UDC 004.822:519.15 https://doi.org/10.15407/pp2021.02.003

J.V. Rogushina, A.Ya. Gladun

DEVELOPMENT OF DOMAIN THESAURUS AS A SET OF ONTOLOGY CONCEPTS WITH USE OF SEMANTIC SIMILARITY AND ELEMENTS OF COMBINATORIAL OPTIMIZATION

We consider use of ontological background knowledge in intelligent information systems and analyze directions of their reduction in compliance with specifics of particular user task. Such reduction is aimed at simplification of knowledge processing without loss of significant information. We propose methods of generation of task thesauri based on domain ontology that contain such subset of ontological concepts and relations that can be used in task solving. Combinatorial optimization is used for minimization of task thesaurus. In this approach, semantic similarity estimates are used for determination of concept significance for user task. Some practical examples of optimized thesauri application for semantic retrieval and competence analysis demonstrate efficiency of proposed approach.

Keywords: domain ontology, task thesaurus, semantic similarity, combinatorial optimization

Introduction

A lot of intelligent applications need in background knowledge about domain. Modeling of domain is often realized by ontologies. But processing of unconditioned ontologies is a complex and hard problem. For many tasks it is reasonable and acceptable to use various simplified domain models, for example, thesaurus of domain that is based on domain ontology but contains the lesser part of domain terms and does not contain relations between them.

Every concept of domain ontology is characterized by properties, relations with other concepts and individuals and other characteristics. We propose to define some initial subset of ontology concepts and then define such other concepts of this ontology that are semantically similar to concepts from initial subset in context of user task. This extended set of terms can be considered as a domain thesaurus and be used for user task solving. We propose to use combinatorial optimization methods (particularly the knapsack task) for development of the optimized domain thesaurus that has minimum quantity of concepts but covers all task-specific needs.

Thesaurus and ontologies as means of domain knowledge representation

By definition, “thesaurus” is the study of term usage in given domains associated to a human activity. A *term* is a sequence of words used in a given domain and which makes sense in this domain. In ontological analysis term corresponds to some concept of ontology. Therefore, thesaurus can be used for domain description.

Domain thesaurus is a sort of terminological base: it is a collection of terms with some set of relations among them. Now many thesauri for medical domain, mathematics, computer science, etc. domains are developed. They are used for unification of terminology, for common interpretation of domain knowledge, for integration of independently developed intelligent software and knowledge bases etc. Thesauri can be used as a bridge from a terminological base to document indexing and for normalization of indexing terms.

Elements of thesaurus can be extracted from natural language (NL) text by means of linguistic analysis. Manual thesaurus building is a hard task and needs much time. But in this way one can guarantee a good quality of the collected terms. Automatic thesaurus building needs less human workforce but the quality is not guaranteed. It relies on the content and structuring of document sources, and also on the methods of NL processing. Another problem deals with selection of NL texts pertinent to analyzed domain.

Domain ontologies. We consider that any human activity that consists of solving different tasks is a characteristic of activity domain. Task solving needs special knowledge, the same for all the tasks that can be represented verbally. Therefore we can speak about special vocabulary of every domain that is used for specification of tasks and their solutions in this domain. A *domain* is considered as a set of the tasks that are solved by specialists of this domain. In process of the task solving all solving subjects (persons, software agents, etc.) use a finite set of objects and a finite set of relations among them. These sets are formed as a result of agreements about understanding among members of the domain community. In the field of the distributed knowledge management the term "ontology" is used for explicit conceptualization of some domain [1]. The focus of ontologies is not only the domain terminology, but also the inherent ontological structure. It shows which objects exist in the application domain, how they can be organized into classes, called concepts, and how these classes are defined and related.

Every domain has phenomena that people allocate as conceptual or physical objects, connections and situations. With the help of various language mechanisms such phenomena contacts to the certain descriptors (for example, names, noun phrases).

At present the usefulness of domain ontologies is generally recognized and causes their wide use. But the elements and the structure of domain ontologies are not defined uniformly in different applications.

Now three main approaches to define domain ontology are used in intelligent information systems (IIS). They are connected with the ways of ontological analysis application and deal with different sciences.

The first one – *humanitarian* approach – suggests definitions in terms understood intuitively but cannot be used for solving of technical problems.

The second one – computer approach – is based on some computer languages (such as OWL, DAML+OIL) for representation of domain ontology and applied software. It realizes the processing of knowledge represented in these languages. Such approach is the most useful for development of knowledge bases (KBs) for IIS.

The third one – mathematical approach – defines the domain ontologies in mathematical terms or by mathematical constructions. This approach is too complex for applied IIS and is used for finiteness of ontology processing algorithm and estimation of their execution time.

Usually at first step of domain ontology building the humanitarian approach is used, then the mathematical model of ontology is constructed, and at last its software realization is developed.

Till now no generally accepted universal definition of domain ontology has been suggested. In [2] different definitions are analyzed. On the meaningful level domain ontology will be understood as a set of agreements (domain term definitions, their commentary, statements restricting a possible meaning of these terms, and also a commentary of these statements). Domain ontology is:

- the part of domain knowledge that is not to be changed;
- the part of domain knowledge that restricts the meanings of domain terms;
- a set of agreements about the domain;
- an external approximation represented explicitly of a conceptualization given implicitly as a subset of the set of all the situations that can be represented.

All these meanings of the notion of domain ontology supplement each other.

For the successful development of IIS it is necessary to present user knowledge about domain of her/his interests in some form suitable for computer processing. The specifications of high-level domain are formed by integration of the domain structures of low-level domains. It is important to achieve an interoperability of domain knowledge representation. Ontological approach is an appropriate tool for solution of this task. Ontology is an agreement about common use of concepts that contains means for representing the subject knowledge and agreements on methods of reasons. It can be considered as the certain description and reflection of the world in some specific spheres of interest. Ontology in the most general representation consists of: 1) domain terms; 2) relations between these terms that define links of domain classes and individuals; 3) rules of their use and interoperation that limit meanings of terms in the context of particular domain [3]. The formal model of domain ontology O is an ordered triple $O = < X, R, F >$, where X – finite set of domain concepts; R – finite set of the relations between concepts of the given subject domain; F – finite set of interpretation functions of given concepts and relations.

