной на синтаксически правильный текст. Если такая проверка обнаруживает дефекты синтаксической структуры, некоторые условия ослабляются. Какие грамматические правила смягчаются, зависит от учитываемых системой ошибок. Например, в русских текстах иногда оказывается пропущенной запятая, обособляющая причастный оборот в постпозиции. Для того, чтобы такое предложение могло быть обработано, требуется временная отмена условия (присутствующего в каноническом правиле) обязательного наличия запятой. Однако ослабление канонических правил неизбежно влечёт за собой возрастание числа возможных интерпретаций. При этом нельзя опознать ошибочный текст прежде, чем будет закончен анализ средствами канонической грамматики. Другой подход предлагает сначала использовать слабую грамматику, а затем подвергнуть обрабатываемое предложение фильтрации на основе строгих требований правильности. Но при этом наличие ошибки предполагается более вероятным, чем соблюдение норм грамматики.

Также отметим, что описанные методы позволяют автоматически обнаружить ошибку только тогда, когда не удаётся построить связный синтаксический граф для рассматриваемого предложения. Однако ошибки, при которых возможно получение формально приемлемой, но по сути неверной интерпретации, остаются невыявленными. При этом никаких сообщений об ошибках не поступает.

3.2. Система комплексного контроля качества текста ЛИНАР

3.2.1. Функции системы ЛИНАР; сценарии работы с системой

Построение автокорректоров сталкивается с рядом принципиальных и не решенных

пока в полном объеме проблем: компактное хранение словарей, эффективные методы морфологического и синтаксического анализа и т.д. Тем не менее на очереди - создание систем, способных производить более сложное по сравнению с автокорректорами автоматическое или автоматизированное редактирование текстов на естественном языке. В идеале же необходима система, выполняющая функции научного редактора - человека, осуществляющего литературную и научную правку научно-технических текстов. Такое направление развития представляет разрабатывавшаяся в 1986-1990 гг. на кафедре алгоритмических языков факультета ВМК МГУ система ЛИНАР (ЛИтературно-НАучный Редактор) - интеллектуальная система комплексного контроля качества и редактирования русскоязычных текстов.

Суть подхода заключалась в существенном расширении возможностей имевшихся в то время автокорректоров за счет:

– ограничения предметной области, к которой относились обрабатываемые тексты (методы, алгоритмы и программы обработки данных телеметрии на многопроцессорных вычислительных комплексах);

– ограничения видов текстов (научно-технические отчеты, деловая переписка);

– использования средств синтаксического и семантического анализа текста;

– привлечения более полных моделей русского языка.

Пользователем ЛИНАР является человек, оценивающий с помощью системы качество некоторого текста с позиций лица, которому адресован этот текст (адресата), и вносящий в текст необходимые исправления. В качестве адресата могут выступать литературный или научный редактор, корректор, потенциальные читатели (конструкторы, программисты, руководители). Пользователем ЛИНАР может быть, например, автор обрабатываемого текста, желающий взглянуть на него "со стороны", или научный руководитель работы, обеспокоенный терминологическими и стилистическими неувязками в текстах разделов, подготовленных различными участниками проекта.

Обработка текста с помощью системы ЛИНАР включает в себя в общем случае несколько циклов (как и при подготовке текста "вручную"), каждый из которых оформляется как самостоятельный сеанс работы с системой. В начале сеанса пользователь формирует задание на обработку текста, для выполнения которого система загружает необходимые информационные модули и вызывает программы контроля текста. Каждая программа проверяет некоторое определенное свойство текста, т.е. реализует одноаспектный контроль текста. Таким образом, в структурном плане систему ЛИНАР можно считать пакетом прикладных программ; сеанс работы с ней состоит из серии одноаспектных проверок текста или его фрагментов.

Основная технологическая схема использования системы ЛИНАР предусматривает, что текст хранится на машинных носителях и обрабатывается программами контроля, формирующими протокол замечаний по тексту (иногда система предлагает свой вариант исправления). Далее пользователь просматривает эти замечания и, если он с ними соглашается, вносит необходимые изменения в текст с помощью текстового редактора. Измененная версия текста может быть объектом обработки в следующем сеансе. В зависимости от объема текста пользователь может выбрать диалоговый или пакетный режим работы с системой. В последнем случае протокол замечаний формируется на внешнем носителе.

Отметим, что используемые в ЛИНАР знания позволяют системе фиксировать различные типы конфликтных ситуаций (и формировать соответствующие замечания). Однако как бы полны ни были знания ЛИНАР, обнаружить все неточности, противоречия, неопределенности система самостоятельно не может. Поэтому часть программ контроля собирает некоторую вспомогательную информацию о тех или иных характеристиках (свойствах) текста, не давая ей оценки.

Например, при написании отдельных фрагментов текста разными авторами для обозначения одной и той же сущности могут быть использованы различные термины, что

усложняет понимание текста. Автоматическое обнаружение подобных конфликтов требует привлечения глубоких знаний о понятийном и терминологическом аппарате предметной области, и в ЛИНАР не реализуется. Однако в составе системы имеется программа контроля, которая может сформировать по фрагментам текста списки используемых терминологических словосочетаний. На основе этой информации решить терминологические проблемы человеку будет значительно проще, чем при обработке текста "вручную".

ЛИНАР не только обнаруживает неточности, ошибки, но и может "объяснить" пользователю суть своих замечаний, а также предложить способы устранения ошибок. Так, например, в случае орфографической ошибки система предлагает свой вариант исправления слова, в случае нарушения естественного порядка слов - правильный порядок слов и т.д. Рекомендации системы призваны помочь пользователю в улучшении текста, направляют его деятельность.

3.3.2. База знаний системы

Контроль текста, осуществляемый системой ЛИНАР, основывается на использовании знаний о том, что такое правильный, хороший текст. Совокупность этих знаний называется контролирующими знаниями, или К-знаниями. При формировании К-знаний учитывались результаты лингвистических, психологических работ, исследований по эргономике; принят во внимание опыт редакторов, корректоров, нормоконтролеров.

