Cybernetics and Systems Analysis, Vol. 50, No. 3, May, 2014

CYBERNETICS

DETERMINING SEMANTIC VALENCES OF ONTOLOGY CONCEPTS BY MEANS OF NONNEGATIVE FACTORIZATION OF TENSORS OF LARGE TEXT CORPORA

A. V. Anisimov,[a†] O. O. Marchenko,[a‡] and T. G. Vozniuk[a] UDC 681.3

Abstract. *This paper describes a method for automatic detection of semantic relations between concept nodes of a networked ontological knowledge base by analyzing matrices of semantic-syntactic valences of words. These matrices are obtained by means of nonnegative factorization of tensors of syntactic compatibility of words. Such tensors are generated in the course of frequency analysis of syntactic structures of sentences taken from large text corpora of English Wikipedia and Simple English Wikipedia entries.*

Keywords: *automatic knowledge acquisition, corpus linguistics, ontology, nonnegative tensor factorization.*

INTRODUCTION

In recent years, nonnegative tensor factorization (NTF) becomes a technology actively used in fields such as information retrieval, image processing, natural language processing, machine learning, and other adjacent areas. It is one of the most promising approaches to the detection and analysis of interrelations and relations in data arrays containing objects of N different types and classes. In computer linguistics, an N -dimensional tensor is realized as a multidimensional array of data obtained as a result of frequency analysis of large text corpora. A tensor is a convenient structure for the representation of data of higher orders. The factorization of an N-dimensional tensor with decomposition rank k forms N two-dimensional matrices consisting of k column vectors representing the mapping of each separate tensor dimensionality onto k factorized dimensionalities of a latent semantic space. This is a unique means for modeling and detecting interrelations between linguistic variables in an array of N-dimensional data.

Tensor factorization is a multilinear analog of singular value decomposition (SVD) of matrices which is used in latent semantic analysis (LSA) for processing two-dimensional data arrays. In a sense, the method of nonnegative tensor factorization can be called an n-dimensional generalization of latent semantic analysis. The structure obtained as a result of factorization of a tensor can be compared with a multilayer neural network consisting of N layers that represent sets of objects of N types and a hidden switching layer consisting of many nodes with different weight coefficients. The latter layer models interrelations between objects of N types and combines N layers into a unified neural network.

At present, nonnegative tensor factorization is a promising method for solving problems of computer linguistics as evidenced by numerous works in this direction [1–4].

Of particular interest are [1, 2] in which models of tensor representation of data on the frequency of different types of syntactic combinations of words in sentences are described, for example, three-dimensional combinations of the type subject — verb — object, four-dimensional combinations of the type subject — verb — direct_object — indirect_object, or other syntactic combinations whose length does not exceed the tensor dimensionality equal to N. In a tensor, to each

[a]Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, [†]*ava@unicyb.kiev.ua*; [‡]*rozenkrans@yandex.ua*.
Translated from Kibernetika i Sistemnyi Analiz, No. 3, pp. 3–16, May–June, 2014. Original article submitted July 25, 2013.

1060-0396/14/5003-0327

dimensionality corresponds some fixed member of sentence, i.e., a subject, a predicate, an object, a definition, an adverbial modifier, etc. *N*-dimensional tensors contain estimates for the frequency of using combinations of different collections of words in natural language sentences in text corpora. In this case, syntactic positions of words are taken into account. After processing large text corpora and accumulating a considerable amount of data, an *N*-dimensional array of description of the behavior of lexical units in sentences of this language is formed, i.e., for the set of words described in the tensor, syntactic relations, words belonging to them, and properties representing frequencies of belonging these words to such relations are described. At the same time, these relations are not binary but multidimensional (*N* is a maximum dimensionality of the relations). Then the stage of nonnegative factorization of the obtained tensor begins. This factorization leads to a considerable transformation of the model of data representation. A multidimensional tensor is initially sparse and huge. Each of *N* axes of the syntactic space contains tens of thousands or hundreds of thousands of points-words. After tensor factorization, its data are represented in the form of *N* two-dimensional matrices consisting of *k*-dimensional vectors (where the value of *k* is much smaller than the number of points-words in each of *N* tensor dimensionalities). The parameter *k* is the degree of factorization, dimensionality of the latent semantic space, and number of feature dimensionalities in it. In addition to a much more compact and convenient representation of a data array, the opportunity is afforded for fast computing the estimated probability of any possible combination of words in different syntactic constructions of a sentence. This can be carried out by the computation of the sum of products of components of *N k*-dimensional vectors corresponding to these words selected from the matrices corresponding to their syntactic positions. For example, in order to check the probability of the sentence "a cook fries a chicken," a *k*-dimensional vector *s* that corresponds to the noun "cook" should be found in the matrix SUBJECT, and then the *k*-dimensional vector *v* corresponding to the verb "fries" should be found in the matrix VERB. Then the *k*-dimensional vector *do* corresponding to the noun "chicken" should be found in the matrix DIRECT_OBJECT, and the sum of products of the corresponding components of these three vectors is computed as follows:

$$x_{svdo} = \sum_{i=1}^{k} s_i v_i do_i \quad \text{(for the case when } N = 3\text{)},$$

where s_i is the *i*th element of the vector *s*, v_i is the *i*th element of the vector *v*, and do_i is the *i*th element of the vector *do*.

If the resulting sum exceeds some threshold level, then the conclusion is drawn that this sequence of words is present in the corresponding sentence. The computation of this estimate for the word combination "a chicken fries a cook" leads to the conclusion that the existence of this variant is impossible.