Domain ontology is a special kind of knowledge base that contains semantic information about some domain in interoperable and formalized representation. It is a set of definitions in some formal language of declarative knowledge fragment focused on common repeated use by the various applications and tasks.

Ontological commitments are the agreements aimed at coordination and consistent use of the common dictionary. The agents (human beings or software agents) that jointly use the dictionary do not feel necessity of common knowledge base: one agent can know something that other ones don't know. Agent that handle the ontology is not required the answers to all questions that can be formulated with the help of the common dictionary.

Every domain with the certain subject of research has it's own terminology, original dictionary used for discussion of typical objects and processes of this domain. The library, for example, involves the dictionary relating to the books, references, bibliographies, magazines etc. Thus, pattern of domain is discovered by its dictionary (the set of NL words that are used in this domain). Clearly, however, that the specificity of domain is shown not only in the appropriate dictionary. Besides, it is necessary:

- to provide strict definitions of grammar managing of combining the dictionary terms into the statements,
- to clear logic connections between such statements.

Only when this additional information is accessible, it is possible to understand both nature of domain objects and important relations established between them.

Task thesauri. For description of some domain is always used the certain set of terms X. Each of terms designates or describes some concept or idea from this domain. Aggregate of terms that describes this domain with pointing the semantic relations between terms is a thesaurus. Such relations in thesaurus always specify the presence of semantic connection between terms. If user needs to solve some task then he/she selects some subset of X dealt with this task. This subset can be considered as a task thesaurus.

The term "thesaurus" for the first time was used still in XIII century by B.Datiny as the name of the encyclopedia. In translation from Greek "thesaurus" means treasure, riches. The thesaurus is the complete systematized data set about some field of knowledge allowing the human or the computer to orient in it. Intelligent information technologies (IIT) consider thesaurus as a dictionary that contains descriptors of the certain field of knowledge with ordering of their hierarchical and correlative relations. These descriptors are represented into thesaurus in alphabetic order but they also are grouped semantically.

Usually thesauri developed for IIS do not contain definitions of terms. Some thesauri can group terms in X (monolingual, bilingual or multilingual) in a hierarchical taxonomy of concepts, others present them in alphabetical order or by a sphere of science.

Task thesaurus is a collection of the domain terms with indication of the semantic relations between them deal with some particular task. Formal model of thesaurus Th is a pair $Th = < T_{Th}, R_{Th} >$, where T_{Th} is a finite subset of the domain terms, $T_{Th} \subseteq X$, where R_{Th} is a finite subset of the relations between these domain terms, $R_{Th} \subseteq R$. Task thesaurus can be considered as a special case of domain ontology.

The expressiveness of the associative relationships in a thesaurus vary and can be as simple as "related to term" as in term A is related to term B [4].

Thesaurus databases, created by international standards, are generally arranged hierarchically by themes and topics.

Formal definition of task thesaurus is a list of terms (single-word or multi-word) important to user task in fixed domain enlarged by the set of related terms for each term from the list.

The structure of thesauri is controlled by international standards that are among the most influential ever developed for the library and information field. The main three standards define the relations to be used between terms in monolingual thesauri (ISO 2788:1986), the additional relations for multilingual thesauri (ISO 5964:1985), and methods for examining documents, determining their subjects, and selecting index terms (ISO 5963:1985). ISO 2788 contains separate sections covering indexing terms, compound terms, basic relationships in a thesaurus, display of terms and their relationships, and management aspects of thesaurus construction. The general principles in ISO 2788 are considered language- and culture-independent. As a result, ISO 5964:1985 refers to ISO 2788 and uses it as a point of departure for dealing with the specific requirements that emerge when a single thesaurus attempts to express "conceptual equivalencies" among terms selected from more than one natural language [5].

Until recently term "thesaurus" was used as a synonym of term "ontology", however now in IISs with the help of the thesauri frequently describe domain lexicon in a semantic projection, and ontologies apply for semantics and pragmatists modeling in a projection to representation language [6]. The models either of ontologies or of thesauruses include (as the basic concepts) the terms and connections between these terms.

Spheres of task thesauri use in IIS. Ontologies that differ by expressiveness, volume, language etc. are widely used in IIS as a source of background knowledge about domain, users and their believes about information processing and representing. Task specifics defines the restrictions on used ontologies. Many researchers differentiate ontologies depending on the complexity of relationships provided by them into "light weight ontologies" and "heavyweight ontologies" [7].

Examples of lightweight ontologies are controlled vocabularies, thesauri and informal taxonomies. Controlled vocabularies are represented by list of domain terms. Taxonomies add hierarchical relations (i.e. "*is-a*" relation) between terms of controlled vocabularies, and therefore we can estimate some semantic similarity of terms by number of steps between them in this hierarchy. Thesauri add additional information to the terms in taxonomies, including preferred names, synonyms and relations to other terms (e.g. "see also").

A lot of thesauri are created for various spheres of human activities – medical domain, mathematics, computer science, etc. Thesaurus can be created for single information resource (IR), natural language (NL) document or the set of documents. It can contain all words of source or some subset of them (for example, nouns, words of reference vocabulary or concepts of domain ontology). Thesaurus terms can be extracted from text by means of linguistic analysis or manually.

Now thesauri are widely used in semantic search [8], e-learning [9], competence analysis [10], and personification of information processing in IIS. User models on base of ontologies can support "personal ontology view" (POV) – ontological representation of individual beliefs about domain conceptualization [11].

Heavyweight ontologies contain not only hierarchical term relation but also domain-specific ones with various sets of characteristics (e.g. transitive or reflexive) that can be used for logical reasoning. Processing of heavyweight ontologies demands more time and calculation facilities but such ontologies are much more expressive as compared with lightweight ontologies. Therefore we try to propose methods that are aimed at automated generation of lightweight ontologies (such as task thesauri) on base of heavyweight ontologies according to needs of particular user task.

