Graph of TAPAZ-2 Semantic Classifier

Aliaksandr Hardzei
Minsk State Linguistic University
Minsk, Belarus
alieks2001@yahoo.com

Anna Udovichenko
LLC "SK Hynix Memory Solutions Eastern Europe"
Minsk, Belarus
anuta.udovichenko@gmail.com

***Abstract*—The article discusses the algorithm for finding the entrance to an arbitrary subject domain based on the TAPAZ-2 Semantic Classifier and the graph thus obtained. The formulas for the exact number of vertices in the graph are derived depending on the number of elements in the Paradigm of Actions and the restrictions imposed while constructing the graph. Various ways are proposed to reduce the number of vertices in order to adapt the power of graph combinatorics for automatic processing with modern technical means.**

***Keywords*—paradigm of actions, semantic classifier, semantic ontology, vertex of the graph, degree of the vertex, artificial intelligence**

I. Introduction

TAPAZ-2 is a tool for generating a model of the world in a form suitable for Natural Language Processing in systems of Artificial Intelligence. The Intellectual Knowledge Base (IKB) built in a computer combines the Semantic Classifier – a final ordered (vector) set of semantic primitives (actions and roles of individs) and the Semantic Ontology – an algorithm for generating new sense units based on the original set of primitives, presented in the form of the Semantic Classifier Graph [1].

An expert (intelligent) search system based on the Semantic Classifier may consist of an intelligent search engine that selects and reviews content on a given topic from the Internet, and a dialog user interface that allows the system to process user requests and transform them in the canonized text corresponding to the machine-readable Model of the World, and the user will confirm whether this conversion was performed correctly, and if not, then offer his own decoding through the Semantic Classifier.

II. Dimension of the Semantic Classifier Graph

Let *n* be the number of actions in the initial generating list (the number of the first degree vertices) of the Paradigm of Actions. *The degree of the vertex S* will be called the number of vertices of the initial generating list directly or indirectly participating in the generation of this vertex. There can be several ways to generate an intermediate vertex from the same set of vertices of the first degree. The subgraph that specifies one of the possible ways to generate a vertex is a binary tree, since exactly two ancestor vertices participate in the generation of each child vertex. Thus, the number of ways to generate an intermediate vertex of degree *k* is expressed by the number of binary trees with the number of leaves equal to *k* (the Catalan number) [2]. For *k* = 4, the number of such trees is 5, all possible configurations are shown in Fig. 1.

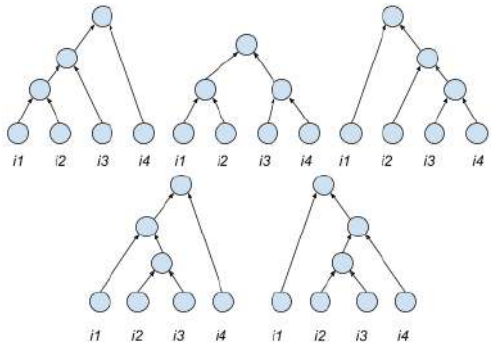

Figure 1. All possible configurations of binary trees for k = 4

The number of *k* degree vertices is determined by the formula (1):

$$N_k = A_n^k \times C_k \tag{1}$$

where A_n^k – is the number of allocations from n vertices with respect to k [3] (that is, the set of all the vertices of the initial list participating in generating a vertex) and C_k – is the Catalan number, which is calculated by the formula (2) or (3):

$$C_{n+1} = \frac{(2n)!}{n!(n+1)} \tag{2}$$

$$C_n = \frac{(2(n-1))!}{(n-1)!n!} \tag{3}$$

where n is the number of leaves of a binary tree [2].

