THE AUTOMATIC GRAMMATICAL TAGGING OF THE LOB CORPUS

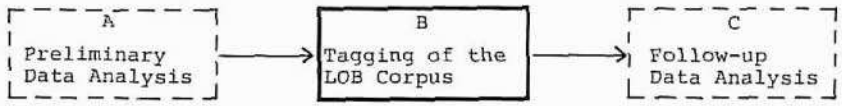
Geoffrey Leech, *Roger Garside*, and *Eric Atwell*
University of Lancaster, England

In collaboration with the English Department, University of Oslo,[1] and the Norwegian Computing Centre for the Humanities, Bergen,[2] we have been engaged in the automatic grammatical tagging of the LOB (Lancaster-Oslo/Bergen) Corpus of British English. The computer programs for this task are running at a success rate of approximately 96.7%,[3] and a substantial part of the 1,000,000-word corpus has already been tagged.[4] The purpose of this paper is to give an account of the project, with special reference to the methods of tagging we have adopted.

1 OVERVIEW OF THE PROJECT

To see the project in its overall context, we must give some attention to the preliminaries which preceded the tagging itself, and also to the follow-up work which we intend to undertake when the tagging is complete:

Fig. 1



1.1 Preliminaries

The first stage of our work was collecting and analysing data from the Tagged Brown Corpus. Our purpose was to make use of, and at the same time to improve on, the automatic tagging of the Brown Corpus (undertaken at Brown University 1971-8).[5] The Tagged Brown Corpus was kindly made available to us by Henry Kučera and Nelson Francis, who also provided us with a copy of the automatic tagging program TAGGIT written by Greene and Rubin (1971). An exploratory run of the program

on the LOB Corpus suggested that a new approach to tag selection would be needed if we were to improve substantially on TAGGIT's performance. For comparability with the Tagged Brown Corpus, we had decided to use largely the same set of tags as were used by TAGGIT; but in practice some changes were advisable, and as a result of these changes, the new Tagset (see Appendix A) consisted of 134 tags (including punctuation tags), as against Brown's 87. For example, we found it desirable to introduce a number of additional tags ("NPL", "NPT", "NNP", "JNP") where Brown had used only the one tag "NP" (proper noun). But where changes were made, we have been careful to preserve general comparability with the Brown Corpus, so that when the LOB tagging is complete, it will be possible to make systematic comparisons between the American and British corpora.

The chief advantage we derived from the Brown tagging project, however, was that we were able to make substantial use of the Tagged Brown Corpus itself as a database for our own Automatic Tagging. From lists provided by the Norwegian Computing Centre for the Humanities, our Oslo colleagues Stig Johansson and Mette-Cathrine Jahr derived lists of word-tag associations and suffix-tag associations which, after revision, formed the kernel of our Tag-Assignment program (see 3.1 below). Also, by means of a group of Context Collecting programs, we were able to derive from the corpus frequency lists of tag-sequences, and these were later adapted for inclusion in our Tag-Selection program (see 3.2).

1.2 Follow-up work

Just as the tagging of the Brown Corpus provided us with a headstart in our own project, so after the tagging of the LOB Corpus it will be possible to use the data derived from the LOB tagging project, including the tagged Corpus itself, as an input to further automatic tagging programs, which will improve on our programs just as these were an improvement on the Brown programs. Corpus-based automatic language analysis is one area of linguistic research where results are cumulative, so we hope, in a follow-up to this project, to revise and improve the programs for implementation on further corpora. For this to happen, however, various frequency listings must be obtained from the Tagged LOB Corpus. Such listings (in particular, a lemmatised word-frequency listing of the LOB Corpus) will also be useful

for other research purposes, e.g. for comparison with the Brown Corpus and with the London-Lund Corpus.

2 THE OVERALL PROCESS OF TAGGING

Having looked briefly at stages (A) and (C) in Fig. 1, we may now examine the middle box (B), dealing with the overall tagging process. The contents of this box we again divide into three stages:

Fig. 2

As may be expected with programs acting on unrestricted language input, the automatic tagging programs require both a pre-editing phase, where the human investigator prepares the corpus for input, and a post-editing phase, where he corrects any errors made by automatic tagging. Manual pre-editing and post-editing are both, however, carried out with the aid of computer programs. We give a brief account of these stages (A and C in Fig. 2) before dealing with the automatic tagging programs themselves.

2.1 Pre-editing

At the start of the process, the Raw Corpus (the Corpus in its untagged orthographic form) exists in a "horizontal" format; i.e. it reads from left to right in the normal way. A Verticalization Program converts this corpus into a "Vertical Corpus" in which one word occurs beneath another in a vertical column. At the same time, the Verticalization Program makes automatic changes which will later help the tagging. These include supplying missing punctuation, splitting enclitic words (*n't*, *'ll*, etc.) from their predecessors, changing capital letters to lower case at the beginning of sentences, in headings, etc.; and marking foreign words, formulae, and other exceptional features of the text. The Verticalization Program also creates a number of columns alongside the text, so that various kinds of information (orthographic, lexical, syntactic) can be recorded for future users of the corpus.

