HIGH-PROBABILITY SYNTACTIC LINKS

Leonid Mitjushin
Institute for Problems of Information Transmission
Russian Academy of Sciences
19 Ermolovoy Street, 101447 GSP-4, Moscow, Russia

1 Introduction

In this paper we consider syntactic relations between words of a sentence that can be strongly predicted by local mechanisms. For instance, if a sentence contains a pair of words

... red block ...,

then the reader immediately makes a conjecture that *red* is an adjective modifier for the noun *block*. The same is true for semantically abnormal pairs such as

... green ideas ... .

Other examples of strong prediction are provided by pairs

... authors describe ... ,
... problem is ... ,

for which a "subject - verb" relation takes place with high probability.

In most cases, such simple hypotheses prove to be correct. However, sometimes they lead to errors, as for the pair *problem is* in the sentence

(1) *The solution of this problem is very simple.*

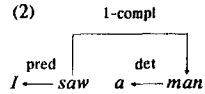
In this example, however, by the moment the word *is* has been read, the word *problem* is already engaged in other strongly predicted constructions, namely the prepositional phrase *of this problem* and even the whole noun phrase *the solution of this problem*. A conflict arises, and plausibility of the new hypothesis becomes much lower.

Such syntactic relations may concern not only adjacent words. For instance, in (1) it is for the pair *solution ... is* that the "subject - verb" relation will be conjectured.

In this paper, strong prediction of syntactic relations is modeled within the framework of dependency syntax (see Mel'čuk 1974, 1988). According to this theory, (surface) syntactic structure of a sentence is an oriented tree whose nodes are the words of the sentence (more precisely, their lexico-morphological interpretations). The arcs of the tree represent syntactic links between words and are labeled by names of syntactic relations. The result of strong prediction is a partial parse of the sentence, in which high-probability syntactic links are established.

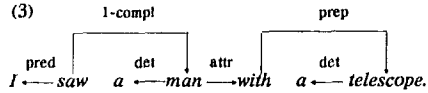
In our opinion, dependency structures are better adapted to partial parsing than constituent structures. The reason is that the dependency structure of a segment is the same both when the segment is considered as isolated and when it is considered as a part of some sentence (by "segment" we understand any sequence of words). Generally, this is not true for constituent structures. For instance, the segment *I saw a man* has the dependency structure *

(2)

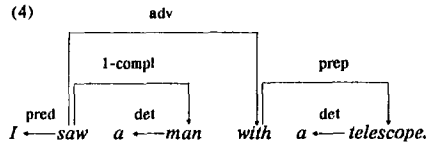


both as a complete sentence and as a part of the sentence *I saw a man with a telescope*. The fact that the latter sentence is ambiguous does not hamper anything, as both its structures contain subtree (2) (and differ only in arcs that go into the word *with*):

(3)



(4)



On the other hand, the constituent structure of the segment *I saw a man* is not fully inherited in the constituent structures of the longer sentence. In our opinion, this comparison demonstrates that, in a certain sense, dependency structures reflect the incremental nature of sentence comprehension from left to right better than constituent structures do.

In this paper we describe a bottom-up, left-to-right algorithm of partial parsing that establishes high-probability syntactic links. It is implemented on a VAX 11/750 computer as a subsystem of a multipurpose linguistic processor developed in the Laboratory of Computational Linguistics of the Institute for Problems of Information Transmission, the Russian Academy of Sciences (Apresjan et al. 1992). The partial parser is employed as a preprocessing unit before the operation of the main filter-type parser. It can also be used for automatic indexing and lemmatization.

The algorithm is language-independent: all language-specific information is recorded in the dictionaries and the rules that establish links.

* Full names of English syntactic relations that appear in examples are: predicative, determinative, 1st completive, prepositional, attributive, adverbial. The number of relations used in complete models of English and Russian syntax varies from 40 to 55 (Mel'čuk 1974; Mel'čuk and Pertsov 1987; Apresjan et al. 1989, 1992).

