Development of a Semantic and Syntactic Model of Natural Language by Means of Non-Negative Matrix and Tensor Factorization

Anatoly Anisimov, Oleksandr Marchenko, Volodymyr Taranukha, and Taras Vozniuk

Faculty of Cybernetics, Taras Shevchenko National University of Kyiv, Ukraine
ava@unicyb.kiev.ua, rozenkrans@yandex.ua, taranukha@ukr.net, taarraas@gmail.com

Abstract. A method for developing a structural model of natural language syntax and semantics is proposed. Syntactic and semantic relations between parts of a sentence are presented in the form of a recursive structure called a control space. Numerical characteristics of these data are stored in multidimensional arrays. After factorization, the arrays serve as the basis for the development of procedures for analyses of natural language semantics and syntax.

Keywords: Information Extraction, WordNet, Wikipedia, Knowledge Representation, Ontologies

1 Introduction

Recently, the non-negative tensor factorization (NTF) method has become a widely used technology in such fields as information retrieval, image processing, machine learning and natural language processing. The approach is most promising for detection and analysis of linkages and relations in the data where objects of N different types are presented. In computational linguistics, the N-dimensional tensor is implemented as a multiway array of data obtained from the frequency analysis of large text corpora. Factorization of N-dimensional tensor with decomposition rank k generates N matrices. Such matrices consist of k columns that represent mapping of each individual dimension of the tensor on k factor-dimensions of latent semantic space. It is a unique tool to model and explore correlations of linguistic variables in an array of N-dimensional data.

The NTF method is looked upon as a promising technique for solving problems of computational linguistics [1,2,3,4]. Two works are of particular interest [1,2]. The authors describe models for the tensor representation of frequency of various types of syntactic word combinations in sentences, such as 3-dimensional combinations "Subject – Verb – Object", or 4-dimensional combinations of "Subject – Verb – Direct_Object – Indirect_Object" and other syntactic combinations no longer than the dimension of tensor N. Each dimension in tensor is responsible for a certain part of a sentence, i.e. Subject, Predicate, Direct Object, etc.

The N-dimensional tensors contain estimates for the frequency of word combinations sets in text corpora. The model takes into account syntactic positions of words. After large text corpora are processed and sufficient amounts of data are accumulated in the

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 324–335, 2014.

tensor, an N-way array is formed. It contains commutational properties of lexical items in the sentences of natural language. For the words presented in the tensor, the properties include: syntactic relations the word tends to be engaged into, other words in the tensor these relations point to, and frequencies of the corresponding relations. Moreover, these relations are multi-dimensional rather than binary, with N being the maximum number of possible dimensions. Then non-negative factorization for the obtained tensor is performed, as it significantly transforms the presentation model. Originally, a multi-dimension tensor is sparse and huge in its volume. Each of the N axes of the syntactic space contains tens of thousands or hundreds of thousands of points representing words. After the tensor has been factorized, its data are represented as N matrices consisting of k columns (where k is much smaller than the number of points in any of the tensor's N dimensions). Parameter k is a degree of factorization, the number of dimensions of the latent semantic space, and the number of attribute dimensions in it. In addition to a much more compact data representation, the probability of every possible word combination can be easily estimated in different syntactic sentence structures. This can be done by calculating the sum of the products of the components for N k-dimensional vectors corresponding to the words chosen from the matrices corresponding, in turn, to their syntactic positions.

For example, one needs to test how feasible is the sentence "The monkey eats banana", in the matrix SUBJECT one finds the k-dimensional vector s, which corresponds to the Noun "monkey", then in the matrix VERB the k-dimensional vector v is found, which corresponds to the Verb "eats". After that in the matrix DIRECT_OBJECT one finds the k-dimensional vector *do* which corresponds to the Noun "banana". The result is calculated as the sum of the products of the corresponding components for three vectors ($N = 3$):

$$x_{svdo} = \sum_{i=1}^{k} s_i v_i do_i,$$

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

Where the sum exceeds a certain threshold level, a conclusion is drawn that such a sequence of words in the sentence is plausible. The calculation for the combination of ("Banana", "eats", "monkey") shows the impossibility of such an option.

This model allows for successful automatic extraction of special linguistic structures from the corpus, such as *selectional preferences* [1] and *Verb SubCategorization Frames* [2], which combine data on the syntactic and semantic properties of relations between verbs and their noun arguments in sentences.