When the Verticalization of the corpus takes place, another set of programs produces "Editlists" of particular text features which have to be checked by a human editor to see whether they have to be altered in order to be suitable input to the Automatic Tagging. The most important lists are those of "CAPITALS" (non-sentence-initial words beginning with a capital letter) and "UNCAPITALS" (sentence-initial words whose capital letter will have been changed to lower case by the Verticalization Program). For example, if a sentence begins with a proper name such as *John*, the Program will have changed this to *john*, and a manual editor will then have to change it back again. The reason for these changes in capitalization is that the Automatic Tagging programs make use of word-initial capitals in deciding what kind of tags to assign to a word (most words beginning with a capital end up being tagged as proper names: see 3.1 and Appendix D).

Although the majority of pre-editing changes are made automatically by the Verticalization Program, Pre-editing has proved to be a time-consuming process, especially since all pre-editing decisions have had to be carefully standardized and entered in a "Pre-editing Manual". In any further tagging projects, we will try to eliminate manual pre-editing, e.g. by enabling the automatic tagging programs to accept a word with an initial capital as a possible variant of a lower case word. For example, if both *Rose* and *rose* occurred in the same text, the capital of the former word would be reduced to lower case; but if *Rose* only occurred in the capitalized version, the capital would be retained, and the word would be analysed as a proper noun. In this way, manual pre-editing could be replaced by automatic pre-editing, and any additional errors which resulted from this would simply add to the number of words requiring correction at the post-editing phase.[6]

2.2 Post-editing

Like Pre-editing, Post-editing currently has both an automatic and a manual aspect. The Vertical Corpus, after automatic tagging, contains, alongside each word, one or more grammatical tags, placed in order of their likelihood of occurring in this context. The tag which the programs have selected as the correct one is clearly indicated (see Fig. 4 below). Thus the task of the manual post-editor is to check the decisions made by the program, and to mark any corrections which

have to be made. With more than a million words to check, this is an exceedingly time-consuming task, and it is therefore worthwhile using the computer to ease the human editor's task in any practicable way. One way of doing this is to present the output in a special form in which the text is arranged in two vertical columns per page, the word and the tag lying alongside one another for ease of reading. Into this "Vertical Output" there is built an additional aid for the post-editor: it is possible to set a threshold below which the likelihood of error is low enough to be disregarded by the initial post-editor. Sample analyses have shown that 60% of the text-words are unambiguously tagged; that of the 40% which are ambiguously tagged, 64% have a likelihood, as calculated by the Tag Selection Program (see 3.2)[7], of more than 90%; and that these have only a 0.5% risk of being erroneous. This means that over the whole sample 86% of words can be unambiguously tagged with less than 1% error. In these relatively safe cases, the output listing simply assumes the one tag to be correct, and gives alternative taggings only for the 14% of words for which the risk of error is relatively high. A specimen of this "Vertical Output" is given in Appendix E.

This facilitates the first manual post-edit, but to ensure that all errors have been caught, a second stage of manual post-editing will take place, this time on a "rehorizontalized" version of the corpus, in which each word in a line has a single tag beneath it, as in Appendix F.

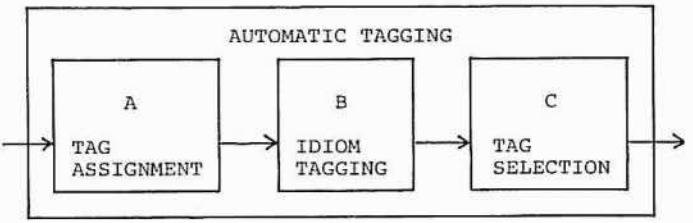
Once it has undergone manual correction, this version of the corpus will be available for distribution to users. There will also, however, be a vertical-format "Rolls-Royce" version of the corpus, which will contain all the information about the original text recorded in the columns of the Verticalization Program (see 2.1) as well as the grammatical tag of each word. This version is the authoritative tagged LOB Corpus, and will enable users to reconstruct the original text. For example, if one wants to study the relation between orthography and grammar, this version will preserve orthographic information excluded from the "rehorizontalized" version.

3 AUTOMATIC TAGGING

We now turn to the Automatic Tagging programs which form the heart of the project, and constitute its main contribution to research.

Once again, the contents of the middle box of the previous diagram (B in Fig. 2) must itself be broken down into three logically separable processes:

Fig. 3



For development purposes, it was convenient to write a separate program for each of these three processes;[8] but it would be easy enough in principle to combine them all into a single program. Logically speaking, the Automatic Tagging divides into Tag Assignment (whereby each word in the corpus is assigned one or more possible tags), and Tag Selection (whereby a single tag is selected as the correct one in context, from the one or more alternatives generated by Tag Assignment). It was as something of an afterthought that we added to the Tag Assignment program (WORDTAG) and the Tag Selection program (CHAINPROBS) a third, intermediate program (IDIOMTAG) to deal with various grammatically anomalous word-sequences which, without intending any technical usage of the term, we may call "idioms".