Experiments with Russian sentences have given promising results: on average, the algorithm establishes 70 - 80 % of syntactic links of a sentence; processing speed (exclusive of morphological analysis) is about 10 words per CPU second. The error rate is less than 1 % (stable estimates have not been obtained yet).

2 Bottom-up Parsing

The processing of a sentence begins with morphological analysis. As a result, each word is given a set of its possible lexico-morphological interpretations, henceforth called "homonyms". A homonym is a list that includes a lexeme identifier, a part-of-speech marker, and morphological features of the wordform. For instance, the morphological module of the ETAP-2 system (Apresjan et al. 1989) will give for the word *saw* the following three homonyms: SEE, V, pt (= past tense); SAW1, V, mf (= main form); SAW2, N, sg.

All morphological data are concentrated in a special morphological dictionary. The key role in parsing proper is played by a combinatorial (syntactic) dictionary that contains versatile information on syntactic properties of lexemes, i.e. on their ability to participate in various syntactic constructions (for details see Mel'čuk 1974, 1988; Apresjan et al. 1989, 1992).

The general scheme of parsing is as follows. After the morphological analysis, for each word there appears one or more homonyms. By "fragment" we shall understand a set of homonyms occupying one or more successive positions in the sentence (one homonym in each position) plus a tree of syntactic links defined on these homonyms as nodes. For instance, an isolated homonym is a trivial fragment; the whole dependency tree of a sentence is also a fragment. It should be noted that in trees (2) - (4) each word is represented by a certain homonym (for example, *saw* is represented by SEE, V, pt).

Lejkina and Tsejtin (1975) described a bottom-up process for constructing dependency trees. It is based on the operation of adjunction. This operation is applied to two adjacent fragments and consists in establishing a link, marked by a certain syntactic relation, from a certain node of one fragment to the root of the other. The result of adjunction is a new fragment on the union of segments occupied by the initial fragments.

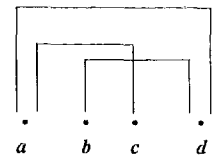
This action is similar to generation of a new constituent from two adjacent constituents. However, unlike constituents, fragments at the moment of adjunction may be "incomplete", i.e. they need not contain all the nodes that will be direct or indirect dependents of their roots in the structure of the sentence. These nodes may be added to them later (also by the operation of adjunction).

Mitjushin (1985) described the class of trees that can be constructed from isolated homonyms by repeated adjunction, i.e. that can be built by the bottom-up process. Consider a tree with an ordered set of nodes. Let a "tangle" be a quadruple of nodes (a, b, c, d) with the following properties:

1) $a < b < c < d$;
2) a and c are linked by an arc (in any direction);
3) b and d are linked by an arc (in any direction);
4) the path between a and d contains neither b nor c (here, orientation of arcs is ignored, so the path always exists).

The following criterion is true: a tree can be constructed from its nodes by repeated adjunction if and only if it contains no tangles.

The simplest tangle looks as follows:



(direction of the arcs does not matter; there can be other nodes between a, b, c, and d). According to the criterion, a tree that contains such a subtree cannot be built by the bottom-up process.

The class of trees obtainable by adjunction is much wider than the class of so-called projective trees (on projectivity see, for example, Gladkij 1985; Mel'čuk 1988). For the model of Russian syntax presented by Mel'čuk (1974) and Apresjan et al. (1992), this class includes all syntactic structures permissible in scientific, technical, and business texts (however, it is not so for spoken language and poetry). We suppose all the structures considered below to belong to this class.

3 Rules

In our system, in contrast to those based on formal grammars, the rules are not a tool for the exhaustive description of the set of correct syntactic structures. We suppose that the correspondence between sentences and their syntactic structures is defined by some other means. The task of the parsing algorithm and the rules it employs is to build, for a given sentence, some set of its syntactic structures or their fragments, without losing the semantically correct ones.

