CHAPTER 3

The CLAWS word-tagging system

Roger Garside

1. Introduction

This chapter describes CLAWS (Constituent-Likelihood Automatic Word-Tagging System), a system for tagging English-language texts: that is, for assigning to each word in a text an unambiguous indication of the grammatical class to which this word belongs in this context. The first version of this system was developed over the period 1981 to 1983 at the Universities of Lancaster, Oslo, and Bergen. This version (CLAWS1) was designed to assign a grammatical tag to each of the million words in the LOB corpus, and achieved 96–97% accuracy (the precise degree of accuracy varying according to the type of text); the remaining errors were removed by a manual post-editing phase. Work is at present proceeding at Lancaster to develop a second generation of the tagging system (CLAWS2), to increase its accuracy, and reduce its reliance on manual pre-editing and the particular coding conventions of the LOB Corpus; this enhancement is described in Chapter 8. The present chapter gives a general overview of the complete CLAWS1 tagging system, and describes in detail the mechanism for assigning a set of candidate tags to each word in the text; a later program in the system selects a preferred tag from this set, and this is described in more detail in Chapter 4.

The tagset used in CLAWS1 was derived from the one used in tagging the Brown Corpus (Greene and Rubin 1971). We wished to retain overall comparability of the tagged LOB Corpus with the tagged Brown Corpus, although we did modify the Brown tagset in the area of proper nouns and pronouns, ending up with a total of 133 possible tags for the syntactic units (words and punctuation marks) of the corpus. For CLAWS2 we decided to develop the tagset in the light of our experience with automatic tagging and parsing systems, resulting in a new set of 166 tags. In both sets the tags each consist of from one to five characters, and are intended to have mnemonic significance; the tagsets are discussed in detail in Appendix B. In most chapters (including this one) we use the first set of tags, called in Appendix B the "LOB tagset"; whenever the second, "Lancaster" tagset is referred to, attention will be drawn to the fact.

There is a general assumption in tagging that there is a one-to-one correspondence between tags and orthographic units. However, this correspondence breaks down with contracted forms (as, for example, in *can't*, *they'd*, *I'll*) and certain idiomatic phrases. It was decided that the CLAWS system would use a different mechanism for dealing with contracted forms from that used in the Brown tagging system. In the Brown system an orthographic unit such as *can't* is assigned two tags representing "modal + *not*": in CLAWS *can't* is split into two syntactic units, *can* and *n't* (with an indication that the two syntactic units are a single othographic unit), and the two units are separately tagged. A late addition to CLAWS allowed for "multi-word units", where two or more orthographic units are assigned a single tag (which the UCREL team generally refer to as a "ditto-tag"): thus *because of* and *such as* are each tagged as a preposition, *as if* as a subordinating conjunction, and *at once* as an adverb. This is discussed further in Chapter 9.

2. Overview of the tagging system

The input to the CLAWS1 tagging system is a text in the format of the LOB Corpus. An example is:

```
D01    2 |^*0With so many problems to solve, it would be a great help to
D01    3 select some one problem which might be the key to all the others, and
D01    4 begin there. ^If there is any such key-problem, then it is undoubtedly
```

This extract is from text category D text extract 1 lines 2 to 4, as indicated by the reference numbers on the left. Notice the symbol ^ marking the beginning of each sentence, and the symbols | and *0 meaning (respectively) "new paragraph" and "normal (roman) typeface". The Corpus is unrestricted in vocabulary and syntax, and contains foreign words and phrases, dialogue, incomplete and non-standard English, etc.

The output from the tagging system is a tagged text. For example, the first sentence in the above excerpt would appear as overleaf in the table on p. 32.

