А.В. Добров

ОАО «Линукс Инк», a.dobrov@linux-ink.ru

# Опыт создания открытого лингвопроцессора AIRE

***Аннотация* — В данной статье описывается система автоматической лингвистической обработки текста на естественном языке, представляющая собой универсальный базовый компонент систем информационного поиска и автоматического перевода, и перспективы ее развития.**

***Ключевые слова* — автоматическая лингвистическая обработка текста, автоматический синтаксический анализ, информационный поиск, автоматический перевод**

## I. Введение

В общем виде задача создания системы автоматической лингвистической обработки текста (лингвопроцессора) сводится к разработке программного обеспечения, способного вычислять смысловое содержание этого текста, имея на входе только сам текст и полное описание того языка, на котором написан этот текст. Это «смысловое содержание» должно иметь сериализуемую форму представления, из которой должно быть возможно породить все эквивалентные по смыслу тексты на том же или другом языке (таким образом можно создать систему машинного перевода). Кроме того, все эквивалентные по смыслу тексты должны получать одинаковую форму представления смыслового содержания. Это свойство обеспечивает возможность создания систем семантического поиска, предполагающих выдачу в качестве результатов поиска всех текстов, соответствующих по смыслу поисковому запросу.

Данная задача не представляла бы собой особой сложности, если бы относилась к формальным языкам. Формальные языки отличаются от естественных однозначностью всех языковых единиц, наличием строгих правил разбиения текстов на составляющие, ограниченностью инвентаря атомарных единиц и взаимнооднозначным соответствием между множеством возможных текстов и множеством возможных смыслов. Естественные языки, напротив, характерны тем, что, за редким исключением, лишь подтверждающим общее правило, каждая единица естественного языка многозначна, множество единиц языка неограничено, и единственное правило, позволяющее предпочесть (но не окончательно выбрать) ту или иную возможную трактовку некоторого текста состоит в том, что эта трактовка должна «больше» или «лучше» других подходить «по смыслу», «по контексту». Этот контекст далеко не всегда наличествует и далеко не всегда достаточен для выбора той или иной гипотезы. Рассмотрим два высказывания на русском языке:

1. *Он увидел их семью своими глазами.*

2. *Эти типы стали есть в цехе.*

Первое высказывание на первый взгляд может показаться однозначным, поскольку существа, обладающие семью глазами, встречаются в жизни крайне редко. Тем не менее, если «он» — это инопланетянин из некоторого фантастического рассказа, то он вполне может обладать каким угодно количеством глаз, и видеть некоторых «их» либо всеми семью своими глазами, либо даже только семью из ста пятидесяти пяти.

Второе высказывание также может показаться однозначным на первый взгляд, так как в цехе, как правило, работают, а не едят. Тем не менее, если некоторые подозрительные типы по какой-то причине все-таки стали есть в цехе, то они не нарушили этим никакие законы физики и, возможно, даже не нарушили законы общества.

Многозначность текста на естественном языке — это вполне естественное его свойство, и отношение между множеством текстов на естественном языке и множеством возможных смыслов не только не взаимнооднозначно, но и даже ни инъективно, ни сюръективно. Таким образом, задача лингвопроцессора состоит не в вычислении одной «правильной» версии разбора каждого текста, а в построении всех возможных версий этого разбора и их сортировке по «правдоподобию».

## II. Задача автоматической лингвистической обработки текста

Чтобы сформулировать более четко задачу автоматической лингвистической обработки текста на естественном языке, видится целесообразным определить, что подразумевается под «смысловым содержанием», «текстом», «языком» и «правдоподобием».

Язык — это система сигнальных единиц (знаков) и отношений между ними, включающих в себя правила объединения знаков в сложные знаки и вычисления возможных означаемых сложного знака из его компонентов. Понятия языка и знака в таком понимании были введены Ф. де Соссюром в работе [1].

В широком понимании текст — это любая последовательность символов, используемая как сложный знак для передачи некоторого сообщения посредством кодирования и декодирования информации в соответствии с правилами языка. С точки зрения лингвопроцессора, текст — это входной поток байтов, подвергающийся декодированию в соответствии с правилами известных лингвопроцессору кодировок и языков. Декодирование текста требует реконструкции структуры составляющих его знаков: текст состоит из частей, части — из абзацев, абзацы — из предложений, предложения — из словосочетаний, более крупные словосочетания — из более мелких, двусловные словосочетания — из слов, слова — из морфем (суффиксов, префиксов,окончаний, корней), морфемы — из слогов, слоги — из звуков, кодирующихся символами, которые, в свою очередь, кодируются байтами или многобайтными цепочками. Структура составляющих текста древовидна.

Смысловое содержание текста — это означаемое текста как сложного знака, вычисляемое из его компонентов в соответствии с правилами языка. Смысловое содержание текста может быть представлено в виде множества понятий (концептов), обозначаемых атомарными сигналами текста, и отношений между этими концептами. Структура смыслового содержания текста представляет собой граф (далее — концептуальный граф).

