А. А. Перфильев

ПОИСКОВАЯ СИСТЕМА С ЭЛЕМЕНТАМИ ЛИНГВИСТИЧЕСКОГО АНАЛИЗА

ВВЕДЕНИЕ

К настоящему времени четко выделились задачи, которые компьютеры пока не умеют решать хорошо. Одной из таких проблемных задач в области информационных технологий и искусственного интеллекта является задача по извлечению информации из текста или, в более широком смысле, задача понимания текста; к ним также можно отнести задачу распознавания образцов текста по смыслу или, более конкретно, задачу эффективного поиска.

Задача эффективного поиска или Интернет-поиска требует вовлечения контекста для ее решения. Если реализация семантического поиска — дело трудоемкое и требующее больших усилий по описанию каждой предметной области и каждого понятия и для описания присущих только этим понятиям характеристик, то использовать поиск с вовлечением контекста можно уже на уровне синтаксиса с грамматикой. Этому и посвящена данная работа.

1. ОБЩИЕ ПРИНЦИПЫ СИСТЕМ, РАБОТАЮЩИХ С ЕЯ

С одной стороны, довлеет требование эффективности, поэтому, как правило, решение строится индивидуально под каждую конкретную задачу, где требуется извлечение смысла из текста; подходы, инновации и ноу-хау глубоко зашиты в программный код и не переносимы на другие схожие задачи.

Несмотря на глубину анализа, общими, ключевыми моментами остаются понятия образца и структуры. Но и сама структура над текстом строится с использованием образцов.

Любая система, имеющая дело с текстом на естественном языке и производящая анализ этого текста, так или иначе сталкивается с задачей построения структуры над этим текстом. Под структурой может пониматься очень широкий класс сущностей: от тривиальной стилевой разметки документа (с которой имеют дело текстовые процессоры) до сложных семанти-

ческих сетей, отражающих смысл текста (необходимых в задачах искусственного интеллекта).

Итак, пониманием текста можно считать построение над текстом некоторой структуры, представляющей смысл этого текста с точки зрения конкретной задачи. После того, как над текстом построена некая первичная структура, она либо является искомой, либо над ней строится другая, более глубинная структура, и так далее до тех пор, пока не будет достигнута требуемая в данной задаче глубина «понимания» текста. Таким образом, структура текста — это иерархическая система объектов, причем терминальными объектами в этой системе являются некие отрезки текста, например, слова. После того, как эта структура построена, она может анализироваться специальными алгоритмами, преобразовываться, модифицироваться и т.п. Задача построения структуры текста не считается самоцелью, она лишь является составляющей более сложных задач по обработке текста, более того, в ряде случаев она считается вспомогательной, и в описании различных систем ей не уделяется должного внимания. Однако благодаря своей базисности (лежит в основе обработки текста) и универсальности (каждая система обработки текста так или иначе реализует данную стадию) она является важной и актуальной в области информационных технологий и искусственного интеллекта. Построение структуры текста является частью любой системы, так или иначе связанной с обработкой текста на ЕЯ [2].

2. ПОСТАНОВКА ЗАДАЧИ

Итак, задача состоит в том чтобы построить алгоритмы, которые, проникая в структуру текста, смогут вывести адекватную оценку релевантности текста. Важно чтобы данная оценка выводилась основываясь на контексте поискового запроса а не ограничивалась только ключевыми словами.

3. МЕТАПОИСКОВАЯ СИСТЕМА INETFINDER

В рамках дипломной работы была создан поисковый бот (iNetFineder an Internet crawler) который автоматизирует работу по поиску информации в сети. Работа сводится к тому чтобы отдать запрос программе и дождаться когда она закончит поиск и сбор информации из сети. При завершении она предложит просмотреть результаты поиска.