Constructing of task thesauri. Construction of task thesaurus includes such main steps (Fig. 1):

1. Definition of user task. At first user has to define particular task that is needed in background knowledge and to fix description of this task (by natural language, in some structured form or by the set of keywords).

2. Selection of domain ontology. Thesauri construction is based on use of domain ontologies of the appropriate areas. Therefore user needs an appropriate ontology $O =< X, R, F >$ that can be retrieved from some ontology repository with the help of matching with user interests description or constructed (manually or semi-automatically) specially for this task.

3. Generation of the set of thesaurus concepts. The main part of task thesauri $Th =< T_{Th}, R_{Th} >$ construction consists in building of set $T_{Th} \subseteq X$ where every $t_i \in T_{Th}$

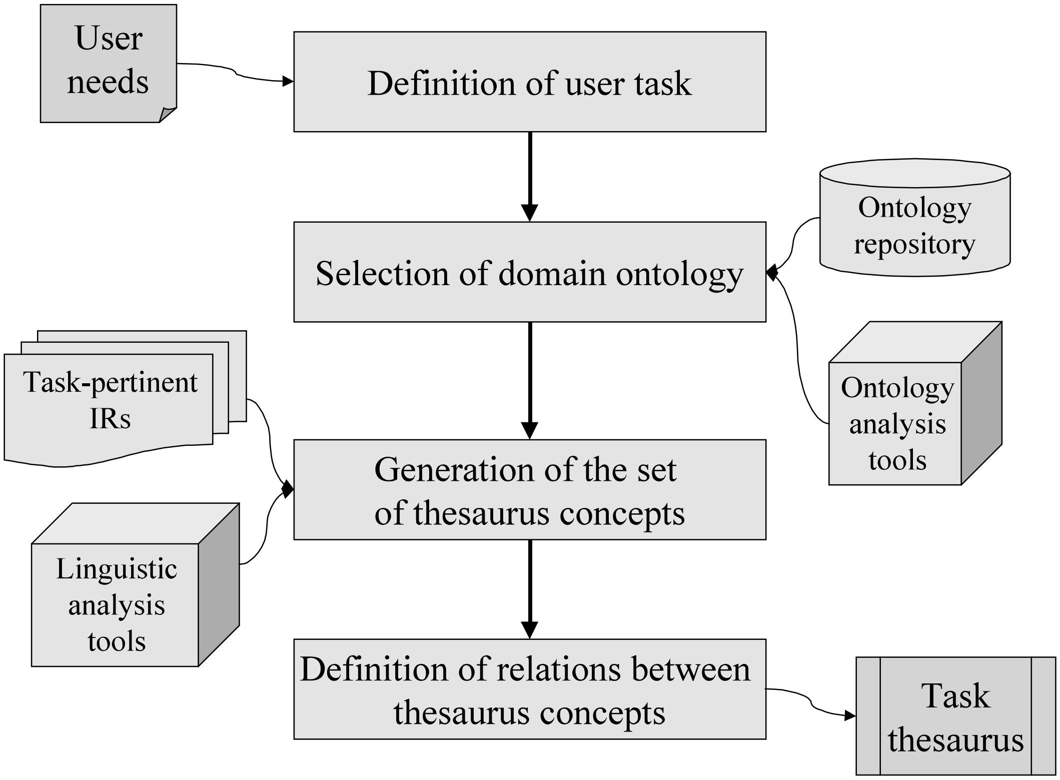

User needs
Definition of user task
Ontology repository
Selection of domain ontology
Ontology analysis tools
Task-pertinent IRs
Generation of the set of thesaurus concepts
Linguistic analysis tools
Definition of relations between thesaurus concepts
Task thesaurus

Fig. 1. Main steps and sources of task thesaurus construction

has some semantic matches with some element $w_i \in W_{ut}$ of user task description W_{ut} that $\forall t_i \in T_{Th}, i = 1, n, \exists w_i \in W_{ut}$. This set can be enriched by processing of pertinent IRs (user should independently select the set of IR that he/she considers relevant to domain of his/her interests). Every IR is described by not empty set of the textual documents connected with this IR - text of content, metadata, results of indexing etc. Task thesaurus is formed as a result of the automated analysis of these documents (the user actions are reduced to constructing of semantic bunches - by linking of each word of the formed thesaurus with some term of domain ontology. Algorithm of NL processing for thesaurus building is proposed in [12].

4. Definition of relations between thesaurus concepts. This step provides identification of hierarchical ("class-subclass", "class-individual", "is-a") and synonymic ("see also") relations from $R_{Th} \subseteq R$ between concepts from $T_{Th} \subseteq X$. These relations can be imported from domain ontology, be extracted from pertinent IRs or be defined manually by user.

In general, task thesaurus can be extended by thesauri of other pertinent IRs and user can edit it manually. This approach is used if task definition is too small and insufficient for retrieval of necessary data but user has some additional information about task (Fig. 2).

This approach provides generation of task thesaurus if user has any information about task, and this thesaurus contains all domain concepts important for task. But such thesaurus can contain a lot of concepts that are not used in task solving. It increase the volume of thesaurus and causes complications of task solving by IIS

Statement of the problem

For the purpose to reduce the time of task solving and complexity of analysis we propose to construct task thesaurus Th available for solving of user task that contains a minimum subset of terms of domain ontology X. $\left|T_{Th_{\min}}\right| \le \left|T_{Th_j}\right|, j = \overline{1, m}$ where Th_j are all possible task thesauri that contain all information from ontology that can be used for task solving and $|A|$ is a number of elements of the set A (sufficiency of information is defined by user and can be estimated by analysis of IIS results).