Следует отметить, что древовидная структура составляющих текста также является частным случаем концептуального графа, так как дерево — это частный случай графа, а языковые единицы (знаки), являющиеся узлами этого дерева, сами по себе являются понятиями. Данное наблюдение дает возможность предположить, что вся информация о языке может быть представлена в единообразном формате описания концептов, а декодирование текста может осуществляться как распознавание и связывание экземпляров этих концептов, представленных в виде потока байтов. Связывание двух экземпляров концептов представляет собой порождение всех возможных маршрутов, состоящих из отношений между концептами, ведущих из первого концепта во второй. Маршрутов, связывающих два концепта, может быть несколько. Каждый маршрут обладает объективной характеристикой длины. Связь между длиной маршрута и правдоподобием той или иной гипотезы может быть достаточно сложной, однако представляется очевидным, что, чем больше длина маршрута концептуального связывания в определенной версии, тем менее правдоподобна эта версия, но

чем выше ее покрытие, то есть количество связанных в этой версии входных (текстовых) экземпляров концептов, тем выше ее правдоподобие.

Таким образом, задача лингвопроцессора сводится к распознаванию и связыванию экземпляров языковых концептов в потоке байтов и представлению гипотез этого распознавания и связывания в виде концептуальных графов, упорядоченных по покрытию в порядке убывания и по длине маршрута в порядке возрастания. Данная задача до сих пор не имела решения в виде готового программного продукта, так как стандартные подходы к ее решению приводили к сложности задачи коммивояжера, а применение различных статистических эвристик не увенчалось успехом: либо система распознает не все языковые сигналы, и тогда отсутствует гарантия достоверности ее работы, либо работает с чрезвычайно низкой производительностью, что делает невозможным ее тестирование.

## III. Алгоритм концептуального связывания

Задача концептуального связывания представляет собой частный случай задачи обхода графа. В данном случае графом, в котором необходимо осуществлять обход, является система всех концептов языка. Отличие задачи концептуального связывания от задачи поиска маршрута на графе в обычном понимании состоит в том, что в случае концептуального связывания найти один маршрут, связывающий два концепта, недостаточно: возникает необходимость полного перебора всех возможных маршрутов. При помощи стандартных вычислений (см. [2]) можно установить, что производительность алгоритма концептуального связывания, если он реализован как поиск в ширину или как поиск в глубину, составляет $O(|E|+|V|)$ , где $E$ - множество вершин графа (то есть языковых концептов), а $V$ - множество дуг графа (то есть отношений между языковыми концептами). Разумеется, при этом предполагается, что память на хранение маршрутов выделена заранее.

Проблема состоит в том, что величина $|E|+|V|$ имеет порядок $10^7$ . Алгоритм концептуального связывания должен вызываться лингвопроцессором для каждой контактной пары входных байтов, а также для каждой контактной пары полученных гипотез с любым покрытием. Производительность лингвопроцессора в этом случае составит $O\left(10^7 \sum_{i=1}^{N-1} i\right)$ , где N - количество байтов входного потока. Для текста объемом 10 Кб (10240 байт) $\sum_{i=1}^{N-1} i = 52423680 \approx 0.5 * 10^8$ , а время работы всего алгоритма имеет неприемлемый порядок $O(10^{15})$ .

Необходимость существенного повышения производительности требует иного подхода к концептуальному связыванию. Выбранный нами подход состоит в том, чтобы хранить информацию о языковых концептах не в виде графов, а в виде т.н. интеллектуальных карт.

## IV. Онтологии и интеллектуальные карты

Для обеспечения формализации хранения информации о понятиях той или иной предметной области часто используются онтологии. Онтологии содержат в себе формальные модели понятий и отношений между ними. Задача онтологии состоит исключительно в хранении этих формальных моделей; задача связывания одного понятия с другим через посредство других понятий, как правило, не рассматривается как задача онтологии.

Интеллектуальные карты отличаются от онтологий именно тем, что поиск и хранение

маршрутов, связывающих между собой хранимые понятия, относится к их непосредственным функциям.

Единицами, хранимыми в интеллектуальных картах, являются концепты. Концепт — это структура данных, представляющая собой модель системы свойств некоторого понятия. Концепт обладает уникальным идентификатором и атрибутами. Атрибут концепта — это концепт, моделирующий то или иное свойство понятия.

Свойства понятий моделируются атрибутами путем указания роли, в которой понятие выступает в соответствующем отношении с другим понятием. Таким образом, атрибут концепта включает в себя роль этого концепта в данном отношении (субъект/объект), само отношение, и другой концепт, на который направлено это отношение. Роли и отношения — это также концепты, обладающие собственными атрибутами; атрибуты связаны с ролями и отношениями через их собственные атрибуты.

Таким образом, интеллектуальная карта обеспечивает:

- хранение концептов (моделей понятий), их атрибутов (моделей отношений между понятиями и их ролей в этих отношениях), в постоянной и оперативной памяти с возможностью их динамического изменения;

- поиск непротиворечивых маршрутов по атрибутам от одного концепта до другого в порядке от кратчайшего до наиболее длинного маршрута.

