An Agreement Corrector for Russian

Leonid Mitjushin
Institute for Information Transmission Problems
Russian Academy of Sciences
Bolshoj Karetnyj Per. 19
101447 GSP-4 Moscow, Russia
mit@ippi.ac.msk.su

Abstract

The paper describes an application-oriented system that corrects agreement errors. In order to correct a sentence with such errors, an extended morphological structure is created which contains various grammatical forms of the words used in the sentence. For this structure the bottom-up parsing is performed, and syntactic structures are found that contain minimal number of changes in comparison with the original sentence. Experiments with real sentences have shown promising results.

1 Introduction

Correction of agreement errors in Russian texts is a problem of real practical interest. Being a language of the inflectional type, Russian has a rich system of word changing. The paradigm of a typical verb contains 30 finite forms and 190 participles, as well as infinitives and certain other forms. The complete adjectival paradigm contains 57 forms, and the paradigm of a noun - 12 forms. Although the number of different graphic words in a paradigm is usually less than the total number of forms, it is also rather large. For that reason, agreement errors give a high proportion of all grammatical errors in Russian texts (here and below the expression 'agreement errors' means the use of words in incorrect forms).

In this paper we describe an application-oriented system that corrects such errors. The system, called below 'corrector', uses a formal description of the Russian syntax in terms of dependency structures. In our case, these structures are directed trees whose nodes represent the words of a sentence, and whose arcs are labelled with names of syntactic relations (see Mel'čuk 1974; Mel'čuk and Pertsov 1987; Apresjan et al. 1992). The corrector is based on the general idea widely used in this kind of systems: if an input sentence is syntactically ill-formed, i.e. it cannot be assigned a syntactic structure (SyntS), the system considers minimal changes that enable it to construct a SyntS, and presents them as possible corrections (see, for example, Carbonell and Hayes 1983; Jensen et al. 1983; Weischedel and Sondheimer 1983; Mellish 1989; Bolioli et al. 1992).

A segment of a sentence is called 'syntactically connected' if a well-formed dependency tree can be constructed on it. (In terms of constituents, connectedness of a segment would mean that it can be parsed as a single constituent.) The 'degree of syntactic disconnectedness' of a sentence is defined as the least number C of connected segments into which the sentence can be partitioned. Hence, $C = 1$ if and only if the sentence can be assigned a SyntS; for an "absolutely disconnected" sentence C would be equal to the number of words. The general idea of correction can be expressed in these terms as follows: for an input sentence, which has $C > 1$, minimal changes are considered that produce sentences with $C = 1$. A more "indulgent" strategy is also possible which only requires that the value of C for new sentences should be less than the initial value, and not necessarily equal to 1.

In the case of correcting agreement errors changes concern only word forms, while the lexical content and word order of the sentence do not vary. At first, the corrector tries to improve the sentence by changing a single word; in case of failure, it tries to change a pair of words, then a triple of words and so on. Actually, particular subsets of words to be changed are not considered, but instead the bottom-up parsing is performed which constructs syntactic subtrees that contain no more than R modified words; here R is a parameter which is succesively assigned values 1, 2 ... .

At present, the linguistic information used by the corrector is not complete. The morphological and syntactic dictionaries, which describe respectively the paradigms and syntactic properties of words, cover about 15 thousand words; the grammar does not cover a number of less frequent syntactic constructions. Nevertheless, experiments show that, if supplied with a large morphological dictionary, the corrector even in its current state could effectively process real texts.

Incompleteness of the syntactic dictionary is overcome by assigning 'standard' entries to the words absent from it (but present in the morphological dictionary). A standard entry describes syntactic properties typical of the words with a particular type of paradigm. Due to incompleteness of the grammar, the corrector fails to construct SyntSs for certain well-formed sentences. (Here and below a sentence is called 'well-formed' if it has one or more SyntSs which are correct with respect to the (hypothetical) complete grammar; otherwise a sentence is called 'ill-formed'.) For that reason, all sentences whose degree of disconnectedness is less than that of the input sentence (C) are regarded as 'improvements'. If $C = 1$, the input sentence is considered correct, and if $C > 1$ and improvements have not been found, it is considered 'quasi-correct'.