Development of such minimized thesaurus $Th_{\min}$ can be based on semantic similarity between domain concepts. They deal

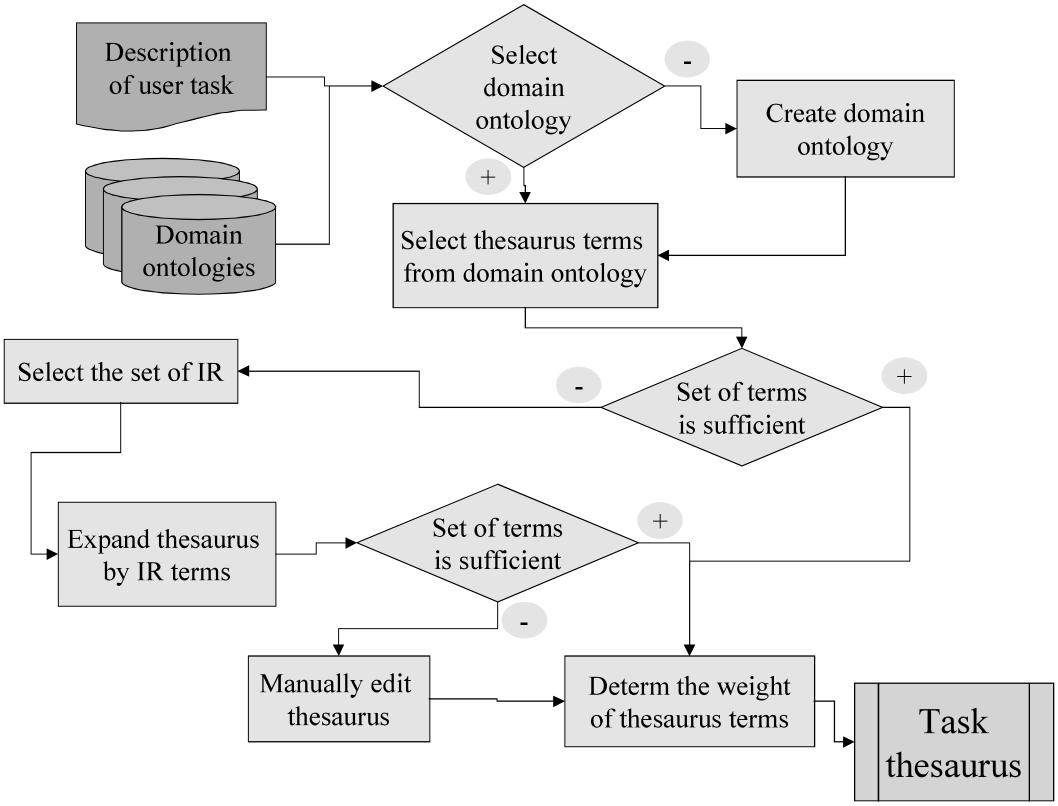

Description of user task
Domain ontologies
Select domain ontology
-
Create domain ontology
+
Select thesaurus terms from domain ontology
Select the set of IR
-
Set of terms is sufficient
+
Expand thesaurus by IR terms
Set of terms is sufficient
+
-
Manually edit thesaurus
Determ the weight of thesaurus terms
Task thesaurus

Fig. 2. Generalized algorithm of task thesaurus generation

with user task concepts and on use of combinatorial optimization methods. Such set can be constructed by combinatorial methods as a comparison of all possible subsets.

Combinatorial methods and knapsack task

Similarity estimates are used in recognition tasks for matching various sets of concept properties; individuals and relations with reference definition of used demands. The accuracy of the matching result depends on adequately selected similarity measures.

Combinatorial optimization uses modeling of processed data with finite numerical sequences. The result is evaluated by correlation approach where an expression that define a total product of the values of these sequences establishes the dependence of input information on the combinatorial configuration (objective function argument) [13].

Mathematical formulation of the general problem of combinatorial optimization. Combinatorial optimization problems are usually defined on one or more basic sets, for example $A=\{a_i\}, i=\overline{1,n}$ and $B=\{b_i\}, i=\overline{1,m}$, n is the number of elements of the set A, m is the number of elements of the set B, the elements of which have any nature [14].

There are two types of combinatorial optimization tasks. In problems of the first type, each of these basic sets is represented in the form of a graph, the vertices of which are elements, and each edge corresponds to the weight of the edge $c_{lt} \in R, l=\overline{1,n}, t=\overline{1,m}$, R is the set of real numbers. There are connections between the elements of these sets A and B, the numerical value of which called scales are set as matrices.

In the second type of task there are no connections between the elements of given set, and the weights are the numbers $v_i \in R, i=\overline{1,n}$ that correspond to some properties of these elements. The numerical values of elements are defined by finite sequences of data.

Knapsack task definition. In this work we use the methods developed for solution of combinatorial task that is known as "knapsack task". This task is formulated as a combinatorial optimization problem like that: a set of items is given, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who processes fixed-size knapsack and must fill it with the most valuable items. The problem often arises in resource allocation where the decision-makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively.

The knapsack problem is a NP-complete combinatorial optimization problem. It got its name from the ultimate goal: to put into knapsack as many valuable things as possible, on conditions that the capacity of the knapsack is limited. Different variations of the knapsack problem can be encountered in economics, applied mathematics, cryptography, logistics, and so on.

The classical formulation of the problem is formulated as follows: there is a set of objects (terms), and each of them has two parameters, weight (significance) and location in the taxonomy of terms. In general, the problem can be formulated as follows: from some given set of items with properties "value" and "weight" we need to select a subset with the maximum total cost, while adhering to the limit on the total weight (adaptation to semantic models is necessary) [15].

A knapsack that has a capacity V must be packed in such a way with n inseparable items with species values $B=\{b_i\}, i=\overline{1,n}$ and capacities $B=\{b_i\}, i=\overline{1,n}$ that the total cost of packaged items would be maximal, and their total capacity would not exceed the value [16]. For the task of thesaurus optimizing we consider items represented by various natural language (NL) information objects (IO). Their values are defined by significance of terms in IO, and capacities – by volume of IO.

