

(С.М. Салибекян, П.Б. Панфилов)

(Название статьи!)¹

АННОТАЦИЯ. В статье приводится описание применения объектно-атрибутного (ОА) подхода организации вычислительного процесса (собственная разработка) для семантического анализа естественного языка. Подход объединяет в себе методы синтеза семантического графа из исходного текста, нотацию для описания правил синтеза семантического графа, математическую модель функционирования вычислительной системы, способ семантического поиска, архитектуру вычислительной системы, осуществляющей семантический поиск. Объектно-атрибутный подход реализует парадигму управления вычислительным процессом с помощью потока данных (dataflow), который обеспечивает массу положительных качеств: параллелизм вычислений, реализация на распределенной вычислительной системе, развитая абстракция, самообучение вычислительной системы и т.д..

Ключевые слова и фразы: вычислительная система с управлением потоком данных, dataflow, моделирование, математическая модель, конечный автомат, сеть Кана, сложно структурированные данные, распределенная вычислительная система, автоматическая обработка естественного языка!

Введение

Проблема семантической обработки естественного языка (Natural Language Processing - NLP) привлекает все большее и большее внимание ученых, ведь уже находит широчайшее применение на практике: автоматическая обработка документов, диалоговые системы, системы интернет-поиска и т.д. Актуальность данного научного направления повышает лавинообразное увеличение объема текстовой информации, которую необходимо обрабатывать, и глобализация мира, увеличивающая потребность автоматическо-

¹ (Рекомендована к публикации.... Поддержана...!)

го перевода с/на иностранный язык. Следует сказать и о чрезвычайно популярной в последнее время концепции семантического Web (The Semantic Web), согласно которой компьютер должен понимать содержание (смысл) веб-сайтов наподобие человека[1].

Для семантической обработки естественно-языкового (ЕЯ) текста необходимы огромные вычислительные мощности – текст изобилует множеством неоднозначностей (плисемия) и характеризуется чрезмерным обилием ассоциативных связей – так что, решение подобных задач под силу только суперкомпьютерам. Причем, желательно, чтобы суперкомпьютерная система была распределенной (MPP-система, GRID- и облачные системы), т.к. она обеспечивает свойство масштабируемости, обладает высокой вычислительной мощностью и имеет относительно низкую цену.

В последние 10 лет наблюдается бум теоретических и практических разработок в области семантического анализа текста, т.к. вычислительные мощности компьютеров выросли и позволили эффективно решать подобные задачи. Однако даже сейчас скорость семантического анализа порой не превышает нескольких слов в минуту. История же компьютерной лингвистики направления насчитывает 50 лет, и за это время было предложено множество методик семантического анализа естественного языка и методик представления знаний. В качестве примеров таких методик можно привести: метод «смысл – текст» Игоря Мельчука [2], семантические сети, , объектно-ориентированный подход или теория фреймов [3], функциональная семантика [4] и т.д.. Уже функционирует множество программных продуктов, ориентированных на семантическую обработку текста: Google graph, поисковые системы Ask, Wolfram|Alpha, hakia и т.д. Однако ни одна из методик, к сожалению, не нашла всеобщего признания, и ни один программный продукт не смог составить ощутимую конкуренцию интернет-поиску по ключевым словам. И работы в рамках данного научного направления до сих пор остаются в поисковой стадии.

В настоящей статье предлагается новый подход (объектно-атрибутивный (ОА) [5]) к семантическому анализу текста, обладающий совокупностью положительных качеств:

- Способность работы с любыми форматами данных, т.е. под конкретную вычислительную задачу создается своя информационная конструкция и, при этом, информационные конструкции могут быть модифицированы непосредственно во время вычислительного процесса.

- Реализация объектного принципа (т.е. возможность объединения в программы и данных, развитая абстракция как программы, так и данных).

- Способность обработки данных от простого к сложному, т.е. семантический анализ проходит несколько стадий – лексический, морфологический, синтаксический и семантический анализ.

- Удобная реализация на распределенных вычислительных системах (ВС).