This model makes it possible to automatically extract linguistic structures from text corpora with much success; examples of such structures are selectional preferences [1] in sentences and verb subcategorization frames [2] which combine data on semantic and syntactic properties of relationships between verbs and their arguments-nouns in sentences of a natural language. This implies the possibility of automatic extraction of semantic relations such as Fillmore semantic "role" cases [5] from the obtained latent semantic space. Semantic role cases form a system of multitype semantic relationships between concepts-nodes located in the hierarchical network of some ontology, for example, the lexico-semantic Wordnet base [6]. Determining semantic relationships between concepts of ontologies during processing and analyzing decomposed tensors of text corpora will make it possible to automate the filling of ontological knowledge bases with contents.

Vectors from matrices of a decomposed tensor are descriptions of frequency distributions of lexemes in sequences of words of sentences.

We call a vector taken from the matrix of a decomposed linguistic tensor and belonging to some word a vector of semantic-syntactic valence of this word in the syntactic position of the corresponding matrix.

The main intricacy is that, in constructing a tensor of semantic-syntactic compatibility of lexemes, basic objects being investigated and analyzed are lexemes, i.e., words that are ambiguous by their very nature. The vector representation of semantic-syntactic valences of any word *W* that is determined by the vector corresponding to it in the matrix of a decomposed tensor is intrinsically the summed component addends of vectors of separate different semantic values of this word *W*, i.e., concepts $Sw_1, Sw_2, \ldots, Sw_t$ in some ontology. In this work, the following problem is set: based on the valence vector $(v_1, v_2, \ldots, v_k)$ of a word *W*, obtain component addends of valence vectors

$$(v_{11}, v_{12}, \ldots, v_{1k}), (v_{21}, v_{22}, \ldots, v_{2k}), \ldots, (v_{t1}, v_{t2}, \ldots, v_{tk})$$

for each of its *t* values. The valence vector of a fixed value, i.e., an ontology concept, is an implicit description of its semantic relations with other concepts of the corresponding ontological knowledge base.

This paper proposes a new method for determining semantic relations between concepts, i.e., synsets of WordNet. It is realized by means of analyzing decomposed tensors formed in processing corpora of entries of the English Wikipedia [7] and Simple English Wikipedia [8] with splitting vectors of semantic valences of words into component vectors of semantic valences of their values and accurately assigning split vectors to the corresponding conceptual nodes of the WordNet ontology. The proposed method is tested for the accuracy of partitioning vectors of semantic valences (VSVs) of words into component addends of **VSVs** of concepts, i.e., values of these words, and also for the accuracy of their assignment to WordNet synsets. The main advantage of this method is the full automation of the process of finding new semantic relations between concepts of semantic knowledge bases in analyzing large text corpora. Despite the fact that relations are specified implicitly, i.e., by vectors of semantic valences, it is this form of notation that makes it possible to solve classical problems of computer linguistics such as word sense disambiguation, measurement of semantic proximity of words, semantic analysis of texts using the technique of constructing the shortest distances in an ontology network, and many others.

METHODS OF AUTOMATIC REPLENISHMENT OF LINGUISTIC DATABASES BY MEANS OF ANALYZING AND PROCESSING TEXTS

At present, the automatic extraction of linguistic data from text corpora is a rather popular direction in computer linguistics. Methods of automation of obtaining different types of information are created, in particular, selective preferences [1], proper names [9], multilinguistic relationships [10], syntactic rules [11–13], collocations [14], and other language structures of data.

Modern methods of automation of extension and replenishment of ontologies by new knowledge on concepts and relationships between them can be divided into the following main groups.

Methods based on properties of distributions of words. This approach consists of investigation and detection of data on joint distribution of words in texts with a view to computing semantic distances between the concepts represented by these words. The constructed metrics can be used for the clusterization of concepts [15], formal conceptual analysis [16], and also for the purposes of classification of words in existing ontologies [17–20]. These methods are used to replenish ontologies by new concepts. Works on the investigation of subordination relations between words in a sentence are also topical since they, together with various heuristics, make it possible to extract non-taxonomic relations in ontologies [21].

Methods based on the extraction and selection of patterns. These methods use lexical and lexico-semantic patterns for detecting ontological and non-taxonomic relations between concepts in texts. In [22–24], regular expressions for the extraction of hyponymic and meronymic relations are manually defined. The number of errors in performing these methods equals approximately 32% [25]. There also are systems that combine methods of analysis of distributions and the pattern-based approach to the detection of hypernymic and other non-taxonomic relations between concepts.

Methods based on the analysis of texts of entries of explanatory dictionaries and thesauruses of electronic encyclopedias. An advantage of these methods [26–29] is that the standard structure of entries and also the relations-relationships that exist between entries can be used for the organization of data structures in ontologies. Definitions of concepts include the most significant body of information on them and also main relationships with other concepts, which is a very convenient notation for machine translation into the form of an ontological database. Recently, to work in this direction, the lexico-semantic ontology of WordNet and the global electronic encyclopedia Wikipedia are often chosen in the capacity of primary data resources. A number of methods are developed for the assignment of Wikipedia entries as new conceptual nodes to WordNet synsets for increasing the coverage of the semantic field of concepts in WordNet [30–32]. In this case, the main problem of finding a place in the WordNet taxonomy for integrating new nodes corresponding to Wikipedia entries is solved. When we are dealing with semantically identical Wikipedia and WordNet nodes, the problem is solved rather simply in contrast to the case when the node of a Wikipedia entry having no existing analog in WordNet should be integrated into the WordNet hierarchy. Then the WordNet node nearest in meaning should be found, and the new node should be assigned to this node as its semantic descendant.