3.1 Tag Assignment

The simplest kind of Tag Assignment procedure would be just a look-up in a WORDLIST or dictionary specifying the tag(s) associated with each word. In addition to such a Wordlist, the Brown Tagging Program TAGGIT has a SUFFIXLIST, or list of pairings of word-endings and tags (for example, the ending -NESS is associated with nouns). We follow Brown in this, using a Wordlist of over 7000 words, and a Suffixlist of approximately 660 word-endings.[9] Further, the LOB Assignment Program contains a number of procedures for dealing with words containing hyphens, words beginning with a capital letter, words ending with *-s*, with *'s*, etc. The advantages of having a SUFFIXLIST are that (a) the WORDLIST can be shortened, since words whose wordclass is predictable from their ending can be omitted from it; and (b) the

set of words accepted by the program can be open-ended, and can even include neologisms, rare words, nonsense words, etc. These advantages also apply to the procedures for dealing with hyphenated and capitalized words.

The Tag Assignment Program reads each word in turn, and carries out a series of testing procedures, to decide how the word should be tagged. The procedures are crucially ordered, so that if one procedure fails to tag a word, the word drops through to the next procedure. If none of the tag-assignment procedures is successful, the word is given a set of default tags. The program's structure can be summarized at its simplest by listing the major procedures as follows (where W = the word currently being tagged):

(1) *Is W in the WORDLIST?*

If so, assign the tags given in the WORDLIST.

(2) *Is W a number, a single letter, or a letter preceded or followed by a number of digits?*

If so, assign special tags.

(3) *Does W contain a hyphen?*

If so, carry out the special procedure APPLYHYPHEN.

(4) *Does W have a word-initial capital (WIC)?*

If so, carry out the special procedure APPLYWIC.

(5) *Does W end with one of the endings in the SUFFIXLIST?*

If so, assign the tags specified in the SUFFIXLIST.

(6) *Does W end in -s?*

If so, apply an *-s* stripping procedure, and check again whether W is in the WORDLIST, or failing that, the SUFFIXLIST. If it is, apply the tags given in the WORDLIST or SUFFIXLIST, retaining only those tags which are compatible with *-s*.

If not, assign default tags for words ending in *-s*.

(7) *If none of the above apply*, assign default tags for words not ending in *-s*.

APPLYHYPHEN and APPLYWIC are 'macroprocedures' which themselves consist of a set of tests comparable to those of the main program. For further details, see the Flowcharts in Appendices B - D.

The output of the Tag Selection Program is a version of the Vertical Corpus in which one or more grammatical tags (with accompanying rarity markers @ or % if appropriate)[10] are entered alongside each word. As an additional useful feature, this program provides a diagnostic (in the form of an integer between 0 and 100) indicating the tagging decision which led to the tag-assignment of each word. This enables the efficacy of each procedure in the program to be monitored, so that any improvement effected by changes in the program can be measured and analysed. In this respect, the program is self-evaluating. It can also be readily updated through revisions to the Tag-set, Wordlist, or Suffixlist.

3.2 Tag Selection

If one part of the project can be said to have made a particular contribution to automatic language processing, it is the Tag Selection Program (CHAINPROBS), the structure of which is described in greater detail in Marshall (1982). This program operates on a principle quite different from that of the Tag Selection part of the program used on the Brown Corpus. The Brown program used a set of CONTEXT FRAME RULES, which eliminated tags on the current word if they were incompatible with tags on the words within a span of two to the left or two to the right of the current word (W). Thus assuming a sequence of words -2, -1, W, +1, +2, an attempt was made to disambiguate W on the evidence of tags already unambiguously assigned to words -2, -1, +1, or +2. The rules worked only if one or more of these words were unambiguously tagged, and consequently often failed on sequences of ambiguous words. Moreover, as many as 80% of the applications of the Context Frame Rules made use of only one word to the left or to the right of W. These observations, made by running the Brown Program over part of the LOB Corpus, led us to develop, as a prototype of the LOB Tag-Selection Program, a program which computes transitional probabilities between one tag and the next for all combinations or possible tags, and chooses the most likely path through a set of ambiguous tags on this basis.

Given a sequence of ambiguous tags, the prototype Tag-Selection Program computed all possible combinations of tag-sequences (i.e. all possible paths), building up a search tree. It treated each possible Tag Sequence or path as a First-order Markov chain, assigning to each

path a probability relative to other paths, and reducing by a constant scaling factor the likelihood of sequences containing tags marked with a rarity marker @ or %. Our assumption was that the frequency of tag sequences in the Tagged Brown Corpus would be a good guide to the probability of such sequences in the LOB Corpus; these frequencies were therefore extracted from the Brown Corpus data, and adjusted to take account of changes we had made to the Brown Tag-set. We expected that the choice of tags on the basis of first-order probabilities would provide a rough-and-ready tag-selection procedure which would then have to be refined to take account of higher-order probabilities. It is generally assumed, following Chomsky (1957:18-25), that a first-order Markov process is an inadequate model of human language. We therefore found it encouraging that the success rate of this simple first-order probabilistic algorithm, when tried out on a sample of over 15,000 words of the LOB Corpus, was as high as 94%. An example of the output of this program (from Marshall 1982) is given in Fig. 4:

Fig. 4

this	DT
task	NN
involved	[VBD]/90 VBN/10 JJ@/0
a	AT
very	[QL]/99 JJB@/1
great	[JJ]/98 RB/2
deal	[NN]/99 VB/1
of	IN
detailed	[JJ]/98 VBN/2 VBD/0
work	[NN]/100 VB/0
for	[IN]/97 CS/3
the	ATI
committee	NN

In this output, the tags supplied by the Tag Assignment Program are accompanied by a probability expressed as a percentage. For example, the entry for the word *involved* ([VBD]/90 VBN/10 JJ@/0) indicates that the tag VBD 'past tense verb' has an estimated probability of 90%; that the tag VBN 'past participle' has an estimated probability of 10%; and that the tag JJ 'adjective' has an estimated probability

of 0%. The symbol @ after JJ means that the Tag Assignment program has already marked the 'adjective' tag as rare for this word (see Note 10). The square brackets enclosing the 'past tense' tag indicate that this tag has been selected as correct by the Tag Selection Program. (The square brackets are used to indicate the preferred tag for every word which is marked as ambiguous; where the word has only one assigned tag, this marking is omitted as unnecessary.)

An improved Tag Selection Program was developed as a result of an analysis of the errors made by the prototype program. We realised that an attempt to supplement the first-order transition matrix by a second-order matrix would lead to a vast increase in the amount of data to be handled as part of the program, with only a marginal increase in the program's success. A more practical approach would be to concentrate on those limited areas where failure to take account of longer sequences resulted in errors, and to introduce a scaling factor to adjust such sequences in the direction of the required result. For instance, the occurrence of an adverb between two verb forms (as in *has recently visited*) often led to the mistaken selection of VBD rather than VBN for the second verb, and this mistake could be corrected by downgrading the likelihood of a triple consisting of the verb *be* or *have* followed by an adverb followed by a past tense verb. Similarly, many errors resulted from sequences such as *live and work*, where we would expect the same word-class to occur on either side of the coordinator - something which an algorithm using frequency of tag-pairs alone could not predict. This again could be handled by boosting or reducing the predicted likelihood of certain tag triples. A further useful addition to the program was an alternative method of calculating relative likelihood, making use of the probability of a word's belonging to a particular grammatical class, rather than the probability of the occurrence of a whole sequence of tags. This serves as a cross-check on the 'sequence probability' method, and appears to be more accurate for some classes of cases. These improvements, together with the introduction of an Idiom Tagging program (see 3.3 below), resulted in an overall success rate of between 96.5% and 97.0%.

Having tried out the heuristic principle that error-analysis of a program's output can be fed back into the program, enabling it to increase its accuracy, we anticipate that a further analysis of errors after post-editing of the LOB Corpus will lead to further improvements.

3.3 Idiom Tagging

The third tagging program, which intervenes between the Tag Assignment and Tag Selection programs, is an Idiom Tagging Program (IDIOMTAG) developed as a means of dealing with idiosyncratic word sequences, which would otherwise cause difficulty for the automatic tagging. One set of anomalous cases consists of sequences which are best treated, grammatically, as a single word: for example, *in order that* is tagged as a single conjunction, *as to* as a single preposition, and *each other* as a single pronoun. Another group consists of sequences in which a given word-type is associated with a neighbouring grammatical category; for example, preceding the preposition *by*, a word like *invoked* is usually a past participle rather than a past tense verb. The Idiom Tagging Program is flexible in the sorts of sequence it can recognize, and in the sorts of operation it can perform: it can look either at the tags associated with a word, or at the word itself; it can look at any combination of words and tags, with or without intervening words. It can delete tags, add tags, or change the probability of tags. It uses an Idiom Dictionary to which new entries may be added as they arise in the corpus. In theory, the program can handle any number of idiomatic sequences, and thereby anticipate likely mistaggings by the Tag Selection Program; in practice, in the present project, we are using it in a rather limited way, to deal with a few areas of difficulty. Although this program might seem to be an *ad hoc* device, it is worth bearing in mind that any fully automatic language analysis system has to come to terms with problems of lexical idiosyncrasy.

4 FUTURE PROSPECTS

Our present overriding objective (in cooperation with our collaborators in Norway) is to complete the grammatical tagging of the LOB Corpus by the summer of 1983, and to make it available for research, through the Norwegian Computing Centre for the Humanities. We hope that its value as a research facility will more than justify the research which has led to the development of the Automatic Tagging programs. But in addition, we believe that the considerable success of these programs has helped to vindicate the value of corpus-based research in the automatic analysis of texts. The strength of computa-

tional corpus-based research is that the programs have to be designed to operate on unrestricted input, and can be progressively enhanced by the 'recycling' of data already analysed into the database.

If resources are available for future research, we hope to eliminate manual pre-editing, and to reduce further the percentage of error to be corrected in post-editing. One method for reducing error would be to derive different tag-pair frequencies from different kinds of text, and to use these in a 'fine-tuning' of the transition matrix for various styles of input text. For example, the frequencies for scientific and for fictional writing can be supposed to differ considerably, and statistical adjustments of the program to deal with these differences can be expected to eliminate additional errors. Even so, there will still be errors which cannot be corrected by enhancement of the present programs. Like Kučera and Francis (see Francis 1980), we have found special problems with certain classes of ambiguity, where the choice of wordclass requires reference to a wide context. Three difficult ambiguities are:

(i) that between IN and CS (e.g. *after* can be a preposition or a conjunction);

(ii) that between IN and RP or RI (e.g. *in* can be a preposition or a prepositional adverb); and

(iii) that between VBD and VBN (e.g. *acquired* can be a past tense verb or a past participle).