- Динамическое распараллеливание вычислительного процесса.

- Масштабируемость.

- Управление вычислительным процессом с помощью потока данных (dataflow), который считается перспективным для создания высокопараллельных ВС.

Совокупность перечисленных качеств оставляет надежду на то, что у данной методики есть будущее и что она найдет практическое применение.

1. Необходимые условия Объектно-атрибутивный подход для анализа естественного языка

ОА-подход [6] и [7] для семантического анализа основан на принципе синтеза семантического графа (служит для описания смысла, заложенного в тексте), вершинами которого являются элементарные понятия (семы), а дугами – смысловые связи между ними, и последующем его анализе. В ОА-подходе семы представляются в виде информационных капсул (ИК), а нагрузки информационных пар (ИП) входящих в ИК, и могут хранить как пара-

метры объекта (или подьобъекта), так и ссылки на другие ИК (эти ссылки реализуют связи между семами). Атрибут же ИП служит идентификатором свойства объекта. ОА-граф может быть легко модифицирован или синтезирован путем добавления или удаления из него определенных ИП и ИК – т.е. ОА-информационная конструкция чем-то напоминает собой детский конструктор, из деталей которого (ИП и ИК) можно собрать всё, что угодно, и легко переделать (модифицировать) эту конструкцию в любой момент, в отличие от фреймового и объектно-ориентированного подходов (ООП), где структура объекта описывается жестко и заранее [8]. ОА- же граф синтезируется непосредственно во время вычислительного процесса, как в ООП (хотя в ООП и имеются средства модификации структуры объектов, однако эта возможность используется программистами не столь широко). К тому же, фреймы и ООП-классы представляют собой древовидный граф, что ограничивает их возможности по описанию объектов и их поведения; ОА- же граф может иметь любую топологию (единственное ограничение – граф должен быть связным).

ОА-подход предполагает две основные стадии анализа текста: 1) синтез семантического ОА-графа; 2) семантический поиск в ОА-графах.

На первой стадии анализа текста ключевую роль играет информационная конструкция «иерархический ОА-список» [9], которая позволяет хранить и обрабатывать список из описаний некоторых объектов (на первой стадии анализа текста объектами будут являться лексемы, таким образом, в ОА-списке будут находиться семантико-морфологические описания слов). Синтез семантического графа, в свою очередь, делится на две фазы: морфологический анализ (распознавание отдельных лексем) и синтаксический анализ.

Морфологический анализ основан на использовании семантико-морфологического словаря, представляющий собой ОА-граф типа «Список», в котором хранятся морфологические и семантические

(смысловые) свойства лексемы, а также ее мнемоника. Ввиду того, что лексемы естественного языка обладают свойством полисемии (многозначности), семантико-морфологическое описание лексемы представляет собой ОА-подсписок, в каждой записи которого помещается одно из толкований лексемы (рис.1): в дальнейшем во время синтеза смыслового графа из него будут отбракованы те толкования лексемы, которые не вписываются в контекст по смыслу или по семантике. При морфологическом анализе производится формирование списка описаний лексем: в список последовательно помещаются описания лексем, формирующих текст (описания берутся из семантико-морфологического словаря). Список толкований лексем текста преобразуется в семантический граф уже на фазе синтаксического анализа.