The corrector was tested on sentences chosen at random from the Russian journal *Computer Science Abstracts*. The experiments are described in detail in Section 5; here we present only the main results. Of 100 sentences chosen, 95 were evaluated as correct or quasi-correct; 3 gave 'false alarms', i.e. wrong corrections were proposed; 2 cases gave system failure (exhaustion of time or memory quotas). The same 100 sentences with single random distortions gave the following results: 14 turned out to be well-formed and were evaluated by the system as correct or quasi-correct; in 79 cases the initial sentences were reconstructed; in 5 cases wrong corrections were proposed; 2 cases gave system failure. The repeated experiment with distorted sentences generated by a different series of pseudo-random numbers gave respectively the figures 10, 84, 5, and 1.

It can be said that in these experiments the difference in performance between the system described and the "ideal" corrector was 5% for correct sentences and 6 - 7% for sentences with single distortions. For more than 90% of ill-formed sentences the right corrections were found.

A natural application of an agreement corrector is to process texts in computer editors. Another possibility is to combine it with a scanner for reading printed texts. Applying this system to problems with "high noise", such as reading handwritten texts or speech recognition, seems more questionable: observations show that when the density of errors increases, the quality of correction becomes rather low.

Further development of the corrector includes as the first step incorporation of a large morphological dictionary (in the experiments the entries of words absent from the morphological dictionary were added to it before running the corrector, i.e. a "complete" dictionary was simulated). Then the syntactic dictionary and the grammar should be extended, and further debugging on real texts should be carried out.

2 Parsing the Input Sentence

The corrector begins its work with ordinary morphological analysis and parsing of the input sentence.

As a result of morphological analysis, for each word all its possible morphological interpretations (called 'homonyms') are constructed. A homonym consists of a lexeme name, a part-of-speech marker, and a list of values of morphological features, such as number, case, gender, tense, voice, and so on. The set of all homonyms built for a sentence is called its morphological structure (MorphS).

The MorphS is regarded as input information for the parser, which is based on the bottom-up principle. The bottom-up method for dependency structures was proposed by Lejkina and Tsejtin (1975). Let 'fragment' be a dependency tree constructed on a certain segment of a sentence. Precisely speaking, a fragment is a set of homonyms occupying one or more successive positions in the sentence (one homonym in each position) together with a directed tree defined on these homonyms as nodes, the arcs of the tree being labelled with names of syntactic relations (such arcs are called 'syntactic links'). A separate homonym and a SyntS are "extreme" instances of fragments.

If two fragments are adjacent in the sentence, then drawing a syntactic link from a certain node of one fragment to the root of the other creates a new fragment on the union of segments occupied by the initial fragments (this is similar to constructing a new constituent from two adjacent constituents). By such operations, starting from separate homonyms, we can construct a SyntS of the sentence provided it does not contain 'tangles' (Mitjushin 1992); though SyntSs with tangles may occur, they are very infrequent and "pathological".

Correctness of links between fragments is determined by grammar rules which have access to all information about the fragments to be linked, including the syntactic entries for homonyms in their nodes.

In the course of parsing, only the most preferred of all fragments are built (see Section 4). Besides that, many fragments are excluded at certain intermediate points. As a result, the amount of computing is substantially reduced, and though the expected total number of fragments remains exponential with respect to the length of the sentence, the cases of "combinatorial explosion" are fairly infrequent (2% in our experiments).

For the set of fragments constructed, the degree of disconnectedness C is counted as the least number of fragments covering all words of the sentence. This value of C will be denoted by $C(0)$. If at least one complete SyntS has been built, then $C(0) = 1$; otherwise $C(0) > 1$. In case $C(0) = 1$ the sentence is regarded as correct and the process terminates, in case $C(0) > 1$ an attempt is made to improve the sentence.

