Unsupervised Parsing of the Russian Sentence

S.B.Potemkin *(potemkin@philol.msu.ru)*
Philological faculty of the Moscow State University, Moscow, Russia,

Abstract:
A statistical approach to the raw text parsing is described. A parsing algorithm builds a projective dependency tree in quadratic time after training on an unannotated corpus.

1 Introduction

In the field of automatic natural language understanding, the problem of connecting syntax and semantics has been faced in different ways. Most researchers have thought that semantics and syntax should be integrated with respect to both the representation and the processing; others have claimed that it is more efficient to build a full-blooded syntactic representation during the parsing process.

The basic schema may look rather classic: the system produces a syntactic analysis of the text, driven on the basis of purely syntactic knowledge. The semantic analyzer checks the syntactic output to see if the semantic relations among words are supported by it. In this paper we deal with the first step of this schema – automatic parsing with keeping in mind the next stage - semantic analysis, based on the formalism of conceptual graphs [11]. The interaction between syntax and semantics should be obtained by exploiting, in a formal way, the isomorphism between syntactic and semantic structures.

The problem of automatic parsing avoiding preliminary manual adjustment and training on the annotated corpora is of great theoretical and practical interest. The resulting grammar rules can support the processes of language acquisition by people and the general structure of language, provide preliminary processing of texts for syntactic marking of large corpora and, in the long term, ensure analysis of texts for natural language processing. This problem attracts essential interest thanks to availability of huge corpora, computing capacities growth and new algorithms of machine training.

The annotated corpora allow to prove the hypotheses which are put forward by grammatical theories, and also to form the syntax rules. The process called as "training" of the formal grammar should terminate at achievement of some small percent of errors. The annotated corpora or «tree banks» are used for grammar training. For the Slavic languages we can mention: Bulgarian (BulTreeBank), Polish (Project CRIT-2), Russian (ETAP-3, IPPI, the Russian Academy of Sciences), and the most advanced one for the Czech language (Prague Dependency Treebank). Tree banks for Balkan (Serbo-Croatian, Slovene, Bosnian) languages are under construction.

The majority of works on parsing are based either on rules, or on supervised training. Good parsers based on the constituent formalism are available for English and some other languages [3]. Some works based on the dependency formalism also exist [6, 8, 9, 10]. However, good parsers or even any parsers are not available for the majority of languages of the world. It is connected with the fact, that the resources necessary for the rule-based parsers or for the example-based parsers for the majority of languages are poor. Development of such resources demands material and labor expenses, so it is desirable to develop some methods for grammatical analysis without training on tree-banks, or for automatic or semi-automatic creation of the tree-banks.

A steady progress in the field of unsupervised parse was observed during the last years, but the majority of works is based on the context-free grammars whereas the classical model of dependencies (Mel'čuk, 1988) [7] is traditionally used for description of syntax of Russian and other Slavic languages. The aim of dependency parsing is to construct a tree structure of a sentence where

nodes represent words, and edges represent links between the words. An advantage of dependency parsing is that dependency trees are a reasonable approximation of the semantics of sentences, and are readily usable in NLP applications. Furthermore, the efficiency of popular approaches to dependency parsing compare favorable with those of phrase structure parsing or deep parsing.

2 Contemporary Reaches

One approach to simplification of the syntactic marking of the national corpus is to use the marked corpus of some other language apply the algorithms specially created for "marking transfer». The English Penn Treebank is used generally as the basic marked corpus. Because for the majority of languages there exists at least the bilingual translation English dictionary, a marking problem, in general, becomes simpler, though results are not ideal. It is especially true for Slavic languages with rather free word order and grammar is usually described by the dependency formalism whereas in the Penn Treebank the constituent formalism is used.

