Construction of a Matrix "Physical Effects – Technical Functions" on the Base of Patent Corpus Analysis

Dmitriy Korobkin(✉), Dmitriy Shabanov, Sergey Fomenkov, and Alexander Golovanchikov

Volgograd State Technical University, Lenin Avenue 28, Volgograd, Russia
dkorobkin80@mail.ru

Abstract. Authors use physical effects (PE) to synthesize the physical operation principle of a technical system. PEs implements the technical functions (TF) that describe the functional structure of the declared technical system. The method finds out relationships between physical effects and technical functions performed by them based on the construction of term-document matrices and the search for hidden dependencies in them. To this end, the authors developed a method for extracting descriptions of physical effects from patents in USPTO and RosPatent databases, as well as a method for extracting technical functions from the natural language texts of the same documents. The developed software has been tested for the tasks of extracting physical effects and technical functions from patent documents.

Keywords: Patent · NLP · Fact extraction · Data mining · Physical effect · Technical function · SAO

1 Introduction

Analysis of the computer-aided innovation systems (CAI) [1] such as Goldfire Innovator, TechOptimizer, Innovation Workbench, Idea Generator, Pro/Innovator, etc., shows these CAI systems do not solve the fundamental problem of updating the information component of the new technical systems generation. Using the world patent database (more than 20 million documents) and open databases of scientific and technical information as a global knowledge base most correctly. The engineer/inventor realizes the required functions of the designed technical system on the basis of heuristic morphological synthesis or the physical operation principle (POP) [2], which is a sequence (network) of physical effects (PE) [3]. Therefore, it is required to extract the following data from the global information space: physical and technical effects, morphological features and their alternatives (elements of functional structure, technical realizations of objects), i.e. information necessary to solve the problems of information support for the synthesis of new technical solutions.

The task of determining the most effective physical operation principle of a technical system with a selected functional structure that synthesized on the basis of a morphological matrix is solved by 2 methods: (1) the method for automating the

A. G. Kravets et al. (Eds.): CIT&DS 2019, CCIS 1084, pp. 52–68, 2019.
https://doi.org/10.1007/978-3-030-29750-3_5

procedures for synthesizing the physical operation principle based on the database of technical functions performed by physical effects, and (2) the developed method for verification the practical realizability [4] of the synthesized POP on based on the criteria parameters of the physical effects that are part of a POP structure.

2 Extracting SAO (Subject-Action-Object) Structures from Natural Language Text

2.1 Segmentation

The “Subject-Action-Object” (SAO) [5] semantic construction is a key concept makes it possible to recognize technical objects and its elements, problems and solutions, technical objects functions and physical effects descriptions in a natural language text.

The following information is extracted from the patent xml-file:

<B110>… </B110> – a patent number;
<B220><date>… </date></B220> – a patent date;
<ru-b542>… </ru-b542> – a patent title;
<ru-b560>… </ru-b560> – a list of citations;
<B721><ru-name-text>… </ru-name-text></B721> – a list of patent authors;
<claims>… </claims> – a patent clame.

The text of the patent claim itself contains in <claim-text></claim-text> tags. Patent claims are based on itself pattern that differs from the usual sentence structure. Most often, the formula is one complex sentence, in which there are several subordinates, each of which extends the properties of the object, which is referred to in the main sentence. Analysis of such a long sentence will be an actually time-consuming procedure. To reduce semantic analysis errors, the sentence is segmented into several parts, and then each part of the original sentence analyzed separately.

The segmentation algorithm [6] consists of the following transformations using regular expressions. To remove the numbering like «A. », «a. », «1. » or «1) » is used «^(\d{1,4}|[a-zA-Z]{1,2})(\.|\))\s» pattern, to remove references like «4. The device according to claim 3…» is used «^.+(of|in|to) claim \d+(,)?» pattern. Sentences are separated by punctuation characters by replacing pattern «(\.|!|\?|:|;)\s?» with the line breaks symbol. Also, the line breaks replace stop words: «, wherein», «, said», «, and», «; and», «, thereby», «if», «else», «thereby», «such that», «so that», «wherein», «whereby», «where», «when», «while», «but». Thus, the sentences will be cut into shorter ones without losing meaning, which will increase the likelihood of the semantic analyzer to work correctly.

2.2 Semantic Text Analysis

We will use the UDPipe library to analyze Russian and English text. The input and output data format for it is CoNLL. For each sentence, a dependency tree is built.

Example of dependency tree from the sentence of patent US20130307109A1: “the incident light causes photoelectric conversion, generating charges”:

```
1 the the DET RD Definite=Def|PronType=Art 4 det _ _
2 incident incident ADJ A Degree=Pos 3 amod _ _
3 light lay NOUN S Number=Sing 4 nmod _ _
4 causes cause NOUN S Number=Plur 6 nsubj _ _
5 photoelectric photoelectric ADJ A Degree=Pos 6 amod _ _
6 conversion conversion NOUN S Number=Sing 0 root _
SpaceAfter=No
```

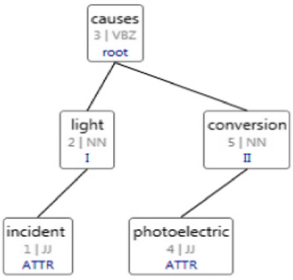
The dependency tree uses Stanford Dependencies [7] such as nominal subject (nsubj), object (pobj), adjectival modifier (amod), tree root (root). After that, vertexes such as "det" with a semantically minor role are removed from the dependency trees. Thus we obtain reduced Collapsed Stanford Dependencies. Based on Meaning-Text Theory (MTT) [8] this reduced trees with Stanford dependencies are converted to Deep Syntactic Structures (DSyntS).