This model is promising and powerful, but lacks flexibility and represents the syntax of natural language in a limited way. The number of dimensions in the tensor restricts the maximum length of sentences and phrases described by this model. Each axis corresponds to a particular syntactic position. Van de Cruys describes the three-dimensional tensor for modeling a syntactic combination: Subject – Verb – Object [1]. Subsequently Van de Cruys and colleagues describe tensors of 9 and 12 dimensions to simulate up to twenty different types of syntactic relations and connections [2]. The mere increase in the tensor dimension number, however, does not seem to be a very convincing way of improving the model and handling more types of extended

arity syntactic relations. It is quite reasonable, therefore, to look for other universal representation models for syntactical structures of natural language sentences.

The control spaces [5] have been chosen from among numerous time-tested classical formal models of language syntax representation due to the fact that in this model an arbitrary complex structure is described using recursion through superposition of two basic syntactic relationships - syntagmatic and predicative. The proposed lexical and syntactic tensor model consists of a 3-dimensional tensor for predicative relations and a matrix for syntagmatic relations. The use of control spaces appears to be an efficient means to reduce arbitrary n-ary syntactic relation to the superposition of binary and ternary relations.

Understanding natural language requires knowledge of the language per se (vocabulary, morphology, syntax), and knowledge of the extralinguistic world. The tensor models include data on communicative properties only of the words from the texts already processed and only within the sentences and phrases in which these words are used. This paper proposes to use the hierarchical lexical database WordNet to generalize descriptions of communicative properties of words using the implicit mechanisms of inheritance by taxonomy tree branches. Assuming a word A belongs to a synset S and has a certain property P, there is a high probability that the other words from S will also have the property P. Also, some words of the children synsets of S will almost certainly have P and words of the parent synsets of S are also likely to have P. These assumptions underpin the implementation of the generalization mechanism that describes communicative semantic and syntactic properties of words applying the principle of taxonomic inheritance.

The training set contains texts from The Wall Street Journal (WSJ) corpus, along with the English Wikipedia and the Simple English Wikipedia articles. The latter two contain the definitions and basic information about concepts, which enhances semantics in the model.

2 Control Space of Natural Language Syntactic Structures

The basic syntactic structures are typically described in classical grammar patterns. Rather subtle relations of government among words are expressed in the linguistic models of constituent systems mostly developed by Chomsky [6] and in the models of dependency trees proposed by Tesnière [7]. These models only approximately describe the actual communication properties of syntactic structures.

Attempts to build models more suitable for machine processing that are able to generalize properties of dependency trees and constituent systems led to the creation of more convenient representations.

In work [5] the author proposed to use space for representation, which is independent of the order of text entries. The space expressing all predicative and syntagmatic relations contained in syntactic structures is named *the control space*.

Let us consider the algorithmic model of natural language sentence in terms of control spaces. A sentence is regarded as a dynamic recursive computational process developing in the control space that connects syntactically grouped parts of the sentence

with informational channels. The structure of the control space reflects the relations of syntagmatic and predicative language constructions.

Apart from having the referential function, language expresses relations that objects enter into, where the Verb establishes relations between the objects involved in the scheme of this Verb; the Adjective specifies the connection of the object to itself. The syntactic model should indicate which parts of the sentence are linked through relations and of what type the relations are. Predicative relation expresses the relation between syntactic objects through the concept that implies an action and is usually expressed by the predicate, a verb. A syntagma is a combination of two syntactic objects, one of which specifies the other, so the model must fully cover these types of relations. Moreover, in the broadest sense, syntagmas should form syntactic groups. An adequate model of the syntactic structure should also reflect the basic property of being recursive [5].

In control space formalism the conventional linguistic approach is intentionally violated. The Verb is not considered as the principal member of the sentence. In the control space model it is more convenient to define the syntactic relations of *generation* and *transmission of relations*, which ensures a more accurate description of the government connections.

When two objects A and B enter into relation C, we distinguish between an object (say A) that brings about relation C, and the object to which the relation is transferred, which is B. Thus, two types of directed links are differentiated: from *the relation generator object* to *the relation* and from *the relation* to *the subordinate*. The first type of connection is the α-connection (*generation connection*). The second is the β-connection (*propagation connection*). Objects A, B and C are placed in relevant locations in the control space and thus the formal representation of the relation C that connects A and B acquires the form: $A - \alpha \rightarrow C - \beta \rightarrow B$.