The following example shows the sort of problem which arises with the last case:

> ... some local authorities ... *have* not only *carried* out a very good business deal for themselves but also *acquired* a beauty spot for their people.

It is notable that if the word *have* were omitted from this sentence, the word *acquired*, which is the fourteenth word following it, would be changed from a VBN to VBD. This is because *carried*, which by virtue of the coordinate construction must be matched by *acquired*, would no longer be marked as the second verb of a perfective (*have* + past participle) construction. In other words, for this disambiguation a span of 14 words to the left of the target word is needed.

Such difficulties inevitably lead us to consider the deficiencies of word-tagging as an autonomous level of analysis. The most obviously

valuable levels of analysis to be added to word-tagging would be (a) syntactic analysis or parsing of a corpus; and (b) semantic tagging, whereby senses of words, as well as their grammatical categories, would be identified. These additional levels, on which work with the LOB Corpus has only recently begun,[11] would have to be added to the LOB Automatic Tagging programs if success in word-tagging were to approach 100%. The VBD/VBN ambiguity cited above, for example, could be successfully resolved only by a program which carried out recognition and tagging of larger-than-word units. There are strong reasons, indeed, for believing that the tagging programs will only reach their full potential when they are implemented in parallel with syntactic and (possibly) semantic analysis programs. These further challenges will remain when the present project is completed.

NOTES

1 Stig Johansson and Mette-Cathrine Jahr (see Johansson and Jahr 1982) have made major contributions to the project in the preparation of the WORDLIST and SUFFIXLIST (see 3.1). They are also undertaking roughly half of the post-editing. The research at Lancaster has been conducted by Ian Marshall, as well as the present authors. The Lancaster project has been supported by the Social Science Research Council (Research Grant HR 7081/1).

2 The Norwegian Computing Centre for the Humanities (director Jostein Hauge) has provided text processing facilities essential to the project. We have particularly appreciated the programming support provided at the Centre by Knut Hofland.

3 The percentage of 96.7% is based on the post-editing of c. 100 texts (i.e. c. 200,000 text words, or 20% of the Corpus). These texts are from categories B, C, F, G and R, representing a varied cross-section of the Corpus. There is little variation in the tagging success-rate between different categories. The figure of 96.7% excludes errors in the output which are not due to automatic tagging (these are chiefly pre-editing errors, and account for c. 0.1% of all words). Punctuation tags (see Appendix A) are discounted in calculating the success-rate.

4 Approximately 55% of the Corpus has been automatically tagged by November 1982.

5 Reported in Francis (1980); for results and analysis of the automatic tagging, see Francis and Kučera (1982).

6 An experiment carried out by Knut Hofland at Bergen in 1982 gave encouraging support to the view that manual pre-editing could be dispensed with. The LOB tagging programs were applied to a machine-readable copy of John Osborne's *Look Back in Anger*, a text not included in the LOB Corpus. Automatic pre-processing followed by

automatic tagging resulted in a success-rate in the region of 90%. This was without modifications to the programs themselves, which are designed to accept the specially pre-edited text of the LOB Corpus. (See p. 7f. above.)

7 See Marshall (1982:10-12) for further details.

8 Each of the three programs was written by a different member of the research team: A by Roger Garside, B by Eric Atwell, and C by Ian Marshall.

9 The Brown Wordlist contained c. 3,000 words, and the Brown Suffixlist contained c. 450 word-endings. See Johansson and Jahr (1982) on the LOB Suffixlist.

10 The marker @ indicates that a tag has (notionally) an intrinsic likelihood of 10% or less; the marker % indicates that a tag has (notionally) an intrinsic likelihood of 1% or less. The tags are also output in order of likelihood, more likely tags being placed to the left of less likely ones. To this extent, the Tag Selection program makes use of probabilities.

11 Roger Garside and Fanny Leech are currently working on programs to be applied in the parsing of the LOB Corpus. Manual work on semantic tagging is being undertaken at Stockholm by Magnus Ljung.

REFERENCES

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Francis, W. Nelson. 1980. 'A Tagged Corpus - Problems and Prospects'. In S. Greenbaum, G. Leech, and J. Svartvik, eds. *Studies in English Linguistics - for Randolph Quirk*. London: Longman. 192-209.

Francis, W. Nelson and Henry Kučera. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston: Houghton Mifflin.

Greene, Barbara B. and Gerald M. Rubin. 1971. 'Automatic Grammatical Tagging of English'. Providence, R.I.: Department of Linguistics, Brown University.

Johansson, Stig and Mette-Cathrine Jahr. 1982. 'Grammatical Tagging of the LOB Corpus: Predicting Word Class from Word Endings'. *In* S. Johansson, ed. *Computer Corpora in English Language Research*. Norwegian Computing Centre for the Humanities, Bergen. 118-46.