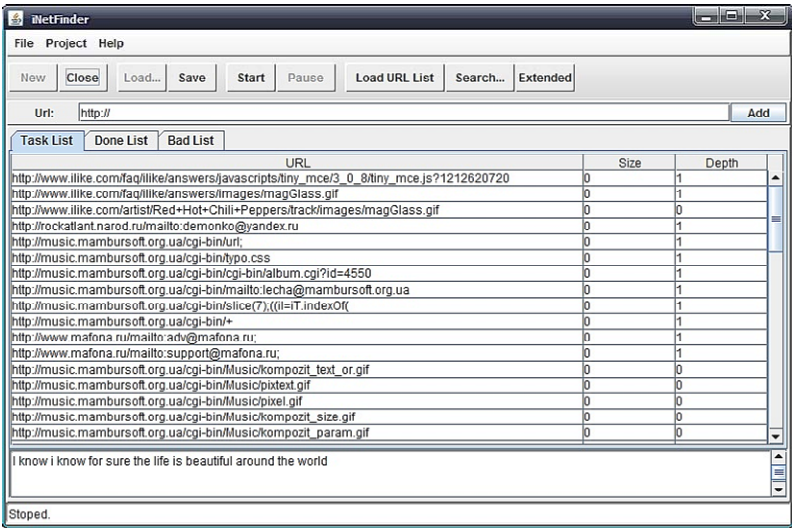


Рис. 1. Скриншот рабочего окна поисковой системы iNetFinder

Особенности ИПС

1. Система находится на стороне пользователя, требует подключения к сети Интернет.
2. Использует результаты запросов к существующим поисковым системам, таким как Yandex, Google. Корректирует и уточняет их.
3. Метаинформации недостаточно. Важна семантическая связность текста. Система просматривает текстовое содержимое Интернет-страничек как базу поиска. Если источник не содержит текста соответствующего запросу — он отбрасывается.

ИПС может работать в нескольких режимах:

- целенаправленная загрузка списка введенных Интернет-адресов, поиск информации соответствующей запросу;
- отправка запроса поисковой системе, получение списка Интернет-адресов, просмотр этого списка, поиск информации соответствующей, запросу;
- рекурсивный просмотр каталога, просмотр файлов, поиск информации соответствующей, запросу;

- целенаправленная загрузка файлов указанных типов с сайта.

Процесс закачки Интернет-страничек предполагает ряд действий.

1. Отправка запроса поисковой системе, разбор полученного гипертекста, пополнение списка ссылок.
2. Загрузка содержимого Интернет-страничек из списка.
3. Просмотр гипертекста поиск и сбор ссылок.
4. Сбор информации, удовлетворяющей запросу пользователя.

4. ПОИСКОВОЕ ЯДРО СИСТЕМЫ INETFINDER

От пользователя в систему поступает текстовый запрос, где из него выделяются ключевые слова и термины. Пополнение запроса с помощью *синонимов* и *гипонимов* (слов с более узким значением) дает более широкий круг поиска слов. При недостаточном количестве найденных документов, поисковый запрос повторяется с привлечением *гиперонимов* (слов с более общим смыслом, например *собака — зверь*). Что значительно расширяет область поиска.

Базой поиска в системе iNetFinder служит текстовое содержимое Интеренет-страничек, которые приходят от встроенного менеджера загрузок. Далее текстовые образцы поступают в систему первичных фильтров, где проходит первичная оценка релевантности текста. Наличие соответствующих ключевых слов говорит о возможной релевантности рассматриваемых образцов. Первичные фильтры ограничивают работу высокопроизводительных алгоритмов синтаксического анализатора, что значительно способствует ускорению работы.

Получаемые на входе предложения транслируются в синтаксические диаграммы. Транслятор проводит лемматизацию слов, приписывание метаинформации словам. Добавление синтаксических связей между словами, типизация этих связей. Приписывание зависимостей между придаточными предложениями. Совокупность этих особенностей дают достаточную информацию о предложении.

Синтаксический анализатор генерирует диаграммы синтаксического разбора, которые используются в системе. Они отображают синтаксическую взаимосвязь между словами.