Verbs define relations between objects, in the typical pattern of the simple sentence: "Noun – Verb – Noun" the α-connection is directed from the first Noun to the Verb, and the β-connection is directed from the Verb to the second Noun. Let us consider an example: ***"Jim bought a ball"***. The object ***"Jim"*** generates the relation ***"bought"*** and directs it to the object ***"ball"***. Therefore the α-β-structure of the sentence has the form: $Jim - \alpha \rightarrow bought - \beta \rightarrow a\ ball$.

In the phrase: ***"Tall Jim"***, the ***"Jim"*** generates the unary relation ***tall*** and transmits the relation onto itself: $Jim - \alpha \rightarrow Tall - \beta \rightarrow Jim$.

Applying similar reasoning to the phrase ***Tall Jim really loves football*** we obtain the following structure:

$$(Jim - \alpha \rightarrow tall - \beta \rightarrow Jim) - \alpha \rightarrow (loves - \alpha \rightarrow really - \beta \rightarrow loves) - \beta \rightarrow football.$$

The sentences have two types of α-β-links: a strictly linear relation and a closed cyclic dependency. The first is called a linear structure, and the second is a definition. The first corresponds to the predicative language constructs, the second to the syntagmatic ones.

For control spaces the formal model oriented to forming complex structures of a required type is constructed as follows.

A base set of objects U is given. Each object is associated with a certain type. The number of types is finite. The types can be expressed as numbers from the interval $[0, N]$.

An ambiguity arises when mapping objects to types where type function φ maps U into the set of all subsets generated by numbers from the interval $[0, N]$. The constructions are either objects of U or obtained from other constructions by substituting the latter in terms of linear or defining dependency. The construction types are calculated after the following rules:

1. If in a linear dependence an i-type object A is α-connected with a j-type object B, and the j-type object B is β-connected to a k-type object C, then the type of construction is $f(i, j, k)$.
2. If an i-type object A is α-connected with a j-type object B in the definition structure, and B is β-connected to A, then the entire construction is attributed to $d(i, j) = i$-type.

Since the set of base types is finite, the functions f and d can be specified by tables. Rule 2 allows us to easily calculate any type of complex construction which will coincide with the type of one of the basic constructions defined by functions f or d.

The type of construction is an incorrect one if we can not calculate it. The entirety of the correct constructions composes the control spaces of set U.

Regarding syntactic structures, the definition reads as follows: the basic objects are words and collocations that represent parts of speech (Nouns, Adjectives, Verbs, Particles, etc.) with the appropriate morphological features, as well as compound relations and correlators, designed to connect subordinate sentences with the principal ones. The type of the word contains its complete grammatical description. For example, the type of the word ***book*** is (”Noun”, ”inanimate”, ”singular”, ”Nominative Case”). The notion of type can be extended with some semantic attributes. The ambiguity of the type definition lies in the ambiguity of of some words taken out of context. For example, the word ***book*** belongs to both the Nouns and Verbs classes. The function f specifies the types of simple sentences and complex sentences, depending on the construction of the upper level. Function d specifies the conditions of matching defined and defining objects. For example, the definitions of the Noun can be defined by Adjectives, Prepositions or a subordinate clause, the Verb can be defined by Adverbs, Gerunds or a subordinate clause, Verbs not forming a cyclic α-β-link with Nouns, etc. Thus, the functions f and d are used as a filter to identify the constructions allowed. In the definition constructions the role of the subordinate part is set to a comment or a clarification of the main part. So the type value for the whole syntagmatic structure is set to the generator value as the main object.

The control space of arbitrary sentences can be converted into a dependency tree and a CFG parse tree [5]. Therefore, the structure of the control space can be regarded as a generalization of both dependency trees and CFG parse trees. So, control spaces can express the syntactic structure of arbitrary complexity and arity as a set of binary and ternary relations. This allows for an accurate recording of all data on the semantic and syntactic relations with a single matrix D and one three-dimensional tensor F.

A special empty word is used instead of the missing object in 3-dimensional tensor for intransitive verbs. The construction ***”The boy runs”*** is written in the tensor F as a triplet (boy, runs, $\emptyset$). For ditransitive verbs, the parallel reduction procedure is used: construction ***”John gave Mary a toy”*** turns into $(John, gave, Mary) + (John, gave, toy)$ [5].