Marshall, Ian. 1982. 'Choice of Grammatical Word-Class without Global Syntactic Analysis for Tagging Words in the LOB Corpus'. Department of Computer Studies, University of Lancaster.

APPENDIX A: A SELECTION OF TAGS FROM THE LOB TAGSET

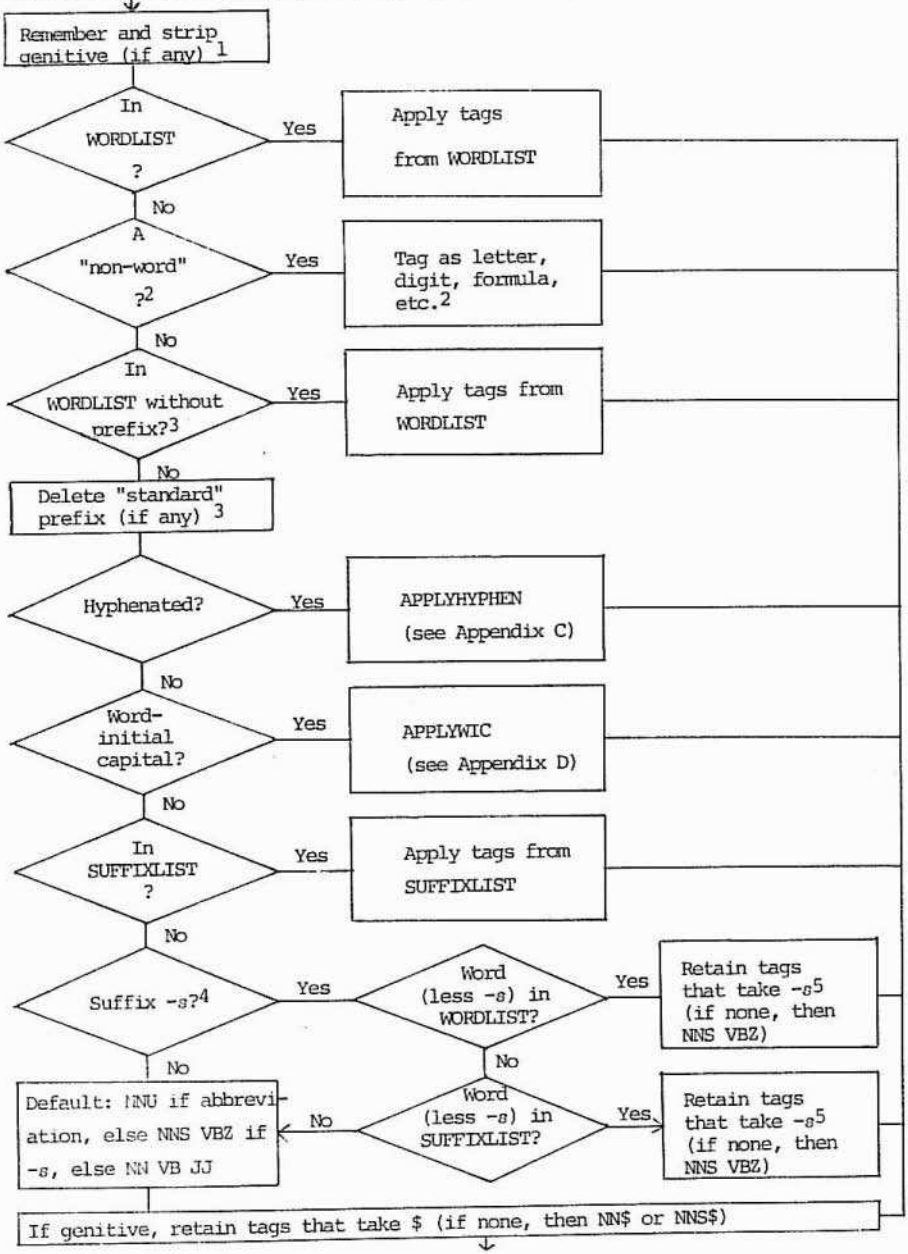
Note 1: The following punctuation tags represent themselves:
"." , "..." , "(" , ":" , "**'" , "*-" , "*'" , ")" , ";" , "?" , ":" , ","

Note 2: The letter "S" added to a tag marks it as plural; e.g. "NNS" = "plural common noun"

Note 3: The dollar sign added to a tag marks it as genitive or possessive; e.g. "NNS$" = "genitive plural common noun".

&FO	formula
AT	singular article (*a, an, every*)
ATI	singular or plural article (*the, no*)
CD	cardinal numeral
CD-CD	hyphenated pair of cardinal numerals
CS	subordinating conjunction
DT	singular determiner
DTI	singular or plural determiner
IN	preposition
JJ	adjective
JJB	attributive adjective
NNU	unit of measurement unmarked for number (e.g. *ft., cc., m.p.h.*)
NN	singular common noun
NNP	singular common noun with word-initial capital (e.g. *Irishman*)
NP	singular proper noun
NPL	singular locative noun with word-initial capital (e.g. *Square*)
NPT	singular titular noun with word-initial capital (e.g. *Mr, Lord*)
NR	singular adverbial noun (e.g. *north, home*)
OD	ordinal numeral
PP1A	*I*
PP1O	*me*
PP2	*you*
PP3	*it*
QL	qualifier (e.g. *very, more*)
RB	adverb
RI	prepositional adverb (homograph of preposition)
RP	prepositional adverb which can also be a particle
VB	verb (uninflected form)
VBD	past tense verb
VBN	past participle
VBZ	verb (3rd person singular present tense)

APPENDIX B: General flowchart of Tag Assignment Program (see 3.1)



NOTES

1 If the word ends in "*s* apostrophe" then strip the apostrophe; if the word ends in "apostrophe *s*" then strip both characters (and any preceding full-stop).