The other, purely statistical approach has certain advantages - it is necessary to have only a limited (about 1 million words) unannotated national corpus, without the parallel corpus and even without the bilingual translation dictionary. It is especially important for small and disappearing languages. The statistical approach to the syntactic analysis of sentences is applied in several, interconnected techniques, including DLM (dependency language model) (Gao, Suzuki, 2003) [4], U-DOP (Unsupervised Data-Oriented Parsing) (Bod, 2006) [2], CCL (Common cover links) (Seginer 2007) [11].

Within the limits of Bod's method it is necessary:

- To construct all possible trees of analysis for all corpus sentences and all subtrees for each tree.
- To find the best (most probable) tree for the given sentence.

A number of computing difficulties arises at the method implementation because the number of subtrees increases tremendously (the Catalan numbers) with the lengthening of the sentence. This problem is resolved by representing subtrees in the form of PCFG (probability context-free grammar) [5] and recording all trees as a "shared forest" [1]. These methods reduce the computation difficulty to an observable, however very large amount of calculations.

In the Seginer's approach the standard representation of a sentence structure in the form of a dependency tree is replaced with the set of Common Cover Links, CCL. Sentence analysis proceeds consistently, word-by-word, by the analysis of the initial sequence of words of the sentence. Results of such partial analysis are not subjected to change afterwards, but only could be supplemented. Each new link is added, if it does not break the certain set of a priori rules and if it possesses maximum weight (among admissible). A lexicon containing the list of left and right neighbors for each word connected with the given word and the frequency of such neighbors is created for determination of the link weight.

In comparison with the dependency structure the CCL structure possesses certain advantages: first, for such sentences as «I know the boy sleeps» with the dependency structure [[I] [know] [[the boy] [sleeps]]] CCL does not establish a link direction in the relation [the boy]. Similarly, for Russian the direction of the preposition-noun group is not established. The second difference is more essential. In traditional methods at the moment of reading a word "boy" the link between "know" and "boy" is established, however at the end of the sentence it is necessary to remove this link and to establish new - [know sleeps] and [sleeps boy]. This problem is known in psycholinguistics as a problem of the repeated analysis. In the CCL structure this problem is bypassed by appointing a value to each link -0 the "depth" of this link. Unambiguity of the bracket structure is achieved, without the necessity of removing the established links. Parser on the basis of CCL, adjusted for English language, it available for noncommercial use, http://staff.science.uva.nl/~yseginer/ccl/.

Next, Gao and Suzuki also have proposed an incremental approach to parsing where the dependency structure is constructed consistently, after input of the sequential word of the sentence and deletion of the links which break acyclic and projectivity features. Their method was applied not to the sentence structure analysis, but to restoration of the hieroglyphic view of the Japanese sentence (kana-kandzi) on the basis of the syllabic record (kana) - this problem and the method of its incremental solution are also applied to speech recognition.

The present work leans basically on the Gao and Suzuki technique, however the algorithm building the spanning tree of the sentence for the analysis of the sentence dependency is developed, without deletion of the links, working at $O(n^2)$ time, while preserving the classical dependency structure. The automatic syntactic marking of unannotated corpus, both for Russian, and for other languages with the sufficient volume of electronic texts with prevalence of projective sentences is possible on the basis of the presented method.

3 Model of local links (MLL)

In the model of local links the dependency structure is bottom-up constructed. Initially the links between the neighboring words (locality) are established; these links form "units". Then the links between the neighboring units are established, and so on, until the last, top level is reached, and the construction of the dependency tree comes to the end. The choice of sequence of association of units which is defined by the link weight between the units is essential.

3.1 Definitions

For a more formal description of our model we define the following:

W - sequence of words of a sentence; $W = \{w_1, w_2..., w_n\}$

T – dependency tree over W; $T = \{(w_i, w_j)\}$, where i, j - numbers of the words connected, $i < j$. T is a projective tree.

U *unit* - subtree of T over an indissoluble subsequence of W; $U_{k0}=w_k$, or $U_{kl} = \{w_k, w_{k+1}, \ldots, w_{k+l}\}$ where each pair of words is connected by a branch of T.

w_m - *Open* node of unit U iff there are no branches (w_i, w_j) of U, $i < m < j$. Otherwise w_m node is *closed*.