The concrete function of the rules is to check whether the given case of adjunction is feasible and, if so, to perform the operation of adjunction. Some additional operations can also be performed. The rules have access to any information about the structure of fragments to be adjoined and the homonyms they contain (their lexeme names, morphological features, and syntactic properties stated in the combinatorial dictionary). The rules may also use data on punctuation and limited data on homonyms not belonging to the given two fragments; they have no access to information about fragments built by the algorithm earlier.

While formally the rules could be strongly context-sensitive within the limits of two given fragments, in most cases they only use information on nodes X and Y (those to be linked) and their nearest syntactic context. In fact, the rules currently emloyed do not

consider nodes for which distance from X or Y exceeds 3 (where distance is the number of links in the path connecting two nodes in the dependency tree of a fragment).

A rule is a program written in the form of a transition graph, with an elementary predicate or operation associated with each arc. The rule interpreter performs ordered search to find a path along "true" arcs that starts at a fixed entry node and ends at one of fixed exit nodes. No backtracking is used: if forward motion from some node proves to be impossible, interpretation is terminated. The fact that backtracking is not necessary has been discovered empirically; it is connected with the nature of syntactic events considered by the rules. On the other hand, when desirable, an explicit return may be made to a point passed earlier, with simple measures taken against infinite cycling.

Each rule contains at least one operation *LINK(X,Y,R)* that establishes a link marked by a certain syntactic relation R between the given node X of one fragment and the root Y of the other (that is, performs the adjunction). The corpus of rules covers only those situations for which the probability that the established links are correct is estimated as close to 1. For instance, the rules do not establish links like attr and adv in structures (3) and (4) because attachment of prepositional postmodifiers is known as a "classical" case of structural ambiguity.

It should be noted that the probability close to 1 characterizes here not individual links (it would be too strong a demand) but all complex of links established for the given words. This can be illustrated by the segment *I saw*, for which two fragments will be built with different homonyms for the word *saw*:

$$\mathrm{I} \xleftarrow{\text{pred}} \mathrm{SEE}_{\mathrm{V,pt}}, \qquad \mathrm{I} \xleftarrow{\text{pred}} \mathrm{SAW1}_{\mathrm{V,mf}}.$$

Both these alternatives are syntactically correct. At the same time, they are mutually exclusive, and it is only their disjunction that has probability close to 1. This ambiguity is also inherited by larger fragments. (As a result, the sentence *I saw a man with a telescope* has four different parses, two of which are semantically abnormal.) Thus, high probability is a "collective" and not an "individual" property of links. Rigorous definitions can be found in the paper by Mitjushin (1988).

4 The Parsing Algorithm

The simplest method of bottom-up parsing is to consider all opportunities for adjunction, starting from adjacent one-element fragments. We employ a faster algorithm, in which certain heuristics are used to reduce search (Mitjushin 1988).

The algorithm builds a growing sequence A of fragments. At any moment of time A contains some homonyms of the sentence and certain fragments constructed of these homonyms. The algorithm moves from the beginning of the sequence A to its end and tries to perform adjunction between the current fragment $F \in A$ and the fragments that appear in A earlier than F. New fragments are added to the end of the sequence.

The fragment considered at the given moment is called active. All fragments of A (including isolated homonyms) become active successively, without leaps or returns.

While the algorithm moves along the sequence A, the sequence grows longer because of the addition of newly built fragments. Nevertheless, a moment will necessarily come when the active fragment is the last in A and further motion is impossible. In this case, the next homonym of the sentence is added to the sequence; it becomes active and the work is continued. When a new deadlock arises, another homonym is added, and so on. If in such a situation it turns out that all homonyms of the sentence are exhausted, then the work is finished.

Homonyms are added to the sequence in the order they are arranged in the sentence from left to right (which is essential), and those occupying the same position are added in an arbitrary order (in this case, the order has no influence on the results). At the initial moment A contains a single element, namely one of the homonyms occupying the leftmost position of the sentence, and it is declared active.