Reduced Stanford dependencies example:

```
ATTR(light-2, incident-1)
I(causes-3, light-2)
OPER(ROOT-0, causes-3)
ATTR(conversion -5, photoelectric-4)
II(causes-3, conversion-5)
```

From deep syntactic structures, we extract the SAO structures in which the root is «Action», its children are «Subject» (I-vertex) and «Object» (II-vertex). There are several Subjects and Objects for one Action, in addition, for each Subject, Object, and Action, its child relations are extracted.

The following SAO structure is extracted (Fig. 1):



Action – «cause» (root vertex);
Subject –«light» (with "incident" as child element);
Object – «conversion» (with "photoelectric" as child element).

Fig. 1. Graphical representation of a dependency tree

2.3 SAO Grouping

To increase the information content of SAO structures, several SAOs are combined into one according to the developed grouping (comparison) algorithm [9]. Consider an example of comparing two SAOs extracted from sentences "An atom with an ionized shell emitted K-series X-ray photons" and "An ionized atom slowly emitted L-series photons" (Fig. 2).

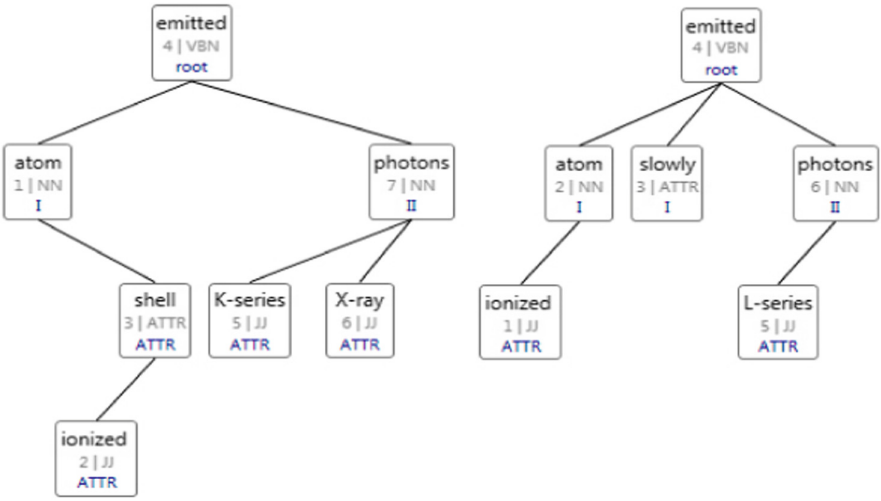


Fig. 2. Comparison of two extracted SAO

The first stage is a comparison of the "Action". If the vertexes of the first SAO (query search case, QSC) and the second SAO (document search case, DSC) do not match, then the SAO trees are not further compared. If the "Actions" match, then the child (not "Subject" and not "Object") elements (children) associated with the root vertex ("Action") are compared.

At each level, if the terms (words) do not match, a significance test occurs. The significance test of a term is made on the basis of a previously prepared table in which IDF factors are defined. If the term IDF is less than the threshold value, then the word is not significant and is not taken into account in the calculation of the similarity factor.

In our case, the «Action» vertices are equal, but their child elements are not (Fig. 2).

Let's introduce the concept of the coefficient of similarity of child elements associated with the root vertex (Action):

$$K^{A}_{Child}(TA_k, TA_l) = \frac{\sum_{i=1}^{N_i} S(t_1, t_2)}{3 \times \max_{Child}(TA_k, TA_l)}, \quad (1)$$

where TA_k, TA_l – subtrees (root related children) for k-th and l-th QSC and DSC respectively;

$max_{Child}(TA_k, TA_l)$ – the maximum number of child elements for QSC and DSC with IDF;

$S(t_1, t_2)$ – an operator defining the similarity of the child elements t_1 and t_2 for the compared subtrees (returns 1 if they completely match);

N_i – number of children in the TA_k semantic tree.

Consider the example on Fig. 2. There are no Action children elements in QSC, DSC has one child element, respectively, so the maximum number of child elements is 1 and the number of matched elements is 0. Including term significance check (IDF of the term is greater than the threshold value) the factor is 0/3 * 1 = 0.

At the second stage, the similarity of the "Subject" is checked. Vertices are compared for I-vertex relations and their children structures.

If the I-vertices of QSC and DSC do not match, then the similarity coefficient of SAO for the "Subject" is 0 and a comparison is made for the next I-vertex. If the I-vertices of QSC and DSC coincides, then their child elements related to the given I-vertex (Subject) are compared (Subject).

