[From: Internat.conf. *Methodology & Techniques of Machine Translation*, British Computer Society, Cranfield, 13-15 February 1984]

The Grammatical Tagging of Unrestricted English Text

Roger Garside, Geoffrey Leech & Eric Atwell.

University of Lancaster

1. Introduction

The LOB (Lancaster-Oslo/Bergen) corpus is a million-word computer-readable collection of written British English texts (Johansson, Leech & Goodluck 1978). It consists of five hundred text extracts, of two thousand words each, organised as a set of fifteen "categories" to cover the different genres of written English. Thus, for example, category A covers newspaper reportage, category J covers learned articles, and category K covers general fiction. The corpus was constructed over the period 1971-78 in the Department of Linguistics at Lancaster University, the Department of English at Oslo University, and the Computing Centre for the Humanities at Bergen University. It was designed to be a parallel corpus to the Brown corpus of American English (Francis & Kuc'era 1979), which it matches in types of category, number and size of text extracts, and the general features of the coding system. The LOB corpus is available from the Computing Centre for the Humanities at Bergen, and has been extensively used as a database of usage in written English for linguistic research, both by itself and in conjunction with the Brown corpus.

The usefulness of the LOB corpus would be much enhanced if it was grammatically tagged; that is, each word in the corpus would have associated with it a symbol indicating its part of speech. Thus "lead" as a verb would be distinguishable from "lead" as a noun by inspecting the tag associated with the word, and it would be passible to search for a pattern of words involving a part of speech rather than a specific word, such as "to adverb verb" to catch split infinitives. We at the University of Lancaster (under SSRC grant HR 7081/1) and our colleagues at Oslo and Bergen are now completing a 23-year project to tag the LOB corpus automatically.

We wished to perform as much as possible of the tagging of the LOB corpus automatically by computer, to reduce the amount of manual processing required and to ensure consistency as far as possible in the tagging decisions taken. The Brown corpus had already been automatically tagged with an accuracy of something like 77% (Greene & Rubin 1971), and we aimed to design algorithms which would ensure a significantly higher success rate than this. There are two problems with this automatic tagging approach; first, the large number of homographs in English, and second, the open-ended nature of English vocabulary. There are about 50,000 word types in the LOB corpus, but we did not wish to rely on a dictionary of this size designed for the LOB corpus, but to have a mechanism involving a smaller dictionary which had the potential of being used on other texts.

As indicated above, the Brown corpus had already been grammatically tagged, and this provided us with three starting tools; (a) a set of tags which had been used for the Brown tagging, (b) the tagged Brown corpus, a database of information about what tags are associated with what words in what contexts, and (c) a tagging program TAGGIT, which did the automatic tagging of the Brown corpus, and which we used to investigate the areas where the automatic tagging system worked least well.

Because of our wish to use the Brown corpus as a data-base of tag information, and because we expected the tagged LOB corpus to be more useful if it was comparable to the tagged Brown corpus, we wished to retain the Brown tagset. However, we felt that there were a number of places where the Brown tagset was deficient, so a new LOB tagset of 134 tags was defined, and this is listed in the appendix. Broadly *it* follows the Brown tagset, but it has been refined in the area of words with an initial capital; thus the Brown tag IMP (proper noun) has been replaced by the LOB tags NP (proper noun), NPL (locative, for example "Wood" in "Burnham Wood"), NPT (titular, for example "President in "President Reagan"), NNP (common noun habitually written with an initial capital, such as "Mexican"), and their derived plural and possessive tags. There are also modifications in such areas as pronouns, adverbs and participles, and a number of minor additional tags for special cases.

2. Verticalising and Pre-Editing

The original LOB corpus consists of a series of lines of running text, with extra information relating to the typographic layout, such as new paragraph, change of typeface, etc., and with markers for words of non-standard English, such as abbreviations, foreign words, sub-standard English. The first phase of the tagging system involves a program which "verticalises" the text, followed by a manual pre-editing stage.

The main task of the verticalising program is to create a separate record for each word or punctuation mark in the corpus, with the word or punctuation mark in a standard place in the record, and with a reference number so that the record can be traced back to its original category, text extract, line and position in the line. However, there are a number of subsidiary tasks for the program;

(a) certain typographic information which is of no help to the automatic tagging system is discarded at this stage. This includes new paragraph symbols, changes of typeface, indications of the position of diagrams, etc.

(b) certain information which may be of use to the tagging system, or which should be retained as possibly of interest in the final tagged corpus, is moved to a subsidiary position in the record. This includes an indication of whether the current word is part of a heading, and the markers for non-standard English mentioned above.