The text has been reformated or "verticalized", with each word or punctuation mark occupying its own line, and being at a fixed position within the line. Each such line has a reference number linking it back to the line, and position within the line, of the word in the original "horizontal" text; punctuation marks are given a reference number subordinating them to the preceding word, and sentences are "framed" by a special new-sentence-marker line. The remainder of the line is taken up with the word itself, the associated tag, and special markers (there are none in this extract) for such things as headings, foreign words, contracted words, etc. The tags are from the LOB tagset, listed and discussed in Appendix B.

There are two problems with an 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; we did not wish to rely on a

D01	2 001	
D01	2 010	with
D01	2 020	so
D01	2 030	many
D01	2 040	problems
D01	2 050	to
D01	2 060	solve
D01	2 061	,
D01	2 070	it
D01	2 080	would
D01	2 090	be
D01	2 100	a
D01	2 110	great
D01	2 120	help
D01	2 130	to
D01	3 010	select
D01	3 020	some
D01	3 030	one
D01	3 040	problem
D01	3 050	which
D01	3 060	might
D01	3 070	be
D01	3 080	the
D01	3 090	key
D01	3 100	to
D01	3 110	all
D01	3 120	the
D01	3 130	others
D01	3 131	,
D01	3 140	and
D01	4 010	begin
D01	4 020	there
D01	4 021	
D01	4 022	

IN
QL
AP
NNS
TO
VB
,
PP3
MD
BE
AT
JJ
NN
TO
VB
DTI
CD1
NN
WDT
MD
BE
ATI
NN
IN
ABN
ATI
APS
,
CC
VB
RN
.

dictionary of this size designed for the LOB Corpus, but preferred a mechanism involving a smaller dictionary which had the potential of being used on other texts. The Brown Corpus had already been automatically tagged with an accuracy of something like 77% (Greene and Rubin 1971), and we aimed to design algorithms which would ensure a significantly higher success rate than this. In achieving this goal we had the benefit of three tools, made available by the Brown team: (a) a set of tags which had been used for the Brown tagging: (b) the tagged Brown Corpus, a database of information about the associations between words, tags and contexts; and (c) a tagging program, TAGGIT, which carried out the automatic tagging of the Brown corpus, and which we used to investigate the areas where the automatic tagging system worked least well.

The CLAWS tagging system consists of five separate stages applied successively to a text to be tagged:

(a) **A pre-editing phase** This stage, partly automatic and partly manual, prepares the text for the tagging system proper; it is described in section 3.

(b) **Tag assignment** Each word in the input text is assigned a set of one or more tags. This assignment phase does not look at the context in which the word appears, so the assigned set of tags should include any tag that could be appropriate to the word in some possible context. This stage (the program WORDTAG) is described in sections 4 and 5.

(c) **Idiom-tagging** This stage looks for a number of special word or tag patterns where a limited amount of context could narrow down the set of possible tags assigned to a word. This stage (the program IDIOMTAG) is described briefly in section 7, and in more detail in Chapter 9.

(d) **Tag disambiguation** This stage inspects all cases where a word has been assigned more than one tag, and attempts to choose a preferred tag by considering the context in which the word appears, and assessing the probability of any particular sequence of tags. This stage (the program CHAINPROBS) is described briefly in section 6, and in more detail in Chapter 4.

(e) **A post-editing phase** This stage involves a manual process in which erroneous tagging decisions by the computer are corrected, followed by a reformating stage (the program LOBFORMAT) to eliminate unnecessary subsidiary information provided by the tagging system and produce a final tagged corpus. This is described in section 8.

In the Brown system, stages (b) to (d) are all executed by a single program TAGGIT. In our system we kept the separate operations as three separate programs (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.

3. Verticalizing and pre-editing

The CLAWS1 system was developed for use primarily on the LOB Corpus, and is now being revised to deal with other formats of input text. The 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 special items such as abbreviations, foreign words, or substandard English. The first phase of the tagging system involves a program (PREEDIT) which "verticalizes" the text, followed by a manual pre-editing stage.

The main task of the PREEDIT Program is to create a separate line for each word or punctuation mark in the corpus, with the word or punctuation mark in a standard place in the line, and with a reference number so that it 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 line. This includes an indication of whether the current word is part of a heading, and the markers for special items mentioned above.