К-знания должны обеспечить возможность оценки текста с различных сторон:

– соответствие общеязыковым нормам;

– соответствие "внешним" нормам, например, требованиям ГОСТов, регламентирующих форму изложения материала в научно-технических документах;

– сложность восприятия текста потенциальным читателем;

– семантическая корректность текста (соответствие выявляемых в тексте семантических отношений и понятийной модели предметной области).

Часть К-знаний (процедурная составляющая) представлена программами одноаспектного контроля. Каждая программа фиксирует строго определенное свойство текста или строго определенный дефект текста (конфликтную ситуацию). Затем формируется соответствующее диагностическое сообщение, которое, в зависимости от выбранного режима работы, либо сразу предъявляется пользователю, либо включается в протокол замечаний.

Важным компонентом информационного обеспечения системы ЛИНАР является и лингвистическая база знаний, содержащая базовые общие знания о русском языке. Кроме того, ЛИНАР использует тематический словарь и тезаурус предметной области, к которой относятся обрабатываемые тексты, и описания нормативных требований, предъявляемых к текстам. Соответствующие информационные массивы создавались разработчиками системы на основе общеязыковых и предметно-ориентированных словарей и справочников, Государственных стандартов и отраслевых инструкций по оформлению текстовых документов.

База знаний ЛИНАР содержит также заранее формируемый - и пополняемый в ходе эксплуатации системы - **банк адресатов** : конкретных читателей или определенных однородных групп читателей (конкретный руководитель научно-исследовательского проекта; конкретный представитель руководства организации-заказчика; инженеры, которые будут создавать описываемый программно-аппаратный комплекс и др.). Настройка на адресата производится в начале очередного сеанса работы с ЛИНАР. При такой настройке могут меняться базовые и тематические лингвистические знания (состав словаря, совокупность грамматических правил), степень жесткости требований по соблюдению тех или иных норм и условий.

Чтобы задать эту информацию, следует указать имя одного из известных ЛИНАР адресатов (или идентификатор известной группы адресатов) и выбрать значения

дополнительных параметров программ контроля.

С помощью такой настройки удается моделировать процесс восприятия текста разными адресатами и, следовательно, оценивать качество текста с разных точек зрения.

Таким образом, К-знания ЛИНАР (которые служат критерием корректности текста и используются для обнаружения "дефектов" текста - отклонений от требований, предъявляемых К-знаниями) формируются динамически в каждом конкретном сеансе работы с системой и являются комплексными по своей природе. Они включают как процедурные знания об исследуемом аспекте текста (воплощенные в соответствующих программах контроля), так и декларативные знания, фильтруемые и конкретизируемые в начале каждого сеанса.

Обнаруженные программой контроля несоответствия текста и К-знаний могут быть устранены двумя способами:

путем внесения изменений в текст (это наиболее частый случай: несоответствие - суть ошибка, допущенная в тексте, которую необходимо исправить);

путем изменения К-знаний системы.Заметим, что изменениям подвергается лишь один компонент К-знаний - лингвистические знания, причем не все, а лишь те, которые соответствуют наиболее подвижной части естественного языка - лексикону. Как правило, такие изменения заключаются в пополнении базы знаний, например, в создании новой словарной статьи для слова, впервые встретившегося в тексте и не знакомого системе. Знания, отображающие требования семантической корректности и простоты интерпретации, общеязыковые и внешние нормы, может изменять только администратор системы.

Для внесения изменений в базу лингвистических знаний используются сервисные программы; для изменения текста - подсистема редактирования ЛИНАРа.

Отметим, что (даже при работе с ЛИНАР в диалоговом режиме) редактирование текста обычно производится по завершении работы программ контроля. Это связано с тем, что исправление фиксируемых системой ошибок и неточностей зачастую требует переделки относительно больших фрагментов текста (разбиение длинной фразы на несколько более простых, устранение неоднозначности трактовки и т.п.). Однако некоторые - локальные - изменения можно внести в текст сразу же в момент обнаружения ошибки. Поэтому в ряде программ контроля, например, в программах орфографического уровня, предусмотрена возможность исправления фиксируемых ошибок в момент их обнаружения.

2.3.3. Программы контроля

Программы контроля текста могут быть классифицированы по нескольким критериям.

Первый критерий связан с анализируемым программой аспектом текста. В соответствии с этим критерием выделяются следующие группы программ одноаспектного контроля:

– контроль орфографии (включая поиск ошибок в склонении и спряжении слов);
– анализ лексического состава текста;
– стилистический контроль;
– проверка выполнения правил структуризации текста;
– контроль синтаксической структуры;
– пунктуационный контроль;
– семантический контроль.

По второму критерию программы одноаспектного контроля подразделяются на программы локального и глобального анализа текста. Программы первой группы обрабатывают мелкие фрагменты текста: отдельные словоформы, словосочетания, специальные символы, не исследуя их контекстные связи или ограничиваясь учетом ближайшего окружения (соседнего слова справа, например). Локальный анализ характерен для программ орфографического, лексического и (частично) стилистического контроля.

Программы, осуществляющие глобальный анализ, исследуют, как правило, структуру более крупных единиц текста: фраз и иногда абзацев (синтаксический и семантический контроль), текста в целом.

Третий критерий связан с характером результата, получаемого программой одноаспектного анализа. Основная часть программ контроля обнаруживает те или иные несоответствия текста и К-знаний, используемых в текущем сеансе. Результатом их работы является список выявленных несоответствий (нарушений). Однако некоторые программы, как уже отмечалось, определяют отдельные свойства текста, не оценивая их. Так, программа ЛЕКС1 составляет частотный словарь исследуемого текста (фрагмента текста). Оценку полученным результатам дает человек - пользователь ЛИНАР, он же принимает решение о дальнейших действиях. Его реакция может быть, например, такой - поработать над текстом пункта 4.5.1., поскольку в этом тексте (занимающем всего две страницы) 26 раз встречается слово *знания* (в различных формах) и 7 раз - слово *соответственно* .