Adjacent units $U_{ap} = \{w_a, w_{a+1}, \ldots, w_{a+p}\}$ and $U_{bq} = \{w_b, w_{b+1}, \ldots, w_{b+q}\}$ are units where $b=a+p+1$, that is the beginning of unit U_{bq} directly follows the end of unit U_{ap}.

Basically, the language model should define probability of sentence W over all possible trees T, that is

$$P(W) = \Sigma P(W, T). \quad (1)$$

where P(W, T) is the probability of the sentence W with sample structure T.

Practically, only one member of the sum, namely P(W, T*) is used for estimation of P(W):

where T* - the most probable dependency structure of the sentence which delivers maximum for P(W, T):

$$T^* = \text{argmax } P(W, T) \quad (2)$$

The parsing purpose is to find the most probable analysis T* of the given sentence W maximizing probability P(T|W). Assuming that links (i, j) are independent from each other (very strong assumption), we have

$$P(T|W) = \mathbf{\Pi} P((i, j)|W) \quad (3)$$

where P((i, j)|W) is probability of link (i, j) in the specific sentence W. It is impossible to estimate directly probability P((i, j)|W) because the corpus does not contain, or contains very few identical sentences. Therefore we will approximate P((i, j|W) as P(i, j) which depends only on occurrence of words w_i, w_j in sentences of the corpus and, probably, from the distance (j-i).

Probability P(i, j) is estimated as

$$P(i, j) = C(w_i, w_j, R) / C(w_i, w_j) \quad (4)$$

where $C(w_i, w_j, R)$ - number of occurrences of link R between words w_i and w_j in the corpus, and $C(w_i, w_j)$ - number of occurrences of words w_i and w_j in the same sentence of the corpus (C stands for Count).

It is possible to consider the probability of link P(i, j) as the link weight d(i, j), that is, link with the higher probability has the higher weight. The problem of the data sparseness is solved as in [3], namely, the following estimation is used:

$$d(i, j) = E = \lambda 1 E1 + (1 \lambda 1)(\lambda 2 E23 + (1 \lambda 2) E4) \quad (5)$$

where

$E1 = CR1/C1$; $E23 = (CR2+CR3) / (C2+C3)$ $E4 = CR4/C4$
$CR1 = C(w_i, w_j, R)$; $C1 = C(w_i, w_j)$,
$CR2 = C(w_i, *, R)$; $C2 = C(w_i, *)$,
$CR3 = C(*, w_j, R)$; $C3 = C(*, w_j)$,
$CR4 = C(*, *, R)$; $C4 = C(*, *)$.

(* means any word, C stands for *Count*, CR stands for *Count of Relations*)

Parameters $\lambda 1$ and $\lambda 2$ are defined experimentally. We accept the values presented in [4], namely $\lambda 1=0.7$, $\lambda 2=0.3$.

3.2 Algorithm of parsing

Traditional methods of parsing use algorithm of dynamic programming which demands $O(n^5)$ operations. For the bi-gram-based parsers $O(n^3)$ algorithms are developed (Smith, Eisner, 2007) [9]. The following algorithm builds projective tree T* over sequence of nodes {1, …, n} in $O(n^2)$, it is very effective and simple in realization.

```
PARSING OF LOCAL DEPENDENCY (W)
1 n = length (W)
2 do while n> 0
3 dmax = max d (i, j) // where i, j there are the open nodes of adjacent units Uap, Ubq
4 (wi, wj) -> T*
5 Ua (p+q+1) = stick_together (Uap, Ubq, i, j))
6 n=n-1
7 end do
8 return (T)
```

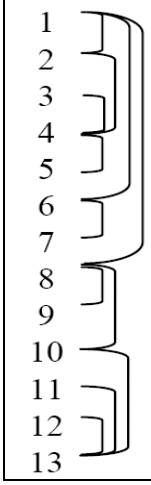
Fig. 1 Algorithm of local dependency parsing

Function *stick_together* (Uap, Ubq, i, j) deletes units Uap, Ubq, creates a new unit Ua(p+q+1) and closes all nodes lying in the interval between i and j. This algorithm of local dependency parsing (LDP) demands $O(n^2)$ operations for analysis of the sentence of n words. We will prove this statement.