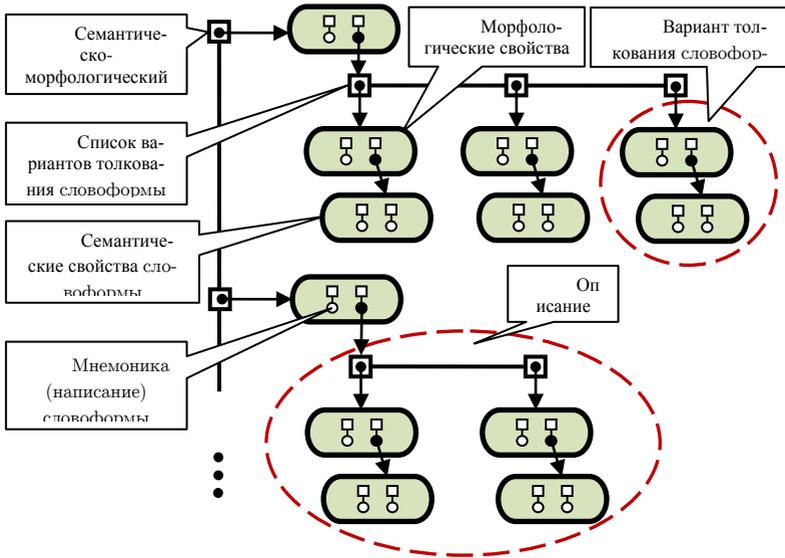


РИС. 1. Семантико-морфологический словарь и формирование списка описаний лексем

Семантический ОА-граф получается из списка толкований лексем помощью применения правил преобразования описывающих

синтаксис естественного языка (в результате применения правил происходит модификация списка лексем). Преобразования ОА-списка несколько напоминают правила преобразования в формальных грамматиках [10]. Однако имеются существенные отличия. Во-первых, обрабатываются не последовательность терминальных и нетерминальных символов, а капсулы (и ОА-графы) с описаниями морфологических и семантических свойств лексем. Во-вторых, обрабатывается не линейная последовательность терминальных и нетерминальных символов, а иерархический список (каждой лексеме соответствует свой подсписок ее семантико-морфологических толкований). В-третьих, итогом процесса вывода является не последовательность терминальных символов, а семантический ОА-граф. В-четвертых, в ОА-системе используется несколько проходов (итераций) преобразования. В-пятых, для модификации ОА-списка может применяться не только операция замены одной последовательности символов на другую, а несколько операций по модификации ОА-списка, например:

- склейка соседних лексем (когда, например, свойства лексемы-прилагательного переходят в описание объекта, заданного существительным);

- сцепка (когда создается связь между лексемами, например, в связке «существительное – глагол – существительное» существительные описывают объекты, в глагол задает связь между ними);

- создание списка альтернатив, когда некоторая языковая конструкция может быть интерпретирована несколькими способами (т.е. несколько толкований), то эта конструкция в списке заменяется на подсписок альтернатив;

- удаление неактуальной альтернативы (когда к определенной лексеме из списка лексем невозможно применить ни одно из правил преобразования, то считается, что ветвь, в которой находится эта лексема, не содержит актуального смысла и она удаляется из списка) (рис. 2).

