Министерство образования Республики Беларусь

Учреждение образования

«Гомельский государственный университет

им. Ф. Скорины»

Филологический факультет

Курсовая работа

Algorithmic recognition of the Verb

Исполнитель:

Студентка группы К-42

Марченко Т.Е.

Гомель 2005

**Content**

Introduction

Basic assumptions and some facts

1 Algorithm for automatic recognition of verbal and nominal word groups

2 Lists of markers used by Algorithm No 1

3 Text sample processed by the algorithm

Examples of hand checking of the performance of the algorithm

Conclusion

References

# **Introduction**

The advent and the subsequent wide use of formal grammars for text synthesis and for formal representation of the structure of the Sentence could not produce adequate results when applied to text analysis. Therefore a better and more suitable solution was sought. Such a solution was found in the algorithmic approach for the purposes of text analysis. The algorithmic approach uses series of instructions, written in Natural Language and organized in flow charts, with the aim of analysing certain aspects of the grammatical structure of the Sentence. The procedures - in the form of a finite sequence of instructions organized in an algorithm - are based on the grammatical and syntactical information contained in the Sentence. The method used in this chapter closely follows the approach adopted by the all-Russia group Statistika Rechi in the 1970s and described in a number of publications (Kovcrin, 1972: Mihailova, 1973; Georgiev, 1976). It is to be noted, however, that the results achieved by the algorithmic procedures described in this study by far exceed the results for the English language obtained by Primov and Sorokina (1970) using the same method. (To prevent unauthorized commercial use the authors published only the block-scheme of the algorithm.)

Basic assumptions and some facts

It is a well known fact that many difficulties are encountered in Text Processing. A major difficulty, which if not removed first would hamper any further progress, is the ambiguity present in the wordforms that potentially belong to more than one Part of Speech when taken out of context. Therefore it is essential to find the features that disambiguate the wordforms when used in a context and to define the disambiguation process algorithmically. As a first step in this direction we have chosen to disambiguate those wordforms which potentially (when out of context, in a dictionary) can be attributed to more than one Part of Speech and where one of the possibilities is a Verb. These possibilities include Verb or Noun (as in stay), Verb or Noun or Adjective (as in pain, crash), Verb or Adjective (as in calm), Verb or Participle (as in settled, asked, put), Verb or Noun or Participle (as in run, abode, bid), Verb or Adjective or Participle (as in closed), and Verb or Noun or Participle or Adjective (as in cut). We'll start with the assumption that for every wordform in the Sentence there are only two possibilities: to be or not to be a Verb. Therefore, only provisionally, exclusively for the purposes of the present type of description and subsequent algorithmic analysis of the Sentence, we shall assume that all wordforms in the Sentence which are not Verbs belong to the non-verbal or Nominal Word Group (NG). As a result of this definition, the NG will incorporate the Noun, the Adjective, the Adverb, the Numeral, the Pronoun, the Preposition and the Participle 1st used as an attribute (as in the best selected audience) or as a Complement (as in we'll regard this matter settled). All the wordforms in the Sentence which are Verbs form the Verbal Group (VG). The VG includes all main and Auxiliary Verbs, the Particle to (used with the Infinitive of the Verb), all verbal phrases consisting of a Verb and a Noun (such as take place, take part, etc.) or a Verb and an Adverb (such as go out, get up, set aside, etc.), and the Participle 2nd used in the compound Verbal Tenses (such as had arrived). The formal features which help us recognize the nominal or verbal character of a wordform are called 'markers' (Sestier and Dupuis, 1962). Some markers, such as the, a, an, at, by, on, in, etc. (most of them are Prepositions), predict with 100 per cent accuracy the nominal nature of the wordform immediately following them (so long as the Prepositions are not part of a phrasal Verb). Other markers, including wordform endings such as -ing and -es, or a Preposition which is also a Particle such as to, etc., when used singly on their own (without the help of other markers) cannot predict accurately the verbal or nominal character of a wordform. Considering the fact that not all markers give 100 per cent predictability (even when all markers in the immediate vicinity of a wordform are taken into consideration), it becomes evident that the entire process of formal text analysis using this method is based, to a certain degree, on probability. The question is how to reduce the possible errors. To this purpose, the following procedures were used: a) the context of a wordform was explored for markers, moving back and forth up to three words to the left and to the right of the wordform; b; some algorithmic instructions preceded others in sequence as a matter of rule in order to act as an additional screening; no decision was taken prematurely, without sufficient grammatical and syntactical evidence being contained in the markers; no instruction was considered to be final without sufficient checking and tests proving the success rate of its performance. The algorithm presented in Section 3 below, numbered as Algorithm No 1 i.Georgicv, 1991), when tested on texts chosen at random, correctly recognized on average 98 words out of every 100. The algorithm uses Lists of markers.

