Т.В. Батура, Ф.А. Мурзин

ОБРАБОТКА ПОИСКОВЫХ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ПОМОЩЬЮ REFAL-ПОДОБНЫХ КОНСТРУКЦИЙ

В статье кратко обосновывается возможность применения модификаций конструкций языка символьных преобразований REFAL для формирования деревообразного представления предложений на естественном языке и схем «вопрос-ответ» и описан алгоритм использования их в поисковых системах. В действительности, сейчас имеется большой список, более сорока схем типа «вопрос-ответ», которые могут быть полезны при реализации программных систем, ориентированных на обработку текстов.

1. КОНСТРУКЦИИ ЯЗЫКА REFAL

Язык программирования REFAL является одним из языков, созданных для проведения символьных преобразований на компьютерах [1]. Это типичный язык продукций, и он аналогичен языку SNOBOL. Его операторы представляют собой продукции вида $\varphi \to \psi$, которые обозначают, что если слово имеет свойство φ, то необходимо применить действие ψ. Отметим, что так называемый функциональный стиль естественным образом присущ языку REFAL. Ниже дано формальное описание синтаксических свойств этого языка, описана виртуальная REFAL-машина, и даны некоторые примеры программ.

Предположим, что зафиксированы: T — алфавит объектных символов, Q — алфавит вспомогательных символов (например, / , $\perp$, $\to$, скобки, двоеточие, запятая), $T \cap Q = \varnothing$ и F — алфавит функциональных символов.

Имеется три типа переменных s_i, e_i, $i \in \omega$, где ω — множество натуральных чисел.

Если S — произвольный алгоритм, то через S^* обозначим множество всех слов над S, включая пустое слово.

1. Формула языка REFAL определяется индуктивно:

a) переменная является формулой,

b) любое $t \in T$ является формулой,

c) если φ — формула, то (φ) — формула,

d) если φ, ψ — формулы, то их конкатенация $\varphi\psi$ — формула,

e) если φ — формула и $f \in F$, то $/ f / \varphi \perp$ — формула,

f) других формул нет.

Если формула φ получена без применения правила e), то мы назовем ее простой. Множество всех переменных, входящих в формулу φ, обозначим var (φ).

2. Оператором называется слово $\varphi \to \psi$, где φ, ψ — формулы, при этом φ — простая и $\text{var}(\varphi) \supseteq \text{var}(\psi)$.

3. Подпрограммой называется любой столбец, имеющий вид:

$$f : L_1$$
$$\vdots$$
$$L_n,$$

где $f \in F, L_1, ..., L_n$ — операторы, $n \in \omega$. При этом, f называется именем подпрограммы.

4. Программа есть столбец вида:

$$F_1$$
$$\vdots$$
$$F_m,$$

где F_i — подпрограммы, $m \in \omega$.

5. Пусть φ — простая формула, $t \in S^*$, где $S = T \cup \{(,)\}$. Определим функцию i, мы будем называть ее функцией отождествления. Функция i паре $<\varphi, t>$ сопоставляет кортеж, с описанными ниже свойствами, если такой кортеж существует. В противном случае она сопоставляет 0. В этом случае говорят, что отождествление невозможно.

Пусть $\varphi = x_1 ... x_k$ — формула, и для любого j выполнено $x_j \in S$, либо x_j — переменная. Мы полагаем $i(\varphi, t) =< c_1, ..., c_m >$, если $k = m$ и выполнены свойства:

a) $x_j \in S \to c_j = x_j$,

b) $x_j = s_i \to c_j \in T$,

$x_j = e_i \to c_j \in S^*$,

c) $x_j = x_i \to c_j = c_i$,

d) любое c_j имеет правильно построенную скобочную структуру,

e) если $< c'_1 , ..., c'_k >$ — произвольный кортеж со свойствами (a)–(d), то

$$< | c_1 |, ... , | c_k |> \leq < | c'_1 |, ... , | c'_k |>$$

в лексикографическом порядке, где $| c_j |, | c'_j |$ — длины слов c_j , c'_j соответственно.