3 Building a Lexical-Syntactical Model of Natural Language

In order to construct the semantic-syntactic model of natural language, the method for automatic filling the three-dimensional tensor F and the matrix D was designed. It is used in the syntactic analysis and post-processing of sentences from large corpora. The method requires the following steps:

- Sentences from a text corpus are taken and parsed by the Stanford Parser module, which generates the syntactic structures of sentences in the form of dependency trees and parse trees for CF phrase structure grammar [8,9];
- The program examines the dependency tree and the CFG parse tree of the current sentence. It constructs the control space of the syntactic structure, analyzing relations between corresponding words to identify predicate combinations of length 3 (e.g., Subject-Verb-Object, etc.) and syntagmatic combinations of length 2 (Noun-Adjective, Verb-Adverb, etc.);
- Having assembled the control space of this sentence for every triad of points (i, j, k) connected with the linear predicative sequence of α-β-links, tensor F receives the value for the cell $F[I, J, K]$: $F[I, J, K] = F[I, J, K] + 1$. The coordinates I, J, K of the tensor cell correspond to pairs (w_i, A_i), (w_j, A_j) and (w_k, A_k), where w means words that are lexical values of the corresponding points (i, j, k), and A is a coded description of the characteristics of these words (part of speech, gender, number of lexical units, etc.).
- Similarly, in the control space of the syntactic structure of the current sentence for each pair of points (i, j) interconnected with the cyclic syntagmatic α-β-link, matrix D receives the value for the cell $D[I, J]$: $D[I, J] = D[I, J] + 1$. The coordinates I, J correspond to pairs (w_i, A_i) and (w_j, A_j), where w stands for words representing lexical values of the corresponding points (i, j), and A is a coded description of these words.

After processing large amounts of text, matrix D and three-dimensional tensor F accumulate sufficient lexical and syntactic communicative information to efficiently implement the lexical and syntactic model of natural language.

An extremely large dimension and sparsity of matrix D and tensor F demand for non-negative matrix and tensor factorization in order to store the data in a more economical way. Matrix D is factorized using Lee and Seung Non-negative Matrix Factorization algorithm [10] that decomposes matrix $D(N \times M)$ as a product of two matrices $W(N \times k) \times H(k \times M)$, where $k << N, M$. Tensor F is factorized using the non-negative three-dimensional tensor factorization parallel algorithm PARAFAC [11]. The factorization yield corresponding matrices X, Y and Z.

4 Properties of a Lexical Model of Natural Language

After matrix D and tensor F factorization, the system forms a strong knowledge base which contains information about the syntactic framework of natural language sentences. The description of semantic relations between the words is integrated into the structures. Apart from the description of general syntax that defines the structure of

the sentences in a general abstract form, the base also contains semantic restrictions that determine which words can form a syntactic connection of a certain type. To determine whether two words a and b form a cyclic syntagmatic relation, one has to take vector-row W_a from matrix W corresponding to word a, and vector-column matrix H_b from matrix H which corresponds to word b, and calculate the scalar product of vectors (W_a, H_b^T). If the product is greater than a certain threshold T, then this relation is defined. In order to determine whether the three words a, b and c enter into predicative relation $(a \rightarrow b \rightarrow c)$, it is necessary to take vector X_a corresponding to word a, vector Y_b corresponding to word b, and vector Z_c corresponding to word c and to calculate the value:

$$S_{abc} = \sum_{i=1}^{k} X_a[i] * Y_b[i] * Z_c[i]$$

If S_{abc} value is greater than a threshold, then this relation is defined. If not, it is considered undefined.

These matrices implicitly define a set of defined language clauses, the set being specified with the input text corpus. The vectors of words from the derived matrices implicitly describe their "structural behavior". They define in which syntactic relation these words may join and which words they have joined. With the resulting matrix, one may parse sentences and generate the control space of their syntactic structures, using the ascending algorithms such as CYK [12]. The control space is built where possible.

