ЛИНГВИСТИЧЕСКИЙ ПРОЦЕССОР В СИСТЕМЕ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА В ИНТЕРНЕТ

С.В. Колосов

Поиск информации в Интернет, основанный на анализе поисковых запросов на естественных языках (ЕЯ), применяется уже довольно давно, однако до сих пор задача интеллектуального поиска далека от окончательного решения [1]. Естественно-языковые средства человеко-машинного интерфейса (ЧМИ) постоянно развиваются, являясь одним из наиболее перспективных способов взаимодействия пользователя со сложными информационными системами. Одним из основных компонентов систем ЧМИ на ЕЯ является лингвистический процессор, выполняющий роль посредника между пользователем и системой семантического анализа. Задачей языкового процессора является преобразование фраз на ЕЯ в некоторую конструкцию, которая является формальным представлением структуры и семантики исходной фразы. Данная работа посвящена алгоритмам формализации фраз ЕЯ.

Любой естественный и искусственный язык представляет собой цепочку лексем, каждая из которых несет в себе определенную семантическую информацию. Задача лексического анализатора – для каждой исходной лексемы определить ее основные характеристики – изображение лексемы и ее тип. Тип лексемы в ЕЯ – это морфологические (словообразовательные) характеристики слова, которые могут изменяться в процессе словообразования. Для реализации ЧМИ на русском языке в качестве ЕЯ достаточно предложить минимальный набор параметров: часть речи как основной параметр, дополнительные параметры определяются в зависимости от части речи (род, число, падеж для существительного, причастия и прилагательного, род, число и время для глагола).

Большая часть русских слов меняет свою форму посредством использования аффиксов (окончаний и возможных суффиксов) и префиксов (приставок). Автоматическое отделение префиксов может привести к ошибкам при лексическом анализе. Кроме того, как правило, использование префиксов меняет семантику слова, поэтому лексический анализатор построен на принципах разбиения слова на основу и аффикс. Те слова, которые изменяют свою форму посредством одних и тех же правил использования аффиксов объединяются в группы, определяемые флективными типами. Задача лексического анализатора, таким образом, заключается в том, чтобы для заданной словоформы определить лексические данные, то есть изображение лексемы и ее тип, который представляет собой определенный набор морфологических характеристик. Реализация предложенной модели может быть выполнена на основе словаря основ и базы данных морфологической информации.

С учетом необходимости наличия в системе различных составляющих языка (лексической, синтаксической и семантической), формально функциональная модель естественного языка представима в виде:

$$L = \{ A, P, S \},$$

где $A = \{ G_m, S \}$ – лексика языка, определяемая морфологической грамматикой G_m и основным лексическим набором (словарем) S;

P – система синтаксически-ориентированного перевода фраз в функциональную форму;

S – наборы семантических функций, отражающие различные аспекты семантики языка.

Структуру словаря основ S, необходимую для работы морфологического анализатора, можно представить следующим образом:

$$S = \{ B, F, T, \delta_1, \delta_2 \},$$

где B – множество основ;

$F = \bigcup_i \left\{ u_{i_1}, u_{i_2}, \ldots, u_{i_n} \right\}$ – класс флективных типов;

$T = \{ t_1, t_2, \ldots t_m \}$ – множество суффиксов;

$\delta_1 \subseteq F \times 2^T$ – система соответствия флективных типов и множества суффиксов;

$\delta_2 \subseteq B \times 2^F$ – система флективных групп.

Большинство основ русского языка принадлежат не одному флективному типу, поэтому вводится понятие флективной группы, которая представляет собой множество из одного или нескольких флективных типов.

Таким образом, у каждой основы имеется своя флективная группа, в которую входят те флективные типы, к которым может принадлежать основа.

Тогда множество словоформ, известных системе в целом, можно представить в виде:

$$L_{осн} = L_{осн}(S) = \{btu \mid b \in B;\ t \in \delta_1^t(\delta_2^g(b));\ u \in \delta_2^u(b)\}.$$

Здесь b, t, u – части словоформы, означающие соответственно основу, суффиксы и окончание.

Следуя предложенной выше модели лексики естественного языка, в данной работе предложена реализация лексического анализатора, структура которого показана на рисунке 1.