(c) As mentioned above, a contracted form such as *he'll* is treated as two separate syntactic units each with its own tag. The PREEDIT program therefore recognizes these cases and splits them into the appropriate units, leaving markers in a subsidiary position in the lines to show that the two units are orthographically 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 Appendix B, the tag symbol associated with a punctuation mark is the punctuation mark itself, so this trivial tagging operation is performed by the PREEDIT 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 first word of a sentence (where the initial capital should be retained only if it would have occurred if the word were found in the middle of a sentence), and in a number of rarer situations with continuously capitalized texts. The PREEDIT program attempts to recognize words where the upper case should be retained, and converts the rest to lower case, relying on manual intervention to correct this where necessary. This is a place where significant manual intervention is currently required, so the new version of CLAWS is being written to attempt to deal with capitalization without manual intervention.

After the PREEDIT program has been run, the verticalized corpus is manually pre-edited to correct the text where necessary, and to tag certain words manually where it is known that the automatic tagging system is likely to fail. In order to help with this manual pre-editing, a suite of programs was written to extract from the original Corpus lists of cases needing consideration. Since the CLAWS1 system was being designed and constructed at the same time as the earlier parts of the pre-editing, several of these lists (such as lists of arithmetic formulae and of abbreviations) were used mainly in suggesting types of linguistic feature which the tagging system had to cope with, and would not be used in pre-editing a new corpus.

Other lists were more central to the pre-editing process, such as lists of words where the verticalizing program retains a word-initial capital letter or where it changes the letter to lower case; the editor would check each example, and correct the verticalized text where the program was in error. As mentioned above, 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 more rarely occurring features (such as non-English words) so that they could be manually tagged. Our policy with CLAWS2 is similarly to

eliminate manual insertion of tags at this stage, in the expectation that consequential errors will be rare and can be dealt with during manual post-editing.

4. The tag assignment program (WORDTAG)

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; however, it may be altered by the IDIOMTAG program or during manual post-editing. If WORDTAG assigns more than one candidate tag, then CHAINPROBS will attempt to choose one of these candidates as the preferred one. An attempt is made by WORDTAG to order such a set of candidate tags in approximately decreasing likelihood, and a "rarity marker" may be attached to a tag (see below).

WORDTAG assigns 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 basic plan of WORDTAG is provided by the first half of the Brown 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. It is designed to be open-ended in the sense of coping as far as possible with unrestricted English, including neologisms, deviant spellings, etc. The program consists of a sequence of rules, ordered so that later rules are applied to a word only if all earlier rules have failed. These rules were constructed by an iterative process, involving the testing of WORDTAG over a portion of the Corpus, an analysis of the results, and subsequent modifications to the WORDTAG rules. The program proceeds as follows:

(a) Some syntactic units will already have been tagged before WORDTAG is reached, either automatically or manually by the first stage of the tagging system, as described above. WORDTAG simply accepts these tagging decisions.

(b) The next step is to look up the syntactic unit in a lexicon. A lexicon of some 7200 words is used, containing the word and up to six possible tags for the word; thus the word *round*, for instance, has possible tags JJ, RI, NN, VB, and IN (i.e. adjective, a certain type of adverb, noun, verb, and preposition). The tags are listed in approximately decreasing likelihood, and may be marked "@" meaning "rare" (likelihood nominally less than 10%) or "%" meaning "very rare" (likelihood nominally less than 1%). Thus the lexicon entry for *round* is in fact:

round JJ RI NN VB@ IN@

If the word is found in the lexicon it is assigned all the tags listed in its entry; otherwise the program applies a sequence of further tagging rules which are to be described.