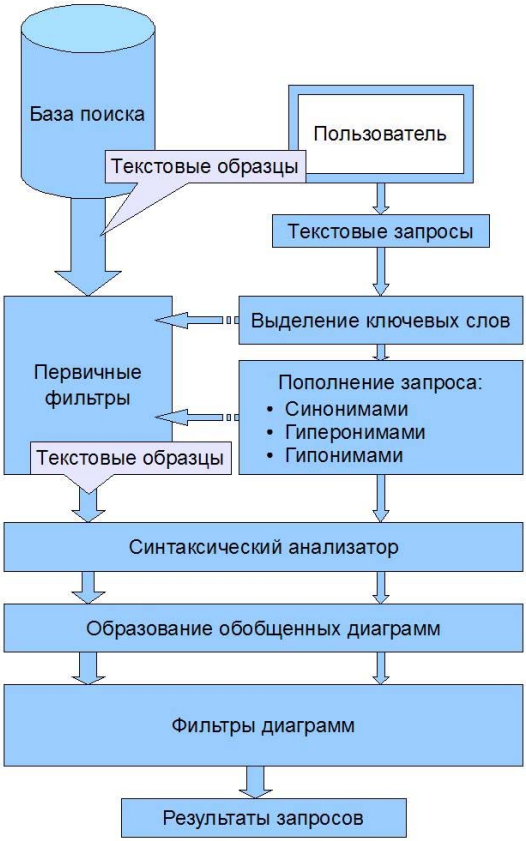


Рис. 2. Схема поискового ядра системы iNetFinder

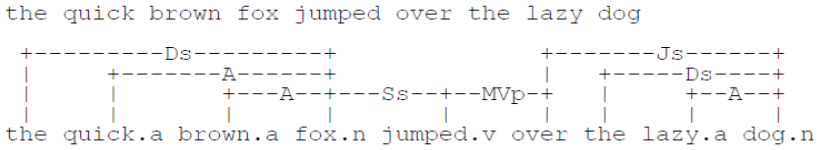
Несколько слов о синтаксическом анализаторе, который был использован в системе. Link Grammar Parser (далее Link) – это синтаксический *парсер* английского языка, основанный на грамматических связях из теории синтаксиса Английского языка. Получив предложение, система приписывает к нему синтаксическую структуру, которая состоит из множества помеченных связей, соединяющих пары слов. Синтаксический анализатор (далее

разборщик или парсер) также создает представление предложения. Link реализован на языке Си и для Unix и Windows, имеет открытый код, распространяется по лицензии совместимой с GNU GPL.

Парсер имеет словарь, включающий около 60000 словарных форм. Он охватывает огромную долю синтаксических конструкций, включая многочисленные редкие выражения и идиомы. Парсер довольно устойчив; он может пропустить часть предложения, которую не может понять и определить некоторую структуру оставшейся части предложения. Он способен обработать неизвестный лексикон, и делать разумные предположения из контекста и написания о синтаксической категории неизвестных слов. У него есть данные насчет различных названий, числовых выражений, и разнообразных знаков препинания.[3]

Внутри парсер использует методы динамического программирования для сопоставления связей между словами. В настоящее время данное программное средство является наиболее перспективным инструментом по работе с текстом.

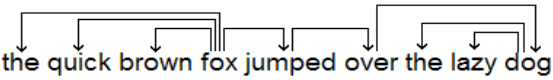
Пример разбора предложения:



- Текстовыми средствами изображаются связи между словами. Каждая связь соединяет только два слова. Скрещивающихся связей нет.
- Каждая связь имеет свою подпись. Например: A ("attributive") – определение
- К некоторым словам добавлены пометки (.n – noun, .a – adverb, .v – verb,) Они указывают на трактовку слов парсером (могут быть различные варианты). Разборщик показывает только те связи, которые найдет в предложении.
- Возможны случаи, когда разборщик выдает несколько вариантов диаграмм, не выдает вообще. Иногда парсер пропускает некоторые слова (для них он не находит грамматических связей).

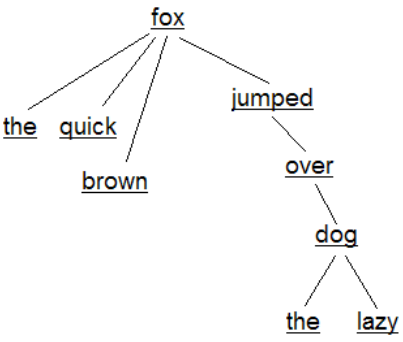
Получаемые диаграммы по сути являются аналогом деревьев подчинения в предложении. В деревьях подчинения от главного слова в предложении

можно задать вопрос к второстепенному. Таким образом слова выстраиваются в древовидную структуру.