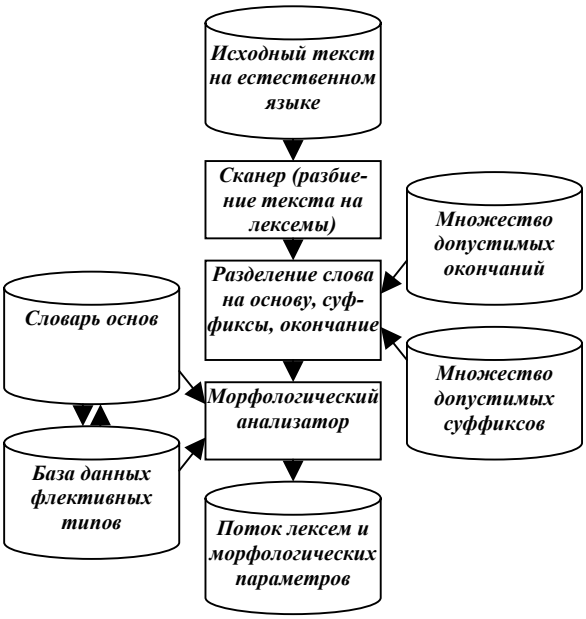

Рисунок 1 – Структура лексического анализатора

Сканер последовательно, по строкам читает текст и разбивает его на лексемы. Полученный поток лексем обрабатывается алгоритмом разделения слова, который каждую лексему разбивает на основу, суффиксы и окончание всеми возможными способами, используя при этом множество допустимых окончаний и суффиксов русского языка. На следующем шаге морфологический анализатор определяет для каждой лексемы ее изображение и тип (набор морфологических параметров). Для работы морфологического анализатора требуется специальный словарь основ, позволяющий по любой известной системе основе определить правила ее словообразования, а также база данных, в которой хранятся правила морфологии языка. Алгоритмы разделения слова и морфологического анализа будут рассмотрены ниже.

На вход морфологического анализатора должны поступать следующие данные для каждой лексемы: основа слова для поиска в словаре основ и окончание слова для определения его морфологических параметров. Возникает задача разделения произвольного слова на основу и постфиксы. Учитывая, что у слова, может быть разное количество суффиксов (в том числе и не одного), возникает проблема неоднозначности отделения суффиксов, что влечёт за собой полный перебор вариантов отсечения суффиксов и анализ этих вариантов, что в свою очередь сильно замедляет процесс анализа. Для решения этой проблемы было принято решение вместо отсечения простых суффиксов отсекать всевозможные пары суффиксов таким образом, чтобы отсекаемая часть была как можно больше. Это может повлечь за собой отсечение лишних букв от основы, однако, учитывая то, что в работе нет задачи формирования словоформ из основы с требуемым набором морфологических характеристик, этим фактом можно пренебречь.

С другой стороны при отделении окончания, нам необходимо учесть все возможные варианты, так как окончание определяет морфологические характеристики словоформы, которые необходимы для следующего этапа анализа (синтаксического).

Итак, морфологическому анализатору для каждой исходной словоформы на вход подаются основа и окончание, которое может быть пустым. Для определения типа лексемы (набора морфологических параметров) используется алгоритм из следующих шагов:

1) производится поиск основы в словаре основ. Если в словаре содержится данная основа, происходит переход к шагу 2, в противном случае алгоритм завершает свою работу с ошибкой;

2) для найденной в словаре основы $b_i \in B$ определяется флективная группа (δ_2). Флективная группа является комбинацией из одного или нескольких флективных типов, которые соответствуют основе;

3) далее для каждого флективного типа флективной группы основы происходит обращение в базу данных флективных типов. Эта база позволяет по выделенному из словоформы окончанию определить все требуемые морфологические параметры. Работа

алгоритма завершается, и в качестве результата возвращаются найденные параметры.

Таким образом, на выходе лексического анализатора мы получаем множество цепочек из лексем (основа с набором морфологических параметров). Эти же цепочки поступают на вход синтаксического анализатора, в задачу которого непосредственно и входит процесс формализации фразы. В качестве формального представления фразы в данной работе предлагается использовать функциональную форму. Пример функциональной формы:

Мама мыла красивую раму.
мыть(Мама, красивый(рама))

Формирование такой функциональной формы было осуществлено при помощи синтаксически ориентированного перевода, реализованного в виде LL(1)-анализатора. Так как естественный язык является достаточно сложным, то его не представляется возможным описать одной LL(1)-грамматикой. Поэтому в данной работе предлагается использовать несколько LL(1)-грамматик, каждая из которых реализует некоторые аспекты синтаксиса русского языка, в том числе возможно и неверные конструкции. При этом синтаксический анализатор пытается провести разбор последовательно по всем грамматикам. Если входные данные удовлетворяют условиям хотя бы одной LL(1)-грамматики, то синтаксический анализ признается успешным (структура синтаксического уровня в целом приведена на рисунке 2).