Algorithm for automatic recognition of verbal and nominal word groups

The block-scheme of the algorithm is shown in Figure 1.1.

|  |  |  |
| --- | --- | --- |
| Recognition of Auxiliary Words, Abbreviations, Punctuation Marks and figures of up to 3-letter length !'presented in Lists) |  | Words over 3-lettcr length: search first left, then right (up to 3 words in each direction) for markers (presented in Lists) until enough evidence is gathered for a correct attribution of the running word |

Output result: attribution of the running word to one of the groups (verbal or nominal)Figure 1.1 Block-scheme of Algorithm No 1 Note: The algorithm. 302 digital instructions in all, is available on the Internet (see Internet Downloads at the end of the book).

1 Lists of markers used by Algorithm No 1

(i) List No 1: for, nei, two, one, may, fig, any, day, she, his, him, her, you, men, its, six, sex, ten, low, fat, old, few, new, now, sea, yet, ago, nor, all, per, era, rat, lot, our, way, leg, hay, key, tea, lee, oak, big, who, tub, pet, law, hut, gut, wit, hat, pot, how, far, cat, dog, ray, hot, top, via, why, Mrs, ..., etc. (ii) List No 2: was, are, not, get, got, bid, had, did, due, see, saw, lit, let, say,met, rot. off, fix, lie, die, dye, lay, sit, try, led, nit, . . ., etc. (iii) List No 3: pay, dip, bet, age, can, man, oil, end, fun, dry, log, use, set, air, tag, map, bar, mug, mud, tar, top, pad, raw, row, gas, red, rig, fit, own, let, aid, act, cut, tax, put, ..., etc.

(iv) List No 4: to, all, thus, both, many, may, might, when, Personal Pronouns, so, must, would, often, did, make, made, if, can, will, shall, ..., etc.

(v) List No 5: when, the, a, an, is, to, be, are, that, which, was, some, no, will, can, were, have, may, than, has, being, made, where, must, other, such, would, each, then, should, there, those, could, well, even, proportional, particular(ly), having, cannot, can't, shall, later, might, now, often, had, almost, can not, of, in, for, with, by, this, from, at, on, if, between, into, through, per, over, above, because, under, below, while, before, concerning, as, one, ..., etc.

(vi) List No 6: with, this, that, from, which, these, those, than, then, where, when, also, more, into, other, only, same, some, there, such, about, least, them, early, either, while, most, thus, each, under, their, they, after, less, near, above, three, both, several, below, first, much, many, zero, even, hence, before, quite, rather, till, until, best, down, over, above, through, Reflexive Pronouns, self, whether, onto, once, since, toward (s), already, every, elsewhere, thing, nothing, always, perhaps, sometimes, anything, something, everything, otherwise, often, last, around, still, instead, foreword, later, just, behind, ..., etc. (vii) List No 7: Includes all Irregular Verbs, with the following wordforms: Present, Present 3rd person singular, Past and Past Participle. (viii) List No 8: -ted, -ded, -ied, -ned, -red, -sed, -ked, -wed, -bed, -hed, -ped -led, -ved, -reed, -ced, -med, -zed, -yed, -ued, ..., etc.(ix) List No 9: -ous, -ity, -less, -ph, -'s (except in it's, what's, that's, there's, etc.), -ness, -ence, -ic, -ее, -ly, -is, -al, -ty, -que, -(t)er, -(t)or, -th (except in worth), -ul8, -ment, -sion(s), ..., etc. (x) List No 10: Comprises a full list of all Numerals (Cardinal and Ordinal).