В итоге мы можем сделать следующие замечания:

- функция i используется, как функция отождествления с образцом,
- переменные типа s_i служат для обозначения символов,
- переменные типа e_i служат для обозначения выражений,
- скобки ( ) используются для того, чтобы фиксировать синтаксическую структуру строк.

6. REFAL-машина состоит из поля зрения и поля памяти и работает в дискретном времени. В текущий момент у нее в поле зрения находится формула, не содержащая переменных, а в поле памяти — программа. Опишем шаг работы машины.

Если в поле зрения находится простая формула, то процесс вычисления на этом заканчивается, и это считается нормальным окончанием.

В противном случае в формуле из поля зрения выделяется самая левая подформула, имеющая вид $/ f / \theta \perp$, такая, что θ простая. После этого отыскивается в программе подпрограмма с именем f. Если такой нет или их несколько, то в поле памяти появляется диагностика этого, и работа машины заканчивается.

В противном случае в подпрограмме с именем f отыскивается самый первый оператор $\varphi \to \psi$ такой, что $i(\varphi, 0) \neq 0$. Если такого оператора нет, то в поле памяти появляется информация об этом, и работа заканчивается.

Допустим теперь, что требуемый оператор имеется. Тогда подформула $/ f / \theta \perp$ в поле зрения заменяется на ψ *. Слово ψ * получается из формулы ψ заменой переменных на значения, которые они получили при отождествлении ψ и θ. Значениями переменных называются c_j из пункта b) в определении функции отождествления.

Рассмотрим теперь несколько примеров. Они показывают, что, используя скобочные структуры специального вида, можно очень коротко записывать довольно сложные программы.

Пример. Пусть в поле зрения REFAL-машины содержится

$/\, f\, /(+a-b+\sin(x))\ \perp$,

а в поле памяти содержится программа

$f : (s_0 e_1 + e_2) \to /\, f\, /(s_0 e_1)\perp /\, f\, /(+e_2)\perp$

$(s_0 e_1 - e_2) \to /\, f\, /(s_0 e_1)\perp /\, f\, /(-e_2)\perp$

$e_0 \to e_0$.

Пусть теперь

$\varphi = (s_0 e_1 + e_2),\ \theta = (+a-b+\sin(x))$.

Тогда имеем

$i(\varphi,\theta) =< (, +, a-b, +, \sin(x),) >$.

Это может быть наглядно представлено в виде

(	s_0	e_1	+	e_2	)
↓	↓	⇓	↓	⇓	↓
(	+	$a-b$	+	$\sin(x)$	).

В этой схеме скобки переходят в скобки, знаку «плюс» соответствует знак «плюс». Переменная s_0 имеет значение +, переменная e_1 имеет значение $a-b$, переменная e_2 имеет значение $\sin(x)$. Поэтому сначала должен быть выполнен первый оператор. В итоге мы получаем в поле зрения REFAL-машины $/\, f\, /(+a-b)\perp /\, f\, /(+\sin(x))\perp$.

Затем исполняется левое вхождение f. Очевидно, что

$$i((s_0 e_1 + e_2),(+a-b)) = 0\,.$$

Поэтому первый оператор программы не может сработать.

Далее имеем

$$i((s_0 e_1 - e_2),(+a-b)) =< (, +, a, -, b,) >,$$

и выполняется второй оператор. Мы можем представить функцию i в виде

(	s_0	e_1	−	e_2	)
↓	↓	↓	↓	↓	↓
(	+	a	−	b	).