(c) enclitics are treated differently in the Brown and LOB corpora. In Brown a word like "he'll" is given the tags for the pronoun "he" and the verb "will" joined with a special symbol. In LOB the orthographic unit "he'll" is treated as two separate syntactic units (or records) each with their own tag. The verticalising program therefore splits enclitics into the appropriate units, leaving markers in a subsidiary position in the records to show that the two units are crthcgraphically joined.

(d) It is the task of the remaining programs in the suite to assign a tag to each word. However, as can be seen from the appendix, the tag symbol associated with a punctuation mark is the punctuation mark itself, so this trivial tagging operation is performed by the verticalising program.

(e) The running text of the corpus is in lower case, but upper case occurs in a number of places; in words where the upper case should be retained "(McDonald", "NATO", "I'm"), but also in the word at the beginning of a sentence (where, because of the way the dictionary lookup works, the initial capital should be retained only if it would have occurred in the middle of the sentence), and also in places where a stretch of text is all in upper case. The latter is fairly rare, but occurs in newspaper headlines, for instance, where the text may actually be in upper case or where the case is indicated by a typeshift. The verticalising program attempts to recognise words where the upper case should be retained, and converts the rest to lower case, relying on manual intervention to correct this where necessary.

After the verticalising program has been run, the verticalises corpus is manually pre-edited to correct the corpus where necessary, and to tag certain words manually where it is known that the automatic tagging system is likely to fail. In addition, since the tagging system was being designed and constructed at the same time as the earlier parts of the pre-editing, the editors also collected information useful for inserting in the tagging system, such as lists of common abbreviations to add to the dictionary.

In order to help with the manual pre-editing, a suite of programs was written to extract from the original corpus lists of cases needing consideration. Several of these (such as the lists of arithmetic formulae and of abbreviations) were used mainly in constructing the tagging system, and would be unlikely to be used in pre-editing a new corpus; and consequential errors would be rare, and could be dealt with in the post-editing process. Other lists were more central to the pre-editing process, such as lists of words where the verticalising program retains or changes a word-initial capital letter; the editor would check each example, and correct the verticalised corpus where the program was in error. It is planned that the enhanced tagging system currently being developed will make more use of automatic methods of selecting the appropriate case-shift in these situations. Lists were also prepared of non-English words to be tagged manually, and graphically emphasised expressions (marked by typeshift or by quotation marks), as these might need tagging as cited words or marking in a subsidiary position as a title (for example, of a book).

3. The Tag Assignment Program

In the Brown system, the automatic tagging is all done by a single program TAGGIT. In our system we kept the separate operations as three separate programs, called WORDTAG, IDIOMTAG and CHAINPROBS. However, when the programs had been developed, a command language procedure was written which automatically applied each program in turn to a portion of the corpus.

It is the task of the WORDTAG program to assign one or more tags to each word in the corpus. If it assigns a single tag, it is assumed that this is the correct tag and it will not be changed by CHAINPROBS or the first stage of post-editing; however, it may be altered by the IDIOMTAG program or by the second, final checking, stage of post-editing. If WORDTAG assigns more than one tag, then CHAINPROBS will attempt to choose some one of these tags as the preferred one. An attempt is made by WORDTAG to order such a set of tags in approximately decreasing likelihood, and the markers @ or % may be attached to a tag to indicate "rare" or "very rare".

WORDTAG assigns these tags to a word considering it in isolation; it is the task of the CHAINPROBS program to select a tag on the basis of the context in which the word appears. The basis of WORDTAG is the first half of the TAGGIT program, but enhanced by the experience of using TAGGIT and by the availability of larger dictionaries derived from the data extracted from the Brown and LOB corpora.

The main mechanism for tagging words is a wordlist of some 7200 words and their associated tags. This wordlist contains all functional words ("in", "of", "who", "can"), and all common words in the LOB corpus. If this look-up fails, the next mechanism is a suffixlist look-up. This is a list of some 700 word-endings which are diagnostic of the appropriate tag for the word. WORDTAG takes each word which has failed to match the wordlist and attempts to match its ending against an entry in the suffixlist, working from more to less specific word endings. Thus there are, for example, entries for -able (adjective], -ble (noun or verb) and -le (noun), which would be matched in turn against a word ending in -le. Any case where this mechanism would fail (for example, "cable" and "enable") must be entered in the wordlist, so that the suffixlist look-up is not invoked. The wordlist and suffixlist must therefore be prepared together; the first versions of the lists were prepared at the Universities of Oslo and Bergen (Johansson & Jahr 1982], and additions and modifications made at Lancaster in the course of running the tagging system over portions of the LOB corpus.