The following key stage consists of extracting non-taxonomic relations between network concepts by analyzing Wikipedia texts. In [31], a method is proposed for the extraction of non-taxonomic semantic relations between concepts. The method lies in sequential processing of the Simple English Wikipedia and extraction of all inputs (entries) after which the disambiguation stage is followed, and then relationships are established between the corresponding nodes-inputs. Disambiguation consists of accurate assignment of each separate input of the Simple English Wikipedia to a concrete synset of WordNet. Next, for each input, its definition is analyzed and words present in the text of the definition are detected. If these words have relationships-relations with a word-input in the WordNet base, then the structure of these sentences is analyzed and the syntactic pattern corresponding to this pair of words is assigned to the type of the corresponding relation

existing in WordNet. Then the syntactic patterns collected at the previous stage and assigned to some type of relations are compared, and the patterns having the same structure and configuration are automatically generalized. Later on, generalized patterns are used for extracting interconceptual relations (nonregistered earlier in WordNet) from Wikipedia texts and subsequently, after detection, are added to the base. Thus, more than 2600 new relations initially absent in WordNet were determined. The accuracy of determining such relations dependent on types of these relations and the degree of generalization chosen for patterns is about 60–70% for the best combinations.

In [32], a method of assignment of Wikipedia entries to WordNet synsets is described. To disambiguate Wikipedia words-inputs, the Personalized PageRank method with an adjustable threshold level was used. In this case, for a test sample, the F1 score of the accuracy of assignment reaches 0.78.

The algorithm proposed in this article differs from the above-mentioned methods in that it uses nonnegative factorization of a multidimensional tensor containing data on word compatibility in a language at different syntactic positions in sentences of texts taken from the corpus of Wikipedia entries. This method is applied to the creation of the vector space for the description of semantic valences of given words. Then vectors of semantic valences of words are decomposed into vectors of semantic valences of separate values of these words with assignment to the WordNet synsets corresponding to them. In this way, an implicit description of semantic relations between conceptual WordNet nodes is formed. To solve this concrete problem, nonnegative factorization of linguistic tensors was not used so far. The comparison of the presented method with other existing approaches within the framework of this paper seems to be a complicated problem since methods of assigning, for example, Wikipedia entries to WordNet nodes apply radically other approaches and an absolutely other problem statement. A simple comparison of figures for the estimates for the accuracy of "assignment" is undoubtedly not correct in view of an absolutely other statement of the problem of assigning not Wikipedia entries but split vectors of semantic valences to WordNet synsets for all the values of some word that are mentioned in the corpus. Nonnegative tensor factorization is widely used to solve other problems that are not algorithmically similar to the problems described in this article. Therefore, the comparison in the line of using algorithms of nonnegative tensor factorization can be mainly performed in terms of time characteristics, which is not the objective of this work.

METHODOLOGY OF COMPOSITION OF A TENSOR FOR A TEXT CORPUS

The methodology presented in [2] was used as basic in composing N-dimensional tensors of text corpora. In this case, some details of the algorithm were modified in view of the specificity of the formulated problem on the investigation and extraction of semantic relations between concepts, i.e., nodes of the ontological network of a knowledge base.

At the initial phase, a corpus passes the stage of syntactic analysis of sentences of texts with the help of the Stanford Parser [33].

Next, parsing the syntactic tree, the post-syntactic parser extracts the following members of the current sentence: the main verb fixed by the root (ROOT-0, ***verb***), subject nsubj (verb, ***noun***), direct object dobj (verb, ***noun***), indirect object iobj (verb, noun), the noun in the prepositional phrase prep_during (verb, ***noun***), prep_on (verb, ***noun***), prep_in (verb, ***noun***) etc., and an interverbal connective xcomp (verb, ***verb*1**). Thus, in analyzing a sentence, the system finds lexemes in the corresponding syntactic positions, fills the tuple (root-verb, nsubj, dobj, iobj, prep _, xcomp, count) of the sentence with these words, and, in this case, writes the noun into prep_ together with its preposition. If some syntactic position is absent in the sentence, then it is filled with the empty word symbol ∅. Only tuples with at least three nonzero fields are entered in a tensor. The position count stores the number of occurrences of such a lexical combination in the corpus. The first six elements of tuples form space coordinates, and the seventh element stores values of the frequency of combinations. As a result, a 6-dimensional tensor of lexical combinations is formed at these syntactic positions. The question of determining lexemes-fillers of fields of a tuple deserves special consideration. It is generally reduced to the problem of analyzing the noun group at a concrete syntactic position and determining a word or a phrase that will be a correct filler for a tuple field. In the case of a complicated noun group, the algorithm determines whether this name group contains the name of a Wikipedia entry at its head position (for example, ***Sulfuric Acid*** or ***Eiffel Tower***). If it contains such a name, then this found Wikipedia entry name is the filler, and if such a name is absent, then the head of this noun group is used.

In [2], a more complicated syntactic model of composition of a tensor with a large number of syntactic patterns of relationships is used and, as a result, a 9–12-dimensional tensor is formed. In this paper, we do not set the problem of constructing a maximally complicated structure. The priority was the increase in the probability of correctly determining

syntactic relations and decrease in the noise level. Moreover, an objective of this publication is the development of methods for determining a semantic relationship of some dimensionality between concepts, i.e., WordNet synsets, and, at this stage, we do not consider the problem of exhaustive detection of all semantic relations described in a corpus. The preference is given to the accuracy of determining the found semantic relations. In any case, the complexity of the syntactic model and increase in tensor dimensionality are questions of technical implementation, and they will increase in further investigations.