Структурное отличие интеллектуальной карты от графа состоит в том, что, во-первых, дуги на интеллектуальной карте могут исходить не только из узлов (концептов), но и из других дуг (отношений), и, во-вторых, интеллектуальная карта хранит в себе информацию о кратчайших расстояниях между каждой парой концептов. Любая связь между двумя концептами регистрируется как трехмерный вектор <идентификатор исходного концепта, идентификатор концепта назначения, кратчайшее расстояние между ними в атрибутах> (напр., <0,1,1>, <0,2,2>, <2,7,2> и т.д.). При регистрации новой связи осуществляется проверка возможности ее объединения с уже зарегистрированными связями (напр., можно объединить <0,2,2> (т.е. от концепта до атрибута расстояние — 2), <2,7,2> (т.е. от атрибута до отношения расстояние — 2) в связь <0,7,4> (т.е. от концепта до отношения расстояние — 4)). Если суммарная связь уже зарегистрирована, то она переустанавливается только в том случае, если ее расстояние меньше, чем у прежде зарегистрированной связи.

Для каждой пары концептов создается список таких атрибутов первого концепта пары, через которые существует нециклический маршрут во второй концепт, причем этот список отсортирован по возрастанию расстояния до второго концепта. Таким образом вычисляются маршруты интеллектуальной карты: на каждом шагу концептуального связывания известны все возможные направления, ведущие к цели, отсортированные по возрастанию расстояния, что дает возможность существенно сократить время связывания.

Интеллектуальные карты хранят свои данные в бинарном формате, обеспечивающем константное время произвольного доступа к любой зарегистрированной связи ($t = O(1)$). Время прохождения одного маршрута, таким образом, линейно зависит от его длины и составляет $O(N)$, где $N$ — длина маршрута.

## V. Реализации системы и их применение

Разработка лингвопроцессора стартовала в 2003 году в лаборатории информационных лингвистических технологий Института Лингвистических Исследований РАН при финансировании из различных источников. Изначальная цель состояла в создании универсального лингвопроцессора с открытым исходным кодом, распространяемого под лицензией GPL. С 2003 по 2009 год проект имел условное название «OOmnik» и лингвопроцессор в различных версиях входил в системы полнотекстового поиска («Буква закона»), кластерного анализа и системы поддержки перевода. В 2008 году стартовал проект создания интеллектуальной поисковой системы для портала Subscribe.ru. Одновременно с этим коллектив разработчиков OOmnik присоединился к компании «Линукс Инк», активно развивающей направление создания поисковых систем и успешно внедрившей одну из версий OOmnik в Справочно-Поисковой Системе (СПС) Библиотеки Российской академии наук (РАН). Вскоре стартовал проект создания универсальной поисковой системы Конституционного Суда РФ (УПС КС РФ), в рамках которого стала очевидной невозможность продолжения развития OOmnik в том направлении, в котором он развивался. Был задействован разработанный для Subscribe.ru лингвопроцессор OOmniklite, реализовавший лингвистическую обработку текста, но ограниченный гипотезами с трехсловным покрытием. OOmniklite не задействовал алгоритм концептуального связывания для анализа структуры составляющих текста, а использовал заранее порожденную базу данных всех возможных сочетаний длиной до трех слов. В результате проект был успешно выполнен. УПС КС РФ осуществляет семантический поиск, ограниченный трехсловными сочетаниями и отношениями концептуальной эквивалентности.

Большая часть коллектива разработчиков OOmnik с 2009 года продолжает развитие лингвопроцессора, основанного на архитектуре OOmniklite, но применяющего интеллектуальные карты для динамического концептуального связывания. Этот проект на сегодняшний день не имеет окончательного названия и существует под кодовым названием «aire» (AI (Artificial Intelligence) Information Retrieval Engine). Текущая версия лингвопроцессора доступна в репозитории https://svn.linux-ink.ru/repos/t

На сегодняшний день реализован анализ структуры составляющих текстов произвольной длины на материале грамматики простого предложения русского языка. После установки пакетов «aire» и «aire-lang», если запущен httpd, тестовый веб-интерфейс лингвопроцессора доступен по адресу http://localhost/cgi-bin/t (вводится тестовая фраза, выдаются графические изображения выявленных структур с наилучшим покрытием).

## VI. Перспективы развития системы

Достигнутые результаты доказывают работоспособность даже нынешней версии aire и возможность его применения в интеллектуальных поисковых системах и системах машинного перевода. Вместе с тем, остается необходимым продолжение его оптимизации и тестирования, так как на более сложной грамматике, чем задействована на сегодняшний день, его производительность может оказаться более низкой. Кроме того, активно развивается процесс создания и пополнения интеллектуальных карт для русского языка (в ближайшем будущем планируется подключение английского, арабского, китайского, турецкого, французского и абхазского языков). Также в ближайшем будущем планируется внедрение aire в ряде действующих информационных систем.

## Литература

[1] Соссюр Ф. Труды по языкознанию. Переводы с французского языка А.А. Холодовича

[2] Breadth-first search. (2010, September 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, October 5, 2010, from http://en.wikipedia.org/w/index.php?title=Breadth-first_search&oldid=386511012