2 "Non-words" are the following:

a letter followed by zero or more digits (0 to 9), possibly followed by a single, double, or triple prime, tagged ZZ

a number* followed by "st", "nd", "rd" or "th", tagged OD

a number followed by "s" tagged CDS

a number containing "-", tagged CD-CD

a number followed by "apostrophe s", tagged CD$

a number followed (possibly) by a letter, tagged CD

a word containing a superscript or subscript, tagged &FO

a word containing letters and digits, but no hyphen, tagged &FO

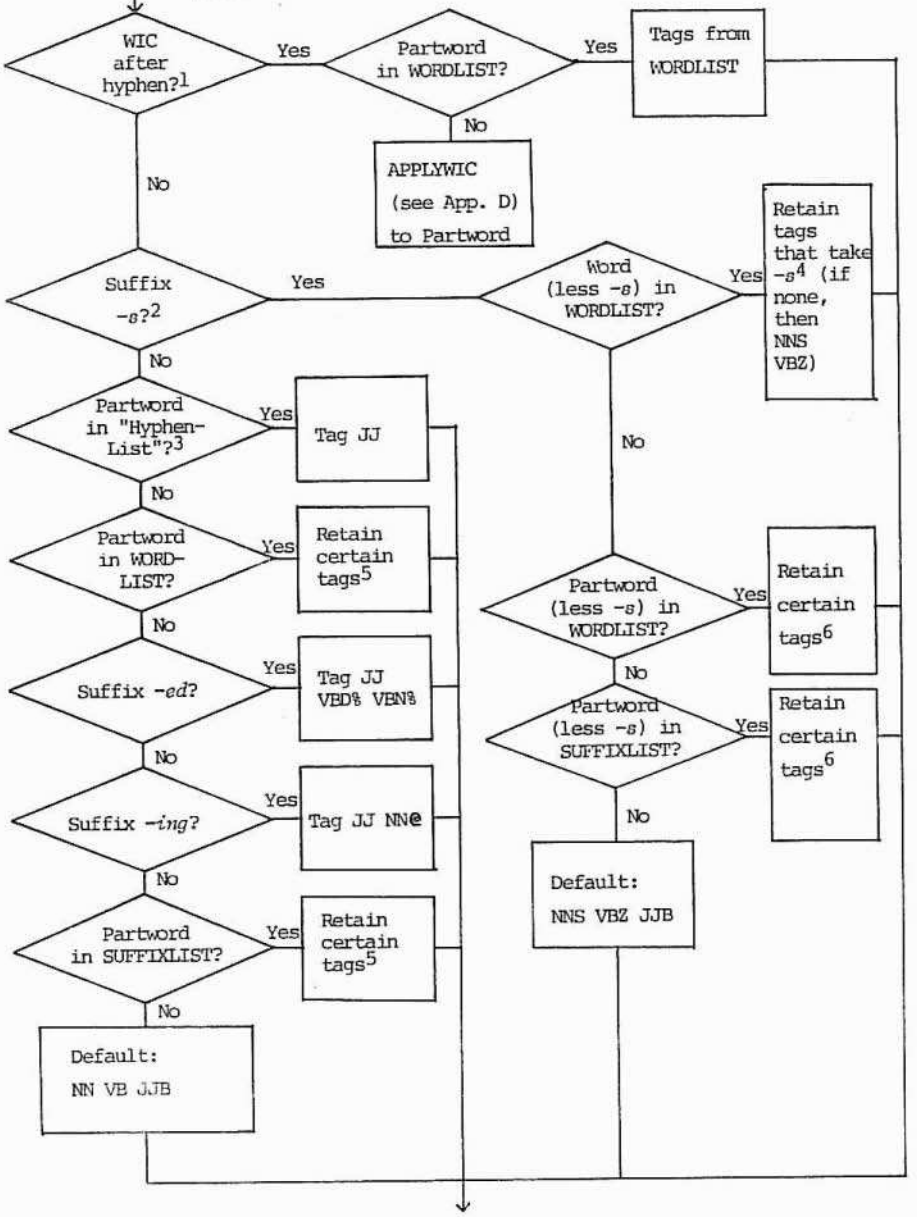
*In this context, a "number" means a sequence of digits (0-9) perhaps also including ".", "," and "/".

3 The "standard" prefixes include "a-", "co-", "counter-", "de-", "hyper-", "mis-", "out-", "over-", "re-", "retro-", "super-", and "trans-".

