Министерство образования Республики Беларусь

Учреждение образования

«Гомельский государственный университет

им. Ф. Скорины»

Филологический факультет

Курсовая работа

Division of the sentence into phrases

Исполнитель:

Студентка группы К-42

Лапицкая Т.Е.

Гомель 2005

**Content**

Introduction

Presentation

Algorithm for division of the sentence into phrases

Lists used by Algorithm No 2

Some examples of the performance of Algorithm No 2

Conclusion

References

**Introduction**

For multiple purposes, in Text Processing and Machine Translation, often there is a need to divide the sentence into smaller units that can be processed more easily than the whole sentence, especially when the sentence happens to be a long one. To that purpose we have devised an efficient algorithm based on the assumptions presented in the next section.

**Presentation**

When we say that we are going to divide the sentence into phrases, we must state first how we will define the phrase and what our understanding of the phrase will be where it starts and where it ends. For the purposes of the present algorithm (and not for any other, especially theoretical, purposes) the phrase is delimited on its left and on its right by Punctuation Marks and Auxiliary words. The phrase usually starts with an Auxiliary word and ends with the appearance of a Punctuation Mark or an Auxiliary word.

The Auxiliary words, marking the boundaries of the phrases, are presented in tables (Lists). Each table lists Auxiliary words of a particular type. It was observed that some Auxiliary words (as well as some sequences of consecutively used Auxiliary words) start usually longer and more independent phrases than others. For example, in a sentence like is often difficult to seek solutions through the curtailment of consumption.

The Auxiliary word through followed by the Article the (another Auxiliary word) starts a phrase that ends with the appearance of a Punctuation Mark, while the Auxiliary word of starts a sub-phrase which is part of a longer phrase. In our algorithm (see Algorithm No 2 in Section 3) this subdivision of the sentence into longer phrases and the subdivision of the longer phrases into smaller constituent phrases is expressed by leaving different lengths of space between one phrase and another. The longer the space left before the phrase, the more self-sufficient and independent the phrase is thought to be. In this study we have established five types of phrases, depending on their relative independence within the sentence. This independence is expressed by a particular Auxiliary word (or words) or by a Punctuation Mark. The longest and the most self-sufficient and relatively independent phrase starts and ends with a Punctuation Mark. The second most independent phrase starts with a word from List No 1 and ends with a Punctuation Mark or with the appearance of another Auxiliary word from List No 1. For example:

(6 spaces left) One US government study estimated

(5 spaces left) that there are 68 large manufacturing complexes

(4 spaces) in the region

(5 spaces left) that have significant idle capacity, (end)

The full stop at the start of the sentence is equivalent to six spaces. In other words, a smaller space following after a larger space to the left means that the phrase starting after the smaller space is dependent on, and a constituent of, the larger phrase. The smaller space in the example above (4 spaces) shows that the phrase following after it is dependent on the previous phrase that there are 68 large manufacturing complexes and explains it (or brings additional information about it, here location), while the five spaces left after region signify that the next phrase is dependent on the previous large phrase (the one that has a longer space left in front), in this case One US government study estimated that there are 68 large manufacturing complexes.

The space left between the phrases depends on the actual Preposition (or Punctuation Mark) used or on the sequence of Punctuation Mark and/or Auxiliary words, as specified (for more details see the instructions for Algorithm No 2 below).

**Algorithm for division of the sentence into phrases**

Input text comparing of each word entry Searching left or right with the Auxiliary words or (up to two words) for Punctuation Marks (presented other Auxiliary words in Lists) and identifying the or Punctuation Marks Auxiliary words or Punctuation Marks Output result: a phrase

Note: The algorithm (27 digital instructions in all) is available for free download on the Internet (see Internet Downloads at the end of the book).

## Lists used by Algorithm No 2

NB The words not registered in the Lists are recorded as they follow, in the same sequence, after those registered in the Lists.

(i) List No 1: besides, therefore, however, whereas, thus, hence, though, despite, with, nevertheless, throughout, through, during, that, only, but, if, otherwise, again, which, although, thereby, already, against, unless, thereafter etc.