Knapsack task can be reduced to the combinatorial optimization task because the knapsack task is given on one set of objects $A=\{a_1,...,a_n\}$, there are no connections between the elements a_j of this set, and the input data are given by the elements of the sets B and E that characterize the properties $a_j \in A$, i.e. the problem belongs to the second type of optimization problem. The argument of the objective function is a combination without repetitions. We set the sequence of sets B and E by numerical functions $\phi(j)|_1^n = (\phi(1),...,\phi(n))$ and $\varphi(j)|_1^n = (\varphi(1),...,\varphi(n))$. We set combinatorial function

$$\beta(f(j), w^k)|_1^n = (\beta_1(f(1), w^k),...$$

$$..., \beta_n(f(n), w^k)),$$

where $\beta_j(f(j), w^k) = 1$, if element a_j is selected from the set A, and $\beta_j(f(j), w^k) = 0$, otherwise. The objective function is reduced to an expression $F(w^k) = \sum_{j=1}^{n} \beta_j(f(j), w^k)\phi(j)$. Knapsack task consists in finding of such combination $w^{k*} \in W$ for which the objective function $F(w^{k*}) = \max_{w^k \in W} F(w^k)$, if $\sum_{j=1}^{n} \beta_j(f(j), w^{k*})\varphi(j) \le V$,

$k, k^* \in \{1,..., 2^n - 1\}$.

Some variants of knapsack task can be separated:

5. Knapsack task: no more than one copy of each item.

6. Bounded knapsack task: no more than the specified number of copies of each item.

7. Unbounded knapsack task: Arbitrary number of copies of each item.

8. Multiple-choice knapsack task: Items are divided into groups, and only one item can be selected from each group.

9. Multiple knapsack task: There are several knapsacks, each with its maximum weight. Each item can be put in any knapsack or left.

10. Multi-dimensional knapsack task: instead of weight, several different resources are given (for example, weight, volume and packing time). Each item spends a given amount of each resource. It is necessary to choose a subset of items so that the total cost of each resource does not exceed the maximum for this resource, and the total value of items is maximum.

11. Quadratic knapsack task: the total value is given by a non-negative quadratic form [13].

Methods of knapsack task solution. As mentioned above, the knapsack task belongs to the class of NP-complete tasks, and there is no polynomial algorithm to calculate it in a reasonable time. Therefore, solving the knapsack task needs to choose between precise algorithms that are not suitable for "large" knapsacks, and approximate ones that work quickly, but do not guarantee the optimal solution to the problem.

Computationally, various approaches have been proposed for solving the knapsack tasks. All these algorithms can be classified into two categories, 1) exact algorithms, and 2) heuristics or meta-heuristics ones [17]. Exact methods for MKP began several decades ago and include branch-and-bound method, special enumeration techniques and reduction schemes, Lagrangean methods and surrogate relaxation methods.

Exhaustive search. As other discrete problems, the problem of the knapsack can be solved by complete processing of all possible solutions. Under the problem conditions there are N items that can be placed in a knapsack, and we need to determine the maximum value of the cargo with weight that does not exceed W .

There are two options for each item: the item is placed in a knapsack, or the item is not placed in a knapsack. Then the search for all possible options has a time complexity of O ($2N$), that allows to use it only for a small number of items [18]. As the number of items increases, the problem becomes unsolvable by this method in a reasonable time.

The method of branches and borders is a variation of the method of exhaustive *search* with the difference that deliberately non-optimal branches of the search tree of complete search are excluded. As well as a method of exhaustive search, it allows to find the optimum decision and therefore concerns exact algorithms.

The original algorithm, proposed by Peter Kolesar in 1967, suggests arranging items by their specific value (in terms of relation of value to weight) and building an exhaustive *search* tree. Its improvement consists in the process of building a tree for each node: the upper limit of the value of the solution is evaluated, and the construction of the tree continues only for the node with the maximum score [19]. When the maximal upper limit is found in the tree leaf, the algorithm ends its work. The ability of the branch and boundary method to reduce the number of search options relies heavily on input data. It is expedient to apply it only if the specific values of items differ significantly [20].

Methods for solving the knapsack problem are subdivided into exact and approximate ones. If exact solution needs too much time then approximate solution may be sufficient for practical application.

Approximate methods for the knapsack problem include:

1. An example of bulleted list is as following.

- greedy algorithms;
- ant colony algorithms;
- genetic algorithms.
- The greedy algorithm for the knapsack problem is as follows:
- the set of items Q is ordered by decreasing the «specific value» of items,
- then, starting from the empty set, objects from the ordered set items are successively added to the approximate solution Q '(initially this set is empty);
- each attempt of adding of item to the knapsack is accompanied with comparison of its weight with empty volume of the backpack;
- the process of constructing an approximate solution to the knapsack problem is ended when all items are considered.

The *ant colony* algorithm is based on the analysis of ant behavior. This algorithm performs the same actions that ants can perform when searching for paths to an object. For each ant, the action of taking an item depends on three components: the ant's memory, importance of item and the virtual pheromone trace. An ant's memory is a list of items taken by an ant that cannot be analyzed iteratively. It is also necessary to include in the list those items that break restrictions on the volume of the backpack. Importance of item is the value inverse of the volume of the item. Ant Colony Optimization (ACO) is a meta-heuristic. And it has been applied to many hard discrete optimization problems. Recently, some researchers have proposed several different ACO algorithms to solve the multidimensional knapsack problem (MKP), which is an NP-hard combinatorial optimization problem.

Special importance is given to local information. It is expressed in a heuristic desire to take an object (the smaller the object, the greater the desire) to put it in a backpack. The virtual trace of the pheromone on the item confirms the ant experience dealt with attempt to process it. To study the entire space of objects, it is necessary to ensure the evaporation of the pheromone: at the beginning of the optimization, the amount of pheromone is taken equal to a small positive number, the number of ants can be assigned equal to the number of items.

Stochastic optimization techniques like evolutionary algorithms, simulated annealing etc., which rely heavily on computational power, have been developed and used for optimization. Among these, evolutionary algorithms, which are randomized search techniques aimed at simulating the natural evolution of asexual species, are found to be very promising global optimizers. The *genetic* algorithm used for knapsack problem is based on the evolutionary principles of heredity, variability and natural selection. This algorithm works with a population of individuals and encodes their chromosomes (genotype) for possible solution to the problem (phenotype).

