

## СИНТАКТО-ЛЕКСИЧЕСКИЙ АНАЛИЗ ТЕКСТА В IP-СИСТЕМЕ МАШИННОГО ПЕРЕВОДА НА БАЗЕ СЛ-ДЕЕРЕВА

Полянский К.В.

Научный руководитель – д-р. техн. наук, профессор Ковалев И.В.

*Сибирский федеральный университет*

Машинный перевод (МП) за историю своего существования развивался по нескольким направлениям. Следует выделить такие из них, как:

- Машинный перевод (МТ-системы)
- Память перевода (ТМ-системы)
- Он-лайн перевод

Если объединить принципы работы систем машинного перевода (СМП) в рамках последних двух направлений, то полученная СМП будет обладать свойствами онлайн ТМ-системы. Однако, в качестве памяти перевода в такой системе будем использовать не базу знаний (как в классических ТМ-системах), а сеть Интернет. Для обращения к сети Интернет данная СМП (назовем ее IP-СМП) должна взаимодействовать с информационно-поисковыми системами (ИПС). Алгоритм перевода IP-СМП следующий:

- Анализ текста на исходном языке (ИЯ-текста)
  - Удаление стоп-термов
  - Стемминг
  - Взвешивание
- Построение запроса к ИПС на основе анализа ИЯ-текста
  - Получение коллекции текстов на целевом языке (ЦЯ-текстов)
  - Анализ ЦЯ-текстов
- Синтез текста-перевода из коллекции ЦЯ-текстов
  - Нахождение цепочек перевода в ЦЯ-текстах
  - Оценка качества перевода

Данная схема показывает, что важным этапом в работе IP-СМП является процесс анализа ИЯ-текста и релевантных ему ЦЯ-текстов. Для проведения качественного анализа текста IP-СМП в данной статье предлагается использование блока синтакто-лексического анализатора. Рассмотрим принцип его работы.

Синтакто-лексический анализ (СЛА) ИЯ/ЦЯ-текстов, включает в себя, в отличие от классического подхода, одновременно две стадии анализа - синтаксический и лексический. В основе СЛА лежит построение синтакто-лексического дерева (СЛ-дерева), структура которого позволяет одновременно хранить информацию как о синтаксических, так и о лексических свойствах рассматриваемых структур текста. Построение СЛ-дерева осуществляется в несколько этапов:

- Вычленение из текстового массива глав (*Part*) и абзацев (*Art*) на основе шаблонов разметки текста.

- Деление абзацев на предложения (*Sent*) с идентификацией знаков препинания (*Punct* : «begin» «,» «:» «?» и т.д.) на основе словаря знаков препинания.
- Определение всех существующих в тексте термов (*Term*), их привязка к знакам препинания и помещение данных термов в упорядоченный двунаправленный список. Каждый терм помечается как главная или служебная часть речи. Список *Term* содержит все термы текста.
- Выполнение стемминга термов – выделения основ слов (*Stem*) и их помещение в упорядоченный двунаправленный список. Каждый *Stem* из списка может быть связан со множеством *Term*, однако каждый *Term* может быть связан только с одним *Stem*. Стемминг терминов осуществляется при помощи словаря окончаний и приставок.
- Построение цепочек *Chain* на основе повторяющихся терминов *Term* в предложениях.

Главными компонентами СЛ-дерева являются двунаправленные упорядоченные списки *Term*, *Stem* и *Chain*, элементы которых связаны друг с другом определенным образом и несут основную лексическую информацию о тексте и его частях. Связанность списка *Term* в СЛ-дерева с элементами *Text*, *Art*, *Part*, *Sent* и *Punct*, представляющими синтаксическую часть дерева, дает возможность определять для *Term*, *Stem* и *Chain* их привязку к синтаксису предложений, абзацев и глав. Сформированное таким образом СЛ-дерево пригодно для проведения частотного анализа текста, а также его составных частей. На основе структуры СЛ-дерева, связей между его элементами, можно получить полную синтаксическую и лексическую информацию о каждом элементе *Term*, *Stem* и *Chain*. Пример СЛ-дерева представлен на Рисунке 1.