После второго шага мы будем иметь в поле зрения:

$$/\, f\, /(+a)\perp /\, f\, /(-b)\perp /\, f\, /(+\sin(x))\perp.$$

Далее может работать только третий оператор. Все дальнейшие изменения содержимого поля зрения изображены ниже, включая последний шаг:

$$(+a)/f/(-b) \perp / f /(+\sin(x)) \perp$$
$$(+a)(-b)/f/(+\sin(x)) \perp$$
$$(+a)(-b)(+\sin(x)).$$

В итоге получаем, что рассмотренная программа осуществляет разбиение выражений на слагаемые.

2. ОБРАБОТКА ПОИСКОВЫХ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Считаем, что поисковый запрос представляет собой совокупность предложений на естественном языке. Эту совокупность предложений можно расширить, используя словарные статьи из толкового словаря (например, словаря Ожегова), т. е. фактически приписать определения отдельных слов. Следующий этап состоит в том, чтобы представить данные предложения в виде помеченных деревьев. Вершины помечаются словами, а ребра — вопросами, задаваемыми от одного слова к другому.

Далее рассмотрим текст достаточно большого объема, из которого необходимо выбрать предложения по тематике поискового запроса и, таким образом, сформировать аннотацию или решить, является ли текст релевантным данному запросу. Для этого предложения данного текста также могут быть представлены в виде деревьев (вообще говоря, необязательно все предложения, а выборочно по некоторым критериям). После этого необходимо сопоставление на похожесть (соответствие) деревьев, полученных из запроса, и деревьев, возникших из текста.

Для аннотации выбираются предложения, которые соотносятся по теме, имеют похожие структуры и т. д. На основании подобных идей можно судить о релевантности.

Обработка поисковых запросов на естественном языке предполагает выполнение ряда действий.

- Семантико-синтаксический разбор запроса.
- Генерация схем возможных ответов.
- Нахождение в тексте фрагментов в соответствии со схемами ответов.
- Анализ контекстной связности между предложениями.

Поисковый запрос может представлять собой либо вопрос, поставленный в явном виде, либо набор фраз, задающих тему. Мы должны в тексте найти ответ на поставленный вопрос, либо выделить фрагменты на заданную тему. Для простоты считаем, что запрос состоит из одного предложения.

2.1. Семантико-синтаксический разбор запроса

Семантико-синтаксический разбор запроса заключается в том, что запросу сопоставляется дерево семантико-синтаксического разбора. В общем случае, это будет размеченное дерево. Пометки ребер будем также называть вопросами. Дерево может быть классического типа, из тех, которые применяются в лингвистике, так и неклассического.

Например, ребра могут помечаться семантическими предикатами, т. е. лексическими функциями по терминологии Мельчука; предикатами, выявленными в процессе изучения структуры словарных статей в словаре Ожегова и т. д. В общем случае можно считать, что ребра помечены двухместными предикатами, которые должны быть истинными на соответствующих пометках вершин.

Результирующее дерево представляется в виде скобочной структуры. Для размеченных деревьев можно использовать следующую конструкцию. Предположим, что (x_1, q, x_2) — помеченное ребро, q — метка. Тогда его скобочное представление будет иметь вид $(x_1\ (q\ (x_2)))$. Понятно, что данное преобразование можно применять рекурсивно.

2.2. Генерация схем возможных ответов

На данной стадии вопросу сопоставляется множество возможных ответов. При этом могут быть использованы конструкции, которые аналогичны конструкциям, применяемым в языке REFAL.

Рассмотрим пример. Вопрос: «Сколько тебе лет?». Ответ: «Мне 15 лет». Договоримся, переменные вида s_i использовать для символов, а переменные e_i использовать для совокупностей слов и для выражений, содержащих скобки.