Typically about 20% of the records or syntactic units processed by WORDTAG have already been tagged (mostly punctuation, but with some manually tagged words], 65% are tagged by searching the wordlist, and 9% are tagged by searching the suffixlist. Of the remainder another few per cent are dealt with by stripping an -s from a potential plural noun or third person singular of a verb, and looking up the result in the wordlist or (failing that] in the suffixlist. There are also special routines to deal with words containing non-alphabetic characters (numbers, formulae, etc.), various forms of hyphenated words and words with an initial capital, and other special cases. If all else fails (which it rarely does] WORDTAG assigns a default tagging of "noun, verb or adjective".

4. The Tag-Disambiguation Program

After WORDTAG has run, every record or syntactic unit has one or more tags associated with it, and about 40% are ambiguously tagged with two or more tags. The program CHAINPROBS attempts to disambiguate such words by considering their context, and then reordering the list of tags associated with each word in decreasing order of preference, so that the preferred tag appears first. With each tag is printed a figure representing the likelihood of this tag being the correct one, and if this figure is high enough CHAINPROBS simply eliminates the remaining tags. Thus some ambiguities will be removed, while others are left for the manual post-editor to check; in most cases the first tag, as preferred by CHAINPROBS, is the correct one.

The second part of the Brown TAGGIT program used what were termed context frame rules to disambiguate words in context. A context frame rule would be an encoded rule of the form:

```
if preceded by tag X
       and followed by tag Z,
              this tag must be a Y
```

or of the form:

```
if preceded by tax G
       and followed by tag Z,
              this tag cannot be a Y.
```

Any number of tags from zero to two could be specified as preceding or following the tag in question, and TAGGIT applied the more specific rules before the less specific. In this way TAGGIT attempted to remove all, or at least some, of the ambiguity.

We tried running the TAGGIT program over *a* portion of the LOB corpus. It became clear that a major problem was the presence of sequences of ambiguously tagged words, since the usable context in *a* frame rule had to be unambiguous. Thus, given a block of ambiguously tagged words, TAGGIT would try to work in from each end of the block applying "one-sided" rules. We wished to be able to take account of the strengths of links between two ambiguous tags as well as between an ambiguous and an unambiguous tag. It was also clear that despite the presence of frame rules taking account of a context of up to four tags, something like 80% of the rule applications involved a context of only one tag. Our plan was therefore to have two stages of disambiguation; the first pass would use co-occurrence information only about pairs of tags, together with a mechanism for dealing with blocks of ambiguously tagged words. The second disambiguation pass would use something more akin to the more specific context frame rules, and apply them to the ambiguities remaining from the first pass. However, the CHAINPROBS program developed for the first pass, with some modification to take account of larger contexts, was more successful than we had anticipated, and we dispensed with the separate second pass.

In order to apply our method of disambiguation, we needed a source of information as to the strengths of links between pairs of tags. This was derived from a sample taken from the tagged Brown corpus, and effectively gave us a matrix of probabilities of tag y occurring given tag x on the immediately preceding word. Some modifications had to be made to this matrix to take account of changes in the tag-set.

Given a sequence of ambiguously tagged words, the CHAINPROBS program uses the one-step probabilities to generate a probability for each sequence of ambiguous tags. Thus given words w_1 and w_4 unambiguously tagged t_1 and t_4 respectively, and words w_2 and w_3 each with two tags:

W_1	W_2	W_3	W_4
t_1	t_{21}	t_{31}	t_4
	t_{22}	t_{32}	

CHAINPROBS calculates the probabilities of the sequences t_1 t_{21} t_{31} t_4, t_1, t_{22} t_{31} t_4 t_{21} t_{32} t_4 and t_1 t_{22} t_{32} t_4, and from these derives a probability for each ambiguous tag. The details are given in (Marshall 1984).

Finally CHAINPROBS arranges the tags in descending order of preference, together with their associated probabilities. If the probability of the preferred tag is high enough, CHAINPROBS will eliminate all the remaining tags.

There are a number of situations where this single-step approach works less well. For example, an adverb often intervenes in a context where the word before the adverb is helpful in disambiguating the word after the adverb. CHAINPROBS is therefore provided with a set of "tag triples", each with an associated weighting factor, and these are used to modify the calculation of the probability of a tag sequence, where the co-occurrence of the three tags has a different probability to that of the occurrence of each of the tag pairs.