3 Search for Corrections

The process used to find corrections is quite similar to the ordinary parsing described in the previous section. The main difference is that the parser gets as the input not the initial MorphS but the 'extended' one, which is constructed by adding new homonyms to the initial MorphS. The new homonyms arise as the result of varying the forms of the words of the input sentence. The process of varying concerns only semantically empty morphological features, such as the case of a noun, the number, gender and person of a finite verb, the number, gender and case of an adjective or participle, and the like. Transforming finite verbs into infinitives and vice versa is also regarded as semantically empty.

As a result, for each homonym of the initial MorphS a 'set of variants' is built, i.e. a certain set of homonyms of the same lexeme that contains the given homonym. For unchangeable words and in certain other cases no real variation takes place, and the set of variants contains a single element, namely the given homonym. The precise rules for constructing variants may be found in (Mitjushin 1993). The extended MorphS is the union of the sets of variants built for all homonyms of the initial MorphS. On average, the extended MorphS is much larger than the initial: for 100 sentences from the *Computer Science Abstracts* the mean number of homonyms in the initial MorphS was $2.4n$, while in the extended one it was $12.2n$, where n is the number of words in the sentence.

The extended MorphS is processed by the parser for various values of the parameter R which limits the number of changed words in the sentence. R is succesively assigned values 1, 2, ... , $R_{\max}$ (where $R_{\max}$ is set by the user), and for each R parsing is repeated from the beginning. Let d be the number of homonyms of a certain fragment which do not belong to the initial MorphS, i.e. the graphic words of which are different from the words of the input sentence. In a sense, d is the distance between the fragment and the input sentence. In the course of parsing only those fragments are considered for which $d \leq R$; one can imagine that the parsing algorithm remains unchanged, but creation of fragments with $d > R$ is blocked in the grammar rules.

For each value of R, the degree of disconnectedness $C = C(R)$ is calculated for the results of parsing. It should be noted that if we put $R = 0$, then parsing on the extended MorphS is actually reduced to that on the initial MorphS, which justifies the notation $C(0)$ introduced in the previous section. Function $C(R)$ does not increase for $R \geq 0$. Behaviour of the corrector may in different ways depend on the values of $C(R)$, which would result in different modes of operation.

If parsing is highly reliable, i.e. the probability to construct at least one SyntS for a well-formed sentence is close to 1, and the same probability for an ill-formed sentence is close to 0, then it is reasonable to regard all sentences with $C(0) > 1$ as incorrect and carry out parsing on the extended MorphS until $C(R) = 1$ is achieved for some R, i.e. at least one SyntS is constructed. Then, replacing the homonyms of each SyntS by their graphic representations (that is, transforming them into ordinary words), we obtain hypothetical corrected sentences; some of them may be identical, as different homonyms may be represented by the same graphic words. Each of the created sentences contains exactly R words changed in comparison with the input sentence.

If $C(R) > 1$ for all $R \leq R_{\max}$, corrections cannot be constructed within this mode. However, the corrector can inform the user that the input sentence is syntactically ill-formed, and indicate the gaps in the SyntS, i.e. the boundaries of the fragments which provide the minimal covering of the sentence for $R = 0$.

In our case, due to incompleteness of the grammar, many well-formed sentences would have $C(0) > 1$. However, the majority of the missing rules describe links which do not require agreement, and so for almost all well-formed sentences $C(R) = C(0)$ for all $R > 0$, i.e. they turn out to be 'unimprovable'. Taking this fact into account, we adopted the following strategy: the least R_1 is found for which $C(R) = C(R_1)$ for all R, $R_1 < R \leq R_{\max}$. In other words, R_1 is the least value of R for which the situation becomes 'unimprovable' (within the extended MorphS constructed for the input sentence). If $R_1 > 0$, i.e. $C(R_1) < C(0)$, then for $R = R_1$ the minimal sets of fragments covering all words of the sentence are considered. Replacing the homonyms of those fragments by their graphic representations, we obtain hypothetical corrected sentences. In case of overlapping fragments, the problem of choosing among several homonyms could arise, but actually overlapping does not occur.

Experiments with the corrector showed from the very beginning that the process described often generated redundant hypothetical corrections. So an additional pruning step was introduced, which will be described for the case of SyntS (for fragments everything is quite similar). The arcs of a SyntS are assigned certain weights which express relative "strength" or "priority" of the corresponding syntactic relations; the weight of the SyntS is equal to the sum of the weights of its arcs. The maximum weight over all constructed SyntSs is counted, and only SyntSs with that weight are retained.