For each active fragment F the algorithm selects in A its left neighbors, i.e. fragments that are adjacent to F on its left. A preference relation is defined between the neighbors of F: fragments of greater length are preferred, and those of equal length are considered equivalent.

For the given F, the algorithm considers its left neighbors E in order of their preference, and for each E tries to adjoin it to F. If for some E adjunction is successful, subsequent search is limited to the neighbors of F equivalent to E; less preferred fragments are not considered.

An attempt to adjoin E to F is made as follows. Links are considered that connect a certain node X of fragment E with the rightmost node Y of fragment F. A preference relation is defined between the links: those of greater length are less preferred, and those of equal length are equivalent. In other words, more preferred are links $X \rightarrow Y$ and $X \leftarrow Y$ with nodes X that are nearer to the right end of E; links with the same X are equivalent.

For the given E and F, nodes $X \in E$ are considered from right to left (i.e. in order of the preference of links between X and Y), and for each X the rules applicable to these X and Y are activated. The list of such rules is determined by parts of speech of X and Y, and by possible direction of the link. If during interpretation of a rule an operation *LINK(X,Y, .)* or *LINK(Y,X, .)* is performed then a new fragment is built which is the result of joining X and Y with the given link. It is placed at the end of the sequence A. After that, for these E and F the search is limited to the links equivalent to the established one; less preferred links are not considered.

When the sequence A is built, its subset C of maximal fragments is formed. A fragment is called maximal if its segment is not a proper part of the segment of any other fragment belonging to A. The set C is the final result of partial parsing. Below, when speaking

about fragments built by the algorithm, we shall always mean exactly the set C.

The first experiments with this algorithm have shown that, in some cases, the preferences and restrictions adopted are too strong and prune away semantically correct parses. To improve the situation, special operations were defined that made it possible to cancel (from inside the rule) priority of longer neighbors or shorter links, and also to make the algorithm consider not only the rightmost node of the right fragment. Owing to them, the search can be made exhaustive in all cases when the rule "considers it desirable". In the real process of parsing, these operations are fired not too often, so the main part of search remains limited.

5 Experiments

At present, after preliminary debugging and tuning of the rules, we have begun to carry out regular experiments with a homogeneous flow of Russian texts. The experiments make use of a computer-oriented combinatorial dictionary of Russian compiled by a group of linguists under the guidance of Ju.D.Apresjan (see Apresjan et al. 1992). It contains over 10,000 entries, mainly general scientific vocabulary and terms from computer science and electrical engineering.

The number of rules in the system is now about 100. Total number of arcs in their transition graphs is about 2,000.

As a source of texts, we have taken several issues of the journal *Computer Science Abstracts* (Referativnyj zhurnal *Vychislitel'nyje Nauki*, in Russian). Sentences are chosen at random. Sentences with formulas, occasional abbreviations, and non-Cyrillic words are excluded. Words absent in the dictionaries (about 8% of all word occurences in these texts) are replaced by "dummy" words that have syntactic properties most probable for the given category. At present, about 300 sentences have been processed.

On the average, fragments produced by partial parsing include 3 - 4 words. It is not infrequent that they have 8 - 10 or more words, or present complete structures of sentences. On the other hand, a substantial part of fragments are isolated homonyms. For instance, subordinate conjunctions remain isolated in most cases because, as a rule, their links with other words are not considered having high probability.

Frequently enough morphological, lexical, and structural ambiguity results in building 2 - 4 different fragments on the same segment. Sometimes their number is 8 - 12 and more, but such cases are relatively rare. The record is now equal to 72 fragments on a segment of 9 words. For such cases, packing techniques can be developed similar to those described by Tomita (1987). Another possible method is to employ numerical estimates of syntactic preference (see, for example, Tsejtin 1975; Kulagina 1987, 1990; Tsujii et al. 1988).

On the average, the number of established links is 70 - 80 % of the total number of syntactic links in the sentence. These figures include links present both in the fragments built and in the semantically correct structure of the sentence; "extra" links that arise due to ambiguity of fragments are not included.

