Semantic and Syntactic Model of Natural Language Based on 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. Factorization of lexical combinability arrays obtained from text corpora generates linguistic databases that are used for analysis of natural language semantics and syntax.

1 Introduction

Recently, the non-negative tensor factorization (NTF) method has become widely used in the natural language processing. From among numerous works in the area of particular interest are two works [1, 2]. They describe models for the tensor representation of the frequency for various types of syntactic word combinations in sentences. After non-negative factorization of tensors such a model allows for successful automatic extraction of specific linguistic structures from a corpus, such as selectional preferences [1] and Verb Sub-Categorization Frames [2], which combine data on syntactic and semantic properties of relations between verbs and their noun arguments in sentences.

The N-dimensional tensors contain estimates for 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 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, which significantly transforms the presentation model. Originally, a multi-dimension tensor is sparse and extensive. Each of the N axes of the syntactic space contains tens of thousands or hundreds of thousands of points that represent 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

A. Przepiórkowski and M. Ogrodniczuk (Eds.): PolTAL 2014, LNAI 8686, pp. 177–184, 2014.

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 more compact data representation, the probability of every possible word combination can be 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.

The number of dimensions in the tensor restricts the maximum length of sentences and phrases described by this model. Van de Cruys describes a three-dimensional tensor for modeling the syntactic combination: Subject – Verb – Object [1]. Van de Cruys and colleagues describe tensors of 9 and 12 dimensions to simulate up to twenty different types of syntactic relations [2]. The mere increase in the tensor dimension number, however, does not seem to be a good way of improving the model and handling more types of complex syntactic relations. It is quite reasonable, therefore, to look for other universal representation models for syntactical structures. The control spaces [3] have been chosen from among numerous time-tested classic formal models of language syntax representation owing to the fact that in this model an arbitrary complex structure is described using recursion through superposition of two basic syntactic relationships – binary syntagmatic and ternary predicative. The lexical and syntactic tensor model proposed here consists of a 3-dimensional tensor for ternary predicative relations (like Subject – Verb – Object) and a matrix for binary syntagmatic relations (like Noun – Adjective, Verb – Adverb, etc.). Sentences have two types of links: a ternary predicative relation and a closed cyclic dependency (binary syntagmatic). 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 language per se (vocabulary, morphology, syntax), and knowledge of the extralinguistic world. The tensor models include data on semantic and syntactic 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 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 Lexical-Syntactic Model of Natural Language

In order to construct a semantic-syntactic model of natural language, a method for automatic filling the three dimensional tensor F(for ternary predicative relations) and the matrix D (for cyclic binary dependencies) was designed. The method calls for the following steps:

- Sentences from a large 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 phrase structure grammar [4, 5].
- 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 cyclic binary combinations of length 2 (Noun – Adjective, Verb – Adverb, etc.).
- In the control space of this sentence for every triad of points (i, j, k) connected with the ternary predicative sequence of links, in tensor F the cell $F[I, J, K]$ receives the value: $F[I, J, K] = F[I, J, K] + 1$. 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 binary link, in matrix D the cell $D[I, J]$ is set to: $D[I, J] = D[I, J] + 1$.

The 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 factored using Lee and Seung Non-negative Matrix Factorization algorithm [6] that decomposes matrix $D(N \times M)$ as a product of two matrices $W(N \times k) \times H(k \times M)$, where $k \ll N, M$. Tensor F is factored using the non-negative three-dimensional tensor factorization parallel algorithm PARAFAC [7]. The factorization yields corresponding matrices X, Y and Z.

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. Along with the description of general syntax that defines the structure of 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 binary relation, one has to take vector-row W_a from matrix W corresponding to the word a, and vector-column matrix H_b from matrix H which corresponds to the word b, and calculate the scalar product of vectors (W_a, H_b^T). If the product is greater than

a certain threshold T, this relation is defined. In order to determine whether the three words a, b and c enter into predicative relation $(a \to b \to c)$, it is necessary to take vector X_a corresponding to the word a, vector Y_b corresponding to the word b, and vector Z_c corresponding to the 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, which is specified with the input 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 ascending algorithms such as Cocke – Younger – Kasami. The control space is built where possible.

3 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 CFG parse trees (for phrase structure grammar) and dependency trees. Also, an algorithm has been developed to construct control spaces by converting the dependency tree and the CFG 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 connections of control space (either predicative or cyclic connections). Each point of the space is assigned a specific lexical value (a word or a 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 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 *hot tea* consists of a pair *(hot, tea)* that has a Noun as the main word. Consequently, the new point will inherit value from *tea*. 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 corresponding titles.

The control space has been built, with matrix D and tensor F filled. 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 yields a 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 ternary predicative connections (consisting of approximately 2.3 million words $\times$ 52 thousand words $\times$ 2.3 million words, with up to 78 million non-zero elements). These arrays were factorized by the non-negative matrix factorization algorithm [6] and the non-negative tensor factorization parallel algorithm PARAFAC [7].