The lexicon contains all function words (*in, my, was, that*, etc.), the most frequent

words in the open classes noun, verb, and adjective, and any words which are exceptions to the general tagging rules which will be applied to unlisted words. Conversely, certain types of derived forms (plurals of nouns, comparatives of adjectives) do not need to appear in the lexicon, since the general tagging rules will correctly assign the appropriate tags. Thus the construction of the lexicon has been an iterative process, taking into account the tagging rules added to WORDTAG and their exceptions. The original version of the lexicon together with the word-ending list or "suffixlist" was constructed in Oslo and Bergen (Johansson and Jahr 1982), and it was extended in Lancaster by adding some 200–300 common abbreviations, some 400–500 common words with a word-initial capital, and a number of other words or syntactic units. This lexicon accounts for a large proportion (65–70%) of the tagging decisions made by WORDTAG.

(c) The next step is to eliminate a wide class of syntactic units which are not strictly speaking words. The types of tagging decision made here cover such cases as:

- *$37.00* and *£2* are tagged NNU (unit of measure)
- *i, b27, x″* are tagged ZZ (letter of the alphabet; *I* and *a*, being exceptions, are in the lexicon)
- *27th, 1st* are tagged OD (ordinal)
- *1940s* is tagged CDS ("plural" cardinal)
- *1950–7* is tagged CD–CD (hyphenated cardinal)
- *1940's, 3's* are ambiguously tagged CDS or CD$ ("plural" cardinal, or cardinal with genitive, as in *Louis 14's reign*: in this example the LOB Corpus includes a marker for "type shift into Roman numerals", which has been ignored by the PREEDIT program)
- Other numbers (*2, 19.6, 4,000,000,* $\frac{1}{2}$, etc.) are tagged CD (cardinal)
- Various expressions like H_2SO_4, $a-4$, $E=mc^2$ are tagged &FO (formula).

All other orthographic units, containing only letters and accents, apostrophes and hyphens, together with such partially-numeric expressions as *12-year-old*, are left for later rules to process. Notice here that a certain amount of care has been taken to assign a correct tag to these non-word structures despite their low frequency, and the examples illustrate the types of thing to be met in unrestricted English text.

(d) The next step deals with words containing hyphens, and is discussed in detail in the following section.

(e) The next step deals with words which retain an initial capital letter after the manual pre-editing phase. (The same set of rules is used within the procedure for dealing with hyphenated words, if there is a capital letter after the (last) hyphen.) The rules are as follows:

- Look the word up in a special list of suffixes or word-endings for words with initial capitals. This includes such endings as *-ish* and *-ian*, which commonly occur in capitalized words, with the appropriate tags.
- Strip any final *-s*, and look the word up in the lexicon or, failing that, the special word-ending list mentioned above.
- Assign a default tag of NP (proper noun). This is the most commonly applied rule for words retaining an initial capital.

(f) If the above steps fail, the next step is to look up the word in a list of word-endings, the "suffixlist"; this consists of about 720 word-endings (for words *without* initial capitals) with their associated tag or tags. The suffixlist contains sequences of up to five letters, including "suffixes" in the ordinary sense, such as *-ness* (noun), but also any word endings which are associated invariably (or at least with high frequency) with certain word classes, for example *-mp* (noun or verb) – the letters *-mp* do not constitute a morphological suffix, but it is a fact that almost all words ending with these letters are either nouns or verbs (the few exceptions, such as *damp* and *limp*, are listed in the lexicon).

The suffixlist is searched for the longest matching word-ending. Thus there are entries in the list for *-able* (adjective), *-ble* (noun or verb), and *-le* (noun), and these will be tested for in that order; exceptions (such as *cable* and *enable*) are in the lexicon. This step succeeds for most of the words not found in the lexicon, typically 7–12% of the words in the text.

While this step is quite successful, it is being extended in the revised tagging system. It is possible to envisage a generalized morphological analysis, which would successively strip a sequence of suffixes. Instead we are concentrating on particular troublesome suffixes; for instance, if the suffix *-er* is stripped and a test made of the word-class of the stem, it enables this step in many cases to disambiguate between agentive noun and comparative adjective.