Though this method is simple, it proved to be quite effective and reliable: in most cases the corrector generates a single hypothetical correction, while the probability of losing the right correction is rather small.

4 Implementation

The system described is based on the "Meaning - Text" linguistic model (Mel'čuk 1974; see also Mel'čuk and Pertsov 1987) and its computer implementation - a multi-purpose linguistic processor developed at the Institute for Information Transmission Problems of the Russian Academy of Sciences (Apresjan et al. 1992).

The corrector employs the morphological and syntactic dictionaries of Russian which are part of the processor. As regards its linguistic content, the grammar of the corrector is similar to the Russian grammar used in the processor, as they describe the same correspondence between Russian sentences and their syntactic structures. However, the corrector uses a different formalism to represent rules, which partly stems from the difference in parsing methods: in the processor an algorithm of the so-called 'filtering' type is implemented, while the corrector uses an algorithm of the 'bottom-up' variety.

It should be noted that, in contrast to certain other systems (for example, Jensen et al. 1983; Weischedel and Sondheimer 1983; Véronis 1988; Chanod et al. 1992), the present corrector does not contain any 'negative' information intended specifically for correcting errors. It contains only 'positive' rules that describe correct SyntSs and their parts and are assumed to be used in ordinary parsing. Correction of errors is reduced to parsing on the extended MorphS, as described in Section 3.

In comparison with the experimental version of the system (Mitjushin 1993), in the present version the grammar is augmented, and it is made possible to process words absent from the syntactic dictionary and to consider quasi-correct sentences. Now, to make the corrector applicable to real texts, it is sufficient to supply it with a large morphological dictionary.

Such a dictionary containing about 90 thousand words has recently been compiled at IITP by Vladimir Sannikov. It is rather close by its lexical content to the *Grammatical Dictionary* by Zaliznjak (1980), but is based on the model of Russian morphology used in the linguistic processor.

Compilation of a large syntactic dictionary is a more labour-consuming task, as its entries contain more complex information. For each word there must be specified a set of relevant syntactic features (from the full list of 240 features), a set of semantic categories (from the list of 50 categories), and a government pattern which expresses the requirements that must be fulfilled by the elements representing in the SyntS the semantic actants of the word (Mel'čuk 1974; Mel'čuk and Pertsov 1987; Apresjan et al. 1992).

Verbs, nouns, adjectives, and adverbs which are present in the morphological dictionary but absent from the syntactic one are assigned one of the following standard entries:

transitive verb,
intransitive verb,
inanimate masculine noun,
animate masculine noun,
inanimate feminine noun,
animate feminine noun,
neuter noun,
adjective,
adverv.

Words of other parts of speech constitute closed classes and must be present in the syntactic dictionary. The standard entries contain "generalized" information which is typical of words of the specified categories. A verb is assumed to be transitive if its paradigm contains passive forms. The gender and animacy of a noun are explicitly indicated in its paradigm.

Although this method is rather approximate by its nature, it works quite well: in most cases standard entries do not prevent the parser from building correct or "almost correct" SyntSs (the latter differing from the former in names of relations on certain arcs). The reason of this is, on the one hand, that the majority of words with highly idiosyncratic behaviour are present in the 15-thousand dictionary of the linguistic processor, and, on the other hand, that syntactic peculiarities of words are often irrelevant to specific constructions in which they occur (for instance, consider the first two occurences of the verb *be* in the sentence *To be, or not to be: that is the question*).

The algorithms by which the corrector constructs the initial and extended MorphSs are similar to the algorithms of morphological analysis and synthesis used in the linguistic processor.

Due to space limitations, we cannot describe the parsing algorithm in detail and give only a sketch. The parsing, i.e. constructing fragments by the bottom-up procedure, is performed in three stages, in the order of decreasing predictability of syntactic links. The parser intensively exploits the idea of syntactic preference used in a wide range of systems based on various principles (see, for example, Tsejtin 1975; Kulagina 1987, 1990; Tsujii et al. 1988; Hobbs and Bear 1990).