■ On the last step of the cycle we need to establish links between two units spanning the whole sentence. For this purpose we shall find the maximum weight link between the open nodes of these units. In the worst case units have equal length and all their nodes are open. We need to do n/2 * n/2 i.e. $n^2/4$ comparisons to choose the maximum link. On the previous step each of units is halved and we need to do $2*n^2/16$ comparisons. On n-i step it is required to do $2^i*(n^2/2^{2i}) = n^2/2^i$ comparisons. Summarizing by i, we receive the overall number of comparisons for the worst case of analysis:

$n^2 * \Sigma\, 1/2^i$

The sum converges to 1, and the overall number of operations = $O(n^2)$ ■



A	B	d
6	7	1.4090
3	4	1.2619
2	4	1.1848
1	2	1.0366
4	5	1.1446
1	6	1.0017
1	8	0.0206
12	13	0.0062
11	13	0.0062
10	13	0.0062
8	9	0.0003
8	10	1.0000

According to values W of weights the links between the neighboring words 6-7, 3-4 are established.
Then link 2-4 is established, thus node 3 becomes closed.
…
After establishing link 4-5 units 2-4 and 4-5 merge
Link 1-6 closes nodes 2, 4, 5
…
Link 8-10 is the last one though its weight is larger than the weight of the previously established links because link 8-9 should be established beforehand.

Fig. 2 Example of the algorithm run

3.3 Creation of the training corpus

Two methods, which were used to mark the raw text corpus for LLM training, are described in this section:

(i) Gathering of statistics of the grammatical features of n-gramms, n=3,

Grammatical features were coded according to the Zalizniak's Grammatical dictionary. The morphological homonymy was not disambiguated, instead the grammatical features of the homographs were split: if a word form was attributed to m various grammatical codes, the statistics of each of these codes is increased by 1/m.

(ii) Gathering of statistics of the k-character endings of n-gramms, k=4, n=5.

As Russian is an inflectional language, the statistics of the k-character endings was used in parallel with the statistics of grammatical features, and also for the internal testing of the method. Collection of texts http://www.lib.ru of about 2 GBytes was used for the statistics gathering.

Iterative training of the model.

1. Each sentence of the training corpus is parsed according to algorithm of Fig. 1. The initial values of weight of link $d(i, j) = C(w_i, w_j, R)/C(w_i, w_j)$, $|i-j| < 5$ are accepted on the basis of the collected statistics (i) or (ii).

2. New values for E1, E23, E4 and E are calculated according to the results of parsing (5).

Parsing of each sentence with the new values of link weights is carried out. Step 2 is repeated until the alternation of link weights becomes less than the preset threshold.

3.4 Results of experiments

Collection of the short stories by A.P. Chekhov about 1 Mb in volume is chosen as the experimental corpus. Usually punctuation marks are an important source of information in the parsing procedure. However we intend to parse the free speech utterances where punctuation is absent. So all punctuation marks were neglected.

One marked sentence is presented in Fig. 3 (the story "Playwright"). Words of the sentence with the word number, the established links and the table "DEPENDENCY" are depicted. Columns A and B contain numbers of the connected nodes, W – the link weight, in the right column - a checkbox for the link. The checkbox allows excluding the false links.