Let's introduce the concept of the similarity coefficient of child elements related to the I-vertex (Subject):

$$K^I_{Child}(TI_k, TI_l) = \frac{\sum_{i=1}^{N_i} S(t_1, t_2)}{\max_{Child}(TI_k, TI_l)}, \tag{2}$$

where TI_k, TI_l – is the subtrees (the child elements related with the I-vertex) for the k-th and l-th sentence in QSC and DSC respectively;

$max_{Child}(TI_k, TI_l)$ – is the maximum number of children for QSC and DSC taking into account the;

$S(t_1, t_2)$ – is the operator defining the coincidence of the child elements t_1 and t_2 for the compared subtrees; it returns 1 if matched;

N_i – is the number of child elements of the semantic tree TI_k for DSC.

Let's introduce the SAO similarity coefficient by "Subject":

$$K^I = \frac{\sum_{i=1}^{N_t=|I_k|} K^I_{M_i}}{\max(I_k, I_l)} + \frac{\sum_{i=1}^{N_t=|I_k|} K^I_{Child_i}}{3 \times \max(I_k, I_l)}, \tag{3}$$

where $K^I_{M_i}$ – is the coincidence coefficient of the i-th I-vertex, if there is a match $K^I_{M_i} = 1$, if there is mismatch $K^I_{M_i} = 0$;

$K^I_{Child_i}$ – is the coincidence coefficient of child elements, related to i-th I-vertex;

$max(I_k, I_l)$ – is the maximum number of I-vertex in QSC and DSC.

Let's consider the example in Fig. 2. There is 2 child element in the QSC for the I-vertex, in the DSC there is 1, respectively, maximum 2. IDF is greater than the limit value for all terms, which means they are meaningful. The coefficient K^I_{Child} for a pair

of "shell, ionized" → "ionized" is 0.5. The SAO similarity coefficient by "Subject" – K^I is 1.1667.

(3) At the third stage, there is a check of the similarity of the "Object" (O) - a comparison of the vertices for the II-vertex relations and related child elements. The SAO comparison algorithm by the "Object" is similar to the comparison algorithm by the «Subject».

Let's introduce the concept of the similarity coefficient of child elements related to the II-vertex (Object):

$$K^{II}_{Child}(TII_k, TII_l) = \frac{\sum_{i=1}^{N_i} S(t_1, t_2)}{\max_{Child}(TII_k, TII_l)}, \tag{4}$$

where TII_k, TII_l – is the subtrees (the child elements related with the II-vertex) for the k-th and l-th sentence in SQC and DSC respectively;

$max_{Child}(TII_k, TII_l)$ – is the maximum number of children for QSC and DSC taking into account the IDF;

$S(t_1,t_2)$ – is the operator defining the coincidence of the child elements t1 and t2 for the compared subtrees; it returns 1 if matched;

N_i – is the number of child elements of the semantic tree TII_k for QSC.

Let's introduce the SAO similarity coefficient by «Object»:

$$K^{II} = \frac{\sum_{i=1}^{N_t=|II_k|} K^{II}_{M_i}}{\max(II_k, II_l)} + \frac{\sum_{i=1}^{N_t=|II_k|} K^{II}_{Child_i}}{3 \times \max(II_k, II_l)}, \tag{5}$$

where $K^{II}_{M_i}$ – is the coincidence coefficient of the i-th II-vertex, if there is match $K^{II}_{M_i} = 1$ –, if there is mismatch $K^{II}_{M_i} = 0$;

$K^{II}_{Child_i}$ – is the coincidence coefficient of child elements, related with i-th II-vertex;

$max(II_k, II_l)$ – is the maximum number of II-vertex in QSC and DSC.

Let's consider the example in Fig. 2. There is 2 child element in the QSC for the II-vertex, in the DSC there is 1, respectively, maximum 2. IDF is greater than the limit value for all terms, which means they are meaningful. The coefficient K^{II}_{Child} for a pair of "X-ray, K-series" → "L-series" is 0. The SAO similarity coefficient by "Object" – KII is 1.

(4) The calculation of the SAO similarity coefficient.

Let's introduce the concept of the similarity coefficient of 2 SAO structures:

$$K_{SAO} = K^A_{Child} + K^I + K^{II}, \tag{6}$$

where K^A_{Child} – is the similarity coefficient of child elements associated with the ROOT vertex (Action);

K^I – is the SAO similarity coefficient by "Subject";

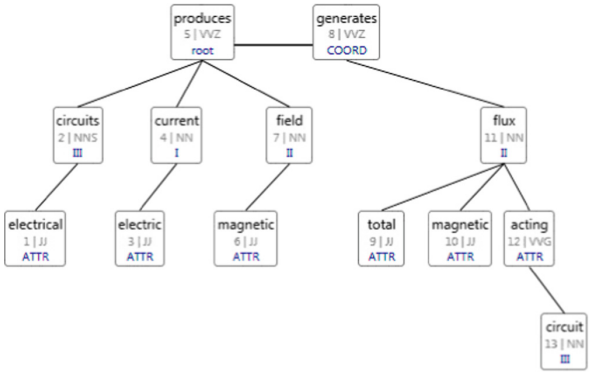
K^{II} – is the SAO similarity coefficient by "Object".

Two SAOs can be grouped if their similarity coefficient is greater than a threshold value of 2, which means that the Action, Subject, Object components have coincided excluding the child elements. The maximum value of this coefficient is 3.