Только что рассмотренный пример (программа ЛЕКС1) можно использовать и для иллюстрации четвертого критерия классификации программ контроля. Эта программа, как и ряд других, выдает некоторую глобальную информацию об исследуемом фрагменте текста, не фиксируя, в каких позициях (абзацах, фразах или строках) были обнаружены в тексте формы различных слов. Другие программы, например программы проверки орфографии, локализуют обнаруживаемые ими свойства (дефекты) текста.

И наконец, отметим еще одно (формальное) различие программ контроля. Для всех программ основным параметром является подлежащий обработке фрагмент текста. Однако для некоторых программ нужно обязательно указать дополнительные параметры, конкретизирующие задание. Например, при вызове программы ЛЕКС2 нужно указать, какие именно грамматические признаки слов интересуют пользователя.

Некоторые программы контроля получают в качестве параметра предельно допустимые (пороговые) числовые значения количественно оцениваемых параметров текста. Отметим, что, меняя порог, можно варьировать уровень требований, предъявляемых к тексту, моделируя тем самым оценку его разными адресатами. Например, можно установить в качестве предельно допустимой длины фразы 25 слов или ограничить число придаточных предложений (в составе сложного предложения) двумя. Фразы, в которых эти пороговые значения превышены, будут классифицированы соответствующими программами контроля как недопустимые.

3.2.3.1. Орфографический контроль

Программы орфографического контроля обнаруживают (и предлагают варианты исправления) мотивированные грамматические ошибки в основах и окончаниях (флексиях) слов, записанных в словарь системы, и слов, встретившихся ей впервые (незнакомых), а также случайные, или немотивированные, ошибки.

Основные классы учитываемых случайных ошибок таковы:

– пропуск одной буквы (*ас емблер*),

– одна лишняя буква (*авт**т** окод*),

– замена одной буквы (*ко**н** пьютер*),

– перестановка двух соседних букв (*агл оритм*).

Признаком ошибки служит появление в обрабатываемом тексте формы незнакомого системе слова.

Предпринимается попытка "свести" такое незнакомое слово к знакомому с помощью преобразований, обратных перечисленным выше (считается, что ошибка могла возникнуть в результате одного из таких "прямых" преобразований знакомого слова). Для предварительной оценки близости слов (основ слов) используется специально разработанная метрика.

Одна из программ обнаруживает ошибки в датах, задаваемых в тексте с помощью конструкций вида ДД.ММ.ГГ. Если задан и диапазон возможных дат, проверяется также

принадлежность всех представленных в исследуемом тексте дат этому диапазону.

Примеры работы программ:

прочитанна - ОШИБКА В СЛОВОИЗМЕНЕНИИ !
ОЖИДАЕМОЕ СЛОВО: прочитана
рассчета - ВОЗМОЖНА ОШИБКА ТИПА "удвоение буквы"
ОЖИДАЕМОЕ СЛОВО : расчета
10.25.89.
ОШИБКА В ДАТЕ - недопустимая дата: месяц: 25

3.2.3.2. Анализ лексического состава текста

Программа ЛЕКС1

Программа подсчитывает, сколько раз в тексте (области) употребляется то или иное слово. Программа формирует полный список всех различных слов текста с указанием частот их встречаемости. Можно задать диапазон частот (например, от 10 до 20 вхождений или ровно 15 вхождений) и сформировать список слов, количество употреблений которых лежит в границах этого диапазона. Если диапазон не задан, формируется полный частотный словарь текста.

Программа ЛЕКС2

Программа формирует список слов, обладающих указанными лексико-грамматическими характеристиками, например, находит все существительные, все причастия или все аббревиатуры, встретившиеся в тексте (области). Слова упорядочиваются по алфавиту, для каждого слова подсчитывается число его вхождений в исследуемый текст. Программа предназначена для анализа словарного состава текста.

Программа ЛЕКС3

Программа находит все вхождения в исследуемый текст (область) любых форм указанного (ключевого) слова и для каждого вхождения выдает контекст установленной длины - цепочку слов, находящихся от ключевого слова на расстоянии, не превышающем заданную длину. Программа удобна для анализа лексического состава текста и контроля используемых терминов и терминологических словосочетаний.

Программа ЛЕКС4

Программа находит в исследуемой области текста все слова, не входящие в формируемый в начале очередного сеанса словарь системы ЛИНАР, - т.е. слова, не знакомые очередному адресату. Для исправления текста следует либо заменить обнаруженные слова синонимами, либо расширить словарь системы. Возможно, что некоторые из обнаруженных слов являются известными системе словами, введенными с ошибками.

Программа ЛЕКС5

Программа осуществляет поиск каждой из обнаруживаемых в тексте (области)

аббревиатур последовательно в трех списках: N 3 - списке аббревиатур, вводимых непосредственно в тексте (этот список формируется динамически самой программой ЛЕКС5);

N 2 - формируемом в начале работы с текстом на основе перечня используемых сокращений;

N 1 - словаре общепринятых сокращений.

В списке N 1 поиск ведется в последнюю очередь так как он, во-первых, самый большой, и во-вторых, если, например, в списках N 3 и N 1 присутствует одно и то же сокращение, но с различными расшифровками, то приоритет имеет сокращение из списка N 3. Результатом работы является список используемых в тексте аббревиатур с указанием их локализации в тексте и типа аббревиатуры.

Программа ЛЕКС6

Программа осуществляет контроль за переопределением известных системе аббревиатур. Если, например, в разделе 1.2. встретилась аббревиатура СВП (с расшифровкой в тексте - "схема внешних прерываний"), а в списке N 2 аббревиатура СВП сопоставлена термину "субкомплекс внешней памяти", фиксируется ошибка: недопустимое переопределение аббревиатуры из перечня.