(g) The suffix *-s* is not dealt with by the above test. Instead the suffix *-s* (but not *-ss*) is stripped; and action taken to construct a putative singular noun or first person singular verb by stripping a trailing *-e* or changing *-ie* to *-y* in the appropriate cases. The resulting character-sequence is looked up first in the lexicon and, if that fails, in the suffixlist. If any tags are assigned by this procedure, then only those compatible with the final *-s* are retained as possible tags for the original word. The possible cases are that the base form of a verb becomes a third person singular form, and various noun classes become plural. Thus the word *kinds* found in a text would have its final *-s* stripped to become *kind*, which is found in the lexicon with tags NN JJ@ (i.e. noun or more rarely adjective); the JJ tag is rejected as incompatible with the *-s* suffix, and the NN tag is converted to NNS (i.e. plural noun), which is therefore the (unique) tag assigned to the word *kinds* by WORDTAG. If none of the tags selected for the s-stripped word are compatible with the *-s* suffix, then the tags NNS VBZ (i.e. plural noun or third person singular verb) are assigned, but the word is marked for possible manual attention.

(h) If all the above rules fail, certain tags are assigned by default. We have just seen that the default tagging for words ending in *-s* is NNS VBZ; all other words are by default tagged NN VB JJ (i.e. noun, verb, or adjective). Very few words receive this default tagging (a total of about 200 out of the million words of the Corpus, mostly foreign words and deviant spellings).

(i) Before step (a), a test is made for the genitive markers *'s* and *s'*; if found they are stripped off the word, and their occurrence noted. Constructions like *1940's* are dealt with in step (c). Any other syntactic unit which is associated with a genitive marker is considered when all the above rules have been tried and some tag assignment made.

Only those tags compatible with a genitive marker are retained; thus, for example, NP (proper noun) becomes NP$. If no tags are compatible, a default tag (either NN$ or NNS$) is assigned and the word marked for possible manual attention.

5. Hyphenated words

If a syntactic unit is a word (by the criteria mentioned in (c) above) and contains one or more hyphens, then the following steps are performed:

(a) The first step is to search before the (first) hyphen for a special set of prefixes which do not generally affect the classification of the word of which they are part. If any of the prefixes *a-*, *co-*, *counter-*, *de-*, *hyper-*, *mis-*, *out-*, *over-*, *re-*, *retro-*, *super-* and *trans-* are found, then the prefix is stripped off and the remaining word tagged by trying all the rules in the preceding section (starting with the lexicon look-up); that is, the word is tagged as if the prefix was not present, so that (for example) the word *a-dying* receives the tags of *dying*.

(b) Similary, if the first letter after the hyphen is a capital letter, that part before the hyphen is ignored and the remaining word is tagged by the rules given in step (e) of the preceding section (for words with an initial capital), so that (for example) the word *un-American* is tagged as if it were *American*.

(c) Next, the part of the word after the (last) hyphen is looked up in the lexicon and, failing that, the suffixlist. If this search is successful, the program attempts to deduce tags for the complete word from the tags found for the "part-word" applying the following rules in sequence:

- If the tags of the part-word include IN (preposition), assign tags NN (noun) and JJ@ (rarely adjective), for example *washing-up, well-off*.
- If the tags of the part-word include VBN (past participle), assign tag JJ (adjective), for example *self-employed, so-called*.
- If the tags of the part-word include VBG (present participle), assign tags JJ (adjective) NN (noun) and VBG@ (rarely present participle), for example *fact-finding, fierce-looking*.
- If the tags of the part-word include NN (noun) unmarked for rarity, assign tag NN (noun) and JJB (attributive adjective), for example *income-tax, long-term*.

A similar sequence of steps is followed for hyphenated words ending in *-s*.