TENSOR DECOMPOSITION

In this article, by the factorization of an N-dimensional tensor T we understand the construction of its representation in the form of the sum of exterior products of N vectors. In linear algebra, the exterior product usually is a denotation of the tensor product of two vectors. The result of application of the exterior product to a pair of vectors is a matrix. For example, the exterior product of a four-dimensional vector u and a three-dimensional vector v is the following matrix:

$$u \circ v^{\mathrm{T}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} [v_1 \; v_2 \; v_3] = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \\ u_4 v_1 & u_4 v_2 & u_4 v_3 \end{bmatrix}.$$

Thus, the exterior product of three vectors yields a 3-dimensional matrix, i.e., a 3-dimensional tensor.

In this case, it is necessary to construct a representation $T = \sum_{i=1}^{k} x_{1i} \circ x_{2i} \circ \ldots \circ x_{ti}$ in which the values of all elements are maximally close to the initial values in the tensor T. Then k vectors x_{1i} are collected in the matrix X_1, k vectors x_{2i} are collected in the matrix X_2 etc., and k vectors x_{ti} are collected in the matrix X_t. Each of these matrices will consist of vectors of length (dimension) k.

The 6-dimensional tensor described in the previous section must be represented in the form of the sum of k exterior products of sextuples of vectors. The vectors of this sum must be added together in the form of the corresponding six matrices each of which will represent the mapping of the set of lexemes located at a definite syntactic position onto the set of k factor dimensions of the latent semantic space of semantic-syntactic relations between words of the text corpus.

To decompose a tensor, the method of nonnegative tensor factorization is used. It is similar to the parallel factor analysis under the constraint that all data items must be nonnegative. Parallel factor analysis is a multilinear analog of singular decomposition of matrices that is used in latent semantic analysis [34]. The essence of the method is the minimization of the sum of squares of the difference between the original tensor and its factored model. For an N-ary tensor $T \in R^{D_1 \times D_2 \times \ldots \times D_N}$, its objective function is defined as follows:

$$\min_{x_{1i} \in R^{D_1}, x_{2i} \in R^{D_2}, \ldots, x_{Ni} \in R^{D_N}} \left\| T - \sum_{i=1}^{k} x_{1i} \circ x_{2i} \circ \ldots \circ x_{Ni} \right\|_F^2, \quad (1)$$

where k is the dimensionality of the factored model and $\circ$ is the exterior product.

For nonnegative factorization, the following constraints on the nonnegativity of values of elements are added:

$$\min_{x_{1i} \in R_{\geq 0}^{D_1}, x_{2i} \in R_{\geq 0}^{D_2}, \ldots, x_{Ni} \in R_{\geq 0}^{D_N}} \left\| T - \sum_{i=1}^{k} x_{1i} \circ x_{2i} \circ \ldots \circ x_{Ni} \right\|_F^2. \quad (2)$$

The result of execution of the algorithm is the representation of a tensor in the form of N matrices describing the mapping of each of tensor dimensionalities onto k factor dimensions of the latent semantic space. As a rule, an NTF model is fitted by the least squares method. At each iteration, $N-1$ dimensionalities are fixed, and the Nth dimensionality is fitted by the least squares method. The process continues until its convergence. The number of factor dimensions of the latent semantic space was taken equal to $k = 150$. In [2], it was experimentally established that it is exactly this value that provides the best results of factorization. To solve problems of nonnegative factorization of a 6-dimensional tensor for the text corpus of Wikipedia entries, the algorithm of parallel factorization PARAFAC [35] was used. To decompose tensors, our own

software implementation of the algorithm was developed. In this case, we managed to reach a considerable acceleration of the process of solving the problem owing to the parallelization of computations on a graphic card according to a technology similar to the technology from [36].

Example of generating of matrices of a factored tensor of a text corpus. The following text corpus is given:

"Mother washed the frame. Victor likes football. Julia likes flowers. Mary listens to an opera. Julia does her homework. Mother planted flowers. Mary likes opera. Mother waters flowers. Julia watches football. Julia likes flowers. Julia likes football. Mary likes flowers. Victor likes opera. Mother employed a cook. The cook fries a chicken. Mother employed a cook. A chicken pecks grains. Mother employed a cook. A chicken pecks a chicken."

To each word in each syntactic category, we assign a unique number *Id* corresponding to its coordinate on the corresponding axis of the three-dimensional space of the tensor.

Subject (1 — Mother, 2 — Victor, 3 — Mary, 4 — Julia, 5 — Cook, 6 — Chicken)

Predicate (1 — washed, 2 — likes, 3 — listens to, 4 — does, 5 — planted, 6 — waters, 7 — watches, 8 — fries, 9 — pecks, 10 — employed)

Object (1 — frame, 2 — football, 3 — opera, 4 — homework, 5 — flowers, 6 — chicken, 7 — grains, 8 — cook)

As a result of parsing the text, the 3-dimensional tensor is generated with the following nonzero elements:

$$T[1,1,1]=1,\ T[2,2,2]=1,\ T[4,2,5]=2,\ T[3,3,3]=1,\ T[4,4,4]=1,\ T[1,5,5]=1,$$

$$T[3,2,3]=1,\ T[1,6,5]=1,\ T[4,7,2]=1,\ T[4,2,2]=1,\ T[4,2,5]=1,\ T[3,2,5]=1,$$

$$T[2,2,3]=1,\ T[1,10,8]=3,\ T[5,8,6]=1,\ T[6,9,7]=1,\ T[6,9,6]=1.$$

All other elements of the tensor are equal to zero and, hence, a sparse tensor is obtained.