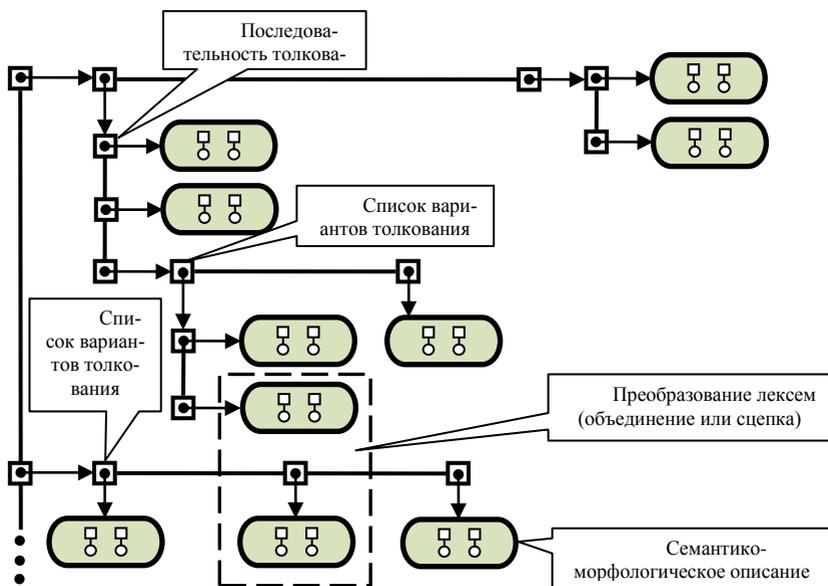


Рис. 2 – СПИСОК ОПИСАНИЙ ЛЕКСЕМ И ЕГО ПРЕОБРАЗОВАНИЕ

Для описания таких правил была разработана собственная нотация, наподобие правил, применяемых в формальных грамматиках. Как и в формальных грамматиках, правило состоит из двух частей: левой (заменяемое выражение) и правой (выражение на которое происходит замена). Эти две части правила разделяются знаком « \rightarrow ». В ОА-правиле преобразования информационная пара обозначается с помощью знака « $=$ »: с левой стороны знака помещается обозначение атрибута, с правой – нагрузка (данные). ИП, формирующие капсулу (ИК), ограничиваются фигурными скобками. ИК могут быть именованными, когда перед первой скобкой стоит обозначение капсулы. В качестве нагрузки ИП могут выступать указатели на ИК, тогда в обозначении капсулы после знака « $=$ » стоит обозначение ИК или последовательность ИП, формирующих капсулу, ограниченных фигурными скобками. Операция склейки двух ИК обозначается знаком « $*$ », т.е. в результате операции $\{mmemo=nil\}\{mmemo=nil\} \rightarrow \{mmemo=nil\}*\{mmemo=nil\}$ к последовательности капсул $\{mmemo="Word1"\}\{mmemo="Word2"\}$

получится капсула {mneмо="Word1" mneмо="Word2"} (nil – обозначение пустой нагрузки). Для обозначения альтернатив толкования лексем или фрагментов текста (т.е списка альтернатив) применяется знак « | », который может присутствовать как в левой, так и в правой частях правила преобразования: в левой части он обозначает, что для выполнения условия применения правила преобразования достаточно, чтобы в списке лексем присутствовала только одна из альтернатив; в правой части это обозначает, что будет сформирован ОА-подписьок альтернатив толкования лексем или фрагментов текста.

Правила применяются в несколько проходов. Например, для русского языка применяется пять проходов:

склейка прилагательных с наречиями степени;

склейка прилагательных и глаголов с качественными наречиями;

склейка прилагательных и существительных;

сцепка существительных и союзов, обозначающих пространственное положение;

сцепка существительных и глаголов.

На рис. 3 приведен фрагмент списка правил преобразования списка лексем для склейки существительных и прилагательных русского языка (Сущ обозначает ИК с описанием существительного, Прил – капсулу с описанием прилагательного). Так, первое правило описывает ситуацию, когда прилагательное стоит между двумя существительными – данный языковой фрагмент можно рассматривать двояко: либо прилагательное описывает признаки первого существительного, либо второго. Поэтому в результате преобразования в список лексем добавляются две альтернативы. На дальнейших стадиях преобразования списка лексем та альтернатива, которая не будет вписываться в контекст предложения, из списка будет удалена. В результате преобразования описание прилагательного будет удалено из списка лексем, однако на него будут указывать ссылки, добавленные в описание существительных (ИП: {Атрибут=Прил}), таки образом, прилагательное станет атрибутом

объекта, заданного существительным. Второе правило описывает ситуацию, когда между существительными стоят два прилагательных, разделенных союзами «и», «да» или запятой. В этом случае на место данной последовательности описания лексем помещается ИК, в которой содержится ИП с атрибутом «Множество» (это нужно для идентификации того, что в ИК содержатся указатели на описания объектов, которые образуют некое множество). Ссылки на объекты, входящие в множество, помещаются в нагрузки ИП с атрибутом «Объект». После обозначения «::» стоят правила преобразования тех ИК, которые в результате преобразования не войдут в список лексем – в нашем случае было необходимо добавить в описание существительных признаки, которые были заданы прилагательными. В отличие от формальных грамматик, правила склейки применяются к списку лексем строго по порядку их нумерации, иначе преобразования могут дать некорректный результат. Также необходимо соблюдать порядок проходов преобразования списка лексем.