Сущность такой взаимосвязи может нести чисто синтаксический характер, но также несет и смысловую компоненту. Если рассматривать языки программирования то они несут сценарную компоненту: это набор объектов, их поля и методы, а также то что нужно с ним и сделать. Разговорные языки несут в себе другие компоненты: смысловую, описательную. В общем случае, чтобы понимать естественный язык требуется установить отображение между объектами речи (в данном случае словами, словосочетаниями, идиомами) и объектами физического и умопостигаемого мира. Важно отметить, что для успешного поиска текста в предложении не обязательно полностью понимать предложение. Вполне достаточно знать как можно перефразировать словосочетания или сами предложения.

Итак, у нас есть дерево разбора. Дальше происходит обобщение таких деревьев. На этом этапе происходит нормализация словоформ. Обратный порядок слов заменяется на прямой. Пассивные формы переделываются (если возможно) в активные. Сложные формы глаголов обрезаются до простой формы. Глаголы переводятся в одну нормализованную форму в настоящем времени в простом виде. Сложные комбинации предлогов объединяются. В результате получается остов дерева, в котором удалены речеобразовательные конструкции. Такие деревья проходят процесс сравнения с диаграммой запроса пользователя.



Фильтрация диаграмм. Перед сличением слова проходят простой фильтр на словоформу – было бы глупо считать глагол и существительное одинаковым словом. Само сличение или наложение слов происходит просто: проверяются гипотезы на соответствие двух слов по набору правил, если все правила проверены, и соответствия не выявлено, то слова считаются далекими по смыслу. Набор правил представляет собой условия, при

которых мы можем считать слова близкими. Туда входят такие правила как прямое равенство слов, совпадение с точностью до окончания, синонимическая близость слов, наличие отношения гипоним-гипероним, слова с трансмутациями и прочие возможные близости между словами.

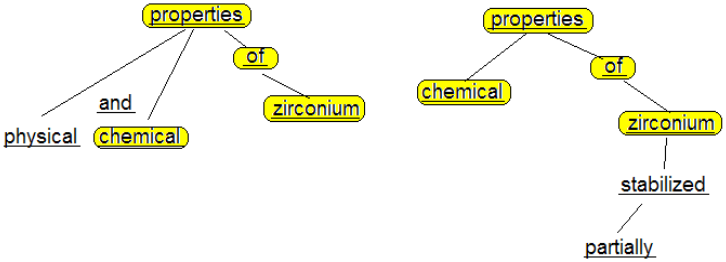


Рис. 3. Пример наложения двух деревьев

Алгоритм сличения выражений, напоминает работу конечного автомата работающего на узлах дерева. По словосочетанию строится конечный автомат. Если в тексте находятся словосочетания удовлетворяющие условиям, это зависит от набора используемых приемов: то автомат переводит свои головки в очередные состояния там где условия удовлетворены. Если хоть одна фраза привела автомат в конечное состояние, значит это выражение из набора и оно релевантно. Количество головок автомата зависит от набора проверяемых схем. Использование автоматов существенно ускоряет обработку выражений. Особенно в сочетании с быстрым синтаксическим анализатором – это делает поиск текста очень быстрым. Тем более, что большинство предложений фильтруются на начальном этапе [1].

Таким образом, приведенная степень оценок позволяет ввести определенную меру близости на предложениях. Она учитывает связь между словами, поиск по словосочетаниям, а также связность слов.

Предложения прошедшие последний фильтр считаются релевантными и выдаются пользователю. Для завершения своей работы система iNetFinder формировала аннотацию из найденного релевантного текста.

5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Помимо этого, возможно формировать конкретные запросы к закачанному содержимому. Например, можно выяснить всевозможные определения, появляющиеся с каким то химическим элементом, и таким образом выяснить его химические и физические свойства.