Factorized data sets allow for efficient computing of probability for cyclic binary relations between any two words using the scalar product of two corresponding vectors. To form ternary predicative relations among 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 corpora sentences and have been processed accordingly. When a pair of words A and B makes a cyclic binary 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 ternary predicative relations. The matter can be easily dealt with by 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.

So, to determine the presence of cyclic binary 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 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 a word a'_k from

$A_k \in \{A_i\}$ and a word $b_j^{'}$ from $B_j \in \{B_i\}$ such that scalar product of vectors $(W_{a_k^{'}}, H_{b_j^{'}}^T) > T$, 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 hyperonym relations in the WordNet. The set $\{B_i\}$ is expanded in the same way. Check the pairs of words formed from elements of $\{A_i\}_{\text{exp}}$ and $\{B_i\}_{\text{exp}}$ (excluding the pairs already checked on step B). If there is a word $a_k^{'}$ from the synset $A_k \in \{A_i\}_{\text{exp}}$ and a word $b_j^{'}$ from the synset $B_j \in \{B_i\}_{\text{exp}}$ such that the scalar product of vectors $(W_{a_k^{'}}, H_{b_j^{'}}^T) > T$, then the link between a and b is **defined**. If it fails: expand $\{A_i\}_{\text{exp}}$ and $\{B_i\}_{\text{exp}}$ recursively 2 or 3 times and repeat step (C).

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

During the expansion of $\{A_i\}$ 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. In expanding $\{A_i\}_{\text{exp}}$ and $\{B_i\}_{\text{exp}}$ with such concepts, the semantic similarity between $a_k^{'}$ and $b_j^{'}$ quickly deteriorates. Inheritance of properties through hyponymy/hypernymy is not correct for such synsets.

For the ternary predicative 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.

4 Experiments

To form a robust syntactical and semantic relations base, it is crucial to have a huge corpus of correctly tagged texts. The WSJ corpus availability has a significant effect on assuring the quality of the resulting model. To construct tagged texts from the English Wikipedia and the Simple English Wikipedia, the Stanford Parser is used. It produces dependency and CFG parse trees. The accuracy of CFG parse trees is about 87%, while the accuracy of dependency trees is about 84%. As some of the trees are incorrect, it natural that they yield some inaccurate descriptions of the control spaces of the syntactic structures. 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 lexical and syntactic 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 Simple English Wikipedia articles; 1,500 sentences – from the English Wikipedia articles (using the texts not included in the 800,000 items 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, applying conversion with the developed algorithm. 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 binary link and each ternary predicative link that were found were 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 hyperonym 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 1500 sentences from the Simple English Wikipedia and 83.4% from 1500 sentences from English Wikipedia were successfully processed in the test set). Also, a 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. 92.7% of sentences from the WSJ corpus obtained complete parse. The results are summarized in Table 1.

Table 1. Precision estimation of cyclic binary links and ternary predicative links on sentences from the Simple English Wikipedia, the English Wikipedia and the WSJ corpus

	Simple Wikipedia	Wikipedia	WSJ corpus
Cyclic binary links (case A)	95.17%	91.23%	93.71%
Cyclic binary links (case B)	91.29%	89.91%	91.05%
Cyclic binary links (case C)	89.17%	83.06%	85.07%
Ternary predicative links (case A)	96.17%	92.24%	94.37%
Ternary predicative links (case B)	93.21%	90.01%	91.33%
Ternary predicative links (case C)	91.03%	87.79%	89.79%

The precision estimates of the ternary predicative links are higher than the precision estimation of the cyclic binary links. It seems natural considering the positional stability for relations of type *Subject – Verb – Object* structure in the sentences. A certain small percentage of errors occurs even in 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 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 for 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.

5 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 the single three-dimensional tensor and the 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 constructed model. The model is of high relevance both for theoretical and practical applications for computational linguistic systems.

References

1. Van de Cruys, T.: A Non-negative Tensor Factorization Model for Selectional Preference Induction. Journal of Natural Language Engineering 16(4), 417–437 (2010)
2. Van de Cruys, T., Rimell, L., Poibeau, T., Korhonen, A.: Multi-way Tensor Factorization for Unsupervised Lexical Acquisition. In: Proceedings of COLING 2012, pp. 2703–2720 (2012)
3. Anisimov, A.V.: Control Space of Syntactic Structures of Natural language. Cybernetics and System Analysis 3, 11–17 (1990)
4. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: Proceedings of ACL 2003, pp. 423–430 (2003)
5. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: Proceedings of LREC (2006), `http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf`
6. Lee, D.D., Seung, H.S.: Algorithms for Non-Negative Matrix Factorization. In: NIPS (2000), `http://hebb.mit.edu/people/seung/papers/nmfconverge.pdf`
7. 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)