At the beginning of the algorithm, the population is formed randomly. In order to assess the quality of solutions, the fitness function is used to calculate the fitness of each individual. According to the results of the evaluation of individuals, the most adapted of them are selected for crossing. As a result of crossing of selected individuals by using a genetic crossover operator new population is formed.

The multidimensional 0-1 knapsack task is a NP-hard combinatorial optimization problem. The problem is an extension of the standard 0-1 knapsack problem with many constraints while the standard 0-1 knapsack problem has only one constraint. The objective of this approach is to maximize the sum of the values of the items to be selected from a given set by taking into account multiple resource constraints.

All these methods can be used for solution of various problems defined in terms of knapsack task. For minimized thesaurus constructing we need in some estimates that define quantitatively the importance of each domain concept for user task in particular IIS. We propose to use ontology-based semantic similarity measures of domain concepts for these purposes.

Semantic similarity and criteria of its estimations

Task thesaurus allows to define that subset of domain which is interesting for user in solving a task as a subset of ontology terms that is generated as certain sub-graph of ontology. Such sub-graph can contain, for example, the concepts which are linked to selected terms with selected subset of relations. They should have some properties with defined values or concepts that are semantically similar to selected terms of ontology.

We define *semantically similar concepts* (SSC) as a subset of the domain concepts joined by some relations, properties, attributes or any other characteristics (for example, joint use or identical elements). There are several ways to build SSC that can be used separately or together. Generation of SSC starts from selection of non-empty initial set of concepts. Then various approaches support retrieval of other concepts that are semantically similar to concepts from initial set. User can define SSC manually according to personal believes about domain.

More often SSC is generated automatically by processing concept links with initial set of concepts (by some subset of the ontological relations) or with the help of matching concept properties. Such processing defines semantic similarity estimation between analyzed concept and concepts from initial set of SSC.

A lot of different approaches used now to quantifying the semantic distance between concepts are based on ontologies that contain these concepts and define their relations and properties. The source [21] classifies methods and their software realizations of such semantic similarity measuring. Methods are grouped by parameters used in estimations and differ within the groups by calculation of these parameters.

Estimations of semantic similarity. Usually generation of task thesaurus starts from the set of task keywords. Domain ontology can be used to define other domain concepts that have semantic links with these keywords. All concepts of ontology have some nonzero value of semantic closeness (they are connected one with the other at least by superclass “Thing”). Therefore we have to define what relations of ontology are important for task, what similarity estimations are used and what threshold value of similarity is acceptable.

The similarity of two entities can be defined on base of information about direct and indirect superclasses of these concepts; and instances of these concepts. The most commonly used way of semantic similarity evaluation in taxonomy lies in measuring the distance (path length from one node to another) between concept nodes – semantic similarity is defined as inverse function to the shortest path length. If elements are connected by multiple paths between them the shortest path length is used. This approach is used also for analysis of thesauri [22]. However, this approach is based on hypothesis that all relations between taxonomy concepts represent equal distances, but real taxonomies have great variability of distances covered by the same taxonomic relation, especially if some taxonomy subsets are much denser than others. Some researchers calculate similarity estimates on base of singular taxonomic relations “*is-a*” and exclude other types of relations.

For example, the source [23] considers ontology as a directed graph. Ontology concepts correspond to graph nodes, and universal and domain-specific relations (mainly taxonomic “*is-a*”) correspond to graph edges. Estimation of semantic similarity between concepts is calculated as a minimum path length that connects the corresponding ontological nodes: $SS_{Rada} = \min | path(c_1, c_2)$. Similarity estimation proposed by Wu and Palmer [24] is based on the analysis of the path between concepts and their depth in the hierarchy: $SS_{WP} = 2H/(N_1 + N_2 - 2H)$, where N_1 and N_2 are calculated as a number of “*is-a*” relations between concepts c_1 and c_2 to the lowest common generic object (subsumer) c, and H is the number of “*is-a*” relations between c and the *root* of taxonomy.

Other researchers take into account also relations “*part-of-part*” [25].

An alternative way of evaluating semantic similarity in a taxonomy, based on the concept of informational content, which is also not sensitive to the different sizes of distances between relations is offered in [26]. Important factor in the similarity of taxonomy concepts is the degree of their information sharing that defines the number of highly specific terms that is applied to both of these concepts. Measures of similarity based on information content determine the similarity of two concepts. It is defined as information content of their lowest common generic object (subsumer).

In general, all semantic similarity estimates provide some function S: that defines quantitative value of similarity for all concepts of domain ontology. Input information for S includes: domain ontology O, initial set of concepts $C_0 \subseteq X$ and analyzed concept $c_i \in X$,

$$\forall c_i \in X \exists S(O, C_0, c_i) = w_i \geq 0 .$$

Optimization of task thesaurus

To reduce task thesaurus $Th =< T_{Th}, R_{Th} >$ by methods of combinatorial optimization we have to represent its characteristics in terms of knapsack task. We analyze the set T_{Th} of concepts that are contained in this thesaurus. For this analysis we propose to use:

- the set of task thesaurus concepts $T_{Th} = \{t_k\}, k = \overline{1, p}$;
- domain ontology O that was used as a base for Th generation;
- initial set of task concepts $C_0 \subseteq T_{Th} \subseteq X$ (these concepts have to be placed into all variants of task thesaurus);
- function of semantic similarity estimation S that defines significance of concept for user task;
- values of some selected semantic similarity estimation for all elements of T_{Th}: $w_i = S(O, C_0, c_i) \geq 0$ that can be used as a value from knapsack task;
- length of concept name $l_i = |c_i| \geq 0$ defined as a number of symbols in this name that can be used as a weight from knapsack task;
- user defined memory capacity that is given for thesaurus storage.