As a result of nonnegative factorization of this tensor, its decomposition into the following sum of products of triples of vectors is obtained ($k=11$, and it is this value that turned out to be optimal since it most accurately brings the model closer to the initial values of the input tensor T):

$$T=\sum_{i=1}^{11}\lambda_i x_i\circ y_i\circ z_i=\lambda_1 x_1\circ y_1\circ z_1+\lambda_2 x_2\circ y_2\circ z_2+\ldots+\lambda_{11}x_{11}\circ y_{11}\circ z_{11},$$

where λ_i is the weight coefficient of the ith factor dimension.

In this example, the algorithm computed the values $\lambda_1=2$, $\lambda_9=3$, and all the other $\lambda_i=1$ for $i\neq 1$ and $i\neq 9$.

Then the vectors $x_1, x_2, \ldots, x_{11}$ form the following matrix:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Mother	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
Victor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Mary	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Julia	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0
Cook	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Chicken	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0.

The vectors $y_1, y_2, \ldots, y_{11}$ form the following matrix:

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}
Washed	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Likes	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
Listens to	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Does	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Planted	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Waters	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Watches	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
Fries	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Pecks	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Employed	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0.

The vectors $z_1, z_2, \ldots, z_{11}$ form the following matrix:

	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}
Frame	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Football	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
Opera	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Homework	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Flowers	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Chicken	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
Grains	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Cook	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0.

Assume that it is required to obtain the corpus frequency estimate for the word combination “The cook fries a chicken.”

From the matrix Subjects X, the following vector of the semantic-syntactic valence of the word “Cook” is taken:

Cook (0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0).

From the matrix Predicates Y, the following vector of the semantic-syntactic valence of the word “fries” is taken:

Fries (0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0).

From Objects Z, the following vector of semantic-syntactic valence of the word “chicken” is taken:

Chicken (0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0).

Computing by the formula $E = \sum_{i=1}^{11} \lambda_i x_i y_i z_i$, we obtain the frequency estimate for the word combination $E = 1$ (intersection of nonzero values only along the tenth coordinate). It testifies to the fact that such a sentence is contained in the text corpus only once. This means that this lexico-semantic relation (cook is the subject, fries is the predicate, and chicken is the object) is admissible, i.e., corresponds to language practice. For the sentence “The chicken fries the cook,” the frequency estimate is $E = \sum_{i=1}^{11} \lambda_i x_i y_i z_i = 0$, which testifies to the fact of absence of the description of such a relation in the input corpus. It is easily verified that all initial values of the tensor T of the input text corpus are similarly computed.

The described method of factorization of text tensors is efficient for large text corpora with a wide scope of subjects when vectors of the matrix of a decomposed tensor implicitly contain information on millions and billions of semantic relations between tens and hundreds of thousands of words. The obtained matrices of tensor factorization are considered as a valuable source of information for solving various problems of linguistic analysis and form a database of lexico-semantic relations between words of a natural language.

ALGORITHM OF SPLITTING VECTORS OF SEMANTIC-SYNTACTIC VALENCES OF WORDS INTO COMPONENT ADDENDS OF VALENCE VECTORS OF THEIR DIFFERENT VALUES

After the factorization of the constructed 6-dimensional tensor of the corpus of Wikipedia entries, the following six matrices were obtained: ROOT_VERB, NSUBJ, DOBJ, IOBJ, PREP _, and XCOMP that consist of vectors of length k ($k = 150$). Each column vector of these matrices corresponds to some word or phrase. These vectors describe the semantic-syntactic behavior of words, namely, syntactic positions of words and also relations to which some word belongs. By analogy with chemical terminology, we call these vectors vectors of semantic-syntactic valences (**VSVs**) of words. By their very nature, words are ambiguous, i.e., each of them, as a rule, has several values (meanings). Thus, the vector of a word is the sum of vectors of all values of the word. To one word may correspond some vectors from different matrices corresponding to different syntactic positions, and the problem of splitting each of these vectors is solved independently. The developed algorithm of splitting the **VSV** of a word onto the set of **VSVs** of all its values, i.e., WordNet synsets, is as follows: a vector is given whose semantic valence equals V, whose length equals k, and that corresponds to some word w in NSUBJ (the

method works similarly in any of six matrices). To the noun w corresponds t values or synsets in WordNet. It is required to divide V into component addends $V_1, V_2, \ldots, V_t$ corresponding to these t synsets.

The algorithm is specified as follows:

for $i = 1$ to k do
begin
if $V[i] \neq 0$ then
for $j = 1$ to t do
begin
{It is necessary to determine the synset out of t synsets to which belongs the ith value $V[i]$. It can belong to one of these synsets or to several of them, and then $V[i]$ should be decomposed into the following sum:
$V_1[i] + V_2[i] + \ldots + V_t[i]$ }
$S = 0$;
Take the jth synset and write the number of words contained in it in r;
for p:= 1 to r do
begin
take the vector W that corresponds to the pth word of the jth synset from NSUBJ;
$S = S + W[i]$;
end;
$S_{\text{mean}} = S / r$; {the average coefficient of assignment of $V[i]$ to words of the jth synset is computed}
Obtain the direct ancestor of the jth synset, i.e., Anc, and, by analogy with S_{mean}, compute the average coefficient of assignment of $V[i]$ to its words $S_{\text{Anc-mean}}$;
Obtain the set of descendants of the jth synset at unit distance {Offs} and, by analogy with S_{mean}, compute the average coefficient of assignment of $V[i]$ to each descendant $S_{\text{Offs-mean}}$ and select the maximum $S_{\text{MaxOffs-mean}}$ among them;
Compute the total estimate for the jth synset
$S_{\text{Fin}}[j] = C_1 * S_{\text{mean}} + C_2 * S_{\text{Anc-mean}} + C_3 * S_{\text{MaxOffs-mean}}$;
{C_1, C_2, and C_3 are coefficients of priorities chosen empirically}
end;
$V_1[i], V_2[i], \ldots, V_t[i]$ are determined from the following system of equations:

1. $V_1[i] = S_{\text{Fin}}[1] * X$; $V_2[i] = S_{\text{Fin}}[2] * X$; …; $V_t[i] = S_{\text{Fin}}[t] * X$;
2. $\sum_{j=1}^{t} S_{\text{Fin}}[j] * X = V[i]$; determine the value of $X = \dfrac{V[i]}{\sum_{j=1}^{t} S_{\text{Fin}}[j]}$;

for $j = 1$ to t do if $S_{\text{Fin}}[j] < R$ then $V_j[i] = 0$ else $V_j[i] = S_{\text{Fin}}[j] * X$;
{R is a threshold level fitted experimentally}
end;

EXPERIMENTAL RESULTS

At the beginning of the experiments, optimal values of the coefficients C_1, C_2, and C_3 of the algorithm described above were fitted. Their approximate values were first assigned empirically. Then the algorithm processed a sample consisting of about 3600 **VSVs** of words uniformly selected from all the matrices of the factorized tensor formed in the course of processing texts of the corpus of Wikipedia entries (approximately 600 **VSVs** from each matrix). Worthy of mention is the fact that, in forming this sample, the obligatory fulfillment of the following condition was provided: for each word from the sample, there guaranteedly was at least one input, i.e., a WordNet synset. After applying the algorithm, a sample of **VSVs** of concepts was obtained with the assignment to concrete WordNet synsets to estimate the correctness of splitting of the **VSV** of a word into the set of **VSVs** of all its different values. To automate the process of estimating the accuracy of splitting vectors of words and assigning component addends of vectors to concepts, i.e., WordNet synsets, a program was developed. Based on a collection of the obtained **VSVs** of WordNet synsets, it generates the set of all possible sequences of words, i.e., sentences with their participation, that are matched with values in these k-dimensional **VSVs** of synsets. Then, with the help of this program, experts carried out the analysis of the correctness of the formed phrases with correction of errors by replacing incorrectly chosen synsets in the generated sentences by correct synsets in places of errors. Thus, a training sample was formed that made it possible to fit the optimal coefficients C_1, C_2, and C_3 for the algorithm of splitting and assignment of **VSVs**.

TABLE 1

Matrices	Wikipedia	Simple Wikipedia
VERB	79.84	73.54
NSUBJ	87.17	81.21
DOBJ	85.62	80.17
IOBJ	86.08	79.09
PREP_	83.45	75.61
XCOMP	73.91	69.08

The optimization problem of fitting C_1, C_2, and C_3 was solved for maximizing the correlation of the constructed set of possible sequences of words in sentences corresponding to values in VSVs obtained by the algorithm of splitting and assignment with the training sample of correct sequences of words in sentences constructed by experts. To solve this problem, the simulated annealing method [37] was used, i.e., a probabilistic heuristics for solving global optimization problems. As the function for maximization, the Spearman correlation coefficient was used.

After fitting optimal values of C_1, C_2, and C_3, the algorithm split and assigned **VSVs** of synsets from the matrices obtained as a result of nonnegative tensor factorization of the corpora of Wikipedia and Simple Wikipedia entries to WordNet. To estimate the quality of operation of the algorithm, by analogy with the stage of formation of the above-mentioned training set, a test sample of **VSVs** of concepts, i.e., Wordnet synsets was formed (its structure was absolutely different from the structure of the training set). This sample consists of approximately 5400 vectors normally distributed over WordNet and assigned to synsets, i.e., approximately 450 vectors from each matrix for each of corpora. With the help of the above-mentioned program of generating sequences of words "allowed" in **VSVs**, the accuracy (semantic correspondence) of the assignment of **VSVs** of concepts to concrete WordNet synsets was estimated. Estimates for the accuracy of operation of the algorithm of splitting valence vectors of words into composite component vectors of their different values and for their assignment to WordNet synsets are presented in Table 1.

An analysis of the obtained test data clearly discloses some trends in estimates for the accuracy of the developed algorithm. First, the decrease in accuracy is appreciable when VSVs of verbs are split (the matrices VERB and XCOMP) in comparison with the accuracy of processing matrices of nouns (NSUBJ, DOBJ, IOBJ, and PREP _). This decrease in the accuracy of the algorithm in processing vectors of verbs is explained by the nature itself of verbs. Verbs form a relatively small class of lexis in comparison with nouns. Moreover, on the average, verbs have much more values per one lexeme in comparison with, for example, nouns. Thus, the problem of splitting VSVs of verbs with further assignment to WordNet synsets is objectively a much more complicated problem. Its solution requires the analysis and processing of text corpora of considerably larger sizes (as to the number of texts and coverage of various subjects) than in the case of nouns. One can also see a relatively small decrease in estimates of the accuracy of the algorithm in processing the matrix PREP, i.e., the matrix of valences of nouns in prepositional phrases. In this matrix, nouns were written with all prepositions combined in pairs, for example, ***at_University***, ***in_Sweden***, and ***to_Granada***. This increases many-fold the number of points on this scale of prepositional phrases with nouns. Such an increase in the tensor space along one of dimensionalities also requires larger sizes of text corpora for a sufficiently uniform filling of a tensor. This implies some decrease in the accuracy of the algorithm operation, but, however, it does not exceed 4–5%, which may be considered as a quite acceptable result.