5. Multiple Syntactic Units and IDIOMTAG

The tagging system as originally conceived consisted of WORDTAG, to assign plausible tags to individual words, followed by the contextual tag disambiguation system. After we had tested this system over some portions of the corpus, it became clear that a useful addition would be a mechanism for assigning plausible tags to groups of words. For simplicity this is a separate program, IDIOMTAG, which modifies some of the decisions made by WORDTAG, and the output of which is fed for disambiguation into CHAINPROBS.

IDIOMTAG looks for any of a specified list of about 150 phrases, and modifies the tags accordingly. For example, if it finds the word "as" followed by a word which WORDTAG has assigned a tentative tag of "adjective" (possibly among others) followed by the word "as", as in "as old as", IDIOMTAG assigns the tag "qualifier" to the first "as" and the tags "preposition or (more rarely) subordinating conjunction" to the second "as"; WORDTAG would have assigned all three of these tags to each of the occurrences of "as".

One minor modification to the tagset was introduced with IDIOMTAG. There are a number of phrases where two or more separate orthographic units function syntactically as a single unit, for example "according to" as a preposition and "so that" as a subordinating conjunction. To deal with this we introduced a "ditto" tag marking which represents a grammatical tag covering two or more records in the tagged corpus, and IDIOMTAG assigns these markings.

6. The Post-Edit Phase and Results

Finally, the corpus is manually post-edited. This is done in two passes; the first is to look at all the remaining ambiguous taggings and decide whether CHAINPROBS's preferred tag is in fact correct, and the second is a manual check of the whole corpus. Corrections are made to the corpus in such a way as to preserve an indication of the type of correction needed; since this version of the corpus also retains information as to how WORDTAG selected the appropriate tags, whether IDIOMTAG was involved, and what probabilities were calculated by CHAINPROBS, it is possible to make a detailed analysis of the source and type of tagging errors; this is currently being done, but it appears that the automatic tagging system selects the correct tag in some 96-7% of cases.

For distribution a further program removes all this tagging information, leaving only the correct tag, and it can if desired return the corpus to a "horizontal" running text form, with the correct tags immediately under the words referred to. It is expected that the complete tagged LOB corpus will be available in the autumn of 1984.

7. Conclusions and Further Work

We have described a system for assigning grammatical parts of speech to words in running text, and to do this with a high degree of accuracy over texts which are unrestricted in vocabulary and contain passages of learned English, dialogue, non-standard English, etc. The system is robust in the sense that, given a text, it will always assign some tag to each word, however complex or erroneous the text.

Our current work at Lancaster includes further development of this tagging system. Our analysis of the errors arising from application of the current system will lead to enhancements to the three main tagging programs, and the tagged LOB corpus will be used to derive a new matrix of probabilities for use by CHAINPROBS. Thus the development of these tagging programs is an incremental process, in that each tagged corpus can be used as a database of information for tagging the next.

One major improvement we expect to make to the tagging system is to reduce the amount of pre-editing done. A lot of the manual pre-editing work reported here involved establishing what types of constructions caused problems, so this would not be repeated. Furthermore our experience in the pre-editing stage suggests ways in which more of the work could be done automatically (especially the decisions as to whether or not to retain word-initial capitals), and that some of the pre-editing could be omitted without significantly increasing the post-editing task. It is interesting that our colleagues in Bergen as an experiment ran the tagging system over a modem dramatic text, replacing the manual pre-editing phase with a small amount of extra automatic processing, and reported a success rate of over 90% (Hofland 1983); i.e. less than our current success rate, but still encouraging.

Finally, we believe that our use of probabilistic methods of grammatical tagging are of more general applicability. We are engaged in using similar techniques in a context-dependent text checking system for word-processors, and in further syntactic analysis of the LOB and other corpora;

i.e. generating what is in effect a surface parse of each sentence. We expect further applications to arise in speech-to-text/text-to-speech systems, and in intelligent front-ends to computer systems.

References

Francis, W.N., and Kučera, H. (1979) Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for Use with Digital Computers, Providence, R.I.: Department of Linguistics, Brown University.

Greene, B.B. and Rubin, G.M. (1971). Automatic Grammatical Tagging of English, Providence, R.I.: Department of Linguistics, Brown University.

Hofland, K. (1983) in 'Seminar on the Use of Computers in English Language Research', ICAME News 7, 1-12.

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

Johansson, S., Leech, G.N. and Goodluck, H. (1978) Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with Digital Computers, Oslo: Department of English, University of Oslo.

Marshall, I. (1984) 'Choice of Grammatical Word-Class without Global Syntactic Analysis for Tagging Words in the LOB Corpus', Computers and the Humanities (to appear).