Тогда может быть предложена схема перехода «вопрос—ответ»

$$e_0\ \textit{тебе}\ e_1 \to \textit{мне}\ s_2 e_1 .$$

Более сложная схема может учесть возможный контекст

$$e_0\ \textit{тебе}\ e_1 \to e_3\ \textit{мне}\ s_2 e_1 e_4 .$$

Если мы будем работать с деревьями синтаксического разбора, то переходу

$$(\textit{сколько}\ (\textit{тебе лет})) \to (\textit{мне}\ (15\ \textit{лет}))$$

соответствует схема

$$(e_0\ (\textit{тебе}\ e_1)) \to (e_3 (\textit{мне}\ (s_2 e_1)) e_4) .$$

Большое количество таких схем можно получить, заглянув в учебники иностранных языков, школьные учебники по конкретным предметам, в учебники по скорочтению. В последних предлагается быстро прочесть текст, а потом с помощью вопросов осуществляется контроль качества его усвоения.

Используя размеченные деревья, можно учесть морфологические особенности слов и их согласование по родам, падежам и т. д. Но в действительности, целесообразно модифицировать REFAL-подобные конструкции, введя новые типы переменных, т. е. сделать их более типизированными и ориентированными на лингвистику. Можно ввести специальные переменные для частей речи. Например, $e[Adj]_0$, где *Adj* — прилагательное и т. д.

В настоящее время выделено более 40 схем, соответствующих стандартным вопросам и ответам. Предложения, в соответствии с которыми строились схемы, взяты из учебника «Do You Speak English?» B.&R. Retman [2]. Каждой из схем сопоставлено представление с вовлечением скобочных структур и использованы расширенные типы переменных языка REFAL. Например,

1. Это красная машина $\rightarrow$ Да, это красная машина

Это $((attr \leftarrow e[Adj]_0))\ sub(e_1)) \rightarrow$ Да, это $((attr \leftarrow (e[Adj]_0))\ sub(e_1))$.

1'. Это красная машина $\rightarrow$ Нет, это не красная машина

Это $((attr \leftarrow e[Adj]_0))\ sub(e_1)) \rightarrow$ Нет, это $((attr \leftarrow (не\ e[Adj]_0))\ sub(e_1))$.

1". Это красная машина $\rightarrow$ Нет, это зеленая машина

Это $((attr \leftarrow e[Adj]_0))\ sub(e_1)) \rightarrow$ Нет, это $((attr \leftarrow (e[Adj]_2))\ sub(e_1))$.

Здесь $attr \leftarrow$ — определение слева от определяемого слова, *sub* — подлежащее.

Более интересными и полезными являются, например, переменные вида f_i, которые будут обозначать, что слова одинаковы с точностью до флексии, т. е. изменений суффиксов и окончаний. При отождествлении левой части продукции с запросом переменная f_i будет отождествлена со словом как s_i. А при отождествлении правой части продукции с предложением в тексте данное слово будет отыскиваться с точностью до флексии.

Можно ввести специальную переменную, которая будет обозначать, что совпадают достаточно длинные начальные части слов, например, не менее 75% каждого из них. Заметим, что слова могут быть разной длины. Такого типа сравнение полезно тем, что оно алгоритмически просто и не требует морфологического разбора. В то же время, для длинных слов указанное частичное совпадение автоматически обозначает, что это одно и то же сло-

во, с точностью до флексии. На ранних стадиях развития ребенка, по-видимому, именно так и происходит.

2.3. Нахождение в тексте фрагментов в соответствии со схемами ответов

Заданному вопросу соответствует несколько возможных ответов. Поэтому можно считать, что схема перехода «вопрос-ответ» имеет вид $\varphi \to \psi_1 \vee \psi_2 \vee \ldots \vee \psi_N$.

После отождествления φ с вопросом переменные, входящие в нее, приобретают значения. Далее в тексте ищем предложения, которые можно отождествить хотя бы с одной из формул ψ_i. Все такие предложения выдаем пользователю как ответы.

Рассмотрим пример, приведенный выше. В нем

φ = Это $((attr \leftarrow e[Adj]_0))\ sub(e_1))$

ψ_1 = Да, это $((attr \leftarrow (e[Adj]_0))\ sub(e_1))$

ψ_2 = Нет, это $((attr \leftarrow (\textit{не}\ e[Adj]_0))\ sub(e_1))$

ψ_3 = Нет, это $((attr \leftarrow (e[Adj]_2))\ sub(e_1))$.