Larger estimates for the accuracy of the operation of the algorithm of splitting **VSVs** of words into component vectors of their different values and their assignment to WordNet synsets in processing matrices of the decomposed tensor of the text corpus of Wikipedia entries in comparison with the estimates for its operation in processing tensor matrices of the corpus of Simple Wikipedia entries are undoubtedly explained by a considerably larger number of entries in the former corpus (4,1 million Wikipedia entries against 98 thousands of Simple Wikipedia entries) and also by their sizes in which Wikipedia also has a considerable advantage. Simple syntactic structures of Simple Wikipedia provide nearly 100% of quality in syntactic analysis and, as a result, a rather high quality of composition of its tensor. Therefore, despite the total advantage of Wikipedia over Simple Wikipedia as to the size, a decrease in estimates of the quality of operation of the algorithm in processing the decomposed Simple Wikipedia tensor does not exceed 6–7% on the average. On the whole, the presented estimates testify to a rather high performance of the proposed algorithm and to promising perspectives of using it in practice.

CONCLUSIONS

This work describes a model of nonnegative factorization of N-dimensional linguistic tensors composed in the course of frequency analysis of syntactic structures of sentences in large text corpora. A decomposition of collected tensors in the form of N matrices of reduced dimensionality k, in addition to a compact and convenient structure of representation of data on the compatibility of sequences of lexemes in some syntactic positions of sentences in a natural language provides an efficient method for the computation of an estimate for the probability of existence of semantic-syntactic relationships between words of different grammatical categories. In this case, k-dimensional vectors from matrices of a factored tensor can be considered as vectors of semantic-syntactic valences (**VSVs**) of words. Since words are ambiguous by their very nature and one word, as a rule, has several values, this work proposes to consider k-dimensional **VSVs** of words as sums of component addends of **VSVs** of different values of these words. This article presents a developed method for splitting **VSVs** of words into component addends of **VSVs** of their different values and assigning these split component addends of VSVs to WordNet synsets as their own **VSV** values. The algorithm uses data on the synonymy of words in WordNet synsets and also hyponymic-hypernymic relationships between synsets in the hierarchy of the ontological network. The implemented algorithm was tested using several experiments with matrices of decomposed tensors of the text corpora of Wikipedia and Simple Wikipedia entries. The estimates of the accuracy of the algorithm that are obtained during testing demonstrate its high efficiency.

It must be emphasized that a considerable advantage of the proposed method consists of a high degree of possible automation of each stage of its operation, namely, parsing of entries, composition of a tensor, nonnegative tensor factorization, and splitting of vectors of semantic valences of words into vectors of their values and their assignment to the corresponding WordNet synsets. All the stages are executed automatically. Experts are involved only at the stage of tuning the algorithm of splitting and assignment. One of topical directions of further investigations is the maximum minimization of participation of experts in tuning the algorithm.

An implicit specification of semantic relations between concepts-nodes of an ontological graph with the help of k-dimensional vectors of semantic valences also has indisputable advantage of the universality of representation of semantic relationships. After detecting the $(n+1)$th type of relationship between existing concepts, the system fixes its existence in the base in vector representation without making any immediate request for entering the new type in the list of relations, completely describing it in the ontology, and associating the corresponding syntactic pattern with it.

These advantages of the method testify to promising perspectives of using it in practice for the automation of methods of filling ontological knowledge bases with contents and for automatic detection of semantic relations between concepts, i.e., nodes of an ontological network, during processing large text corpora.

REFERENCES

1. T. Van de Cruys, “A Non-negative tensor factorization model for selectional preference induction,” J. Natural Language Engineer., **16**, No. 4, 417–437 (2010).
2. T. Van de Cruys, L. Rimell, T. Poibeau, and A. Korhonen, “Multi-way tensor factorization for unsupervised lexical acquisition,” in: Proc. COLING 2012, Mumbai, India (2012), pp. 2703–2720.
3. S. B. Cohen and M. Collins, “Tensor decomposition for fast parsing with latent-variable PCFGs,” NIPS, 2528–2536 (2012).
4. W. Peng and T. Li, “On the equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis,” Appl. Intel., Springer J., **35**, No. 2, 285–295 (2011).
5. C. J. Fillmore, “The Case for CASE,” in: E. Bach and R. Harms (eds.), Universals in Linguistic Theory, Holt, Rinehart, and Winston, New York (1968), pp. 1–88.
6. G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, Introduction to WordNet: An On-Line Lexical Database, http://wordnetcode.princeton.edu/5papers.pdf.
7. http://en.wikipedia.org/wiki/Main_Page.
8. http://simple.wikipedia.org/wiki/Main_Page.
9. A. Mikheev, C. Grover, and M. Moens, “Description of the ltg system used for muc-7,” in: Proc. 7th Message Understanding Conference (MUC-7) (1998), pp. 1–12.
10. I. Dagan, A. Itai, and U. Schwall, “Two languages are more informative than one,” in: Proc. ACL-91, Berkeley, California (1991), pp. 130–137.
11. J. Hockenmaier, G. Bierner, and J. Baldridge, “Providing robustness for a ccg system,” in: Proc. Workshop on Linguist. Theory and Grammar Implement., Birmingham (2000), pp. 97–112.