2 Text sample processed by the algorithm

Text Word Group

She NG

Nodded VG

Again and NG

Patted VG

My arm, a small familiar gesture which always NG

Managed to convey VG

Both understanding and dismissal. NG

## **3 Examples of hand checking of the performance of the algorithm**

Let us see how the following sentence will be processed by Algorithm No 1, word by word: Her apartment was on a floor by itself at the top of what had once been a single dwelling, but which long ago was divided into separately rented living quarters. First the algorithm picks up the first word of the sentence (of the text), in our case this is the word her, with instruction No 1. The same instruction always ascertains that the text has not ended yet. Then the algorithm proceeds to analyse the word her by asking questions about it and verifying the answers to those questions by comparing the word her with lists of other words and Punctuation Marks, thus establishing, gradually, that the word her is not a Punctuation Mark ('operations 3-5), that it is not a figure (number) cither (operation 5 7i, and that its length exceeds two letters (operation 8). The fact that its length exceeds two letters makes the algorithm jump the next procedures as they follow in sequence, and continue the analysis in operation No 31. Using operation No 31 the algorithm recognizes the word as a three-letter word and takes it straight away to operation No 34. Here it is decreed to take the word her together with the word that follows it and to remember both words as a NG. Thus: Her apartment~NG Then the algorithm returns again to operation No 1, this time with the word was and goes through the same procedures with it till it reaches instruction No 38, where it is seen that this word is in fact was. Now the algorithm checks if was is preceded (or followed) by words such as there or it (operation No 39, which instructs the computer to compare the adjacent words with there and it), or if it is followed up to two words ahead by a word ending in -ly or by such words as never, soon, etc., none of which is actually the case. Then, finally, operation No 39d instructs the computer to remember the word was as a VG

Was =VG

And to return to the start again, this time with the next word on. Going through the initial procedures again, our hand checking of this algorithm reaches instruction No 9 where it is made clear that the word is indeed on. Then the algorithm checks the left surroundings of on, to see if the word immediately preceding it was recognized as a Verb (No 10), excluding the Auxiliary Verbs. Since it was not (was is an Auxiliary Verb), the procedure reaches operation Nos 12 and 12a, where it becomes known to the algorithm that on is followed by a. The knowledge that on is followed by an Article enables the program to make a firm decision concerning the attribution of the next two words (12a): on and the next two words are automatically attributed to the NG:

On a floor NG

After that the program again returns to operation No 1, this time to analyse the word by. The analysis proceeds without any result till it reaches operation No 11. Where the word by is matched with its recorded counterpart (see the List enumerating the other possibilities). In a similar fashion (see on), operation No 12b instructs the computer to take by and the next word blindfoldedly (i.e. without analysis) and to remember them as a NG. Thus we have:

By itself= NG

We return again to operation No 1 to analyse the next word at and we pass, unsuccessfully, through the first ten steps. Instruction No 11 enables the computer to match at with its counterpart recorded in the List (at). Since at is followed by the (an Article), this enables the computer to make a firm decision: to take at plus the plus the next word and to remember them as a NG:

At the top =NG

We deal similarly with the next word - of - and since it is not followed by a word mentioned in operation No 12, we take only the word immediately following it (12b) and remember them as a NG:

Of what —NG

Since the next word - had - exceeds the two-letter length (operation No 7), we proceed with it to operation No 31, but we cannot identify it till we reach operation No 38. Operation No 39 checks the immediate surroundings of had, and if we had listed once with the other Adverbs in 39b, we would have ended our quest now. But since once is not in this list, the algorithm proceeds to the next step (39d) and qualifies had as a VG:

Had =VG

Now we proceed further, starting with operation No 1, to analyse the next word, once. Being a long word once jumps the analysis destined for the shorter (two- and three-letter) words and we arrive with it at operation No 55. Operations No 55 and 57 ascertain that once does not coincide with either of the alternatives offered there. Through operation No 59 the computer program finds once listed in List No 6 and makes a correct decision - to attribute it to the NG:

Once =NG

Now we (and the program) have reached the word been in the text. The procedures dealing with the shorter words are similarly ignored, up to operation No 61, where been is identified as an Irregular Verb from List No 7 and attributed (No 62b) to the VG:

Been =VG

Next we have the word a (an Indefinite Article) which leads us to operations No 11 and 12 (where it is identified as such), and with operation No 12b the program reaches a decision to attribute a and the word following it to the NG: a single — NG Next in turn is dwelling. It is somewhat difficult to tag, because it can be either a Verb or a Noun. We go with it through all the initial operations, without significant success, until we get to operation No 69 and receive the instruction to follow routines No 246-303. Since dwelling does not coincide with the words listed in operation No 246, is not preceded by the syntactical construction defined in No 248 and does not have the word surroundings specified by operations No 250, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272. 274, 276, 278 and 280, its tagging, so far, is unsuccessful. Finally, operation No 282 finds the right surrounding - to its left there is, up to two words to the left, an Article (a) - and attributes dwelling to the NG:

Dwelling ~NG

However, in this case dwelling is recognized as a Gerund, not as a Noun. If we were to use this result in another program this might lead to problems. Therefore, perhaps, here we can add an extra sieve in order to be able to always make the right choice. At the same time, we must be very careful when we do so, because the algorithms arc made so compact that any further interference (e.g. adding new instructions, changing the order of the instructions) might well lead to much bigger errors than this one. Now, in operation No 3, we come to the first Punctuation Mark since we started our analysis. The Punctuation Mark acts as a dividing line and instructs the program to print what was stored in the buffer up to this moment. Next in line is the word but. Being a three-letter word it is sent to operation No 31 and then consecutively to Nos 34, 36, 38 and 40. It is identified in No 42 and sent by No 43 to the NG as a Conjunction:

But =NG

Next, we continue with the analysis of the word which, starting as usual from the very beginning (No 1 ) and gradually reaching No 55, where the real identification for long words starts. The word which is not listed in No 55 or No 57. We find it in List No 6 of operation 59 and as a result attribute it to the NG:

whuh - NG

The word long follows, and in exactly the same way we reach operation No 55 and continue further comparing it with other words and exploring its surroundings, until we exhaust all possibilities and reach a final verdict in No 89:

long -= NG

Next in turn is the word ago. As a three-letter word it is analysed in operation No 31 and the next operations to follow, until it is found by operation No 46 in List No 1, and identified as a NG (No 47): Following is the word was, which is recognized as such for the first time in operation No 38. After some brief exploration of its surroundings the program decides that was belongs to the VG: ext in sequence is the word divided. Step by step, the algorithmic procedures pass it on to operation No 55, because it is a long word. Again, as in all previous cases, operations No 55, 56, 57, 59, 61 and 63 try to identify it with a word from a List, but unsuccessfully until, finally, instruction No 65 identifies part of its ending with -ded from List No 8 and sends the word to instructions No 128-164 for further analysis. Here it does not take long to see that divided is preceded by the Auxiliary Verb was (No 130) and that it should be attributed to the VG as Participle 2nd (No 131):

divided = VG

The Preposition into comes next and since it is not located in one of the Lists examined by the instructions and none of its surroundings correspond to those listed, it is assumed that it belongs to the NG (No 89):

Into =NG

Next, the ending -ly of the Adverb separately is found in List No 9 and this gives enough reason to send it to the NG (No 64):

Separately =NG

Now we come to a difficult word again, because rented can be either a Verb or an Adjective, or even Participle 1st. Since its ending -ted is found in List No 8, rented is sent to instructions No 128-164 for further analysis as a special case. With instructions No 144 and 145 the algorithm chooses to recognize rented as a Participle (1st) and to attribute it to the NG:

Rented = NG

Next comes living. At first it also seems to be a special case (since it can be Noun, Gerund, Verb - as part of a Compound Tense - Adjective or Participle). Instruction No 69 establishes that this word ends in -ing and No 70 sends it for further analysis to instructions No 246-303. Almost towards the end (instructions No 300 and 301), the algorithm decides to attribute living to the acknowledging that it is a Present Participle. If the program were more precise, it would be able also to say that living is an Adjective used as an attribute. The last word in this sequence is quarters. The way it ends very much resembles a verbal ending (3rd person singular). Will the algorithm make a mistake this time? Instruction No 67 recognizes that the ending -s is ambiguous and sends quarters to instructions No 165 245 for more detailed analysis. Then the word passes unsuccessfully (unrecognized) through many instructions till it finally reaches instruction No 233, where it is evidenced that quarters is followed by a Punctuation Mark and this serves as sufficient reason to attribute it to the NG:

Quarters = NG

Finally, our algorithmic analysis of the above sentence ends with commendable results: no error. However, in the long run we would expect errors to appear, mainly when we deal with Verbs, but these are not likely to exceed 2 per cent. For example, an error can be detected in the following sample sentence: .Not only has his poetic fame - as was inevitable - been overshadowed by that of Shakespeare but he was long believed to have entertained and to have taken frequent opportunities oj expressing a malign jealousy oj one both greater and more successful than himself.

This sentence is divided into VG and NG in the following manner:

Text Word Group

Not VG

Only NG

Has VG

His poetic fame NG

As NG

Was VG

Inevitable NG

Been overshadowed VG

By that of Shakespeare NG

But he NG

Was long believed to have entertained VG

And NG

To have taken VG

Frequent opportunities of expressing NG

A malign jealousy of one both greater NG

And NG

More successful than himself. NG

As is seen in the above example, the word long was wrongly attributed to the VG (according to our specifications laid down as a starting point for the algorithm it should belong to the NG). The reader, if he or she has enough patience, can put to the test many sentences in the way described above (following the algorithmic instructions), to prove for himself (herself) the accuracy of our description. Though this is a description designed for computer use (to be turned into a computer software program), nevertheless it will surely be quite interesting for a moment or two to put ourselves on a par with the computer in order to understand better how it works. Of course, that is not the way we would do the job. Our knowledge of grammar is far superior, and we understand the meaning of the sentence while the computer does not. The information used by the computer is extremely limited, only that presented in the instructions (operations) and in the Lists. Further on we will try to give the computer more information (Algorithm No 3 and the algorithms in Part 2) and correspondingly increase our requirements.

Conclusion

* Most of the procedures to determine the nominal or verbal nature of the wordform, depending on its context, are based on the phrasal and syntactic structures present in the Sentence (for example, instructions 11 and 12, 67 and 68, 85, etc.), i.e. structures such as Preposition + Article + Noun; will (shall) + be + (Adverb) + Participle; to + be + (not) + Participle 2nd + to + Verb; -ing + Possessive Pronoun + Noun, etc. (the words in brackets represent alternatives).
* When constructing the algorithm it was thought to be more expedient to deal first with the auxiliary and short words of two-letter length, then with words of three-letter length, then with the rest of the words - for frequency considerations and also because they represent the main body of the markers.
* The approach presented in this study is not based on formal grammars and is to be used exclusively for text analysis (not for text synthesis). One should not associate the VP (Verbal Phrase) with the VG and the NP (Noun Phrase) with the NG - for these are completely different notions as has been shown by the presentation.
* The algorithm can be checked by feeding in texts through the procedures (the instructions) manually and if the reader is dissatisfied he or she may change the instructions to improve the results. (See Section 3.3 for details of how the performance of the algorithms can be hand checked.) The algorithm can be easily programmed in one of the existing artificial languages best suited for this type of operation.

**References**

1. Brill, E. and Mooney, R.J. (1997), ‘An overview of empirical natural language processing', in AI Magazine, 18 (4): 13-24.

2.Chomsky, N. (1957), Syntactic Structures. The Hague: Mouton. Curme, G.O. (1955), English Grammar. New York: Barnes and Noble.