1. $Суц_1 \text{ Прил } Суц_2 \rightarrow (Суц_1*\{Атрибут=Прил\} Суц_2)/(Суц_1 Суц_2*\{Атрибут=Прил\})$
2. $Суц_1 \text{ Прил}_1 (и | да | ,) \text{ Прил}_2 Суц_2 \rightarrow$
 $\{Множество=? \text{ Объект}=Суц_1 \text{ Объект}=Суц_2\}$
3. $Суц_1*\{Атрибут=Прил_1\} :: Суц_2*\{Атрибут=Прил_2\}$
4. $Суц_1 \text{ Прил } Суц_2 \rightarrow Суц_1*\{Атрибут=Прил\}$
- ...

Рис. 3 – ПРАВИЛА СКЛЕЙКИ СУЩЕСТВИТЕЛЬНЫХ И ПРИЛАГАТЕЛЬНЫХ
РУССКОГО ЯЗЫКА

2. Формализация алгоритма синтезе семантического графа

Формально этап преобразования списка лексем можно задать с помощью тройки:

$\{A, L, P\}$, где

A – множество атрибутов ИП

L – множество нагрузок ИП $L = \{Q \cup \text{nil} \cup S \cup \Omega\}$, где Q – множество рациональных чисел, nil – обозначение пустой нагрузки, Ω – множество ссылок на ИК; $S \in \Sigma^*$ – множество цепочек символов, принадлежащих алфавиту Σ ; S является цепочкой, если удовлетворяет следующим правилам:

ε – пустая цепочка символов ($\varepsilon \in \Sigma$), ε – цепочка в алфавите Σ .

если α – цепочка в символов алфавите Σ и a – символ этого алфавита, то αa – цепочка в алфавите Σ (αa – операция конкатенации цепочки α и символа a);

s – цепочка символов в алфавите Σ тогда и только тогда, когда она является таковой в силу (1) и (2).

$P = \{P_1, P_2, \dots, P_n\}$ где P_1, P_2, \dots, P_n – множество правил преобразования списка лексем для каждого прохода преобразования.

Как говорилось ранее, итогом преобразований списка лексем будет семантический граф, узлами которого являются описания объектов и связей между ними. Каждый узел представляет собой ИК. Узлы связаны между собой ссылками, расположенными в нагрузках ИП, входящих в состав ИК (рис. 3). Как правило, в ОА-графе между его узлами используются двунаправленные связи. Это делается для того, чтобы обеспечить его связность, т.к., начав обход графа с любого узла, мы сможем по связям (ссылкам) попасть в любой другой узел (связь в реальной ВС представляет собой не что иное, как указатель на ячейку памяти, где хранится ИК, описывающая какой-либо объект, или часть объекта).

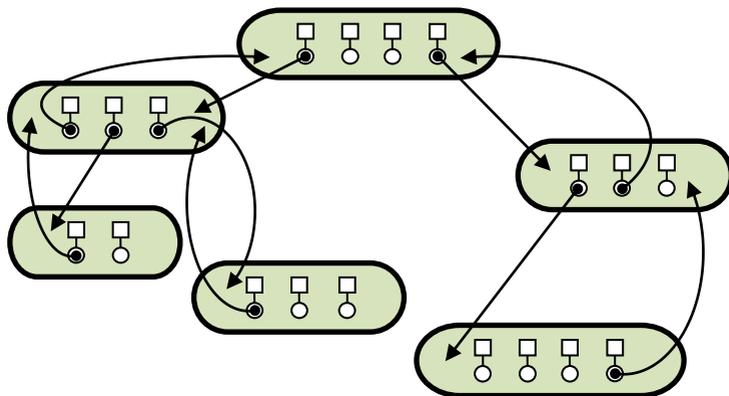


Рис. 3 – Семантический ОА-граф

Семантический ОА-граф, в частности, можно использовать для интернет-поиска, следующим образом. Вначале для всех текстов, в которых будет производиться поиск, строятся ОА-графы. Затем поисковый запрос, введенный пользователем на естественном языке, также преобразуется в семантический ОА-граф. И если граф-запрос будет являться подграфом ОА-графа, описывающего смысл (семантику) одного из текстов, то поисковый запрос считается успешным – и пользователю выдается положительный ответ за его запрос.