At the first stage the parser constructs fragments containing 'high-probability' links; as a result, on average 70 - 80% of all syntactic links of a sentence are established (for details see Mitjushin 1992). At the second stage the fragments are connected with "weaker" and more ambiguous links, like those between a verb or noun and a modifying prepositional phrase. At the third stage "rare" and/or "far" links are established, such as coordination of independent clauses. At the second and third stage attempts are also made to establish links of previous stages, as they could be not established at their "own" stage because of missing intermediate links of the later stages.

At each stage the sentence is looked through from left to right, and attempts are made to link each fragment with its left neighbours. A strong system of preferences is used which substantially reduces the number of arising fragments. Its main points are: longer fragments are preferred to shorter ones; links of earlier stages are preferred to those of later stages; shorter links are preferred to longer ones. The general rule requires that only the most preferred of all possible actions should be considered. Only if they all fail, the actions of the next priority level are considered, and so on.

After each stage only 'maximal' fragments are retained (a fragment is maximal if its segment is not a proper part of the segment of any other fragment). The process terminates after the stage at which complete SyntSs have arisen; otherwise the

Table 1. Results for distorted sentences which are not correct or quasi-correct (P is the set of corrections proposed).

	Number of sentences in P	Number of cases	
		1st series	2nd series
Right correction: P contains the initial sentence	1	74	70
	2	5	10
	3	0	4
Wrong correction: P does not contain the initial sentence	1	5	5
	2	0	0
	3	0	0
System failure		2	1
	Total:	86	90

fragments left after the third stage are regarded as the final result of parsing.

It should be noted that grammar rules, by means of special operations, can change priorities of links and fragments in order to widen the search if there is a danger to "lose" correct fragments. They can also mark the fragments which must be retained after the stage even if they are not maximal. As the rules have access to all information about the fragments they consider, this makes it possible to control the parsing process effectively enough depending on the specific situation in the sentence.

5 Preliminary Experiments

In order to evaluate performance of the corrector, 100 sentences were chosen at random from the journal *Computer Science Abstracts* ('referativnyj zhurnal *Vychislitel'nye Nauki*', in Russian). The sentences had to have no more than 50 words and to contain no formulas or words in Latin alphabet. The words absent from the morphological dictionary were added to it before the experiments (such words covered about 5% of all word occurences in those sentences).

The chosen 100 sentences were processed by the corrector. Then a single random distortion was made in each sentence, and the 100 distorted sentences were processed (this was made twice, with different series of pseudo-random numbers used to generate distortions). As only single distortions were considered, it was fixed $R_{\max} = 2$.

The 100 initial sentences gave the following results. In 75 cases SyntSs were built; 20 sentences were evaluated as quasi-correct, i.e. they had $1 < C(0) = C(1) = C(2)$; for 3 sentences wrong "corrections" were proposed; in one case the time limit (120 seconds) was exceeded; one case gave an overflow of working arrays. Thus, the corrector's reaction was right for 95 sentences.

Distortions were generated as follows. A word of the sentence was chosen at random for which the number of homonyms in the extended MorphS was greater than that in the initial one (the mean number of such "changeable" words in a sentence was 14.3, while the mean length of a sentence was 17.6 words). A list of different graphic words corresponding to those homonyms was built (on average, it contained 7.7 words), and one of the words different from the initial word was chosen at random. All random choices were made with equal probabilities for the results. An additional condition was imposed that the initial word should belong to the set of variants of the new one (sometimes it may not hold). If this was not fulfilled, generation of a distorted sentence was repeated.

Some of distorted sentences turn out to be well-formed (for the distortions described, the probability of this is about 10 - 15%). In most cases such sentences are semantically and/or pragmatically abnormal. However, it cannot be established on the syntactic level, just as a spelling corrector is helpless if a word is transformed by an error into another existent word.

There were 14 well-formed sentences in the first series of distorted sentences, and 10 in the second series. The corrector evaluated all those sentences as correct or quasi-correct. The results for the other distorted sentences are shown in Table 1. On the whole, for the first series of distorted sentences the corrector's reaction was right in 93 cases, and for the second in 94 cases.