Программа ЛЕКС7

Программа проверяет правильность расшифровки, то есть тот факт, что аббревиатура читается в расшифровке по началам слов, причем некоторые слова расшифровки могут не участвовать в образовании аббревиатуры. Пример работы программы:

Эта организация - центр переводов (ВЦП).
НЕСООТВЕТСТВИЕ АББРЕВИАТУРЫ И РАСШИФРОВКИ:
ВЦП - центр переводов

Программа ЛЕКС8

Программа ЛЕКС8 (без параметров) проверяет правильность оформления списка используемых в тексте аббревиатур (для отчета по НИР - это "Перечень условных обозначений, символов, единиц и терминов"). Предполагается, что каждая пара "аббревиатура - расшифровка" в перечне представлена одной строкой. В процессе обработки перечня заполняется список замечаний. Пример работы программы:

ОБРАБАТЫВАЕТСЯ ПЕРЕЧЕНЬ АББРЕВИАТУР:
БНК - бортовой нейрокомпьютер
БНФ - бекусовская нормальная форма
КПД - канал прямого доступа
ОЗУ
МПК - микропрограммируемый контроллер
ОРЗ - общий регистр записи
ПНП - перейти в неустойчивое положение
СВП - субкомплекс внешней памяти
СПТ - субкомплекс рабочего таймера

ЗАМЕЧАНИЯ:
4 : ОЗУ * НЕТ РАСШИФРОВКИ
5 : МПК * НАРУШЕНИЕ АЛФ. ПОРЯДКА
7 : ПНП * РАСШИФРОВКА НЕ ЯВЛЯЕТСЯ ГРУППОЙ СУЩЕСТВИТЕЛЬНОГО
9 : СПТ * НЕСООТВ: АББР.-РАСШ.

3.2.3.3. Стилистический контроль

Программы данного блока фиксируют внешние характеристики фраз, свидетельствующие о сложности их структуры, а следовательно, и о сложности восприятия смысла. Имеются, например, программы, контролирующие длину фраз, количество запятых, количество придаточных предложений, наличие во фразах текста длинных цепочек слов в родительном падеже (например, *значений аргументов программы пользователя*) или цепочек однокоренных слов (*пользователь может воспользоваться, транслятор транслирует*). Есть программы контроля стилистической окраски слов. В научно-технической литературе нежелательно употребление устаревших слов и канцеляризмов (*ибо, вышепоменованный*), жаргонизмов (*виндуза*), разговорных оборотов (*этот алгоритм, уж поверьте, . . .*). При обнаружении таких слов в тексте их рекомендуется убрать или заменить более нейтральными синонимами. Особый класс составляют слова, явно характеризующие специфику темы (предметной области), раскрывать которую иногда нежелательно. Например, в документе для внутреннего пользования можно употребить термин *военно-космический* , а в тексте сообщения, передаваемого по открытым каналам связи его целесообразно заменить (соответствующая программа предлагает слово-замену *специальный*).

3.2.3.4. Контроль структуры текста

Данные программы контролируют правильность оформления отдельных структурных частей текстового документа с точки зрения соответствующих нормативных требований (например, требований ГОСТа 7.32-81, регламентирующего правила оформления научно-технического отчета). Проверяется оформление титульного листа, списка исполнителей, реферата и других разделов документа.

3.2.3.5. Синтаксический контроль

Программа СИНТ1

Программа СИНТ1 находит в указанной области именные словосочетания вида вЂ№прилагательноевЂє + вЂ№существительноевЂє и вЂ№существительноевЂє + вЂ№существительное в форме родит. падежавЂє и др. Программа может оказаться полезной при анализе лексического состава текста и при поиске терминологических словосочетаний, особенно в тех случаях, когда различные фрагменты текста написаны разными авторами (возможно, использующими близкие, но не совпадающие термины). Найденные программой словосочетания группируются вокруг "ключевого слова" - существительного, играющего роль синтаксической вершины словосочетания. Ряд программ синтаксического контроля обнаруживает нарушения обычного (нейтрального) порядка слов и взаимного расположения групп слов. Такие нарушения могут затруднить восприятие текста.

Например : "*Раздел второй посвящен описанию новых алгоритмов".* или "*Использует этот алгоритм всего две вспомогательные переменные* ."

Отметим, что иногда нарушение нейтрального порядка слов может намеренно

использоваться автором текста с целью изменения логического ударения, усиления ("*Алгоритм этот очень эффективен!"*).

Программа СИНТ2

Программа СИНТ2 осуществляет контроль придаточных предложений с союзным словом который , а именно, проверяет однозначность установления связи между союзным словом и его словом-хозяином из главного предложения. В случае, когда таких слов-хозяев не обнаружено или их более одного, выдается соответствующая диагностика. Пример работы программы:

Рассмотрим структурупамяти вычислительной машины, в **которой** хранятся команды.
СЛОВО **которой** ИМЕЕТ БОЛЕЕ ОДНОГО СЛОВА-ХОЗЯИНА В ГЛАВНОМ ПРЕДЛОЖЕНИИ: машины, памяти, структуру
Каждому каналу соответствует свое устройство, **которые** в свою очередь связаны с главной ЭВМ.
СЛОВО **которые** НЕ ИМЕЕТ СЛОВА-ХОЗЯИНА В ГЛАВНОМ ПРЕДЛОЖЕНИИ
Мощь языка Си - результат выявления его авторамипотребностейпрограммистов, **которые** возникают при программировании на языке ассемблера.
СЛОВО **которые** ИМЕЕТ БОЛЕЕ ОДНОГО СЛОВА-ХОЗЯИНА В ГЛАВНОМ ПРЕДЛОЖЕНИИ: программистов, потребностей, авторами

3.2.3.6. Пунктуационный контроль

Пунктуационные ошибки в реальных предложениях русского языка встречаются довольно часто. Разделим их условно на две группы. Ошибки одной группы связаны с уровнем пунктуационной грамотности и появляются в основном в тех типах текстов русского языка, которые не проходят этап профессионального редактирования (например, в репликах в диалоге пользователя с ЭВМ).

Причиной ошибок другого рода является несовершенное владение навыками клавиатурного набора. Такие ошибки принято называть «типографскими».

Блок пунктуационного контроля системы ЛИНАР разработан на основе весьма полной пунктуационной модели русского языка. Полнота и корректность базовых знаний является основой достижения устойчивости и эффективности программных средств, реализованных на основе данной модели.