Appendix Tagset for the LOB Corpus

.	punctuation tag - full stop
...	punctuation tag - ellipsis
(	punctuation tag - left bracket
!	punctuation tag - exclamation mark
&FO	formula
&FW	foreign word
**'	punctuation tag - close quotes
*-	punctuation tag - dash
*'	punctuation tag - open quotes
)	punctuation tag - right bracket
;	punctuation tag - semicolon
---	punctuation tag - new sentence marker
,	punctuation tag - comma
?	punctuation tag - question mark
:	punctuation tag - colon
ABL	pre-qualifier
ABN	pre-quantifier
ABX	pre-quantifier/double conjunction (BOTH)
AP	post-determiner
AP$	post determiner + genitive
APS	plural post-determiner (OTHERS)
APS$	plural post-determiner + genitive (OTHERS')
AT	singular article (A , AN , EVERY)
ATI	singular or plural article (THE , NO)
BE	BE
BED	WERE
BEDZ	WAS
BEG	BEING
BEM	AM
BEN	BEEN
BER	ARE
BEZ	IS
CC	coordinating conjunction
CD	cardinal
CD$	cardinal + genitive
CD-CD	hyphenated pair of cardinals
CDS	plural cardinal
CD1	ONE
CD1$	ONE'S
CD1S	ONES
CS	subordinating conjunction
DO	DO
DOD	DID
DOZ	DOES
DT	singular determiner
DT$	singular determiner + genitive
DTI	singular or plural determiner
DTS	plural determiner
DTX	determiner/double conjunction (EITHER,NEITHER)

EX	existential THERE
HV	HAVE
HVD	HAD past tense
HVG	HAVING
HVN	HAD past participle
HVZ	HAS
IN	preposition
JJ	adjective
JJB	attributive adjective
JJR	comparative adjective
JJT	superlative adjective
JNP	adjective with word-initial capital
MD	modal
NC	cited word
NN	singular common noun
NN$	singular common noun + genitive
NNP	singular common noun with word-initial capital
NNP$	singular common noun with w.i.c. + genitive
NNPS	plural common noun with w.i.c.
NNPS$	plural common noun with w.i.c. + genitive
NNS	plural common noun
NNS$	plural common noun + genitive
NNU	abbreviated unit of measurement unmarked for number
NNU$	abd. unit of measurement unmarked for number + genitive
NNUS	abb. plural unit of measurement
NNUS$	abb. plural unit of measurement + genitive
NP	singular proper noun
NP$	singular proper noun + genitive
NPL	singular locative noun with w.i.c.
NPL$	singular locative noun with w.i.c. + genitive
NPLS	plural locative noun with w.i.c.
NPLS$	plural locative noun with w.i.c. + genitive
NPS	plural proper noun
NPS$	plural proper noun + genitive
NPT	singular titular noun with w.i.c.
NPT$	singular titular noun with w.i.c. + genitive
NPTS	plural titular noun with w.i.c.
NPTS$	plural titular noun with w.i.c. + genitive
NR	singular adverbial noun
NR$	singular adverbial noun + genitive
NRS	plural adverbial noun
NRS$	plural adverbial noun + genitive
OD	ordinal
OD$	ordinal + genitive
PN	nominal pronoun
PN$	nominal pronoun + genitive
PP$	first possessive personal pronoun
PP$$	second possessive personal pronoun
PPL	singular reflexive personal pronoun
PPLS	plural reflexive personal pronoun
PP1A	I
PP1AS	WE
PP10	ME
PP10S	US
PP2	YOU
PP3	IT
PP3A	HE,SHE

PP3AS	THEY
PP3O	HIM,HER
PP3OS	THEN
QL	qualifier
QLP	post-qualifier ENOUGH,INDEED
RB	adverb
RB$	adverb + genitive (ELSE'S]
RBR	comparative adverb
RBT	superlative adverb
RI	adverb (homograph of preposition)
RN	nominal adverb HERE, THERE, NOW, THEN
RP	adverb which can also be a particle
TO	infinitival TO
UH	interjection
VB	verb
VBD	verb past tense
VBG	present participle
VBN	past participle
VBZ	verb 3rd person singular
WDT	wh- determiner
WP	wh-pronoun, neutral between nomin. & obj.
WP$	possessive wh-pronoun
WPA	nominative wh-pronoun (WHOSOEVER)
WPO	objective wh-pronoun (WHOM.WHOMSOEVER)
WRB	wh-adverb
XNOT	NOT or N'T
ZZ	letter of the alphabet