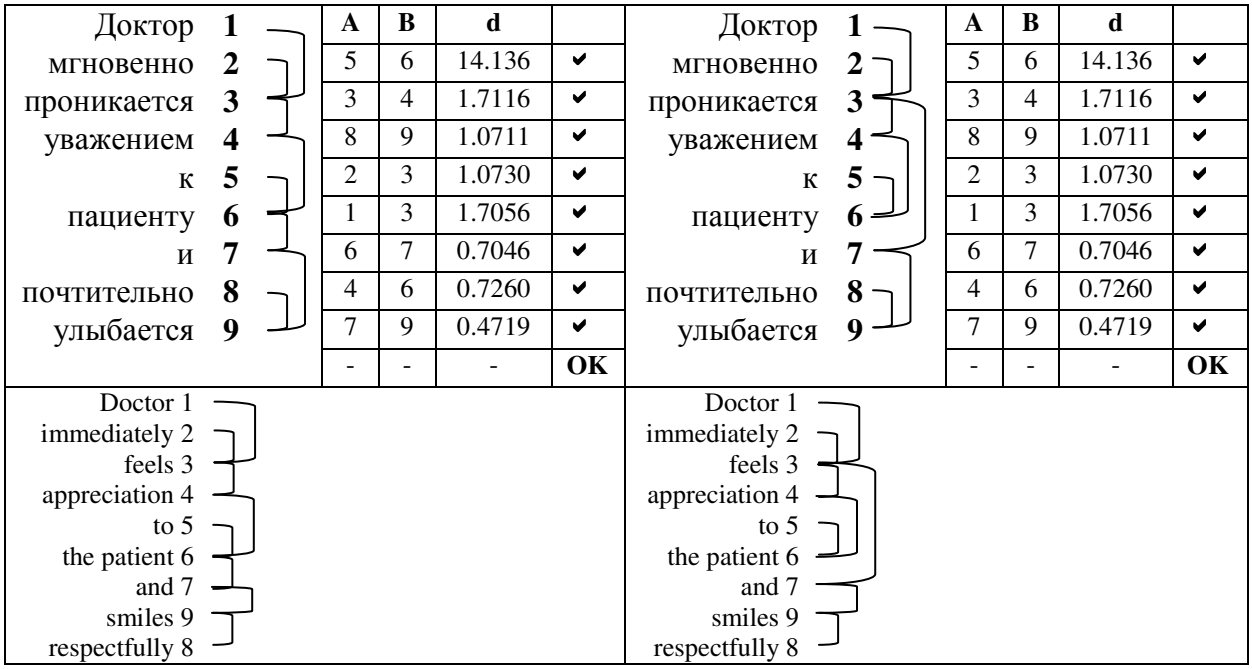


Fig. 3 Stucture of the sentence after the 1st and the 4th iteration

This example represents achievement of correct analysis after a small number of iterations. Analysis of the majority of sentences, however, contains false links which are not eliminated even after the 10th iteration. Counting of correct and false links is carried out usually by comparison with the «gold standard», i.e. with the corpus of the certainly correctly parsed sentences. Unfortunately, such gold standard for Russian is not available in the public domain. Therefore we expect to execute expert check of the parse trees. The preliminary evaluation of the results gives the following figures:

Number of the analyzed sentences (99 short stories by A.P. Chekhov)	14058
Number of words	191307
Average sentence length	~ 13.6 words
Number of the dependencies established	177131
Number of manually reviewed randomly selected sentences	1000
Ratio of the correct dependencies to all established dependencies	~ 0.746

A group of experts will be asked to check the rest of sentences to assess the algorithm, and, foremost, to improve the weights of links.

4 Incorporating semantic knowledge

The semantics of syntactic role fillers are usually determined by their lexical, semantic and morpho-syntactic properties, instead of position in the sentence especially for such languages as Russian with free constituent order. Case frame for the Russian predicate is an entry of a case frame lexicon. Such entry should contain semantic features of the word and of its valences. These features serve to impose constrains on the links between the words in the sentence. Within our approach rigid constrains are not aloud, instead we can decrease the weight of dependency d_{ij} if semantics of words w_i and w_j is incompatible, i.e. if w_i is the master and w_j is the slave, no semantics of valence of w_i coincide with semantics of w_j.