В то же время блок пунктуационного контроля является «открытым», т.е. построен таким образом, чтобы обеспечить возможность работы средств адаптации и, при необходимости, введения новых правил пунктуации. Адаптация позволяет автоматически либо модифицировать правила анализа (чтобы новый вариант был применим к встретившейся ситуации), либо обнаружить и исправить пунктуационную ошибку в рассматриваемом предложении. Открытость блока - одна из предпосылок его устойчивости к появлению случайных и мотивированных пунктуационных ошибок, вариативных форм. Система ЛИНАР готова к возможности появления в тексте незнакомых пунктуационных ситуаций и к соответствующей адаптации своих лингвистических знаний (изменению модели) или к исправлению ошибки (изменению текста).

При проверке пунктуации можно использовать любое количество программ контроля, выбирая их при этом по различным признакам. Например, можно осуществлять проверку только тех правил, которые выявляют лишние знаки препинания, можно только тех, которые выявляют пропущенные знаки препинания и т.д. При подобной настройке может меняться совокупность пунктуационных правил, степень жесткости требований по соблюдению каких-либо условий и т. д., что позволяет оценивать качество текста с точки зрения

различных категорий пользователей. Набор желаемых для данного сеанса работы модулей формируется в начале работы пользователем.

Пример работы программ пунктуационного контроля:

В ПРЕДЛОЖЕНИИ:

Только и развлечений **,** что кино раз в неделю

ЗАМЕЧЕНА ПУНКТУАЦИОННАЯ ОШИБКА.

В выделенном месте не должно быть данного знака препинания. В рассматриваемом случае запятая перед что не ставится .

Необходимо пояснение ошибки? (Д/Н)

Д

В безглагольном предложении перед союзом что в выражении **только и ... что** , за которым следует имя существительное или местоимение, запятая не ставится.

Необходимы примеры правильного применения данного правила? (Д/Н)

Д

Только и денег что пятак в кармане.

Только и разговоров что о них двоих.

3.2.3.7. Семантический контроль

Программа СЕМ1

Программа обнаруживает несовпадение ожидаемых семантических признаков актантов (подлежащее, дополнения) глагола и признаков слов (групп слов), реально занимающих соответствующие позиции. Такое несовпадение мешает завершить анализ фразы, поскольку синтаксически допустимая связь не может быть установлена из-за семантических противоречий. Проверяя употребление в тексте глаголов, программа обращает внимание пользователя на "подозрительные" актантные конструкции.

Пример работы программы:

Все рассматриваемые программы написаны на **ассемблере .**

НЕСОВПАДЕНИЕ СЕМАНТИЧЕСКИХ КЛАССОВ!

В ОПИСАНИИ ГЛАГОЛА "написать" СЕМ.-КЛАСС АКТАНТА:

=язык_программирования=

РЕАЛЬНЫЙ АКТАНТ **ассемблере** ИМЕЕТ СЕМ.-КЛАСС: =транслятор=

Схема прерываний подключается к магистрали.

НЕСОВПАДЕНИЕ СЕМАНТИЧЕСКИХ КЛАССОВ!

В ОПИСАНИИ ГЛАГОЛА "подключаться" СЕМ.-КЛАСС АКТАНТА:

=устройство=

РЕАЛЬНЫЙ АКТАНТ **схема прерываний** ИМЕЕТ СЕМ.-КЛАСС:

=структура2=

Программа СЕМ2

Программа проводит полный синтактико-семантический анализ фраз указанной области текста. При этом фиксируются случаи, когда фраза имеет (в контексте предметной области, к которой относится текст) более одной интерпретации, т.е. допускает неоднозначное толкование.

Пример работы программы:

Снижение напряжения вызвало отключение принтера.

НЕОДНОЗНАЧНАЯ ИНТЕРПРЕТАЦИЯ!

1 трактовка:

=причина= : снижение напряжения
=следствие= : отключение принтера
2 трактовка:
=причина= : отключение принтера
=следствие= : снижение напряжения

Программа CEM3

Программа CEM3 проверяет однозначность установления связи между личным местоимением и его антецедентом (словом, на которое ссылается данное местоимение). В случаях, когда такой антецедент не найден или их найдено более одного, выдается соответствующая диагностика.

Пример работы программы:

Каждому каналу сопоставлено определенное устройство. **Они** , в свою очередь, связаны с главной ЭВМ.

ДЛЯ МЕСТОИМЕНИЯ **они** В ПРЕДШЕСТВУЮЩЕЙ ФРАЗЕ НЕ НАЙДЕНО СЛОВ, НА КОТОРЫЕ ЭТО МЕСТОИМЕНИЕ ССЫЛАЕТСЯ

Рассмотрим структурупамятиЭВМ. Она состоит из двух основных частей.

ДЛЯ МЕСТОИМЕНИЯ **она** В ПРЕДШЕСТВУЮЩЕЙ ФРАЗЕ НАЙДЕНО БОЛЕЕ ОДНОГО СЛОВА,

НА КОТОРОЕ ССЫЛАЕТСЯ ЭТО МЕСТОИМЕНИЕ: ЭВМ, памяти, структуру

Программа CEM4

Программа проверяет, принадлежат ли значения количественно оцениваемых свойств описываемых в тексте объектов заданному диапазону. В случае, если значение свойства выходит за границы диапазона, процедура выдает соответствующую диагностику.

Пример работы программы:

Информация передается в **сопроцессор АК-34** по **16 каналу** .