(d) Various special-purpose procedures are inserted in this sequence of steps. For example, before step (c) a check is made to see whether the part of the word after the (last) hyphen is one of the set *-class*, *-free*, *-hand*, *-like*, *-price*, *-proof*, *-quality*, *-range*, *-rate* and *-scale*; any one of these causes the full word to be tagged JJ (adjective). An example would be *middle-class*; exceptions like *price-range* must be in the lexicon.

Failing all else a default (NN VB JJB) is assigned; there were about 100 words tagged in this way in the Corpus.

6. The tag-disambiguation program

After WORDTAG has run, every syntactic unit has one or more tags associated with it, and about 35% 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 associated 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.

This disambiguation mechanism requires a source of information as to the strengths of links between pairs of tags; much of this information was derived from a sample taken from the tagged Brown Corpus, and effectively gives us a matrix of probabilities of tag y occurring given tag x on the immediately preceding word. Given a sequence of ambiguously tagged words, the CHAINPROBS program uses these 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:

$$\begin{array}{llll} w_1 & w_2 & w_3 & w_4 \\ t_1 & t_{21} & t_{31} & t_4 \\ & t_{22} & t_{32} & \end{array}$$

CHAINPROBS calculates the probabilities of the sequences:

$$t_1\ t_{21}\ t_{31}\ t_4$$
$$t_1\ t_{21}\ t_{32}\ t_4$$
$$t_1\ t_{22}\ t_{31}\ t_4$$

and

$$t_1\ t_{22}\ t_{32}\ t_4$$

and from these derives a probability for each ambiguous tag. More details of this process are given in Chapter 4.

7. Multiple syntactic units and IDIOMTAG

The tagging system as originally conceived consisted of WORDTAG, to assign plausible tags to individual words, followed by CHAINPROBS to disambiguate the tags in context. 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, since with this we could eliminate certain obvious classes of error. 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, suppose it finds the word *as*, followed by a word to which WORDTAG has assigned a set of candidate tags which include JJ (adjective), followed by the word *as*, for example *as old as*. IDIOMTAG assigns the tag QL (qualifier) to the first *as* and the tags IN CS@ (preposition or rarely subordinating conjunction) to the second *as*. WORDTAG would have assigned all three of these tags to each of the occurrences of *as*, so the amount of ambiguity to be dealt with by CHAINPROBS is reduced.

One minor modification to the tagset was introduced with IDIOMTAG. There are a number of cases where two or more separate orthographic units function syntactically as a single unit. A number of examples were given in section 1 (p. 31), and another is *as well as*, which is an exception to the pattern described in the previous paragraph. To deal with this we introduced a "ditto-tag" marking which represents a single grammatical tag covering a sequence of two or more orthographic units in the tagged Corpus, and these markings are assigned by IDIOMTAG; *as well as* would for example be tagged CC (conjunction). Chapter 9 of this book discusses the IDIOMTAG program in more detail, and some of the problems it raises.

8. The post-edit phase

After the LOB Corpus had been processed by CLAWS1, it was manually post-edited. This was done in two passes: the first was to look at all the remaining ambiguous taggings, and decide whether CHAINPROBS's preferred tag was in fact correct, and the second was a manual check of the whole Corpus, since we required the tags assigned to the words of the LOB Corpus to be as accurate as possible. For other uses of the tagging system this manual post-editing phase might be reduced in scope or even omitted. Subsequently a third phase of checking has been performed on the tagged LOB Corpus in Oslo and Bergen; this has involved extracting various lists of particular tags in context, in order to check the consistency of the final published tagged Corpus.

Corrections were 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 all tagging errors. The results of such an error analysis have guided the construction of CLAWS2.

For distribution a further program (LOBFORMAT) removes all the extra

information, leaving only the correct tag, and it can if desired reformat the corpus into a "horizontal" running text form, with the correct tag immediately next to the word to which it refers.

9. Conclusions

This chapter has described a system for assigning grammatical parts of speech to words in running text, a task which it performs 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 is being 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.