Sometimes the fragments overlap, that is, their segments intersect. It happens approximately in one tenth of sentences. As a rule, in such cases the correct result is a combination of one of the overlapping fragments with its "truncated" competitor.

A fragment is called correct for a given sentence if it is a subtree of the semantically correct dependency tree of this sentence (or of one of such trees, in the rare cases of real semantic ambiguity like (3) - (4)). A fragment is called feasible if it is a subtree of some dependency tree of some sentence of the given language. The algorithm makes an error in the following two cases: (a) if a non-feasible fragment is built; (b) if all fragments built on some segment are feasible but none is correct. (Here we do not take into account semantically abnormal sentences or the possibility of overlapping; these situations would require more accurate definitions.)

In most cases, an error means that some link of a fragment is established erroneously, while all the others are correct. The experiments have shown that the frequency of errors for the algorithm described is fairly small. For the last 100 sentences, 12 errors were made (9 of the first type and 3 of the second), which is less than 1 % of the total number of links established in correct fragments. A stable estimate is not yet obtained because at this stage of experiments tuning of the rules is continued, and the error frequency decreases steadily.

Errors of the first type are caused by inaccuracy of the lexicographic descriptions and imperfection of the rules. In the presence of adequate lexicographic information, these errors in principle are avoidable, as the rules may fully control internal properties of the fragments being created.

The second type of error is intrinsic to our approach. The rules employed are local in two respects: they take no (or almost no) account of the context outside the fragments being adjoined, and they take no account of a very large part of syntax that concerns less probable links. The first restriction means that fragments may appear which are grammatically feasible but do not agree with the context. The second one implies that we do not intend to obtain complete structures of sentences, and therefore shall not be able to reject a fragment for the reason that it is not engaged in any complete structure.

In general, it is not at all surprising that a certain part of syntactic links can be reliably revealed by local mechanisms. Any flow of texts in any language must contain chains of words the parse of which weakly depends on the context ("weakly" can be understood here in the statistical sense: the share of those occurences for which the parse differs from the most probable one is small). The possibility of examining fragments in any detail permits to avoid situations in which the risk of creating a non-feasible fragment is too large.

A more surprising fact is that the number of reliably established links is rather high - about 75 %. For the most part, these are links typical of the basic, most frequent syntactic constructions such as "adjec-

tive + noun", "preposition + noun", "numeral + noun", "adverb + verb", and also a large group of links connecting predicate words with their arguments. As regards the last type, preference for the predicate-argument interpretation of word combinations was often noted in the literature (this preference is a particular case of the Most Restrictive Context Principle proposed by Hobbs and Bear (1990)).

Observations show that the number of established high-probability links noticeably depends on the type of text. The general trend is as follows: the more "formal" the text is, the more links are established. From this point of view, the language of scientific abstracts suits the given approach quite well.

As regards comparative frequency of high-probability links in different languages, it would be natural to expect these links to be more typical of languages with rich morphology than of analytical ones (such as English). Nevertheless, preliminary experiments have shown no substantial difference in this respect between English and Russian scientific texts.

We suppose that in case of high-probability links, the efficiency of local approach is additionally augmented due to factors "of the second order" concerning general mechanisms of text comprehension and generation. This opinion is based on the following assumptions. If someone reading a text sees that a high-probability link is possible between certain words and this link is compatible with the previous part of the text, then he makes a conjecture that this link is correct; such conjecture is abandoned only if some counter-evidence is obtained. When people generate texts, they take into account this property of the comprehension mechanism and tend not to disappoint expectations of the readers. In other words, they are careful not to create high-probability links that would prove to be incorrect. This can be regarded as an instance of cooperation in language performance (cf. the Cooperative Principle in pragmatics formulated by Grice (1975)).