We understand that memory needed for thesaurus storage is not a problem now. For NP-complete combinatorial optimization size of processed data it defines the calculation time. Therefore we try to add to C_0 concepts with bigger values of semantic similarity according to one of knapsack task solution methods till then their length l_i is less then the free space in memory for thesaurus. Selection of optimization method and function of semantic similarity estimation depends on task specifics and user needs.

Practical use of optimized task thesaurus

Practical use of optimized task thesaurus. Approach to generation and optimization of task thesaurus for IIS we test on problem of personified information retrieval. Intelligent retrieval system "MAIPS" [27] use thesauri generated semi-automatically on base of domain ontologies selected by users. This IIS use task thesaurus defined by user to filter retrieval results received from retrieval systems. Every user can select one or more domain ontologies and generate one ore more task thesauri for each of them. Moreover, users can combine thesauri based on different ontologies by set-theoretic operations. Now we enrich functions of MAIPS dealt with thesauri by optimization operation (Fig. 3). Thesauri in MAIPS are visualized by tag cloud where font size represents the significance of concept for user task.

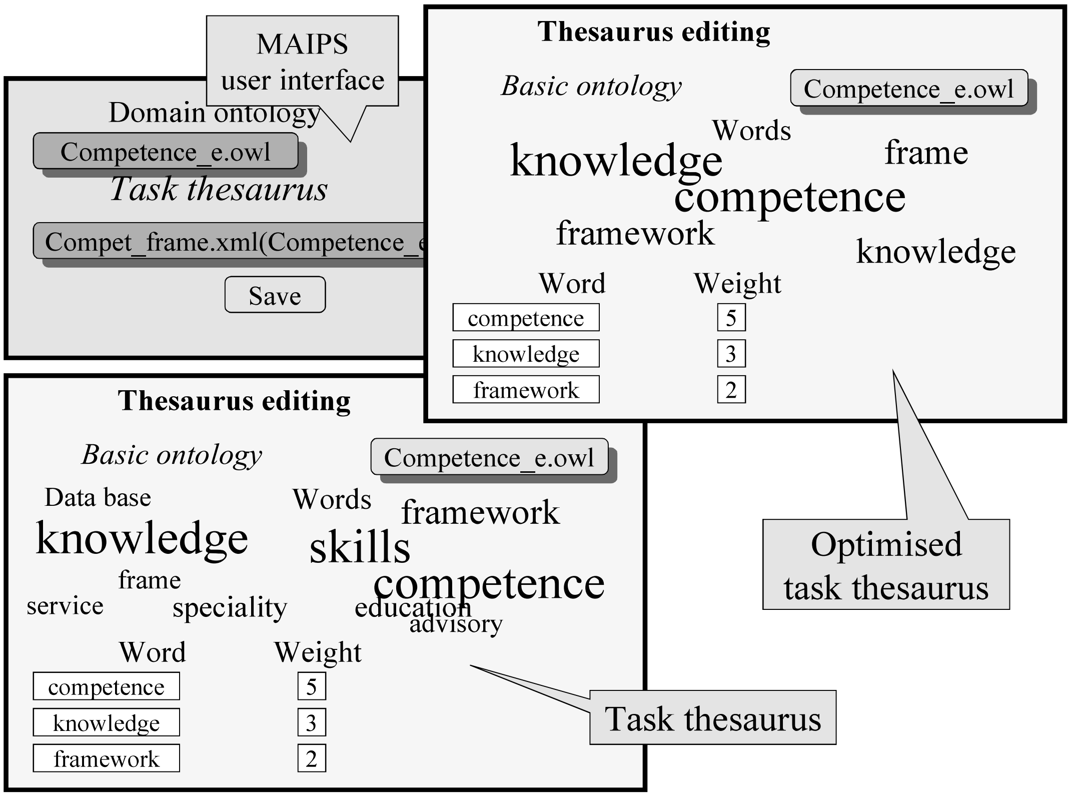
We compare the time and quality of retrieval with usual task thesaurus and with optimized one and draw a conclusion that processing of optimized thesaurus distinctly accelerates data processing. And use of this filtering of concepts with low semantic similarity estimates not influences substantially retrieval results.

Prospects for further use. We consider that use of combinatorial methods to form optimized user profiles that meet user conditions can be applied in various IIS that work with sets of competencies [30, 31], [28].

For example, if for a certain problem it is necessary to use a set of competencies K, then the problem is solved by construction of minimized set of items (courses, learning disciplines, experts, employees, etc.) P such that $\bigcup_{i=1}^{n} c(p_i) \subseteq K$, where $c(p_i)$ is a set of competencies of the *i*-th participant R. Now we plan to include appropriate service into the advisory system "Advisont" [29].

Acknowledgements

The article was prepared in the framework of basic research "Development of meth-



Word	Weight
competence	5
knowledge	3
framework	2

Word	Weight
competence	5
knowledge	3
framework	2

Fig. 3. Use of optimized task thesaurus in MAIPS

ods and tools for integrating combinatorial optimization and semantic modeling for information technology," topic code VF 170.26 (2017-2022), which was performed at the International Research Center for Information Technology and Systems of the National Academy of Sciences of Ukraine and the Ministry of Science and Education. Results of research work №: III-2-17 "Methods and tools for creating intelligent service-oriented information and support systems in the Semantic Web environment" provided by Institute of Software Systems of National Academy of Sciences of Ukraine were used in theoretical and practical parts of this research.