5 Implementation

As the initial training text corpus, sets of articles from the English Wikipedia, the Simple English Wikipedia and the WSJ corpus are used. The texts are processed sequentially with the parser and with the program that constructs the control space of syntactic structures. First, the sentences are analyzed with the Stanford Parser yielding a CFG parse trees (for phrase structure grammar) and a dependency trees. Also, an algorithm has been developed to construct control spaces by converting a dependency tree and a parse tree into the control space of a sentence. The algorithm is a recursive traversal from left to right of the sentence tree which creates points of the control space in each node of the CFG parse tree and performs conversion of corresponding relations of the dependency tree into α-β-connections of control space (either predicative or syntagmatic connections). Each point of the space is assigned a specific lexical value (a word or phrase) and characteristics (part of speech, gender, number, etc.). At the outset every word is an isolated point in the control space. When points A and B are connected to form a new point S in the space, representing the α-β-relationship between A and B, this new point gains its own lexical value. This value can be inherited from the main element of the pair (A, B), e.g., the phrase *cold water* consists of a pair *(cold, water)* that has a Noun as the main word. Consequently, the new point will inherit value from *water*. Also, the merger of two points may result in their lexical value forming a fixed collocation. For example, the combined value of point *A (Weierstrass)* and *B (theorem)* is the *Weierstrass theorem*, which is the lexical value of the new generated point C. Fixed collocations are obtained based on Wikipedia articles with a corresponding titles.

After the control space has been built, for each cyclic α-β-syntagmatic link the value $d[I, J]$ is increased by 1 in cyclic links matrix D (where I is the index of the first word, J is the index of the second word): $d[I, J] = d[I, J] + 1$. For each of the triplets in linear relations $A - \alpha - B - \beta - C$ the three-dimensional tensor F cell $f[I, J, K]$ is increased by 1 (where I is the index of word A, J is the index of word B, and K is the index of word C): $f[I, J, K] = f[I, J, K] + 1$.

800,000 articles from the English Wikipedia and the Simple Wikipedia have been processed, along with the WSJ corpus. As the WSJ corpus is annotated manually and contains correct syntactic structures, a high number of quality syntactic structures control spaces are received.

The processing yielded the large matrix D for cyclic links (numbering approximately 2.3 million words $\times$ 2.3 million words, with up to 57 million non-zero elements) and the large three-dimensional tensor F for linear predicative connections (consisting of approximately 2.3 million words $\times$ 52 thousand words $\times$ 2.3 million words, with about 78 million non-zero elements). These arrays were factorized by the non-negative matrix factorization algorithm [10] and the non-negative tensor factorization parallel algorithm PARAFAC [11].

Factorized data sets allow for efficient computing of probability for cyclic syntagmatic relations between any two words using the scalar product of two corresponding vectors. To form linear predicative relations between any three words the probability can be efficiently and easily calculated.

To investigate the applicability of this model for practical NLP tasks a parser for the English language based on the obtained arrays of lexical-syntactic combinability has been implemented. This parser, based on the Cocke-Younger-Kasami algorithm, directly constructs the control space of a sentence.

The model describes only the relations among those words which actually occur in the corpus sentences and have been processed accordingly. When a pair of words A and B make a cyclic syntagmatic link and has value in the array, the pair A_1 and B_1 (where A_1 is synonymous with A and B_1 is synonymous with B) will not have the link if A_1 and B_1 are absent in the data. The same holds for linearly predicative relations. The matter can be easily dealt with using synonym dictionaries. In the system we developed the WordNet is used to this end. We assume that if between A and B a relation exists, it also exists between an arbitrary pair of A_i and B_i, where A_i is any word from the synset that contains A, while B_i is any word from the synset that contains B. However, the question of homonymy arises when one word corresponds to several synsets in the WordNet. Every time a sentence is parsed, the point at issue is how to determine whether a pair or triplet of synsets is correct.

On the one hand, there are several standard approaches to solving this classic problem of ambiguous words (WSD). On the other hand, the two matrices W and H resulting from the non-negative matrix factorization of D can be considered powerful tools for determining the degree of semantic similarity between words according to the methods of latent semantic analysis [13].

So, to determine the presence of cyclic syntagmatic α-β-connections and to solve the problem of ambiguous words the following steps are carried out:

A: Take vector W_a corresponding to word a from term matrix W, vector column H_b which corresponds to the word b from matrix H, and calculate the scalar product of the vectors (W_a, H_b^T). If the value $(W_a, H_b^T) > T$, then this link is **defined**. T is the threshold. The optimal value of T is found experimentally. If it fails:

B: Take synsets for words a and b from the WordNet. The set of synsets $\{A_i\}$ refers to word a, and the set of synsets $\{B_i\}$ refers to word b. Check the pairs of the words formed from the elements of $\{A_i\}$ and $\{B_i\}$. If there is word $\acute{a_k}$ from $A_k \in \{A_i\}$ and word $\acute{b_j}$ from $B_j \in \{B_i\}$ such that scalar product of vectors $(W_{\acute{a_k}}, H_{\acute{b_j}}^T) > T$, then this link between a and b is **defined**. If not:

C: The set $\{A_i\}$ is expanded with synsets linked with nodes from $\{A_i\}$ with hyponym and hypernym relations in the WordNet. The set $\{B_i\}$ is expanded in the same way. Check the pairs of words formed from elements of $\{A_i\}_{exp}$ and $\{B_i\}_{exp}$ (excluding the pairs already checked on step B). If there is a word $\acute{a_k}$ from the synset $A_k \in \{A_i\}_{exp}$ and a word $\acute{b_j}$ from the synset $B_j \in \{B_i\}_{exp}$ such that the scalar product of vectors $(W_{\acute{a_k}}, H_{\acute{b_j}}^T) > T$, then the link between a and b is **defined**. If it fails: expand $\{A_i\}_{exp}$ and $\{B_i\}_{exp}$ recursively 2 or 3 times and repeat step (C).

If it is always $(W_{\acute{a_k}}, H_{\acute{b_j}}^T) < T$, then the link **does not exist**.

When expanding $\{A_i\}$ and $\{B_i\}$, one should avoid adding synsets from the list of the concepts with the most general meanings from the top of the WordNet hierarchy. If $\{A_i\}_{exp}$ and $\{B_i\}_{exp}$ are extended with such concepts, the semantic similarity between $\acute{a_k}$ and $\acute{b_j}$ quickly deteriorates. Inheritance of properties through hyponymy/hypernymy is not correct for such synsets.

For the linear predicative α-β-link this algorithm works in the same way.

The taxonomic hierarchy of the WordNet lexical database together with the mechanism of inheritance allows us to generalize this representation model of syntactic and semantic relations of natural language. This turns the constructed system into a versatile tool for syntactic and semantic analysis of natural language texts.

6 Experiments

To form a robust syntactical and semantic relations base, it is crucial to have a huge corpus of correctly tagged texts. Usage of the WSJ corpus has a significant effect on the quality assurance of the resulting model. To construct tagged texts from the English Wikipedia and the Simple English Wikipedia, the Stanford parser is used. The accuracy of parse trees is about 87%, while the accuracy of dependency trees is close to 84%. As some of the trees are incorrect, it is natural that they yield some inaccurate descriptions of the syntactic structures control spaces. The algorithm for converting CFG parse trees and dependency trees into control spaces of syntactic structures shows no errors on correct trees.

The development of the system for parsing and control spaces generation for natural language sentences based on created syntactic and semantic relation databases was followed by experiments. The accuracy was measured by computing control spaces of the syntactic structures. To generate test samples, 1,500 sentences were taken from the

Table 1. Precision estimation of cyclic α-β-syntagmatic links and linear predicative α-β-links on sentences from the Simple English Wikipedia, the English Wikipedia and the WSJ corpus

Cyclic α-β-syntagmatic links	Simple Wikipedia	Wikipedia	WSJ corpus
Case A	95,17%	91,23%	93,71%
Case B	91,29%	89,91%	91,05%
Case C	89,17%	83,06%	85,07%
Linear predicative α-β-links	Simple Wikipedia	Wikipedia	WSJ corpus
Case A	96,17%	92,24%	94,37%
Case B	93,21%	90,01%	91,33%
Case C	91,03%	87,79%	89,79%

Simple Wikipedia articles; 1,500 sentences - from the Wikipedia articles (using the texts not included in the 800,000 articles processed for constructing matrix D and tensor F).

The syntax trees of the sets of texts from the Wikipedia and the Simple Wikipedia that were processed with the Stanford parser were automatically transformed into control spaces. The obtained control spaces were manually verified and corrected by experts. This annotated text corpus was formed for the purpose of checking the quality of parsing and generating syntactic structure control spaces for the Simple Wikipedia and the English Wikipedia texts.

The system for parsing and control spaces generation constructs control spaces of syntactic structures for sentences from the annotated corpus. Subsequently, the obtained control spaces were compared with the corresponding correct control spaces from the annotated test corpus.