3 Search for Descriptions of Physical Effects in Natural Language (NL) Documents

For the problem of extracting descriptions of physical effects and POP structures from Russian and English texts used previously developed procedures for segmentation of complex sentences of patent texts, the building of dependency trees and deep syntactic structures based on the Meaning-Text Theory for reduced Stanford dependencies.

An example of deep-syntactic structures from patent proposal US20130307109A1 «In electrical circuits, any electric current produces a magnetic field and hence generates a total magnetic flux acting on the circuit» is shown in Fig. 3.




```
ATTR(circuits-2,
electrical-1)
III(produces-5, cir-
cuits-2)
ATTR(current-4, elec-
tric-3)
I(produces-5, cur-
rent-4)
OPER(ROOT-0, produc-
es-5)
ATTR(field-7, magnet-
ic-6)
II(produces-5, field-
7)
COORD(produces-5,
generates-8)
```

```
ATTR(flux-11, total-9)
ATTR(flux-11, magnetic-10)
II(generates-8, flux-11)
ATTR(flux-11, acting-12)
III(acting-12, circuit-13)
```

Fig. 3. Example of deep-syntactic structures

According to the model of the physical effect [10], developed at the CAD department of VSTU, in a Natural-Language text containing a description of the PE, it is necessary to find predicates (verbs) that have some "effect" on arguments that have definite active relations with the predicate inside this "influence". There were identified in a single class all predicates characteristic for describing PE in the physical profile text, such as "влияние", "воздействие", "зависимость", etc. (for Russian language), "change", "increase", "decrease", "depend", "change", "generate", "act", "cause", etc. (for English).

During the work on the project as a result of the analysis of Russian and English texts containing descriptions of PE, for each predicate, the actant relations "I" (what

influences), "II" (what the effect is aimed at), "III" (where the effect is implemented) were correlated with description elements of the PE.

There was modified the model for the representation of the description of the physical effect in the Russian and English text, previously developed by the authors [11].

$$M_{PE} = <P, Act, E, R_P, R_E>, \quad (7)$$

where P – is the set of predicates (verbs), specifically describe the PE in the Russian and English text, $p_i \in P$;

Act – is the set of actant relations {I, II, III} of arguments and predicate p_i;

$a_j \in Act$;

E – is the set of elements describing PE (A – PE input, B – PE object, C – is PE output), $E_k \in E$,

$$\forall p_i \in P \quad \exists a_j \in Act \quad [a_j \xrightarrow{def} E_k]$$

where $E_k \in \{A, B, C\}$, def – is the operator assigning to the actant relation of the argument a_j with the predicate d_i the element/set of elements E_k describing the PE;

R_P – is the relation on P × Act, the couple $(p_i, a_j) \in R_P$ uniquely identifies the element/elements of the PE description, that realizes the actant relation a_j for the predicate p_i;

R_E – is the relation on R_P × E, the couple $((p_i, a_j), E_k) \in R_E$ defines a set of concepts of the subject domain "Physical Effect", corresponding to the element of the description of the PE e_l, $e_1 \in E_k$;

According to the modified model, a database of patterns of representations of descriptions of structured physical knowledge in Russian (104 patterns) and English texts (36 patterns) was created.

For example,

P = PRODUCE; Act = {I,II,III}; E = {PE input (A), PE object (B), PE output (C)}; $I \xrightarrow{def} A$; $II \xrightarrow{def} C$; $III \xrightarrow{def} B$;

P = ACT; Act = {I,II,III}; E = {A, B, C}; $I \xrightarrow{def} A$; $II \xrightarrow{def} A,C$; $III \xrightarrow{def} B$;

Next is the construction of a semantic network for the description of the PE in the text.

The vertices of the semantic network $V_i = (T_i, E_i)$, where T_i is the Natural Language representation of the argument (term) for the predicate d_j; E_i is an element of PE description, represented in the text as T_i, $E_i \in R_P$.

When constructing a semantic network, the presence of the Natural Language representation of the argument T_i is tested at the thesauruses of "Physical Effect" ontology that developed by the authors of the paper.

Let us give an example of the semantic network obtained from the sentence «In electrical circuits, any electric current produces a magnetic field and hence generates a total magnetic flux acting on the circuit» (Fig. 4).

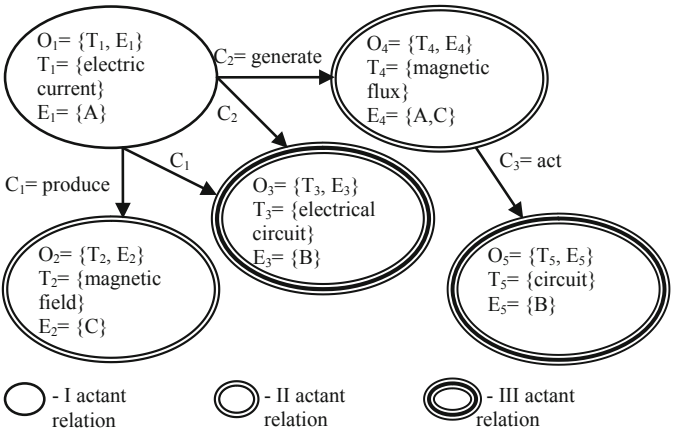


Fig. 4. An example of a semantic network

The extraction of structures of PE is based on the constructed semantic network using the procedure of combining concepts based on the taxonomy of the ontology «Physical Effect».