References

Apresjan, Ju.D., I.M.Boguslavskij, L.L.Iomdin, A.V.Lazurskij, N.V.Pertsov, V.Z.Sannikov, and L.L.Tsinman. 1989. *Lingvisticheskoje Obespechenije Sistemy ETAP-2.* Nauka, Moscow. ('The linguistics of the ETAP-2 system', in Russian)

Apresjan, Ju.D., I.M.Boguslavskij, L.L.Iomdin, A.V.Lazurskij, L.G.Mitjushin, V.Z.Sannikov, and L.L.Tsinman. 1992 (forthcoming). *Lingvisticheskij Protsessor dlja Slozhnykh Informatsionnykh Sistem.* Nauka, Moscow. ('A linguistic processor for complex information systems', in Russian)

Gladkij, A.V. 1985. *Sintaksicheskije Struktury Estestvennogo Jazyka v Avtomatizirovannykh Sistemakh Obshchenija.* Nauka, Moscow. ('Syntactic structures of natural language in automatic dialogue systems', in Russian)

Grice, H.P. 1975. Logic and Conversation. In P.Cole, J.L.Morgan, editors. *Syntax and Semantics*, Vol. 3, Academic Press, New York, pp. 41 - 58.

Hobbs, J.R. and J.Bear. 1990. Two Principles of Parse Preference. In *Proceedings of COLING-90*, Vol. 3, Helsinki, pp. 162 - 167.

Kulagina, O.S. 1987. *Ob Avtomaticheskom sintaksicheskom Analize Russkikh Tekstov.* Preprint No. 205, Institute for Applied Mathematics, Moscow. ('On automatic parsing of Russian texts', in Russian)

Kulagina, O.S. 1990. *O Sintaksicheskom Analize na Osnove Predpochtenij.* Preprint No. 3, Institute for Applied Mathematics, Moscow. ('On preference-based parsing', in Russian)

Lejkina, B.M. and G.S.Tsejtin. 1975. Sintaksicheskaja Model' s Dopushchenijem Ogranichennoj Neprojectivnosti. In *Mezhdunarodnyj Seminar po Mashinnomu Perevodu*, Moscow, pp. 72 - 74. ('A syntactic model allowing limited non-projectivity', in Russian)

Mel'čuk, I.A. 1974. *Opyt Teorii Lingvisticheskikh Modelej "Smysl ↔ Tekst".* Nauka, Moscow. ('Toward a theory of Meaning ↔ Text linguistic models', in Russian)

Mel'čuk, I.A. 1988. *Dependency Syntax: Theory and Practice.* State University of New York Press, Albany.

Mel'čuk, I.A. and N.V.Pertsov. 1987. *Surface Syntax of English: A Formal Model within the Meaning ↔ Text Framework.* John Benjamins, Amsterdam.

Mitjushin, L.G. 1985. Dlina Sintaksicheskikh Svjazej i Induktivnyje Struktury. In *Semiotika i Informatika*, No. 26, Moscow, pp. 34 - 51. ('Length of syntactic links and the class of inductive structures', in Russian)

Mitjushin, L.G. 1988. O Vysokoverojatnykh Sintaksicheskikh Svjazjakh. In *Problemy Razrabotki Formal'noj Modeli Jazyka* (series "Voprosy Kibernetiki", No. 137), Moscow, pp. 145 - 174. ('On high-probability syntactic links', in Russian)

Tomita, M. 1987. An Efficient Augmented-Context-Free Parsing Algorithm. *Computational Linguistics*, Vol. 13, No. 1 - 2, pp. 31 - 46.

Tsejtin, G.S. 1975. Metody Sintaksicheskogo Analiza, Ispol'zujushchije Predpochtenije Jazykovykh Konstruktsij: Modeli i Eksperimenty. In *Mezhdunarodnyj Seminar po Mashinnomu Perevodu*, Moscow, pp. 131 - 133. ('Parsing methods based on preference of the language constructions: models and experiments', in Russian)

Tsujii, J., Y.Muto, Y.Ikeda, and M.Nagao. 1988. How to Get Preferred Readings in Natural Language Analysis. In *Proceedings of COLING-88*, Vol. 2, Budapest, pp. 683 - 687.