Таким образом, для данного примера схема перехода «вопрос-ответ» кратко запишется $\varphi \to \psi_1 \vee \psi_2 \vee \psi_3$.

Заметим, что, говоря об отождествлении, мы можем рассматривать как сами предложения, так и результаты синтактико-семантического разбора, и работать с ними, что, безусловно, более интересно, и может привести к более качественным результатам.

Целесообразно предусмотреть специальные метки, которые позволят управлять областями отождествления и выдаваемыми областями. Например, формулу ψ_i можно отождествлять не с отдельным предложением, а с целым абзацем. Другой вариант, когда ψ_i отождествляется с предложениями, но после нахождения соответствующего предложения пользователю выдается весь абзац, в котором оно найдено. Поэтому можно считать, что схемы переходов «вопрос-ответ» имеют вид

$l : \varphi \to \psi_1 \vee \psi_2 \vee \ldots \vee \psi_N$, где l — метка.

2.4. Анализ контекстной связности между предложениями

В текстах на естественном языке наблюдается явление, называемое контекстной связностью. Например, мы хотели бы выделить в тексте все пред-

ложения, в которых идет речь о птице вороне. В ряде предложений встречается слово «ворона» с точностью до флексии, а в ряде предложений может встречаться «эта птица». Если слова «эта птица» в соответствующих предложениях заменить на «ворона», то полученные предложения можно анализировать в соответствии с методами, описанными выше.

Связать «ворона» и «эта птица» можно, если заглянуть в толковый словарь. Там написано, что «ворона — большая всеядная птица…».

В ряде случаев способы обнаружения контекстной связности более-менее простые. В данном случае имеется указательное местоимение «эта», слово при нем «птица», как правило, согласовано в роде и падеже со словом «ворона», но вообще говоря, это не обязательно.

Самое главное, что слово «птица» встречается в соответствующей словарной статье толкового словаря. Последнее легко проверить на компьютере. Фактически, в толковом словаре содержится информация о том, что имеет место истинность лексического предиката $Gener(ворона, птица)$, который обозначает, что «птица» является более общим понятием, чем «ворона».

Отметим в заключение, что вопрос о контекстной связности требует дополнительного изучения. В целом он очень сложный, но типовые ситуации можно достаточно легко описать и реализовать на компьютере.

ВЫВОДЫ

На основе изложенного выше могут быть сделаны следующие выводы. При формировании деревообразного представления предложений на естественном языке предлагается использовать модифицированные конструкции языка символьных преобразований REFAL:

1. Целесообразно использовать новые типы переменных, связанные с частями речи, частичным совпадением слов и т. д.;

2. В языке REFAL для любого оператора $\varphi \to \psi$ выполнено $\mathrm{var}(\varphi) \supseteq \mathrm{var}(\psi)$. У нас это нарушается, и таким образом пытаемся учесть контекст.

3. Заданному вопросу соответствует несколько возможных ответов. Поэтому можно считать, что схема перехода «вопрос—ответ» имеет вид $\varphi \to \psi_1 \vee \psi_2 \vee ... \vee \psi_n$.

4. После отождествления φ с вопросом переменные, как и в обычном языке REFAL, входящие в нее, приобретают значения. Далее в тексте ищем

предложения, которые можно отождествить хотя бы с одной из формул ψ_i. Все такие предложения выдаем пользователю в качестве ответов.

СПИСОК ЛИТЕРАТУРЫ

1. Murzin F.A. Syntactic properties of the REFAL language // Int. J. Computer Math. — 1985. — N17. — P. 123 — 139.
2. Retman B.&R. Do You Speak English? — Warszawa: Wiedza Powszechna, 1977. — 160 p.