(ii) List No 2: over, as, what, toward(s), for, into, about, by, so, from, at, above, under, beside, below, onto, since, behind, in front of, beyond, around, before, after, then, altogether, among(st), between, beneath etc.

(Hi) List No 3: both, neither, none etc.

(iv) List No 4: of, to (as Preposition)

(v) List No 5: the, a, an

(vi) List No 6: so much as, so far as, so far, as long as, as soon as, so long as, in order that, in order to, lest, as well as, and, or, nor etc.

(vii) List No 7: such, than, onto, until, all, near, even, when, while, within, last, next, also, less, more, most, whether, much, once, one, any, many, some, where, another, other, each, then, whose, who, whoever, till, until, what, across, whence, according, due to, owing, whereby, prior, wherever, whenever, already, moreover, likewise, however etc.

(viii) List No 8: out, in, on, down etc.

## Some examples of the performance of Algorithm No 2

Below we will present a text divided into phrases according to the instructions for the algorithm:

(i) Many countries also have established or have under construction a free zone, where exporters have access to shipping facilities, a pool of labour and freedom from exchange controls.

(ii) The Caribbean Basin Initiative, a US package of aid and trade incentives to encourage manufacturing, has given an added boost to industrial development in this region.

The analysis of the sentence starts with checking the contents of the memory and taking to print any information stored up to this moment (this is done at the start of each new sentence), also with ascertaining whether the sentence has ended or not and recording the analysed word in the memory if it is not recorded yet ia procedure carried out after each word). Then the algorithm reads the next word (in No 4a), which in the case of (i) above is many, and proceeds to analyse it in 5. Since it is not a full stop or any other Punctuation Mark (5, 7), nor a word specified in 9, 11, 13, 15, 17 or 19, the analysis yields no result until the program gets to operation No 21, where the word many is located in List No 7. Here the program, through operation No 22, checks whether many is followed by yet another word from the Lists. Operation 22ab certifies that it is not, and instructs the program to cut the sentence at this point and to leave three spaces (before many) when recording it, then to return to operation No 2 to start the analysis of the next word. The next word, countries, could not be identified (it is not registered in the Lists), therefore operation 27 instructs the program to record it in the memory as the next consecutive word of the phrase and to return to 2 to continue the analysis of the sentence.

The word also follows next. The program cannot locate the word and proceeds further, after registering it. The next words have and established are dealt with in a similar way. Next comes the Conjunction or. The program locates the word in operation No 17, then it checks if other words from the Lists follow (18). A single space is left before recording it (No 18b). The word have is registered next and the program reaches under (15) to draw a dividing line by leaving four spaces (16ab), and this carries on till the end of the text.

These procedures can be applied to any English language texts. The actual users of the algorithm can improve it by adding new words to the Lists or by changing the dividing lines to suit other strategies and other interpretations of the boundaries of the English phrase.

**Conclusion**

Algorithm No 2 was developed with the special purpose of aiding the overall automatic analysis of the sentence. The division of the sentence into smaller units helps us understand better its meaning, though the division, as presented in this section, is not based on meaning but on formal features. The reader will find somewhat different and much more accurate interpretation of the existing boundaries within a sentence in Part 2.

In the course of this study it was observed that each foregoing phrase finds further interpretation of its meaning in the next phrase. In other words, the first phrase of a sentence carries a certain meaning, which with each successive phrase becomes more and more clear and complete - the next phrase simply adds more information to the meaning of the previous phrase. The phrases have varied mutual interdependence, which we tried to express with a margin left between them. We will express this graphically in Figure 2.2, which considers two sentences.

The brackets show the dependence of each succeeding phrase both on the previous one and on all preceding ones. In the second sentence, the phrases are separated with equal space left between them. In those cases where the space left is smaller, this means that the tie with the previous phrase is stronger (i.e. the next phrase is an integral part of the preceding one). A sudden surge of the interval signals the division between two phrases, as in the example in Figure 2.3. In this example, the second large phrase (Clause) explains the meaning of the first. This is indicated with the interval left and with the brackets.