12. T. Briscoe and J. Carroll, “Automatic extraction of subcategorization from corpora,” in: Proc. 5th Conf. on Appl. Natural Language Proces. (ANLP-97), Washington DC, USA (1997).
13. F. Xia, “Extracting tree adjoining grammars from bracketed corpora,” in: Proc. 5th Natural Language Proces. Pacific Rim Symp. (NLPRS-99), Beijing, China (1999).
14. K. Church, W. Gale, P. Hanks, and D. Hindle, “Using statistics in lexical analysis,” in: U. Zernik (ed.), Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon, Ch. 6, Lawrence Erlbaum Associates, Hillsdale-New Jersey (1991), pp. 115–164.
15. L. Lee, “Similarity-based approaches to natural language processing,” Ph.D. Thesis, Harvard University Techn. Rep. TR-11-97 (1997), http:/www.cs.cornell.edu/home/llee/papers/thesis.pdf.
16. P. Cimiano and S. Staab, “Clustering concept hierarchies from text,” in: Proc. LREC (2004), pp. 1721–1724.
17. P. M. Hastings, “Automatic acquisition of word meaning from context,” Ph.D. Dissertation, University of Michigan (1994), http://reed.cs.depaul.edu/peterh/papers/hastingsdiss.pdf.
18. U. Hahn and K. Schnattinger, “Towards text knowledge engineering,” in: Proc. 15th National Conference on Artificial Intelligence AAAI-98 (1998), pp. 524–531, URL citeseer.nj.nec.com/43410.html.
19. V. Pekar and S. Staab, “Word classification based on combined measures of distributional and semantic similarity,” in: Proc. of Research Notes of the 10th Conf. of the European Chapter of the Assoc. for Comput. Linguistics, Budapest (2003), pp. 147–150.
20. E. Alfonseca and S. Manandhar, “Extending a lexical ontology by a combination of distributional semantics signatures,” in: Knowledge Engineering and Knowledge Management, Lecture Notes in Artificial Intelligence, **2473** (2002), pp. 1–7.
21. A. Maedche and S. Staab, “Discovering conceptual relations from text,” in: Proc. 14th Europ. Conf. on Artificial Intel. (2000), pp. 1–17.
22. M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora,” in: Proc. COLING-92, Nantes, France (1992), pp. 539–545.
23. M. A. Hearst, “Automated discovery of WordNet relations,” in: Ch. Fellbaum (ed.), WordNet: An Electronic Lexical Database, MIT Press (1998), pp. 132–152.
24. M. Berland and E. Charniak, “Finding parts in very large corpora,” in: Proc. ACL-99 (1999), pp. 57–64.
25. J. Kietz, A. Maedche, and R. Volz, “A method for semi-automatic ontology acquisition from a corporate intranet,” in: Workshop “Ontologies and text” co-located with EKAW’ 2000, Juan-les-Pins, French Riviera (2000), pp. 2–6.
26. Y. Wilks, D. C. Fass, C. M. Guo, J. E. McDonald, T. Plate, and B. M. Slator, “Providing machine tractable dictionary tools,” J. of Comput. and Translat., No. 2, 99–154 (1990).
27. G. Rigau, “Automatic acquisition of lexical knowledge from MRDs,” Ph.D. Thesis, Departament de Llenguatges i Sistemes Inform’atics, Universitat Polit’ecnica de Catalunya (1998), URL http://adimen.si.ehu.es/~rigau/publications/thesis-rigau.pdf.
28. S. D. Richardson, W. B. Dolan, and L. Vanderwende, “MindNet: Acquiring and structuring semantic information from text,” in: Proc. COLINGACL’98, 2, Montreal, Canada (1998), pp. 1098–1102.
29. W. Dolan, L. Vanderwende, and S. D. Richardson, “Automatically deriving structured knowledge bases rfon on-line dictionaries,” in: PACLING 93 Pacific Association for Comput. Linguistics (1993), pp. 5–14.
30. M. Ruiz-Casado, E. Alfonseca, and P. Castells, “Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets,” in: Advances in Web Intelligence, Springer, Berlin–Heidelberg (2005), pp. 380–386.
31. M. Ruiz-Casado, E. Alfonseca, and P. Castells, “Automatising the learning of lexical patterns: An application to the enrichment of Wordnet by extracting semantic relationships from Wikipedia,” Data & Knowledge Engineering, **61**, No. 3, 484–499 (2007).
32. E. Niemann and I. Gurevych, “The people’s web meets linguistic knowledge: Automatic sense alignment of Wikipedia and WordNet,” in: Proc. 9th Intern. Conf. on Comput. Semantics (IWCS) (2011), pp. 205–214.
33. http://nlp.stanford.edu/software/lex-parser.shtml.
34. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” J. of the American Soc. for Inform. Sci., **41**, No. 6, 391–407 (1990).
35. R. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis,” UCLA Working Papers in Phonetics, **16**, 1–84 (1970).
36. J. Antikainen, J. Havel, R. Josth, A. Herout, P. Zemcik, and M. Hauta-Kasari, “Nonnegative tensor factorization accelerated using GPGPU,” IEEE Trans. Parallel Distrib. Syst., **22**, No. 7, 1135–1141 (2011).
37. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Sci. 220, 671–680 (1983).