For example, after analyzing a piece of text: «In electrical circuits any electric current produces a magnetic field and hence generates a total magnetic flux acting on the circuit» we obtain the structure of the PE, which has as an input effect «electric current», as an object of the PE – «electrical circuit» and as an output effect – «magnetic flux» (Fig. 5).

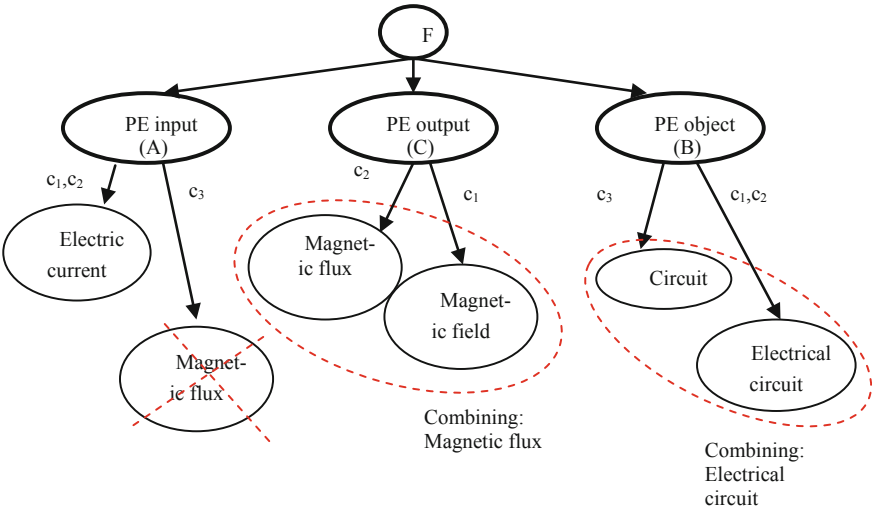


Fig. 5. Extracting information on the structure of PE

In this case, since the full structure of the PE, that contains the input action and no more than one output effect and object, was extracted from the text fragment, then we can speak of the generated description of the new or existing physical effect in the database. Otherwise, if there are several extracted output effects or objects (the latter to a lesser extent, since a structural change of the object, can be observed), we can speak of the extracted description of the elements of the structures of the physical principle of action.

4 The Method of the Automated Construction the Matrix of Technical Functions Performed by Physical Effects Based on Analysis of the Patent Corpus

Methods for extracting descriptions of technical functions (TF) and physical effects from patent texts are used to form the two term-document matrices.

The terms (functions of the technical object $TechFunc_i$) are extracted from the aggregate of all documents {Pat} of the patent array and the frequency of their occurrence in patent documents is determined.

At the same time, the device from the description of the claims is taken as the name of the technical object (TO) (the "claim" field): "A battery containing a body,…" (the TO is "Battery"). This information is stored as an attribute of the patent Pat.

Several TFs are combined into one according to the developed grouping (comparison) algorithm to reduce the space of extracted technical functions (TF).

The normalized term frequency weighting was chosen as the localized weighting of the i-th $TechFunc_i$ technical function in the j-th Pat_j patent document:

$$t_{ji} = 0.4 \times \chi(f_{ij}) + 0.6 \times \left(\frac{f_{ij}}{max_k f_{ij}}\right), \tag{8}$$

where f_{ij} – the frequency of occurrence $TechFunc_i$ in Pat_j,

k – the number of technical function in j-th patent document Pat_j,

$$\chi(f_{ij}) = \begin{cases} 1, & \text{if } f_{ij} > 0, \\ 0, & if\, f_{ij} = 0 \end{cases}.$$

A value of 0.4 is assigned to all terms, included in the document, and 0.6 is added to it, depending on the frequency of occurrence of the term (technical function).

Inverted document frequency was chosen as global weighing $TechFunc_i$, which is equal to 0 if the technical function appears in all patent documents of the array, and this weight increases as the number of documents in which the technical function occurs decreases:

$$g_i = \log\left(\frac{N}{\sum_{j=1}^{N} \chi(f_{ij})}\right), \tag{9}$$

where N – the number of documents in the patent array.

It's helpful to normalize the columns of the final matrix after calculating local and global weights. If this is not done, short patent documents may not be recognized as relevant:

$$d_i = \left(\sum_{j=1}^{M} \left(g_j t_{ji} \right)^2 \right)^{-1/2}, \tag{10}$$

where M – the number of terms (technical functions) in the patent array,

t_{ji} is the local weighting of TechFunc_i in Pat_j,

g_i – the global weighting TechFunc_i.

Calculate the reduced frequencies TFIDF_ij of the occurrence of the technical functions TechFunc_i in patent documents Pat_j:

$$TFIDF_{ij} = t_{ji} \times g_i \times d_i, \tag{11}$$

Further, by means of the latent semantic analysis (LSA), the space of technical functions ("noise") is reduced and hidden dependencies between terms (technical functions) are revealed. The LSA uses the term-document matrix as initial information (Table 1). The elements of this matrix contain the reduced frequencies TFIDF_ij of the occurrence of the i-th technical function TechFunc_i in the j-th patent document Pat_j.