Для определения же критерия того, что ОА-граф G_1 является подграфом ОА-графа G_2 следует ввести несколько определений.

Индекс ИК, входящей в ОА-граф: пронумеруем все ИК, входящие в ОА-граф G , подряд, начиная с 1, эти номера будут являться индексами ИК. Будем обозначать ИК с индексом i ($i=1,2,\dots,|G|$, где $|G|$ - количество вершин (ИК), входящих в ОА-граф G) так: G_i .

Адрес в нагрузке ИП. Будем считать, что адресом ИК, хранящемся в нагрузке ИП, является индекс ИК, т.е. $\Omega = \{1, 2, \dots, |G|\}$, где Ω - множество адресов ИК.

Переиндексация ОА-графа – это изменение индексов графа G , т.е. перенумерация всех ИК из ОА-графа G . Переиндексация производится с помощью функции переиндексации f , однозначно отображающая множество первоначальных индексов I на множество новых индексов I' (отображение биективно): $I'=f(I)$. Во время переиндексации изменяются и индексы, расположенные в нагрузках ИП, входящих в ОА-граф.

ИК W_1 равна ИК W_2 ($W_1=W_2$), в том случае, когда любой ИП из W_1 имеется эквивалентная ей ИП в W_2 (ИП эквивалентны, когда у них совпадают атрибут и нагрузка).

ОА-графы G_1 и G_2 изоморфны ($G_1 \approx G_2$), если выполняются следующие условия:

$$|G_1|=|G_2|;$$

Можно сделать такую переиндексацию ОА-дерева, что $\forall i \in 1,2,\dots,|G_1|$, что $G_{1i}=G_{2i}$.

ОА-граф G_1 частично изоморфен G_2 ($G_1 \approx > G_2$), если можно найти такую переиндексацию графов G_1 и G_2 , что:

$G_{1i} \cap G_{2i} = G_{1i}$, где $i=1,\dots,|G_1|$, где \cap - операция пересечения графов.

Задача поиска подграфов в ОА-графе упрощается тем, что вершины графа помечены (меткой является ИК, которая приписана к вершине). Здесь для сокращения числа переборов можно применить разработанный нами метод спектра атрибутов ОА-графа. Метод основан на том, что для ОА-графа текста и ОА-графа запроса подсчитывается количество ИП с определенным атрибутом, все атрибуты индексируются и строится график, в котором по оси абсцисс откладывается индекс атрибута, а по оси ординат – число вхождений ИП с данным атрибутом в ОА-граф (рис. 4). Естественно, сравнение подграфов следует начинать с тех вершин ОА-графа, в которых находятся ИП с наиболее редко встречающимися атрибутами (далее сравнение ОА-графов будет происходить по рекурсии). Если в результате анализа спектров атрибутов обнаружится, что в графе текста нет ИП с определенным атрибутом из графа-запроса,

то из этого сразу можно заключить, что граф запроса не является подграфом графа текста и результат поиска отрицателен.