Типичные приемы:

1) Отбор словосочетаний. Имеется словосочетание: существительное и несколько прилагательных. Ищем словосочетания, в которых присутствует то же существительное и частичный набор прилагательных.

2) Подлежащее-сказуемое. Имеется подлежащее и сказуемое или просто глагол и существительное. Строим всевозможные времена глагола, форму в пассивном залоге, также герундий. Если в просматриваемом тексте находятся похожие конструкции, то подбираем их.

3) Преобразование аббревиатур, идиом, поиск синонимов. Искомое словосочетание пополняем всевозможными сокращениями, аббревиатурами, синонимами. Если в просматриваемом тексте находятся похожие конструкции, то подбираем их.

Нечеткий поиск слов и исправление ошибок в словах:

Режим *нечеткого* поиска позволяет найти документы, которые содержат слова, похожие по написанию на слова запроса — например, слова с опечатками: вкрапления, пропуски букв, перестановка рядомстоящих букв, замена символа на неправильный, перепутанная раскладка клавиатуры, просторечные выражения, сокращения, транслитерация и пр. Режим также включает в себя исправление слов написанных похожими символами из других языков и специальными символами, такие приемы обычно используются хакерами для маскировки слов.

Типы ошибок в словах:

1. приставка/суффикс;
2. вкрапление/вакансия;
3. перестановка пары рядом стоящих букв;
4. замена символа на другой символ;
5. “мутации” замены символами из другого набора;
6. “мутации” при рисовании специальными символами;
7. перепутанная раскладка клавиатуры;
8. просторечные выражения и сокращения.

6. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СИСТЕМЫ INETFINDER

Для демонстрации эффективности работы системы были произведены испытательные загрузки с помощью данной системы. Были сформированы десять простых запросов из области неорганической химии. По каждому запросу произведены загрузки и поиск релевантного запросу текста. Для сравнения с поисковой системой nigma.ru в таблице можно увидеть результаты запросов. Первая колонка — количество ссылок, выданных системой nigma.ru; из них iNetFinder выбрала некоторое число (вторая колонка) действительно релевантных ссылок. Третья и четвертая колонки — это ошибки системы iNetFinder; уровень ошибок около 5%.

Запрос	Всего ссылок получено от поисковой системы	Количество релевантных ссылок, одобренных системой	Количество релевантных ссылок, пропущенных системой	Количество не релевантных ссылок, одобренных системой
the burning rate of rocket fuels	99	15	8	1
using of liquid nitrogen	85	29	2	0
physical and chemical properties of zirconium	96	8	2	9
raw material for produce of medicine	121	26	7	9
use of zirconium in medicine	97	9	1	1
harmful influence of strontium on a man	102	6	0	0
molecular structure of products of diseintegration of alcohol	85	20	1	12

ways of getting glycerin	89	3	2	0
physical properties of oxides	95	17	4	8
classifying of separation techniques	107	10	0	1

ЗАКЛЮЧЕНИЕ

Когда реализация сложных семантических поисковых запросов очень сложна сама по себе ввиду необходимости проникать вглубь в каждое понятие, обладать необходимым набором знаний о нем, и даже в этом случае система все еще остается “туповатой”. Это является следствием огромной выразительной силы естественного языка. В то же время данная система является не единственной возможностью по улучшению поиска, что в свою очередь, свидетельствует о необходимости развития данного направления. Также, несмотря на примитивность алгоритмов применяемых в прототипе iNetFinder, система показала довольно хорошие результаты, что свидетельствует о перспективности развития данной системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Батура Т.В., Корда О.В., Мурзин Ф.А.** Исследовательская система для анализа текстов на естественном языке // Методы и инструменты конструирования и оптимизации программ. – Новосибирск, 2005. – С. 7–21.
2. **Шевченко М.И.** Технология построения многовариантных объектно-ориентированных структур текстов // Диссертационная работа на соискание ученой степени кандидата физико-математических наук. – МГУ, Москва, 2005. – С. 5–21.
3. **Daniel D.K.Sleator, David Temperly** Parsing English with a Link Grammar // School of Computer Studies. – Carnegie-Melon University, Pittsburg, PA, 1991.