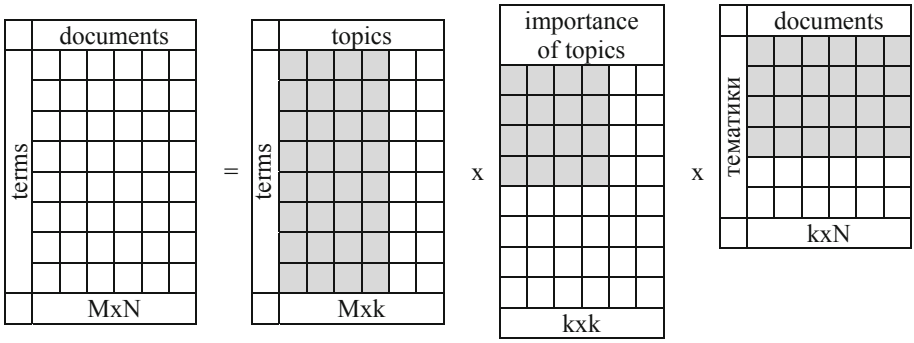
Table 1. Matrix of reduced frequencies of technical functions in documents

Term	Document			
	Pat_1	Pat_2	...	Pat_N
$TechFunc_1$	$TFIDF_{11}$	$TFIDF_{21}$	...	$TFIDF_{N1}$
$TechFunc_2$	$TFIDF_{12}$	$TFIDF_{22}$	...	$TFIDF_{N2}$
...	...	...	...	
$TechFunc_M$	$TFIDF_{1M}$	$TFIDF_{2M}$	...	$TFIDF_{NM}$

The singular decomposition of a term-document matrix into a product of three matrices is used in LSA:

$$A = UDV^T, \tag{12}$$

where U – the term vector matrix, D – the matrix of singular values, V^T – the matrix of vectors of patent documents.



If we leave k the largest singular values in the matrix D, and the columns/rows corresponding to these values in the U/VT matrices, then the product of the resulting matrices $A_k = U_k x D_k x V_k^T$ will be the best approximation of the original matrix A by the rank-k matrix. All terms from the matrix A_k will be an abridged space of technical functions, i.e. the most significant for the patent array will be automatically determined from the entire set of technical functions.

The value of k is chosen empirically, and since the patent array is about a million documents, then k is about 2%.

In the aggregate of all documents {Pat} of the patent array, the physical effects (PE_i) are searched (defined as terms) and the frequency of their occurrence is determined in patent documents.

Normalized term frequency weighting selected as the local weighing of i-th physical effect PE_i in j-th patent document Pat_j:

$$t_{ji} = 0.4 \times \chi(f_{ij}) + 0.6 \times \left(\frac{f_{ij}}{max_k f_{ij}}\right) \times r_i, \tag{13}$$

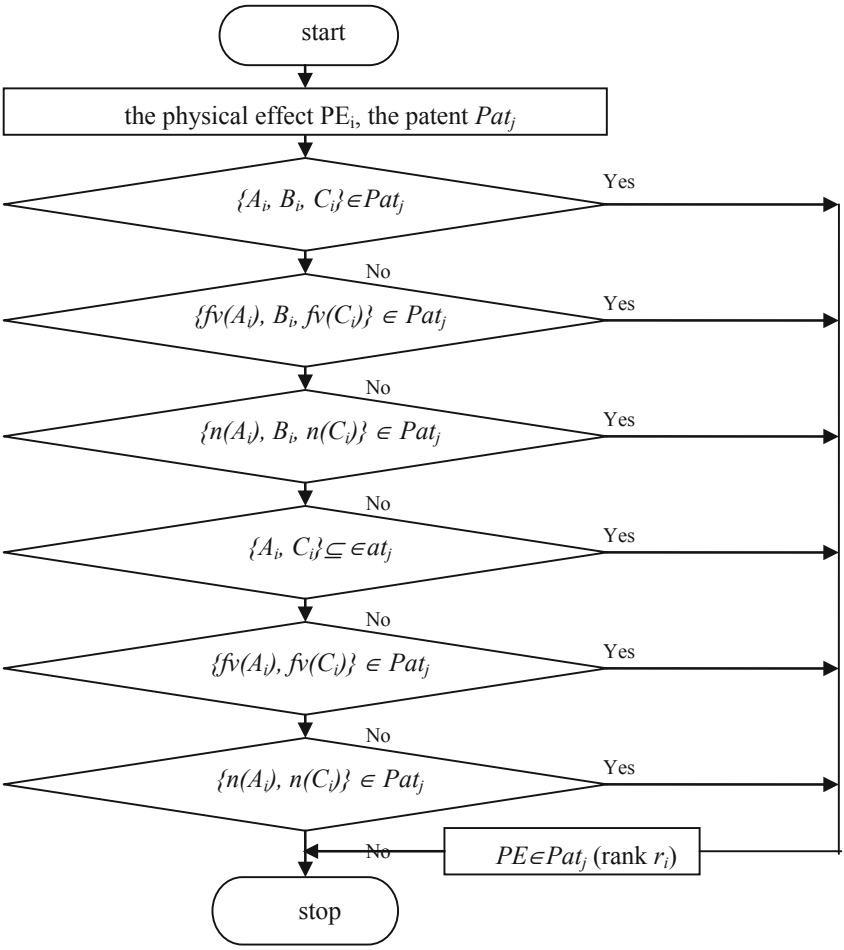
Where f_{ij} – the frequency of occurrence of PE_i in Pat_j,

k – the number of PE in j-th patent document Pat_j,

r_i – coefficient of completeness of the description of physical effect (PE_i) in patent document Pat_j,

$$\chi(f_{ij}) = \begin{cases} r_i, & \text{if } f_{ij} > 0, \\ 0, & if\, f_{ij} = 0 \end{cases}$$