4 Words ending "ches", "shes", "sses", "zzes", "oes", "xes" have the "es" removed: words with 5 or more letters and ending in "ies" have the "ies" changed to "y"; words ending in "full-stop s" have both characters removed; other words ending in "s" (unless they end in "ss") have it removed.

5 Tags that take -*s* are VB (becoming VBZ) and CD, NN, NNP, NNU, NP, NPL, NPT, NR (becoming CDS, NNS, NNPS, NNUS, NPS, NPLS, NPTS, NRS).

APPENDIX C: Tagging decisions of APPLYHYPHEN
(Note: "Partword" means the characters after the last hyphen)



NOTES

1 "WIC" means "Word-initial Capital".

2 See Note 4, Appendix B.

3 The "Hyphen-List" consists of "class", "hand", "like", "price", "proof", "quality", "range", "rate", and "scale".

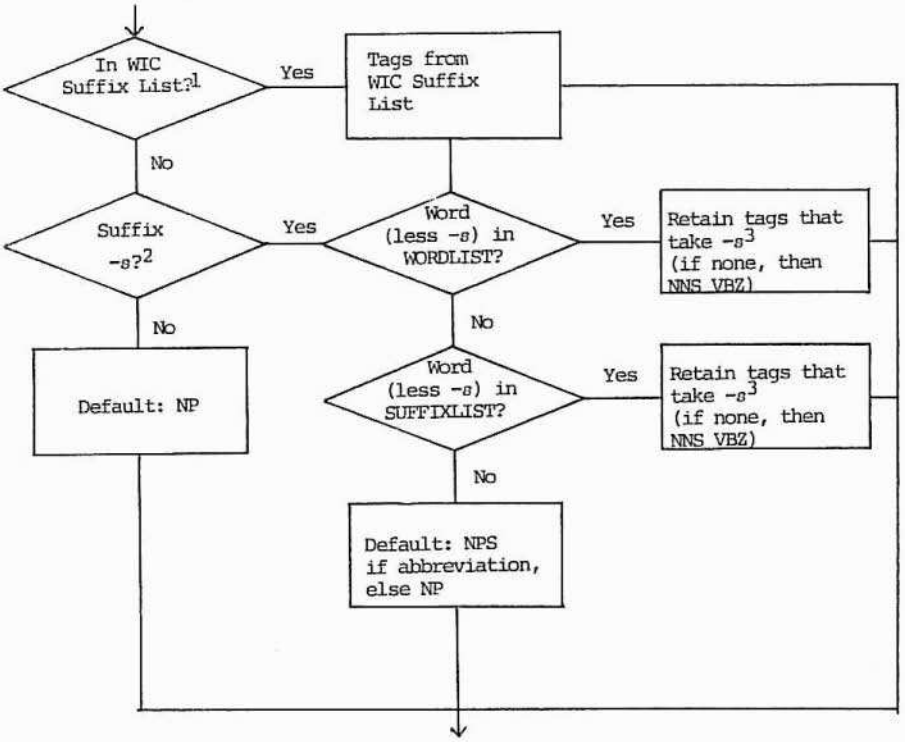
4 See Note 5, Appendix B.

5 For words not ending in "s", if IN is one of the tags, tag the word NN JJ@; if VBN is one of the tags, tag the word JJ; if VBG is one of the tags, tag the word JJ NN VBG@; if NNU is one of the tags, tag the word JJB; if NN with "normal" probability is one of the tags, tag the word NN JJB; otherwise leave the tags unchanged.

6 For words ending in "s", if IN is one of the tags, tag the word NNS; if VBG is one of the tags, tag the word NNS; if NNU is one of the tags, the tag is JJB; if NN with "normal probability" is one of the tags, the tag is NNS; otherwise retain tags that take "s" (see Note 5, Appendix B). If there are none, then tag the word NNS VBZ.

APPENDIX D: Tagging decisions of APPLYWIC

("WIC" means "Word-initial Capital")



Notes

1 The WIC Suffix List contains the following endings: "ic", "ese", "ite", "esque", "ish", "ism", "ean", "ian", "woman", "women", "ation", "ist".

2 See Note 4, Appendix B.

3 See Note 5, Appendix B.

APPENDIX E: SPECIMEN OF VERTICAL OUTPUT (before post-editing)

```
--------------------------------
        thus  RB
          it  PP3
          is  BEZ
       clear  [JJ]/73 RB%/23 NN@/3 VB@/1
        that  CS
         the  ATI
 predominant  JJ
organization  NN
           ,  ,
particularly  RB
          in  IN
         the  ATI
distribution  NN
          of  IN
manufactured  JJ
       goods  NNS
           ,  ,
          is  BEZ
         the  ATI
   wholesale  JJ
    merchant  NN
         who  WP
     carries  VBZ
      stocks  NNS
           .  .
```

APPENDIX F: THE SAME PASSAGE AS REHORIZONTALIZED OUTPUT

```
^ thus it  is  clear that the predominant organization, particularly
^  RB  PP3 BEZ  JJ    CS  ATI      JJ           NN     ,       RB

in the distribution of manufactured goods, is  the wholesale merchant
IN ATI      NN      IN      JJ       NNS , BEZ ATI     JJ       NN

who carries stocks.
WP    VBZ     NNS .
```