The maximum degree of a vertex in the Semantic Classifier Graph is equal to the number of vertices in the initial set (all vertices of the initial set participate in generating a vertex of degree n). Thus, in order to obtain the number of vertices in the Graph, it is necessary to sum up the number of vertices of all degrees from 1 to n:

$$S_n = \sum_{k=1}^{n} A_n^k \times C_k = \sum_{k=1}^{n} \frac{n!}{(n-k)!} \times \frac{(2(k-1))!}{(k-1)k!} \quad (4)$$

Python program code, calculating the function $S(n)$ by the formula (4):

```
from math import factorial as fact

def S(n):
  sum = 0
  for k in xrange(1, n + 1):
    A = fact(n) / fact(n - k)
    C = fact(2 * (k - 1)) / (fact(k - 1) * fact(k))
    sum += A * C
  return sum


print "%e" % S(112)
```

The Paradigm of Actions, which is used to build the Semantic Classifier Graph, contains 112 elements. Substituting this number into formula (4) as n, we get:

$$S_n \approx 8,2 \times 10^{245}$$

At present, there are no devices whose computational capacities are capable of processing and storing a similar amount of information, therefore it is necessary to select the most significant part of the Graph in order to sequentially generate, process and store this graph using computer technology.

III. Ways to reduce vertices in the Semantic Classifier Graph

Now we can investigate the procedure for generating new actions similar to that described earlier, but in which the active action of degree k will be refined only by first-degree actions, i.e., a vertex of degree k and a vertex of degree 1 will always be involved in generating a vertex of degree k + *1*. Such a graph is called a *graph of the standard form*. Obviously, it will contain fewer vertices than the Semantic Classifier Graph, since the graph of the standard form is a subgraph of the Semantic Classifier Graph. The subgraphs defining the method for generating vertices of degree k have the following form (Fig. 2).

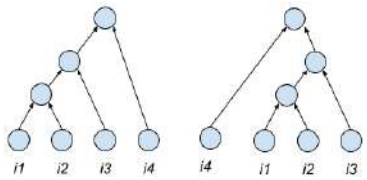

Figure 2. Subgraphs defining the generation of vertices in a graph of the standard form

At the same time, the order in which the clarification occurs is important: "warm by linking" is not the same as "link by warming" [4]. We define the dimension of the graph obtained in this way using the dynamic programming method [5] – the number of vertices of degree k will be expressed in terms of the number of vertices of degree $k-1$.

1) The number of vertices of degree 1 is n:
$$N_1 = n \quad (5)$$
2) The number of vertices of degree 2 is equal to the number of ways to form pairs, taking into account the order:
$$N_2 = n \times (n-1) \quad (6)$$
3) Fix a vertex of degree $k-1$ (for $k > 2$) and we will successively generate new vertices from this vertex and all the remaining vertices of the first degree that are not active in its generation. The number of ways to generate a vertex of degree k from one vertex of degree $k-1$, taking into account the order, is equal to:
$$n_k = 2 \times (n - (k-1)) = 2 \times (n-k+1) \quad (7)$$
where $(n - k + 1)$ is the number of vertices of the first degree that are not involved in the generation of the fixed vertex of degree $k-1$.
Then the recurrence formula for the total number of ways to generate a vertex of degree k is as follows:
$$N_k = 2 \times (n-k+1) \times N_{k-1} \quad (8)$$
4) Sum up the number of vertices of all degrees from 1 to n:
$$S_n = \sum_{k-1}^{n} N_k \quad (9)$$

Python program code:

```
def S(n):
  f = [0] * (n + 1)
  result = [0] * (n + 1)
  f[1] = n
  result[1] = n
  for k in xrange(2, n + 1):
    f[k] = f[k-1] * (n - k + 1)
    if k > 2:
      f[k] *= 2
    result[k] = result[k-1] + f[k]
  return result[n]


print "%e" % S(112)
```

As a result of calculating the dimension of the graph for $n = 112$, we obtain:

$$S_{112} \approx 4,2 \times 10^{215}$$

Despite the fact that the number of vertices in such a graph is less than in a full graph without restrictions, it is still extremely large and cannot be processed using modern computational tools.

In order to further reduce the dimension of the Semantic Classifier Graph, we impose an additional constraint: we fix one active vertex of the first degree (active action) and will consistently refine it with the rest (passive) actions. Then the *subgraph of new actions with one active vertex* will acquire the following properties:

- for any vertex of degree k, the first generating ancestor of degree $k-1$ is a refined new action of the active vertex, in other words, if you depict a method of generating a vertex of degree k as a subtree, then its first leaf will always be the active vertex;
- any vertex in the graph can be uniquely defined by an ordered list of first-degree vertices participating in its generation, with the first vertex always being the active vertex and the degree of the vertex k being the length of this list.