The search for a description of the physical effect in a patent document is carried out by performing iterative steps with a consistent simplification of the query (Fig. 6):



A, B, C – formalized descriptions of the PE input, object, and output,

fv(A), fv(C) - physical values of input A and output C for PE (for non-parametric effects),

n(A), n(C) names of respectively input and output effects.

Fig. 6. Algorithm to simplify the search query

(a) search for full descriptions of input effect A, of object B, output effect C, $r_i = 1$;
(b) search for full descriptions B, physical values A and C (for non-parametric effects), $r_i = 0.8$;
(c) search for full descriptions A and C, $r_i = 0.6$;
(d) search for physical values A and C (for non-parametric effects), $r_i = 0.4$;
(e) search for full descriptions B, names of effects A and C, $r_i = 0.3$;
(f) search for names of effects A and C, $r_i = 0.2$.

The inverted document frequency is chosen as the global weighting of PE_i, which is 0, if the physical effect (at one of the 6 stages of detailing (Fig. 6), which was found in the patent Pat_j) appears in all patent documents of the array, and this weight increases as the number of documents, in which the physical effect occurs, decreases:

$$g_i = \log\left(\frac{N}{\sum_{j=1}^{N}\gamma(f_{ij})}\right), \tag{14}$$

where N – the number of documents in the patent corpus,

$$\gamma(f_{ij}) = \begin{cases} 1, & if\ f_{ij} > 0, \\ 0, & if\ f_{ij} = 0 \end{cases}$$

Normalization:

$$d_i = \left(\sum_{j=1}^{Q}\left(g_j t_{ji}\right)^2\right)^{-1/2}, \tag{15}$$

where Q – the number of terms (physical effects) in the patent array,

t_{ji} – the local weighting of PE_i in Pat_j,

g_i – the global weighting of PE_i.

Calculate the reduced frequency of occurrence of the physical effects PE_i in patent documents Pat_j.

$$TFIDF_{ij} = t_{ji} \times g_i \times d_i, \tag{16}$$

A term-document matrix is formed (Table 2), the elements of which contain the reduced frequencies $TFIDF_{ij}$ of the occurrence of the i-th physical effect PE_i in the j-th patent document Pat_j.

Table 2. Matrix of reduced frequencies of occurrence of physical effects in the documents

Term	Document			
	Pat_1	Pat_2	…	Pat_N
PE_1	$TFIDF_{11}$	$TFIDF_{21}$	...	$TFIDF_{N1}$
PE_2	$TFIDF_{12}$	$TFIDF_{22}$	...	$TFIDF_{N2}$
...	...	...	...	
PE_M	$TFIDF_{1M}$	$TFIDF_{2M}$	...	$TFIDF_{NM}$

Since each patent Pat stores information about a technical object (TO) as an attribute, then we will create a «TO-PE» matrix based on physical effects PE_i, for which the coefficient $TFIDF_{ji}$ of occurrence in the j-th patent document Pat_j is greater than a certain threshold value equal to 0.9.

The singular decomposition of the term-document matrix is carried out, and all terms from the matrix Ak will be the reduced space of physical effects (PE), i.e. the

most significant PE for the patent array will be automatically determined from the whole set of PE.

Thus, the method of latent-semantic analysis allows you to submit patent documents as vectors distributed in space:

(a) technical functions $Pat_j = (TechFunc_1, TechFunc_2, \ldots TechFunc_M)$,
(b) physical effects $Pat_j = (PE_1, PhE_2, \ldots PE_Q)$.

To build a database performed by the physical effects of technical functions, it is necessary to establish a connection between the physical effects (PE) and technical functions (TF) distributed in the common patent space $(Pat_1, Pat_2, \ldots Pat_N)$.

The construction of the matrix of semantic connections "PE –TF" is carried out with two reduced and cleared of "noise" term documentary matrices. Since in term documentary matrices each term (physical effect or technical function) is a row vector, then the semantic connections between any two terms (PE_i and $TechFun_j$) can be interpreted as proximity or distance the vectors corresponding to these terms, using any known measures of proximity or distance.

An assessment was made of the effectiveness of methods for comparing patent vectors distributed over a term space [12]:

– storage vectors by two different methods (storage in HDFS distributed file system, storage in a PostgreSQL database) [13];
– comparing the resulting vectors by four different methods [14] (based on the standard deviation of the vectors, based on the element-by-element comparison of vectors, based on the cosine method, based on the comparison of the lengths of vectors) to search for patents-analogs.

The efficiency test of the methods (Table 3) was carried out on a test dataset generated on the basis of the patent databases of Rospatent and the USPTO. For each patent, patents from his citation list have been added to the test array as relevant. Irrelevant "noise" patents are also included in the test set.

Table 3. Results of efficiency test of the methods.