**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.

4. Curme, G.O. (1955), English Grammar. New York: Barnes and Noble.

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

6. Garside, R. (1986), 'The CLAWS word-tagging system', in R. Garside,

7. G. Leech and G. Sampson (eds) The Computational Analysis of English. Harlow: Longman.

8. Gazdar, G. and Mellish, C. (1989), Natural Language Processing in POP-11. Reading, UK: Addison-Wesley.

9. Georgiev, H. (1976), '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.

10. Georgiev, H. (1991), 'English Algorithmic Grammar', in Applied Computer Translation, Vol. 1, No. 3, 29-48.

11. 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.

12. 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.

13. Georgiev, H. (1994—2001), Softhesaurus, English Electronic Lexicon, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/ Windows.

14. Georgiev, H. (1996-2001a), Syntcheck, a computer software program for orthographical and grammatical spell-checking of German texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/Windows.

15. Georgiev, H. (1996-200lb), Syntparse, software program for parsing of German texts, produced and marketed by LANGSOFT, Sprachlernmittel, Switzerland; platform: DOS/Windows.

16. 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.

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

18. 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.

19. Giorgi A. and Longobardi G. (1991), The Syntax of Noun Phrases: Configuration, Parameters and Empty Categories. Cambridge: Cambridge University Press.

20. Graver B.D. (1971), Advanced English Practice. Oxford: Oxford University Press.

21. Grisham R. (1986), Computational Linguistics. Cambridge: Cambridge University Press.

22. Harris Z.S. (1982), A Grammar of English on Mathematical Principles. New York: Wiley.

23. Hausser R. (1989), Computation of Language. Berlin: Springer.

Hornby. A. S. (1958), A Guide lo Patterns and Usage in English. London: Oxford University Press.

24. 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.

25. Koverin A.A. (1972), 'Grammatical analysis, on a computer, of French scientific and technical texts' (in Russian), PhD thesis, Leningrad University, Russia.

26. Leech, S. and Svartvik, J. (1975), A Communicative Grammar of English. London: Longman.

27. Manning C. and Schutze H. (1999), Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press.

28. Marcus M.P. (1980) A Theory of Syntactic Recognition for Natural Language. Cambridge, MA: MIT Press.

29. McEnery T. (1992), Computational Linguistics. Wilmslow, UK: Sigma Press.

30. 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.

31. 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. Piotrovskij (ed.) Statistika Teksta, II. Minsk: Politechnical Institute.

32. Pullum, G.K. (1984), 'On two recent attempts to show that English is not a CFL', Computational Linguistics, 10 (3-4), 182-6.

33. Quirk, R. and Greenbaum, S. (1983), A University Grammar of English. London: Longman.

34. Quirk R., Greenbaum S., Leech G. and Svartvic J. (1972), Grammar of Contemporary English. London: Longman.

35. Reichman R. (1985), Getting Computers to Talk like You and Me. Cambridge, MA: MIT Press.

36. Sestier A. and Dupuis L. (1962), 'La place de la syntaxe dans la traduction automatique des langues. Esquisse d'un nouveau systeme de description grammaticale et de son utilisation pour la reconstruction des structures grammaticales', Inge'nieurs et Techniciens, No. 1555, 43-50.

37. Schank R. and Fano A. (1992) 'Knowledge, memory, learning and teaching. A survey of our research', in t.a.l., Traitement Automatique des Langues, Vol. 33, No. 1-2.

38. Shanks D. (1993) 'Breaking Chomsky's rules', New Scientist, February, 26-30.

39. Shieber S. M. (1985) 'Evidence against the non-context-freeness of natural language', in Linguistics and Philosophy, 8, 333-43.

40. Stannard A. (1974), Living English Structure. London: Longman.

41. Urdang L. (ed.) (1968) The Random House Dictionary of the English Language (College Edition). New York: Random House.