Thus, the number of vertices in the graph will be equal to the number of ordered subsets of the first degree vertices' set, excluding the active one (since it is always fixed). This value is calculated by the recurrence formula [6]:

$$T_n = 1 + n \times T_{n-1} \tag{10}$$

Then the total number of vertices in the graph (taking into account all the vertices of the first degree) is:

$$S_n = n - 1 + T_{n-1} \tag{11}$$

Python program code:

```
def S(n):
    f = [0] * (n + 1)
    f[1] = 2
    for k in xrange(2, n + 1):
        f[k] = 1 + k * f[k - 1]
    return f[n - 1] + n - 1

print "%e" % S(112)
```

As a result, the dimension of the graph for $n = 112$ is:

$$S_{112} \approx 4,8 \times 10^{180}$$

If we build a separate graph, alternately choosing each of the 112 actions active, we get 112 graphs, containing in total the following number of vertices:

$$\sum_{n=1}^{112} S_n = 5,4 \times 10^{182}$$

which is much less than the number of vertices in the Semantic Classifier Graph of the same 112 actions. Although some of the possible meanings are lost, but such a construction can significantly reduce the number of vertices in the graph.

The Paradigm of Actions contains 56 physical and 56 information elements [7]. Let us analyze *two separate graphs constructed on these two sets*. Each of them will contain the following number of vertices:

Table I
THE DIMENSION OF SEPARATE GRAPHS OF PHYSICAL OR INFORMATION ACTIONS

	Semantic Classifier Graph	Graph of the Standard Form	Graph with One Active Vertex
S_{56}	$1,6 \times 10^{105}$	$2,1 \times 10^{91}$	$3,4 \times 10^{73}$
$2 \times S_{56}$	$3,2 \times 10^{105}$	$4,2 \times 10^{91}$	$3,4 \times 10^{73}$

Despite the fact that the number of vertices is still extremely high, the total number of vertices for graphs built on half of the set of actions is significantly less than the number of vertices for the Semantic Classifier Graph. Thus, due to partial losses of some variants of meanings, it is possible to significantly reduce the amount of information to be processed.

Now we will try to reduce the number of vertices in the Semantic Classifier Graph from another angle: instead of the restriction on the number of generated vertices of k degree, we introduce a restriction on the degree of generated vertices. We will find the k degree at which the depth of detailing of the new actions is sufficient to achieve the required semantic power, but the number of vertices in the graph remains within the limits allowing it to be processed and stored by modern computing means.

Table II
THE DIMENSION OF GRAPHS WITH LIMITED DEPTH OF DETAILING

Depth of detailing k_{max}	Semantic Classifier Graph		Graph of the Standard Form	
	S_n	$S_n \times 31$	S_n	$S_n \times 31$
2	12544	388864	12544	388864
3	2747584	85175104	2747584	85175104
4	748045984	23189425504	598986304	18568575424

Let us investigate the number of vertices in the Semantic Classifier Graph and in the graph constructed by the method of sequential detail (Table 2). Already at the level of $k_{max} = 3$ detailing, the number of generated vertices, multiplied by the number of k = 31 roles of individuals, exceeds 85 million (compare to the Dictionary of the modern Russian literary language in 17 volumes which contains 120,480 words, the declared volume of the Large Academic Dictionary of the Russian language which consists of 150,000 words, and the available electronic resources of the Institute of Linguistic Studies of the Russian Academy of Sciences for 1.4 billion of word usage which contain about 5 million Russian words of the XVIII – XXI centuries) [8].

At the level of $k_{max} = 4$ detailing, the number of vertices of the Semantic Classifier Graph, multiplied by the number of roles of individuals, exceeds 23 billion.