References

1. Gruber T. R. A translation approach to portable ontology specifications. Knowledge Acquisition 5, (1993), pp. 199-220.
2. Kleshche A., Artemjeva I. A Structure of Domain Ontologies and their Mathematical Models. URL: www.iacp.dvo.ru/es/.
3. Gladun A., Rogushina J. Mereological Aspects of Ontological Analysis for Thesauri Constructing. In Buildings and the Environment, Nova Science Publishers. (2010), pp. 301-308.
4. The differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a metamodel. URL: www.metamodel.com/article.php?story=20030115211223271.
5. Matthews B.M., Miller K., Wilson M.D. A Thesaurus Interchange Format in RDF. – URL: www.limber.rl.ac.uk/External/SW_conf_thes_paper.htm.
6. Thesaurus Links URL: https://www.w3.org/2001/sw/Europe/reports/thes/thes_links.html.
7. Lassila O., McGuinness D. The role of frame-based representation on the semantic web. Linköping Electronic Articles in Computer and Information Science, 6(5), (2001).
8. Gladun, A., & Rogushina, J. Use of Semantic Web technologies in design of informational retrieval systems. In book Building and Environment, Nova Science Publishers, (2010), pp. 289-299. http://www.novapublishers.org/catalog/product_info.php?cPath=23_67_742&products_id=10117
9. Gladun, A., & Rogushina, J. Use of Semantic Web technologies in design of informational retrieval systems. In book Building and Environment, Nova Science Publishers, (2010), pp. 289-299. URL: https://www.researchgate.net/profile/Anatoly-Gladun/publication/287721726.
10. Gladun, A., & Rogushina, J. Distant control of student skills by formal model of domain knowledge. International Journal of Innovation and Learning, 7(4), (2010), pp. 394-411.
11. Use_of_semantic_web_technologies_in_design_of_informational_retrieval_systems/links/569ff66c08ae4af52546d9cc/Use-of-semantic-web-technologies-in-design-of-informational-retrieval-systems.pdf.
12. Gladun A., Rogushina J."Distant control of student skills by formal model of domain knowledge". International Journal of Innovation and Learning, 7(4), (2010), pp. 394-411.
13. Rogushina J., Priyma S. Use of competence ontological model for matching of qualifications. Chemistry: Bulgarian Journal of Science Education, Volume 26, №2, (2017), pp. 216-228.
14. Kalfoglou Y., Schorelmmer M. "Ontology mapping: the state of the art." The Knowledge Engineering Review 18(1), (2003), pp. 1–31.
15. Gladun A., Rogushina J. Semantic search of Internet information resources on base of ontologies and multilinguistic thesauruses. Information Theories & Applications, Vol.14, (2007), pp. 48-55.
16. Timofieva N. On some approaches to estimating the optimal solution of combinatorial optimization problems, USiM, Control systems & computers, №3 (281). (2019), pp. 3–13. URL: doi.org/10.15407/csc.2019.03.003. (in Ukraininan).
17. Sergienko I., Kaspshitskaya M. Models and methods of solving combinatorial optimization problems on a computer. - K.:Naukova dumka, 1981, 281 p. (in Russian)
18. Sigal I., Ivanova A. Introduction to Discrete Application Programming: Models and Computing. algorithms.: M.: Fizmatlit, 2002, 237 p.
19. Korbut A., Finkelstein Yu.Discrete programming. - M .: Nauka, 1969. 368 p. (in Russian)
20. Kilincli Taskiran G., An Improved Genetic Algorithm for Knapsack Problems (Doctoral dissertation, Wright State University). 2010. URL: core.ac.uk/download/pdf/36754668.pdf.
21. Okulov S. Programming in algorithms. Binom. Laboratory of Knowledge, 2007. - ISBN 5-94774-010-9. (in Russian)
22. Martello S., Toth P. Knapsack problems: algorithms and computer implementations. John

Wiley & Sons Ltd., 1990. - P. 29.50. - 296 p. ISBN 0-471-92420-2.

23. Burkov V., Gorgidze I., Lovetsky S. Applied problems of graph theory / ed. J. Gorgidze - Tbilisi: Computing Center of the USSR Academy of Sciences, 1974. 231 p. (in Russian)
24. Taieb M., Aouicha A. H., Hamadou M. B. "Ontology-based approach for measuring semantic similarity." Engineering Applications of Artificial Intelligence, 36, (2014), pp. 238-261.
25. Rada R., Bicknell E. Ranking documents with a thesaurus. JASIS, V.10(5), (1989), pp. 304-310.
26. Rada R., Mili H., Bicknell E., Blettner M. Development and application of a metric on semantic nets. IEEE transactions on systems, man, and cybernetics, 19(1), (1989), pp. 17-30.
27. Wu Z., Palmer M. Verbs semantics and lexical selection. Proceedings of the 32-nd Annual Meeting on Association for Computational Linguistics, ACL'94, Association for Computational Linguistics, Stroudsburg, PA, USA, (1994), pp. 133–138.
28. Richardson R., Smeaton A. F., Murphy J. Using WordNet as a knowledge base for measuring semantic similarity between words. Working paper CA-1294, Dublin City University, School of Computer AppUcations, Dublin, (1994).
29. Resnik P. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11, (1999), pp. 95-130.
30. Rogushina J. Analysis of Automated Matching of the Semantic Wiki Resources with Elements of Domain Ontologies. International Journal of Mathematical Sciences and Computing (IJMSC), 3(3), (2017), pp. 50-58.
31. Rogushina J., Gladun A., Pryima S., Strokan O. Ontology-Based Approach to Validation of Learning Outcomes for Information Security Domain. CEUR Vol-2577, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security", (2019), pp. 21-36. URL: ceur-ws.org/Vol-2577/paper3.pdf.

Received: 11.05.2021

About authors:

Rohushina Julia Vitalievna,
Candidate of Physical and Mathematical Sciences, Senior Research Fellow.
Number of scientific publications in Ukrainian publications - 150.
Number of scientific publications in foreign publications - 31.
http://orcid.org/0000-0001-7958-2557,

Gladun Anatoliy Yasonovych,
Candidate of Technical Sciences, Senior Research Fellow.
Number of scientific publications in Ukrainian publications - 160.
Number of scientific publications in foreign publications - 45.
http://orcid.org/0000-0002-4133-8169,

Affiliation:

Institute of Software Systems, National Academy of Sciences of Ukraine.
03680, Kyiv, Ukraine.
Academician Glushkov Avenue, 44.
Tel:066 550 1999.
E-mail: ladamandraka2010@gmail.com,

International Research and Training Center of Information Technologies and Systems, National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine.
03187, Kyiv, Ukraine.
Academician Glushkov Avenue, 40.
Tel: 044 502 6366.
E-mail: glanat@yahoo.com