3. Dowty, D.R., Karttunen, L. and Zwicky, A.M. (eds) (1985), Natural Language Parsing. Cambridge: Cambridge University Press. Garside, R. (1986),

4. 'The CLAWS word-tagging system', in R. Garside, G. Leech and G. Sampson (eds) The Computational Analysis of English. Harlow: Longman. Gazdar, G. and Mellish, C. (1989), Natural Language Processing in POP-11. Reading, UK: Addison-Wesley. Georgiev, H. (1976),

5. 'Automatic recognition of verbal and nominal word groups in Bulgarian texts', in t.a. information, Revue International du traitement automatique du langage, 2, 17-24. Georgiev, H. (1991), 'English Algorithmic Grammar', in Applied Computer Translation, Vol. 1, No. 3, 29-48.

6. Georgiev, H. (1993a), 'Syntparse, software program for parsing of English texts', demonstration at the Joint Inter-Agency Meeting on Computer-assisted Terminology and Translation, The United Nations, Geneva.

7. Georgiev, H. (1993b), 'Syntcheck, a computer software program for orthographical and grammatical spell-checking of English texts', demonstration at the Joint Inter-Agency Meeting on Computer-assisted Terminology and Translation, The United Nations, Geneva. Georgiev, H. (1994—2001), Softhesaurus, English Electronic Lexicon, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/ Windows. Georgiev, H. (1996-2001a),

8. Syntcheck, a computer software program for orthographical and grammatical spell-checking of German texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/Windows. Georgiev, H. (1996-200lb), Syntparse, software program for parsing of German texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS Windows.

9. Georgiev, H. (1997—2001a), Syntcheck, a computer software program for orthographical and grammatical spell-checking of French texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS Windows.

10. Georgiev, H. (1997-2001b), Syntparse, software program for parsing of French texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/Windows.

11. Georgiev, H. (2000 2001), Syntcheck, a computer software program for orthographical and grammatical spell-checking of Italian texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/Windows.

12. Giorgi, A. and Longobardi, G. (1991), The Syntax of Noun Phrases: Configuration, Parameters and Empty Categories. Cambridge: Cambridge University Press. Graver, B. D. (1971), Advanced English Practice. Oxford: Oxford University Press.

13. Grisham, R. (1986), Computational Linguistics. Cambridge: Cambridge University Press. Harris, Z.S. (1982)

14. A Grammar of English on Mathematical Principles. New York: Wiley. Hausser, R. (1989), Computation of Language. Berlin: Springer. Hornby. A.S. (1958)

15. A Guide lo Patterns and Usage in English. London: Oxford University Press. Kavi, M. and Nirenburg, S. (1997), 'Knowledge-based systems for natural language', in A.B. Tucker (ed.) The Computer Science and Engineering Handbook. Boca Raton, FL: CRC Press, Inc., 637 53.

16. Koverin, A.A. (1972), 'Grammatical analysis, on a computer, of French scientific and technical texts' (in Russian), PhD thesis, Leningrad University, Russia. Leech, S. and Svartvik, J. (1975)

17. A Communicative Grammar of English. London: Longman. Manning, C. and Schutze, H. (1999), Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press. Marcus, M.P. (1980)

18. A Theory of Syntactic Recognition for Natural Language. Cambridge, MA: MIT Press. McEnery, T. (1992), Computational Linguistics. Wilmslow, UK: Sigma Press.

19. Mihailova, I.V. (1973), Automatic recognition of the nominal group in Spanish texts' (in Russian), in R. G. Piotrovskij (ed.) Injenernaja Linguistika. St Petersburg: Politechnical Institute, 148-75.

20. Primov, U.V. and Sorokina, V.A. (1970), 'Algorithm for automatic recognition of the nominal group in English technical texts' (in Russian), in R.G.

21. Piotrovskij (ed.) Statistika Teksta, II. Minsk: Politechnical Institute. Pullum, G.K. (1984), 'On two recent attempts to show that English is not a CFL', Computational Linguistics, 10 (3-4), 182-6. Quirk, R. and Greenbaum, S. (1983),