ОБ'ЕКТ: сопроцессор АК-34

ГРУППА: 16 каналу

ВЫХОД ЗНАЧЕНИЯ ЗА ВЕРХНЮЮ ГРАНИЦУ ДИАПАЗОНА

(СОПРОЦЕССОР АК-34 ИМЕЕТ КАНАЛЫ: 0,1,2, ... 15)

4. Информационно-поисковые системы

Поиск информации является одной из основных составляющих человеческой деятельности, с ним мы сталкиваемся ежедневно: изучая театральную афишу, чтобы выбрать интересный спектакль, подбирая в расписании поездов удобную электричку, листая телефонную книгу. Человеку, в силу своей профессии или увлечений часто сталкивающемуся с подбором и поиском какой-либо тематической информации, рано или поздно (с возрастанием ее объема) приходится применять некоторые принципы систематизации и классификации имеющихся данных, обеспечивающие более удобный и эффективный поиск. Так, в библиотеках составляют картотеку: сведения о книге по определенной схеме записываются на карточку, туда же помещается шифр - несколько букв и цифр, по которым можно определить местоположение книги (хранилище, стеллаж, полку); карточки расставляются в алфавитном или тематическом порядке. Применение ЭВМ дает

более широкие возможности для работы с большими массивами информации.

4.1. Основные определения

Информационно-поисковая система (ИПС) - программная система для хранения, поиска и выдачи интересующей пользователя (абонента) информации. Абонент обращается к ИПС с ***информационным запросом*** - текстом, отражающим ***информационную потребность*** данного абонента, например, его желание найти список книг по теории информационного поиска или список аптек, в которых можно купить нужное лекарство. Поиск информации ведется в ***поисковом массиве*** , который формируется (и по мере необходимости обновляется) разработчиками или администраторами системы. Элементы поискового массива вводятся в информационно-поисковую систему на естественном (или близком к нему) языке, а затем обычно подвергаются ***индексированию*** , т.е. переводу на формальный ***информационно-поисковый язык*** .

Индексирование - выражение центральной темы или предмета какого-либо текста или описание какого-либо объекта на информационно-поисковом языке[1].

Предмет - объект (материальная вещь, понятие, свойство или отношение), который рассматривается или упоминается в документе/информационном запросе.

Тема документа/информационного запроса - раздел науки или техники, область практической деятельности или проблема, которой посвящен документ/ информационный запрос.

По характеру поискового массива и выдаваемой информации ИПС подразделяют на ***документальные*** и ***фактографические*** .

Документальная ИПС предназначена для отыскания документов (статей, книг, отчетов, описаний к авторским свидетельствам и патентам), содержащих необходимую информацию. Поисковый массив такой ИПС состоит из поисковых образов документов (т.е. элементов, каждый из которых передает основное содержание документа) или из самих документов. В ответ на предъявляемый информационный запрос ИПС выдает некоторое множество документов (или адреса их хранения), содержащих искомую информацию. Документом называют любой осмысленный текст, который обладает определенной логической завершенностью и содержит сведения о его источнике и/или создателе.

Фактографическая ИПС обеспечивает выдачу непосредственно фактических сведений, затребованных потребителем в информационном запросе. Поисковый массив состоит из фактографических записей, т.е. из описаний фактов, извлеченных из документов и представленных на некотором формальном языке.

Например, если бы Служба знакомств решила создать документальную ИПС, поисковый массив состоял бы непосредственно из писем ее клиентов типа: *"Меня зовут Илья Муромец. Просидел я сиднем на печи 33 года, а теперь у царя в охранниках..."*. Для создания фактографической ИПС по письмам клиентов заполнялись бы таблицы вида: *"Фамилия - Муромец. Имя - Илья. Возраст - 33. Должность - секьюрити"* . Соответственно и запросом в первом случае будет служить часть письма клиента с пожеланиями относительно его партнера: *"Невесту хочу моложе меня, но премудрую и чтоб хозяйством домашним интересовалась"* , а во втором - составленная по ней таблица: *"Возраст вЂ№33, интеллект - высокий, интересы - домашнее хозяйство"* .

В настоящее время фактографические ИПС (как специальный класс поисковых систем) практически не разрабатываются, выполняемые ими действия реализуются с помощью штатных СУБД. Далее, говоря ИПС, будем иметь в виду документальную информационно-поисковую систему.

Одним из популярных способов перевода документа на внутренний язык системы является ***координатное индексирование*** - присвоение документу набора ключевых слов или кодов, определяющих его содержание. Возможны два способа индексирования: свободное, когда непосредственно из текста документа извлекаются ключевые слова без учета всех видоизменений их форм и отношений между ними; и контролируемое, когда в

поисковый образ документа включаются только те слова, которые зафиксированы в ***информационно-поисковом тезаурусе*** , где указаны их синонимические, морфологические и ассоциативные отношения.

4.2. Тезаурус

Тезаурус - специально организованный нормативный словарь лексических единиц информационно-поискового и естественного языка. Лексическими единицами информационно-поискового языка являются ***дескрипторы*** . Дескриптор ставится в однозначное соответствие группе ключевых слов естественного языка, отобранных из текста определенной предметной области. Например, в качестве дескриптора может быть выбрано любое (предпочтительно наиболее часто используемое или короткое) ключевое слово или словосочетание или же цифровой код. Многозначному слову естественного языка соответствует несколько дескрипторов, а нескольким синонимичным словам и выражениям - один дескриптор. Тезаурус учитывает семантические связи между словами: антонимы, синонимы, гипонимы, гиперонимы, ассоциации.

Синонимы - слова (словосочетания), разные по написанию, но одинаковые (в рассматриваемой предметной области) по значению: *ведьма = злая волшебница* . ***Антонимы*** - слова с противоположным значением: *добрый - злой* . ***Гипоним*** - термин, являющийся частным случаем другого, более общего понятия. ***Гипероним*** - термин, наоборот, являющийся общим для ряда других, частных понятий.

Солдат = гипоним (*военный*); *человек* = гипероним (*военный*)

гипероним (*вкусно готовит*)= гипероним (*содержит дом в чистоте*)=

гипероним (*умеет шить*)= *хорошая хозяйка.*

В Государственном стандарте на "Тезаурус информационно-поисковый одноязычный" определены следующие типы связей:

– род-вид: *средства передвижения - телега , ковер-самолет , сапоги-скороходы , печка*

– часть-целое: *стена , дверь , курья ножка* - части *избушки* ;

– причина-следствие: *опустил меч - голова с плеч* ;

– сырье-продукт: *сталь - меч* ;

– административная иерархия: *султан - визирь - стражник* ;

– процесс-субъект: *казнить - палач* ;

– процесс-объект: *казнить - жертва* ;

– функциональное сходство: *печка Емели - джип Cherokee* ;

– свойство - носитель свойства: *огнедышащий - дракон* ;

– антонимия;

– синонимия.