Each cyclic syntagmatic α-β-link and each linear predicative α-β-link that were found was automatically tested. The test was carried out with due regard for the algorithmic case in which a particular syntactic relation was found. Case **A** describes the identification of the direct link between words through the scalar product of their vectors; case **B** describes the usage of synonyms to compute the probability of the link. Case **C** describes the usage of the hyponym and hypernym WordNet connections for these words to find the probability of the link. The test was performed only for the sentences that had been successfully processed with the complete building of the syntactic structure control spaces (94.1% from 1,500 sentences from the Simple English Wikipedia and 83.4% from 1,500 sentences from English Wikipedia were successfully processed in the test set). Also, the test was performed on the WSJ corpus using cross-validation (when checking the quality of the system on 1 part of the corpus out of 10, the corresponding data obtained from the above mentioned part were temporarily excluded from the base of the model). The test on the WSJ corpus was performed automatically and 92.7% of sentences from the WSJ corpus obtained complete parse. The results are summarized in Table 1.

It should be noted that the precision estimates of the linear predicative α-β-links are higher than the precision estimates of the cyclic syntagmatic α-β-links. It seems natural to consider the relative positional stability for relations of type *Subject-Verb-Object* structure in the sentences. A certain small percentage of errors occurs even in the simplest case **A**. It indicates that errors must be present in the training set of control

spaces of sentences that served as the base for constructing the cyclic links matrix D and the three-dimensional linear predicative relations tensor F. The model can be improved by checking and correcting the training set. The best estimates correspond to sentences from the Simple Wikipedia, which is quite understandable due to the simple and clear syntactic structure of its sentences. The English Wikipedia sentences are much more complicated, leaving more room for different interpretations of grammatical structures. Hence the precision of processing the WSJ corpus sentences is higher than that for the English Wikipedia sentences. It indicates that the high quality training data from the WSJ corpus allows for improving the model to a great extent.

7 Conclusions

The recursiveness of syntactic structures control spaces allows us to describe sentence structures of arbitrary complexity, length and depth. This enables the development of a semantic-syntactic model based on a single three-dimensional tensor and a single matrix instead of increasing the number of dimensions of connectivity arrays for lexical items. To investigate the applicability of this model for practical NLP tasks a system for analysis and constructing syntactic structure control spaces has been developed on the basis of factorized arrays. It shows high quality and accuracy, thus proving the correctness and efficiency of the developed model.

References

1. Van de Cruys, T.: A Non-negative Tensor Factorization Model for Selectional Preference Induction. Journal of Natural Language Engineering, 16(4), pp. 417–437 (2010)
2. Van de Cruys, T., Rimell, L., Poibeau, T., Korhonen, A.: Multi-way Tensor Factorization for Unsupervised Lexical Acquisition. Proceedings of COLING-2012, pp. 2703–2720 (2012)
3. Cohen, S. B., Collins, M.: Tensor Decomposition for Fast Parsing with Latent-Variable PCFGs. NIPS-2012, pp. 2528–2536 (2012)
4. Wei, P., Tao, L.: On the equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis. Applied Intelligence, Springer Journals, October, Vol. 35, Issue 2, pp. 285–295 (2011)
5. Anisimov, A.V.: Control space of syntactic structures of natural language. Cybernetics and System Analysis, 93, pp. 11–17 (1990)
6. Chomsky, N.: Syntactic Structures. Mouton & Co. 117 pages. (1957)
7. Tesnière, L.: Èlèments de syntaxe structurale, Klincksieck, Paris (1959)
8. Klein, D., Manning, C. D.: Accurate Unlexicalized Parsing. Proceedings of ACL-2003, pp. 423–430 (2003)
9. de Marneffe, M.-C., MacCartney, B., Manning C. D.: Generating Typed Dependency Parses from Phrase Structure Parses. Proceedings of LREC (2006), `http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf`
10. Lee, D.D., Seung, H.S.: Algorithms for Non-Negative Matrix Factorization. NIPS(2000), `http://hebb.mit.edu/people/seung/papers/nmfconverge.pdf`
11. Cichocki, A., Zdunek, R., Phan, A.-H., Amari, S.-I.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. J. Wiley & Sons, Chichester, (2009)

12. Kasami, T.: An efficient recognition and syntax-analysis algorithm for context-free languages. Scientific report AFCRL-65-758. Air Force Cambridge Research Lab, Bedford, MA, (1965)
13. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, Th.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41 (6), pp. 391–407 (1990)