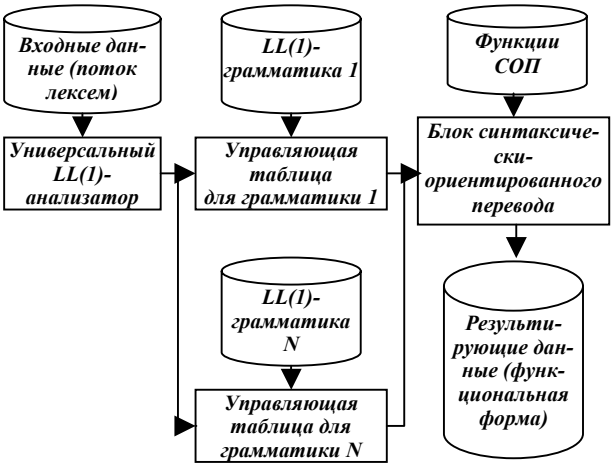

Рисунок 2 – Структура синтаксического анализатора

Множество LL(1)-грамматик, предложенных в данной работе, построено по принципу расширения ядра: каждая грамматика описывает какие-либо особенности синтаксиса естественного языка. Это множество можно условно разделить на уровни следующим образом: грамматики первого уровня описывают простейшие конструкции естественного языка, второй уровень является расширением первого и так далее. Некоторые грамматики по множеству поддерживаемых предложений могут пересекаться, некоторые грамматики целиком содержатся в других, грамматики могут также выходить за пределы естественного языка. Вместе взятые, эти грамматики достаточно полно описывают множество фраз естественного языка.

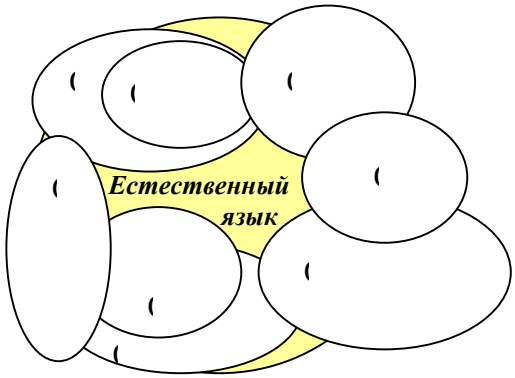

Рисунок 3 – Сравнительная мощность множества LL(1)-грамматик по отношению к естественному языку

Таким образом, естественный язык задается множеством контекстно-свободных грамматик:

$$G_k = (V_t, V_n^{(k)}, P, H, S^{(k)}),$$

где $V_t = \{ g_i \mid g_i = \delta_2^g (b); b \in B \}$ – множество терминальных символов;

$V_n^{(k)}$ – множество нетерминальных символов;

H – функция синтаксически-ориентированного перевода;

$P = \{ A \rightarrow a_1 a_2 \ldots a_m \}$ – множество правил.

Результатом синтаксического анализа является функциональная форма, которая используется на семантическом уровне для уже непосредственного поиска информации, семантическая нагрузка которой соответствует поисковому запросу (функциональной форме). Для этого системе необходима некая база знаний о мире, которая бы содержала в себе семантику ЕЯ. В качестве такой базы используется некий гибрид семантических

сетей и функционального представления данных. Формально это выглядит следующим образом:

$$S = <I, C>,$$

где I = (T, F) – множество информативных единиц (узлы);

$C=(C_1, C_2)$ – множество связей между элементами.

T – подмножество простых элементов (как правило сущ.).

F – подмножество элементов-функций (как правило глаголы, прилагательные, причастия).

Функции определяют семантическую структуру исходной синтаксической конструкции. Каждое слово в исходном тексте представимо некоторым элементом словаря и является либо функцией, либо ее аргументом в построенной суперпозиции.

Множество C_1 – это связи типа IS-A, то есть каждая такая связь связывает простые элементы. C_2 – это функциональные связи, т.е. указатели на аргументы. Например, связь C_1 может существовать м/у основами «мальчик» и «человек» («мальчик» - это «человек» / «boy» IS A «man»). В свою очередь связи C_2 могут использоваться, например, м/у глаголом и существительным или м/у прилагательным и существительным («мама мыла красивую раму» - мыть(мама), красивый(рама)).

Однако, при такой организации базы знаний достаточно сложно связывать координаты текста (URL) с той семантической информацией, которая представлена в этом тексте. Для решения этой проблемы был введен еще один слой базы знаний, называемый классификатором. По структуре он полностью совпадает с основной базой знаний, но содержит в себе информацию только о конкретной области (медицине, журналистике, образовании, химии и т. д.). В этом случае URL привязывается именно к классификатору, а не к узлам базы знаний.

Таким образом, при поиске происходит последовательное преобразование фразы в функциональную форму, затем эта форма ищется в одном или нескольких классификаторах (при этом найденная форма может отличаться от искомой), после чего выдаются ссылки на информацию, отнесённую к найденному классификатору.

СПИСОК ЛИТЕРАТУРЫ

1. Некресьянов И., Пантелеева Н. Системы текстового поиска для ВЕБ – http://ir.apmath.spbu.ru.
2. Зализняк А.А. Грамматический словарь русского языка. – М.: Русский язык, 1980.
3. Ножов И.М. Реализация автоматической синтаксической сегментации русского предложения. Автореф. дисс. к.т.н. – М., 2003.
4. Тестелец Я.Г. Введение в общий синтаксис. – М: РГГУ, 2001.