Ассоциативное отношение является объединением других отношений, не входящих в иерархические отношения или в отношения синонимии (то есть любые виды связей между словами, возможно весьма специфичные, существующие только в определенной предметной области).

Словарная статья (на неформальном уровне) могла бы выглядеть так:

ПРЕМУДРАЯ = умная

АНТОНИМ - глупая

ГИПОНИМЫ: знающая, образованная, догадливая, начитанная

ВИД - показатель интеллекта (высокий)

Тезаурус и грамматика составляют ***информационно-поисковый язык*** . Грамматика содержит правила образования производных единиц языка (семантических кодов, синтагм, предложений) и регламентирует использование средств обозначения синтаксических отношений (например, указателей связи).

В рассмотренной выше сказочной информационной службе знакомств тезаурус должен

описывать всевозможные качества и характеристики, встречающиеся в письмах клиентов, правила их классификации. Грамматика и тезаурус должны быть составлены таким образом, чтобы система могла понимать, что задает, скажем, число, указанное в запросе: рост, возраст или количество зубов (это может определяться по ключевому слову - единице измерения), уметь отличить сведения, сообщаемые клиентом о себе, от его требований к партнеру (здесь помогут словосочетания *хотел бы познакомиться , должен соответствовать*).

На основании тезауруса и правил грамматики формируются поисковые образы документа и запроса (поисковое предписание). ***Поисковое предписание*** - текст на информационно-поисковом языке, содержащий признаки документов, затребованных пользователем в запросе.

Поисковый образ документа - текст на информационно-поисковом языке, поставленный в однозначное соответствие документу и отражающий его признаки, необходимые для поиска его по запросу. Кроме поисковых признаков, раскрывающих содержание документа или, как минимум, определяющих его тему, поисковый образ документа обычно содержит также идентифицирующие и некоторые дополнительные сведения (выходные данные, тип документа, его язык и т.д.). Поисковые предписания формируются при поступлении запросов, а поисковые образы документов могут создаваться как при пополнении системы новыми документами, так и при поиске ответа на запрос. В системах, где потоки информации велики и часто обновляемы, нет необходимости тратить ресурсы на индексирование, и за поисковый образ документа часто принимается сам документ или же его название.

4.3. Релевантность

Целью ИПС является выдача документов, ***релевантных*** (семантически соответствующих) запросу (по-английски relevant - относящийся к делу). Различают релевантность***содержательную*** и ***формальную*** . Релевантность содержательная трактуется как соответствие документа информационному запросу, определяемое неформальным путем (Василиса Премудрая сама прочитает письма всех добрых молодцев и выберет кандидатов в женихи, отвечающих ее требованиям), а релевантность формальная - как соответствие, определяемое алгоритмически путем сравнения поискового предписания и поискового образа документа на основании применяемого в информационно-поисковой системе ***критерия выдачи*** .

Критерий выдачи - формальное правило, совокупность признаков, по которым определяется степень формальной релевантности поискового образа документа и поискового предписания и принимается решение о выдаче/невыдаче некоторого документа в ответ на информационный запрос.

Информационная потребность

è

Формулировка информационного запроса

è

Поисковое предписание

ô Релевантность Релевантность ô

содержательная формальная
: Документы
è

Индексирование

è

**Поисковый
массив**

В автоматизированных системах поиск основан на формальной релевантности, содержательная релевантность в них определяется, например, путем экспертных оценок и используется для получения данных об ***эффективности информационного поиска в системе*** (качестве ее работы). В качестве критерия выдачи может быть выбрано полное совпадение поисковых образов документа и запроса, включение множества ключевых слов запроса во множество ключевых слов документа, пересечение этих множеств и др.

В рассматриваемом примере при выборе в качестве критерия выдачи полного совпадения ключевых слов документа и запроса клиенту должны быть предоставлены письма персонажей, полностью отвечающих его требованиям. Навряд ли это их удовлетворит, так как явно выбор будет не слишком велик. Этот критерий больше бы подошел для системы, где необходима точность, например, определяющей выбор лекарства при лечении определенной болезни (пусть их будет немного, зато все подходящие), здесь же, наверное, уместен критерий на пересечение.

Дескрипторам могут быть приданы весовые коэффициенты в зависимости от степени их соответствия запросу; при поиске коэффициенты дескрипторов, обнаруженных и в запросе и в документе, суммируются, и документы выдаются в зависимости от значения этой суммы (например, если она превысила некоторое значение). Таким образом, если указать, что наиболее весомыми являются характеристики *богатство* и *могущество* , а не *доброта* и *возраст* , можно заполучить в женихи Кощея Бессмертного. При использовании весов также может применяться ***эшелонированная выдача*** - отобранные документы предъявляются пользователю не в произвольном порядке, а по степени релевантности (по убыванию сумм весов), право окончательного выбора релевантных документов - за пользователем.

Идеальная ИПС должна выдавать документы, содержательно релевантные запросу, и ничего кроме них. Однако на практике это обычно не достигается, наблюдаются молчание ИПС (невыдача некоторого количества релевантных документов) и шум (выдача лишних документов). Массив документов разделяется на **выданные** и **невыданные** - по одному критерию, и на **релевантные** и **нерелевантные** - по другому.