№	Comparison of vectors	Storage	T, s	Precision	Recall
1	Cosine method	HDFS	0,5	0,98	0,87
2	Comparison of the lengths of vectors	HDFS	0,7	0,85	0,68
3	The standard deviation of vectors	HDFS	0,7	0,95	0,81
4	The element-by-element comparison of vectors	HDFS	1,1	0,88	0,75
5	Cosine method	PostgreSQL	0,8	0,98	0,87
6	Comparison of the lengths of vectors	PostgreSQL	1,3	0,85	0,68
7	The standard deviation of vectors	PostgreSQL	1,2	0,95	0,81
8	The element-by-element comparison of vectors	PostgreSQL	1,6	0,88	0,75

$$T = t_{end} - t_{start},\ precision = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|},\ recall = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|},$$

where t_{end} – search end time, t_{start} – search start time, D_{rel} – set of relevant documents in the database, D_{retr} – set of documents found by the system.

According to the test results, we can conclude: the cosine method is the most effective method for comparing vectors.

Since we are only interested in the strongest and most stable semantic relations "PE-TF", we will not take into account the proximity values of two vectors, representing the i-th physical effect (PE_i) and j-th technical function ($TechFun_j$) distributed in a common space of patents ($Pat_1, Pat_2, \ldots Pat_N$), below a certain threshold value of 0.85, determined empirically.

5 Conclusion

Authors developed the method for extraction the physical effects descriptions from the patents of USPTO and RosPatent databases, and the method for extracting of technical functions from Natural Language documents including patent texts.

The method of automated construction of a matrix of physical functions performed by physical effects is based on the detection of latent dependencies in the consolidated matrix "Physical Effects – Technical Functions". The consolidated matrix is formed from two term-document matrices: (a) the first matrix describes the frequency of occurrence (TFIDF) of each term (technical functions in SAO form) in all patent documents, (b) the second matrix describes the frequency of occurrence of physical effects in all patens.

Acknowledgments. The reported study was funded by RFBR (research project 18-07-01086), RFBR and Administration of the Volgograd region (projects 19-47-340007, 19-41-340016).

References

1. Orloff, M.: Inventive Thinking through TRIZ: A Practical Guide, p. 352. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-33223-7
2. Vayngolts, I., Korobkin, D., Fomenkov, S., Golovanchikov, A.: Synthesis of the physical operation principles of technical system. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.) CIT&DS 2017. CCIS, vol. 754, pp. 575–588. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_42
3. Korobkin, D., Fomenkov, S., Kravets, A.: Methods for extracting the descriptions of sci-tech effects and morphological features of technical systems from patents. In: IISA 2018 (2018). https://ieeexplore.ieee.org/document/8633624
4. Davydova, S., Korobkin, D., Fomenkov, S., Kolesnikov, S.: Modeling of new technical systems using cause-effect relationships. In: IISA 2018 (2018). https://ieeexplore.ieee.org/document/8633683
5. Park, H., Yoon, J., Kim, K.: Identifying patent infringement using SAO based semantic technological similarities. Scientometrics **90**, 515 (2012)

6. Korobkin, D., Fomenkov, S., Kravets, A., Kolesnikov, S.: Prior art candidate search on base of statistical and semantic patent analysis. In: Xiao, Y., Abraham, A.P. (eds.) Multi Conference on Computer Science and Information Systems, pp. 231–238 (2017)
7. Manning, C., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
8. Mel'čuk, I.: Dependency Syntax Theory and Practice. SUNY, New York (1988)
9. Yufeng, D., Duo, J., Lixue, J.: Patent Similarity Measure Based on SAO Structure. Chin. Sentence Clause Text Inf. Process. **30**(1), 30–36 (2016)
10. Korobkin, D., Fomenkov, S., Kolesnikov, S., Lobeyko, V., Golovanchikov, A.: Modification of physical effect model for the synthesis of the physical operation principles of technical system. In: Kravets, A., Shcherbakov, M., Kultsova, M., Shabalina, O. (eds.) CIT&DS 2015. CCIS, vol. 535, pp. 368–378. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23766-4_29
11. Fomenkova, M., Korobkin, D., Fomenkov, S.: Extraction of physical effects based on the semantic analysis of the patent texts. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.) CIT&DS 2017. CCIS, vol. 754, pp. 73–87. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_6
12. Korobkin, D., Fomenkov, S., Kravets, A., Kolesnikov, S.: Methods of statistical and semantic patent analysis. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.) CIT&DS 2017. CCIS, vol. 754, pp. 48–61. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_4
13. Ustugova, S., Parygin, D., Sadovnikova, N., Yadav, V., Prikhodkova, I.: Geoanalytical system for support of urban processes management tasks. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.) CIT&DS 2017. CCIS, vol. 754, pp. 430–440. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_31
14. Ustugova, S., Parygin, D., Sadovnikova, N., Finogeev, A., Kizim, A.: Monitoring of social reactions to support decision making on issues of urban territory management. In: Procedia Computer Science: Proceedings of the 5th International Young Scientist Conference on Computational Science, YSC 2016, Krakow, Poland, 26–28 October 2016, vol. 101, pp. 243–252. Elsevier (2016). https://doi.org/10.1016/j.procs.2016.11.029