Since the generation of separate graphs for physical and informational actions significantly reduces the number of values obtained, this method of calculation will increase the possible depth of detailing. These results are presented in tab. 3, similar to tab. 2, only for a graph built on a set of 56 vertices of the first degree:

Table III
THE DIMENSION OF THE GRAPHS OF PHYSICAL AND INFORMATION PROCESSES WITH LIMITED DEPTH OF DETAILING

Depth of detailing k_{max}	Semantic Classifier Graph		Graph of the Standard Form	
	S_n	$S_n \times 31$	S_n	$S_n \times 31$
2	3136	97216	3136	97216
3	335776	10409056	335776	10409056
4	44410576	1376727856	35595616	1103464096
5	6461701456	200312745136	3702618976	114781188256

At the depth of $k_{max} = 4$ detailing, the number of vertices in the Semantic Classifier Graph, multiplied by the number of roles of individuals, does not exceed two billion. Thus, we get an additional level of detailing that improves the accuracy of calculating the subject domains.

IV. CONCLUSION

A search procedure of generating new actions through the set of first degree actions may be represented as a graph, a matrix, or a vector system. In the graph the relations of active first degree action and clarifying first degree action are represented as follows (Fig. 3):

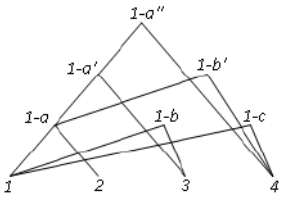

Figure 3. TAPAZ-2 Semantic Classifier Graph

where: *1* – active first degree action; *2,3,4* – clarifying first degree actions; *1-a* , *1-b*, *1-c* – derivative second degree actions with *1-a* as the active derivative second degree action; *1-a'* and *1-b'* – derivative third degree actions with *1-a'* as the active derivative third degree action; *1-a''* – the active derivative fourth degree action [4].

The total number of vertices in the TAPAZ-2 Semantic Classifier Graph is expressed by a number of 10^{245}, which, of course, is too large not only for manual, but also automatic processing. Reducing the number of vertices is achieved in three ways:

- restrictions on the generation of vertices;
- division of the complete graph into two separate subgraphs;
- limiting the depth of actions' detailing.

The combination of all three methods allows you to adjust the number of processed vertices, while, however, some of the meanings are lost. The question of limiting the depth of detail without significant loss of meaning remains open for further research.

The second version of the Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2) is one of the possible models for calculating semantics. Despite the fact that the model does not have analogues in calculating of the subject domains, it does not claim to be exclusive. Linguistic semantics is versatile and allows different ways of formalizing.

However, all methods, like Euclidean and non-Euclidean geometry, should be consistent and effective in its problem solving, and those who argue with that, as Reichenbach aptly said, only "confuse a rigor of the method with a limitation of a goal" [9].

REFERENCES

[1] A. Hardzei, Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2, Minsk, 2017, 50p.
[2] R. P. Stanley, Catalan addendum. Available at: http://www-math.mit.edu/ rstan/ec/catadd.pdf.
[3] R. P. Stanley, Enumerative combinatorics. Cambridge, 2012, vol.1, 585p.
[4] A. Hardzei, Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2. Minsk, 2017, 35p.
[5] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, Cl. Stein, Introduction to Algorithms, 3d ed. Cambridge, Massachusetts, London, 2009, 1292p.
[6] B. Schroeder, Ordered Sets: An Introduction, New York, 2003, 391p.
[7] A. Hardzei, Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2, Minsk, 2017, 34p.
[8] L. E. Kruglikova, El gran diccionario académico de lalengua rusa» como continuador de las tradiciones de la lexicografía académica rusa. Cuadernos de Rusística Española, 2012, No.8.
[9] M. Reichenbach, Philosophie der Raum-Zeit-Lehre, De Gruyter, 1928, 386 p. English translation: The Philosophy of Space and Time, New York, Dover Publications, 1958, 295p.

ГРАФ СЕМАНТИЧЕСКОГО КЛАССИФИКАТОРА ТАПАЗ-2

А. Н. Гордей, А. М. Удовиченко

В статье рассматривается алгоритм поиска входа в произвольную предметную область на базе семантического классификатора ТАПАЗ-2 и получаемый таким образом граф. Выводятся формулы точного количества вершин в графе в зависимости от количества элементов в таблице макропроцессов и ограничений, накладываемых при построении графа. Предлагаются различные способы сокращения количества вершин графа в целях адаптации мощности его комбинаторики для автоматической обработки современными техническими средствами.

Ключевые слова: таблица макропроцессов, семантический классификатор, семантическая онтология, вершина графа, степень вершины, искусственный интеллект.

Received 10.01.19