Таким образом, для каждого запроса получаем 4 группы документов:

Соотношение количества документов в каждой из этих групп определяет эффективность информационного поиска. Для оценки эффективности используют следующие характеристики:

Рв

Полнота выдачи =

tabletable--

x 100%

Рв+Рн

Рв

Точность выдачи =

tabletable--

x 100%

Рв+Нв

Рн

Потери информации =

tabletable--

x 100%

Рв+Рр

Нв

Информационный шум =

tabletable--

x 100 %

Рв+Нв

Рв

Чувствительность =

tabletable--

x 100 %

Рв+Рн

Нн

Специфичность =

tabletable--

x 100%

Нн+Нв

В идеальной ИПС Рн=Нв=0 и поэтому полнота и точность= 100%, а шум = 0 (найдены все документы и ни одного лишнего). В реальных системах коэффициент полноты достигает 70%, а коэффициент точности поиска колеблется в очень широких пределах, иногда снижаясь до 10%. Величины этих коэффициентов зависят от целого ряда факторов: как внутренних свойств собственно поисковой системы (объема и характеристик информационного массива, информационно-поискового языка, критерия выдачи), так и от многих "внешних" условий: степени специфичности информационных запросов, способности пользователя правильно сформулировать свои информационные потребности на естественном языке, правильности построения конкретного запроса, а также от субъективного представления пользователя о том, что такое нужная ему информация. Из-за ошибок и неточностей, возникающих на каждом из этапов работы как пользователя, так и системы, результаты могут сильно отличаться от того, что хотел получить пользователь, обращаясь к ИПС.

Существует понятие ***устойчивость поиска*** - характеристика изменения полноты и точности при малых (семантически незначительных) изменениях запроса. Средние значения полноты и точности для конкретной системы обычно вычисляют путем тестирования ее на эталонной базе документов.

В зависимости от требований к количеству и качеству выдаваемой ИПС информации выбираются разные критерии выдачи. Если важно не упустить нужную информацию (патентная экспертиза) - нужно повысить полноту, если надо сократить объем выдаваемой информации (библиотека) - следует улучшить точность.

Английским ученым С.Клевердоном выявлена обратная зависимость между полнотой и точностью поиска в одной системе (при использовании одного и того же информационно-поискового языка), т.е. повышение точности ведет к увеличению шума и, наоборот, при уменьшении шума снижается точность. Улучшить оба эти показателя одновременно можно, только внося изменения в информационно-поисковый язык, делая грамматику и тезаурус более лингвистически развитыми. При этом достижение максимально возможной полноты поиска связано с огромными сложностями. Последние 5-10% требуют такого же усложнения языкового аппарата системы, как и предыдущие 90-95%, что влечет за собой увеличение трудоемкости обработки входной информации и времени поиска.

4.4. Языковой компонент

Увеличению эффективности ИПС в большой степени помогает более детальная обработка текста документа. Так, существуют системы, которые для простоты в качестве поискового образа документа принимают его название, однако оно в силу разных обстоятельств не всегда формально отражает содержание текста. Например, при подготовке данного материала была использована статья "А глаз как у орла", не имеющая никакого отношения ни к орнитологии, ни к окулистам. Также большое значение имеет применение программ, производящих лингвистически содержательную обработку текстов на естественном языке (учитывающую морфологию, синтаксис). Только с их помощью можно установить, являются ли похожие слова (почти все буквы одинаковые) формами одного слова или же это совершенно разные слова, в соответствие которым поставлены разные семантические единицы.

Более примитивные, лежащие на поверхности приемы могут подвести разработчика ИПС. Так, если система не учитывает никакие правила русского языка и работает с шаблонами (типа var*, text*.exe), то при поиске для Золушки кавалера, интересующегося бальными танцами, в качестве ключевого слова-шаблона придется выбрать *бал* * (чтобы не было потери информации, иначе можно пропустить эту характеристику, высказанную словами *люблю танцевать на балах*). Тогда в результате поиска ей может быть предложено

познакомиться со всеми любителями *балета* , *балыка* , *Бальмонта* , *Бальзака* , со всеми, живущими около *Балтийского* моря, в домах с *балконом* , а также со всевозможными *баловниками* и *баловнями судьбы* .

Все эти претенденты будут отсеяны, если в качестве ключевого слова будет задано прилагательное *бальный* и система сможет распознавать его во всех его формах (применение морфологического анализа слов также дает возможность уменьшить объем тезауруса, избавив его от избыточной информации - иначе все формы одного слова приходится определять как синонимы). Еще один способ уменьшения шума и повышения точности - введение в информационно-поисковый язык аппарата работы с однокоренными словами. В нашем примере при задании ключа-корня *бал* выданными оказались бы только документы, содержащие разные формы слов *бал* и *бальный* . Однако и в этом случае письмо желанного принца затеряется между сообщениями о *салонах бального платья* , *владельцах бальных залов* , музыкантах и официантах, *обслуживающих балы* . С помощью синтаксического анализа можно более точно определять словосочетания (например, распознавать их не только когда слова стоят друг за другом, но и когда они разделены рядом других слов). В приведенном примере в системе с синтаксическим компонентом можно было бы вести поиск документов со словосочетаниями *бальный танец* и *танцевать на балу* . Конечно, и это не обеспечивает 100% точности (например, ничто не запрещает выдачу сообщений об *учителях бальных танцев*), однако понятно, что количество выданных документов значительно сократится, и Золушка уже не превратится в старую деву, просматривая предложенную ей системой информацию.

Развитые информационно-поисковые языки допускают использование логических связок: *дурак* =NOT(*умный*), *добрый молодец* =(*мужчина*) AND (*молодой*). В перспективе - возможность описания на информационно-поисковом языке смысла целой фразы (который не всегда складывается из смыслов входящих в нее слов) и возможность формулировки соответствующих семантически сложных запросов.

[1] Отметим, что в рекламе или обзорах поисковых средств часто можно встретить слова "индексирование" или "индексация". Там эти термины означают создание общего глоссария по всему массиву для увеличения скорости поиска. Для всей текстовой базы составляется список встречающихся в ней терминов, и каждому из них ставится в соответствие некоторый индекс (координаты в текстовой базе); чаще всего это номер документа и номер слова в документе. При поступлении запроса слово сначала ищется в этом списке, и по найденным координатам выдаются нужные документы. Если слов в запросе несколько, над их координатами производится операция пересечения. Именно так организован поиск статей, включающих заданное слово, в подсистемах помощи Windows.