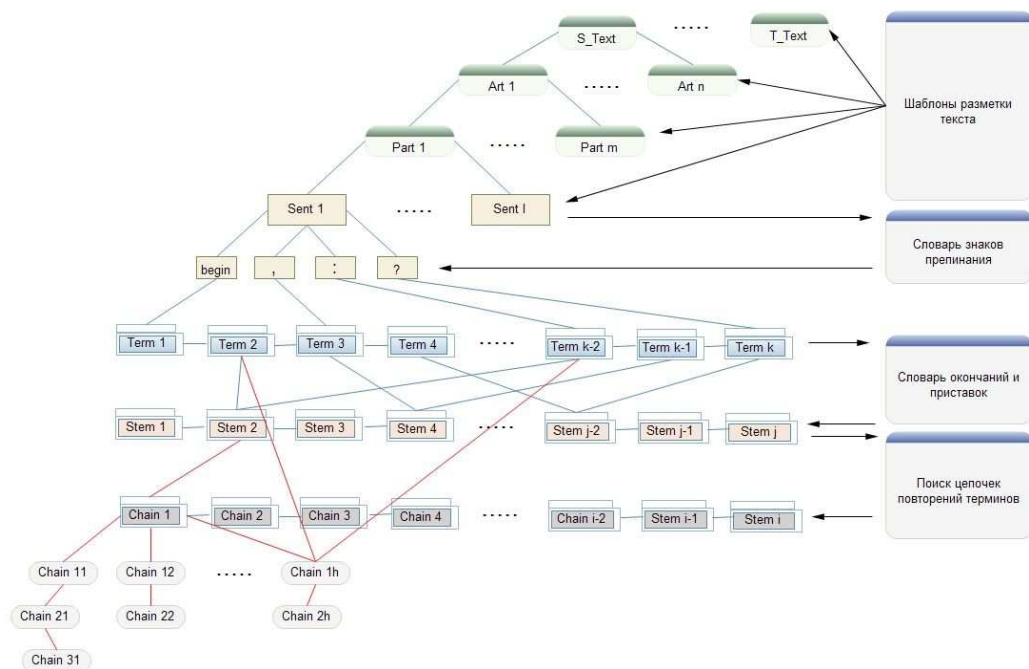


Рис. 1. Структура СЛ-дерева

На основе СЛ-дерева осуществляется построение цепочек повтора термов. Механизм образования таких цепочек приведен на Рисунке 3. Формирование цепочек повтора происходит во время построения СЛ-дерева. Каждый добавленный в дерево узел-терм *Term* проходит этап стемминга, в процессе которого происходит вычленение его основы. Все новые основы заносятся в список основ *Stem*, располагающийся на уровень ниже списка термов. Между парой *Term* и *Stem* ставится связь. В случае если узел *Stem* уже имелся в списке, то добавления нового *Stem* не происходит, а связывание *Term* происходит с уже существующим *Stem* из списка. Если ситуация с существующими *Stem* наблюдается два или более раза подряд, и эти последние *Stem* указывают на соседние *Term* из списка, имеет место образование цепочки повтора термов *Chain*. Цепочка повтора имеет организацию в виде древовидной структуры, а любая последовательность элементов данного дерева от заданного узла к корню и представляет собой цепь *Chain*. Каждый элемент цепи связывается с двумя или более *Term* в списке, которые фактически являются реальными отображениями цепочки на пространство *Term*.

Для построения цепочек повтора *Chain* используется механизм автоматной грамматики. Структура автоматной грамматики  $G$  приведена на Формуле 1.

$$G = (V = \{root, sheet, \varepsilon\}, N = \{A, S\}, S, R), \text{ где} \quad (1)$$

$V$  -алфавит грамматики (состоит из элемента вершины дерева *root*, листьев этого дерева *sheet* и пустого символа  $\varepsilon$ ),

$N$  -множество нетерминальных символов (состоит из стартового символа  $S$  и нетерминала вывода листа дерева  $A$ ),

$R$  -набор правил, в соответствии с которыми осуществляется вывод грамматики  $G$ . Набор правил вывода представлен на Формуле 2.

$$R: \quad (2)$$

$$S \rightarrow rootA$$

$$A \rightarrow sheetA$$

$$A \rightarrow \varepsilon$$

Схематично автоматную грамматику, реализующую цепочки повтора *Chain*, можно представить в виде конечного автомата, а вывод такой грамматики в виде дерева. Их структура представлена на Рисунке 2.