We have chosen only about 130 semantic features with tree-like hierarchic structure. A word may have more than one semantic feature. Certainly, one has to attach case frame to each predicate

and semantic features to each word manually. We expect to do it for 1000 most frequent verbs and 5000 most frequent nouns of Russian. After incorporating semantics the syntactic parsing will map the semantic structure of the sentence.

5 Conclusion

The model of local dependency in which the linguistic restrictions of the sentence structure – the probability of links, and also projective character of the sentence was presented. The new algorithm of grammatical analysis which searches the dependency tree in the bottom-up order is proposed. The algorithm establishes local links between the neighboring words and groups of words.

After analysis of all sentences of the corpus the links weights are improved, then analysis of all sentences is carried out, etc. – in an iterative mode. Experiments show, that results of analysis improve after several iterations, however not for all variants of grammatical and lexical structure of sentences.

There are some possibilities for the model perfection. In particular, at formation of the unit, it is possible to check, whether it is a steady or a terminological word-combination, and to process it accordingly. It is supposed to include check of grammatical restrictions explicitly in the algorithm (e.g., the noun and adjective coordination, an interdiction for link of a preposition with more than one noun, etc.). Further, it is possible to transform an undirected tree into a directed one by considering each open node of a tree (that is, node with no links over it) as a root of the tree, calculating statistics for the formed directed links and choosing the most probable variant.

In order to avoid the lack of efficiency characterizing a syntax parser, it avoids exploding the structural ambiguities, supplies the next stage - semantic interpreter with knowledge about syntactic connections between the words occurring in the text. The isomorphism between syntax and semantics should be accounted into a limited set of formal mapping rules and conditions. Prepositional phrase attachment, apposition, determination of conjunction's scope and modification of a NP through other NPs are dealt in a satisfactory way. Other complex linguistic phenomena (as anaphora, quantification and ellipsis) require a more extensive use of heuristics. The future work will concentrate on these specific aspects in order to check the adequacy of the hypothesis of isomorphism between syntactic and semantic structures to larger fragments of the Russian language. As the model of local dependency is applicable to the languages with projective sentences, and thanks to high speed of parsing, this model and LLD algorithm can be used for languages with the limited linguistic resources, even in absence of the morphological analyzer.

References

1 Billot S., Lang B. The Structure of Shared Forests in Ambiguous Parsing // Proceedings ACL 1989.

2 Bod R. An all-subtrees approach to unsupervised parsing // Proceedings of COLINGACL

3 Collins M., Hajic J., Brill E., Ramshaw L., Tillmann C. A statistical parser for Czech // Proceedings of the 37th Meeting of the Association for Computational Linguistics (ACL), pp. 505–512

4 Gao J., Suzuki H.: Unsupervised learning of dependency structure for language modeling // ACL 2003, pp. 521–528.

5 Goodman J. Efficient algorithms for parsing the DOP model // Proceedings Empirical Methods in Natural Language Processing 1996, Philadelphia, PA: 143-152.

6 McDonald R., Satta G. On the complexity of non-projective data-driven dependency parsing // Proceedings of the International Conference on Parsing Technologies (IWPT)

7 Mel'čuk I. Dependency Syntax: Theory and Practice // Albany, N.Y.: The SUNY Press, 1988

8 Nivre J An efficient algorithm for projective dependency parsing. // Proceedings of International Workshop on Parsing Technologies, pp. 149–160

9 Smith D.A., Eisner J. Bootstrapping feature-rich dependency parsers with entropic priors // Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 667–677

10 Seginer Y Fast Unsupervised Incremental Parsing // Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 384–391, Prague, Czech Republic, June 2007.

11 Sowa J.F., Conceptual Structures // Addison Wesley, 1984.

12 Ножов И.М. Реализация автоматической синтаксической сегментации русского предложения // Дисс. Канд. Техн. Наук – М.: РГГУ, 2003 (Nozhov I.M. Implementation of automatic syntactic segmentation of the Russian sentence, PhD thesis, Moscow, 2003; http://bankrabot.com/work/work_7895.html)