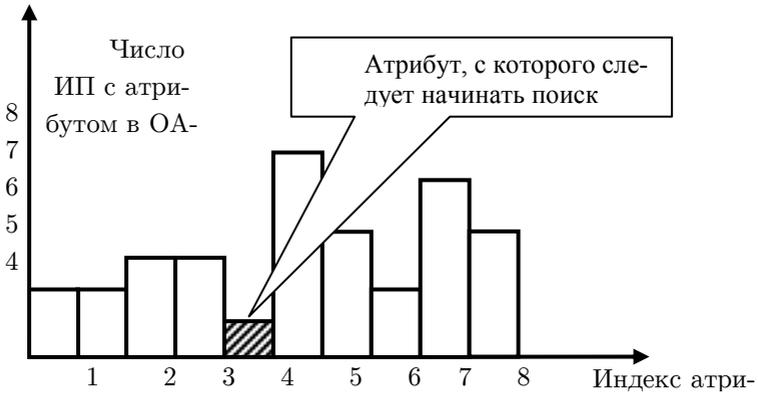


Рис. 4. Спектр атрибутов АО-графа

ОА-система синтаксического анализа обладает весьма полезным свойством параллелизма и реализации на распределенной ВС, что позволит эффективно реализовать ее на суперкомпьютерных ВС. Параллельность вычислений достигается во-первых, благодаря применению парадигмы *dataflow*, обеспечивающей динамическое распараллеливание вычислений. Успешность реализации на распределенной ВС достигается, опять-таки, благодаря применению парадигмы *dataflow*, при которой не происходит жесткой привязки операндов к ячейкам оперативной памяти. Также следует отметить тот факт, что ОА-ВС может быть реализована на стандартных вычислительных элементах (в отличие от существующих универсальных *dataflow-ВС*), что существенно снизит стоимость системы.

Для ОА-системы была разработана формальная модель, описанная в [11]. Модель позволяет проводить моделирование распределенной ОА-ВС.

Для реализации семантического поиска был разработан специальный тип ФУ «Список». С помощью этого класса ФУ можно реализовать правила преобразования списка лексем и произвести синтез семантического графа. На рис. 5 приведен фрагмент ОА-

программы, описывающей семантико-морфологический словарь для английского языка, где

Vocabulary.Set – милликоманда установки ссылки на ОА-писок;

> - обозначение новой записи в списке;

Fork – атрибут вариантов толкования лексем.

Vocabulary.Set=

```
>{Sepator="." Fork=>{Mnemo="." PartOfSpeech=Dot}}
>{Mnemo="is" Fork=>{Mnemo="is" PartOfSpeech=Is}}
>{Mnemo="it" Fork=>{Mnemo="it" PartOfSpeech=It}}
>{Mnemo="there is" Fork=>{Mnemo="there isthere is" ThereIs}}
>{Mnemo="the" Fork=>{Mnemo="the" PartOfSpeech=Article The_Article }}
>{Mnemo="a" Fork=>{Mnemo="a" PartOfSpeech=Article A_Article }}
>{Mnemo="on" Fork=>{Mnemo="on" PartOfSpeech=Conjunction On }}
>{Mnemo="above" Fork=>{Mnemo="above" PartOfSpeech=Conjunction Above }}
>{Mnemo="under" Fork=>{Mnemo="under" PartOfSpeech=Conjunction Under }}
>{Mnemo="opposite" Fork=>{Mnemo="opposite" PartOfSpeech=Conjunction Opposite }}
>{Mnemo="in" Fork=>{Mnemo="in" PartOfSpeech=Conjunction In }}
```

Рис. 5. Фрагмент ОА-программы описания семантико-морфологического словаря английского языка

Заключение

Семантического анализ естественного языка была выбрана нами для реализации на ОА-ВС ввиду того, что он является одной из задач искусственного интеллекта, и данная работа служит научным заделом для создания других систем искусственного интеллекта на базе ОА-подхода.

Итогом проделанной работы явилось создание:

- методики семантического разбора естественного языка;
- нотации для описания алгоритма преобразования списка лексем в семантический граф;

- математической модели, описывающей функционирование ОА-системы;
- программной реализации экспериментальной ВС для семантического анализа естественного языка;
- экспериментальных баз знаний для семантического анализа русского и английского языков.

Предложенная методика семантического анализа показала свою работоспособность. И теперь требуется продолжение работы для расширения экспериментальной базы знаний, чтобы расширить ее лексические и синтаксические возможности.