Рис. 2. Граф автоматной грамматики вывода цепочек повтора Chain (слева) и дерево вывода этих цепочек (справа)

Следует отметить, что грамматика  $G$  описывает построение лишь одной ветки дерева для элемента  $Stem$  с вершиной  $root$ . В случае появления альтернативной ветки цепочки повтора  $Chain$  вывод повторяется по тому же алгоритму, а вершина остается неизменной. В итоге, несколько цепочек построенных по принципу автоматной грамматики образуют дерево с общей вершиной  $root$ . Если при идентификации системой следующей цепочки повтора ее начало отлично от  $root$ , то строится новое дерево, где  $root[i] = root[i + 1]$ , а  $i$  - порядковый номер построенных деревьев. В результате набор цепочек повтора  $Chain$  может быть представлен, как

$$\{root[1], (sheet[1][1], sheet[1][2], \dots, sheet[1][a]), (sheet[m][1], sheet[m][2], \dots, sheet[m][b]), \dots, \\ \{root[k], (sheet[1][1], sheet[1][2], \dots, sheet[1][c]), (sheet[n][1], sheet[n][2], \dots, sheet[n][d])\} \quad (3)$$

где

$k$  - количество созданных деревьев с уникальными вершинами  $root$ ,

$a, b$  - длины первой и последней цепочки с общей вершиной  $root[1]$ ,

$c, d$  - длины первой и последней цепочки с общей вершиной  $root[k]$ .

Наглядное представление формирования цепочек  $Chain$  представлено на Рисунке 3.

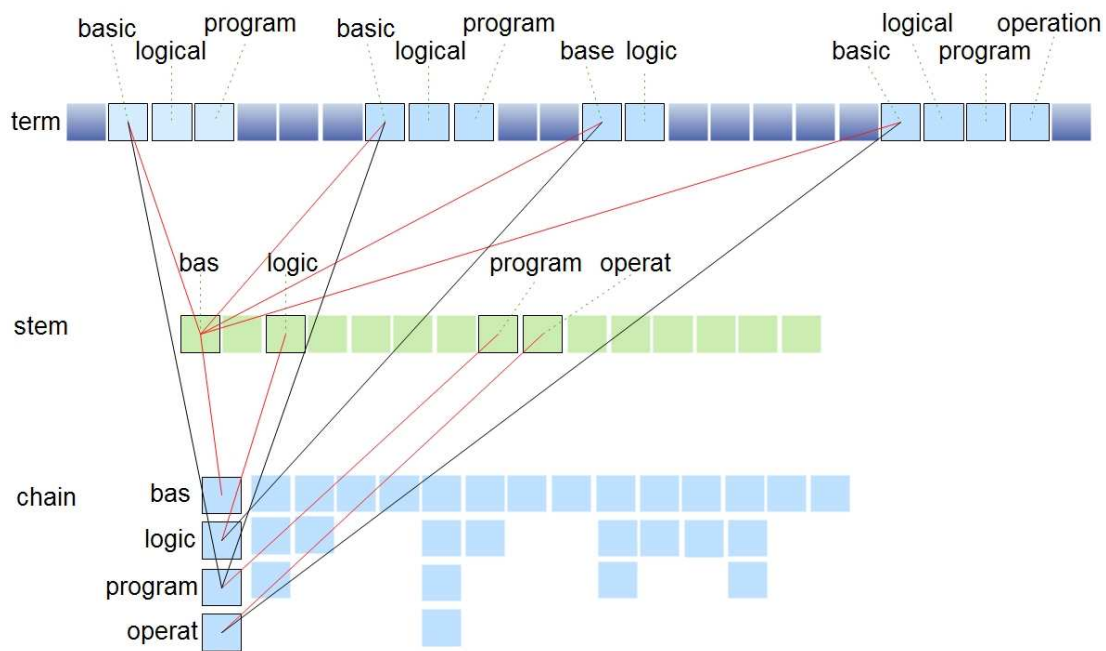


Рис. 3. Построение цепочек повторов Chain

После построения СЛ-дерева на его базе выполняется дальнейший анализ текста. Возможность получения единой синтакто-лексической информации о тексте играет важную роль в процессе перевода IP-СМП. Формирование цепочек повтора позволяет объединять термины в семантически связанные группы, а значит осуществлять более качественный перевод.