No regular experiments were carried out for sentences containing more than one distortion. Our experience suggests that if the number of distorted words is small and they are syntactically isolated, i.e. the corresponding nodes are not too close to each other in the SyntS of the original sentence, then the system corrects each distortion independently of the others, as if it were the only one in the sentence. On the other hand, for massively distorted (and not too short) sentences probability of good results is rather low.

The mean processing time on the MicroVAX 3100 computer was 11.2 seconds for an initial sentence (0.64 seconds per word) and 11.4 seconds for a distorted one. Faster performance may be expected when the grammar is enlarged, because the proportion of sentences with SyntSs in comparison with quasi-correct ones will become higher. For quasi-correct sentences parsing must be performed for all $R \leq R_{\max}$, while for sentences with SyntSs it must be done only for $R = 0$ (if a correct sentence is to be checked) or for $R \leq K$ (if K distortions are to be corrected). In our experiments, for initial sentences with SyntSs the mean processing time was 2.6 seconds (0.17 seconds per word, the mean length of such sentences being 15.5 words), and the mean time of parsing was 0.6 seconds.

Acknowledgements

This work was supported by INTAS grant 94-3509. I am grateful to the anonymous referees for comments on the preliminary version of the paper.

References

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

Bolioli, A., Dini, L., and Malnati, G. 1992. JDII: Parsing Italian with a robust constraint grammar. In *Proceedings of COLING-92*, Vol. 3, Nantes, pp. 1003 - 1007.

Carbonell, J.G. and Hayes, P.J. 1983. Recovery strategies for parsing extragrammatical language. *American Journal of Computational Linguistics*, Vol. 9, No. 3 - 4, pp. 123 - 146.

Chanod, J.-P., El-Bèze, M., and Guillemin-Lanne, S. 1992. Coupling an automatic dictation system with a grammar checker. In *Proceedings of COLING-92*, Vol. 3, Nantes, pp. 940 - 944.

Hobbs, J.R. and Bear, J. 1990. Two principles of parse preference. In *Proceedings of COLING-90*, Vol. 3, Helsinki, pp. 162 - 167.

Jensen, K., Heidorn, G.E., Miller, L.A., and Ravin, Y. 1983. Parse fitting and prose fixing: getting a hold on ill-formedness. *American Journal of Computational Linguistics*, Vol. 9, No. 3 - 4, pp. 147 - 160.

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 Tsejtin, G.S. 1975. Sintaksicheskaja model' s dopushchenijem ogranichennoj neprojektivnosti. 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. and Pertsov, N.V. 1987. *Surface Syntax of English: A Formal Model within the Meaning - Text Framework*. John Benjamins, Amsterdam.

Mellish, C.S. 1989. Some chart-based techniques for parsing ill-formed input. In *Proceedings of the 27th Annual Meeting of ACL*, Vancouver, pp. 102 - 109.

Mitjushin, L.G. 1992. High-probability syntactic links. In *Proceedings of COLING-92*, Vol. 3, Nantes, pp. 930 - 934.

Mitjushin, L.G. 1993. O korrektsii oshibok soglasovanija v russkikh tekstakh. *Nauchno-Tekhnicheskaja Informatsija*, series 2, No. 10, pp. 28 - 32. ('On correction of agreement errors in Russian texts', in Russian)

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., Muto, Y., Ikeda, Y., and Nagao, M. 1988. How to get preferred readings in natural language analysis. In *Proceedings of COLING-88*, Vol. 2, Budapest, pp. 683 - 687.

Véronis, J. 1988. Morphosyntactic correction in natural language interfaces. In *Proceedings of COLING-88*, Vol. 2, Budapest, pp. 708 - 713.

Weischedel, R.M. and Sondheimer, N.K. 1983. Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics*, Vol. 9, No. 3 - 4, pp. 161 - 177.

Zaliznjak, A.A. 1980. *Grammaticheskij Slovar' Russkogo Jazyka. Slovoizmenenije*. Nauka, Moscow. ('Grammatical dictionary of the Russian language. Word changing.', in Russian)