Список литературы

- [1] Berners-Lee T., J. Hendler and O. Lassila. The Semantic Web // Scientific American, May, 17, 2001.
- [2] Мельчук И.А. Опыт теории лингвистических моделей «СМЫСЛ <->ТЕКСТ» - М.: Школа «Языки русской литературы», 1999.
- [3] Минский М. Фреймы для представления знаний: Пер. с англ.- М.: Энергия, 1979.
- [4] А.В. Тузов Компьютерная семантика русского языка. СПб.: Изд-во СПбГУ, 2004. 400 с.
- [5] Салибебян С.М., Панфилов П.Б. Объектно-атрибутная архитектура – новый подход к созданию объектных систем // *Информационные технологии*.—2012.—№. 2 (186).—С.8-13.
- [6] Салибебян С.М., Панфилов П.Б. Объектно-атрибутная архитектура – новый подход к созданию объектных систем // *Информационные технологии*.—2012.—№. 2 (186).—С.8-13.
- [7] Салибебян С.М., Панфилов П.Б. ОА-архитектура – новый подход к созданию объектных систем // Объектные системы - 2011: материалы III Международной научно-практической конференции (Ростов-на-Дону 10-12 мая 2011 г.) / Под общ. ред. П.П. Олейника. - Ростов-на-Дону, 2011. - С. 73-79 URL: http://objectsystems.ru/files/Object_Systems_2011_Proceedings.pdf
- [8] Gabriel, R. 2002. Objects Have Failed: Notes for a Debate. (retrieved 17 May 2009). <http://www.dreamsongs.com/Files/ObjectsHaveFailed.pdf>

- [9] Салибекян С.М., Панфилов П.Б. Анализ языка с помощью вычислительной системы объектно-атрибутивной архитектуры. // Объектные системы - 2012: материал VI Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2012 г.) / Под общ. ред. П.П. Олейника. - Ростов-на-Дону: ШИ ЮРГТУ (НПИ), 2012. - С. 31-37 URL: http://objectsystems.ru/files/2012/Object_Systems_2012_Proceedings.pdf
- [10] Ахо, Ульман Теория Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. - М.: Мир, 1978
- [11] Салибекян С.М., Панфилов П.Б. Формализация dataflow-модели вычислительного процесса. // Объектные системы - 2013: материал V Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2013 г.) / Под общ. ред. П.П. Олейника. - Ростов-на-Дону: ШИ ЮРГТУ (НПИ), 2013. - С. 87-93 URL: http://objectsystems.ru/files/2012/Object_Systems_2013_Proceedings.pdf
- [12] С.М. Салибекян, П.Б. Панфилов Моделирование суперкомпьютерной вычислительной системы объектно-атрибутивной архитектуры с управлением потоком данных // Информационные технологии и вычислительные системы. №1, 2013 – стр. 3-10

Об авторах:



Сергей Михайлович Салибекян

(Доцент МИЭМ НИУ ВШЭ, к.т.н., Разработчик объектно-атрибутивной архитектуры вычислительной системы, создатель объектно-атрибутивного подхода анализа естественного языка. Область научных интересов: параллельные вычисления, системы искусственного интеллекта, семантическая обработка естественного языка.

e-mail:

ssalibekyan@hse.ru

Пётр Борисович Панфилов

Доцент Минэкономразвития РФ, к.т.н., Область научных интересов: параллельные вычисления, моделирование, системы виртуальной реальности.

e-mail:

panfilow@miem.edu.ru

Образец ссылки на публикацию:

С.М. Салибекян, П.Б. Панфилов. Семантическая обработка естественного языка на распределенной объектно-атрибутивной вычислительной системе // Программные системы: теория и приложения: электрон. научн. журн. 2013. Т. 4, № 3(17), с. ??-??.

URL:

<http://psta.psir.ru/read/???>

S.M. Salibekyan, P.B. Panfolow. Natyral language processing with distributed object-attributed computer.

ABSTRACT. In the paper it is description of object-attribute (OA) approach to computation process organization for natural language processing. The approach involves the method of semantic graph synthesis, formal model of computation process, the semantic search method, the architecture of computer for semantic search. The OA-approach realizes the dataflow paradigm of computation process which provides a several benefits such: concurrent computing, distributing of computation, height level of abstraction of program and data structures, self-learning of computer etc.

Key Words and Phrases: dataflow, formal model, automata, Kahn process networks, complex data structures, distributed computer, natural language processing.