11 The robust tagging of unrestricted text: the BNC experience

ROGER GARSIDE

11.1 Introduction

The production of annotated machine-readable corpora has been a central activity of the UCREL team at Lancaster University, led by Geoffrey Leech, since the early 1980s. This commenced with the annotation of the LOB (Lancaster-Oslo/Bergen) corpus with part-of-speech information over the period 1981–84 (Garside, Leech and Sampson 1987). The work has continued with corpora which introduced syntactic annotation at the constituent level (Leech and Garside 1991) and at the level of anaphora (Fligelstone 1992; Garside 1993), with corpora marked with word sense and key semantic relationship information (Wilson and Rayson 1993), and with aligned multi-lingual corpora (McEnery *et al* 1994). Most recently it has resulted in the development of the British National Corpus (BNC), a corpus of one hundred million words of varied written and spoken texts annotated with part-of-speech information (Leech 1993). The BNC was constructed by a team of publishers (Oxford University Press, Longman, and Chambers Harrap), academic institutions (Lancaster University and Oxford University Computing Services) and the British Library, over the period 1991–94; Lancaster was responsible for the grammatical tagging of the corpus.

The various types of annotation listed above have always been inserted in the text by a mixture of automatic and manual procedures, generally making use of some form of probabilistic technique to assign an analysis to a text at the appropriate linguistic level, and this can then be manually post-edited if necessary to achieve the desired level of accuracy. Because large quantities of text are involved, the post-editing will always involve some assistance by the computer to minimize the time taken to implement the analyst's decision, and to check the validity of each change. The results of the manual correction

10.6.5 Conclusions

Machine learning of language models from corpus resources is a fledgling research field, with much to learn![4] The pioneering developers of corpus resources had backgrounds in English linguistics and language teaching, so naturally saw this as the prime application area for corpus linguistics. Lancaster University continues to be the leading centre for the production of corpus resources (see, for example, Leech and Garside 1991, Eyes and Leech 1993, Black *et al* 1993). The rapidly-growing field of Information Technology, and specifically Speech And Language Technology, is finding new uses for corpus resources as training data for machine learning of robust large-scale language models; Geoffrey Leech's work is finding applications in a whole new field of research.

Notes

1 ICAME = International Computer Archive of Modern English, an international network of corpus linguists, with a base at Bergen University. ICAME publishes an annual Journal, and holds annual International Conferences; see Souter and Atwell (1993) for a list of past venues and proceedings.

2 DTI = Department of Trade and Industry, which subsidises Industrial SALT research and development projects; EPSRC = Engineering and Physical Sciences Research Council, which funds University SALT research projects; HEFCs' NTI = Higher Education Funding Councils' New Technologies Initiative, which funds IT infrastructure for UK Universities, including SALT training resources.

3 As fellow Star Trek fans will know, the Undiscovered Country is the Future.

4 The Association for Computational Linguistics has recently set up a Special Interest Group on Natural Language Learning, ACL-SIGNLL; for further details, contact its President, David Powers, Flinders University (powers@cs.flinders.edu.au) or Secretary, Walter Daelemans, Tilburg University (walter@kub.nl).

check the validity of each change. The results of the manual correction (for example, revised probability values) can also be used to improve the subsequent automatic processing of other similar texts.

A preliminary task in all the work described above has been the preparation of the texts (tokenization, explicit marking of sentence breaks, etc.) and the assignment of part-of-speech markers (word-tags or, often, simply 'tags'). Since the early 1980s this has been carried out at Lancaster by a program or suite of programs called Claws (Constituent Likelihood Automatic Word-tagging System). This was one of the first part-of-speech taggers to make use of probabilistic information, achieving a 95–6% accuracy rate (Marshall 1983; see also Church 1988 for another early system). A consequence of the assignment of part-of-speech markers is the wish to associate a single tag with a sequence of two or more contiguous words, such as 'Preposition' with 'according to' or 'Conjunction' with 'as well as' (at least in some cases). For historical reasons we use the term 'dittotag' for this concept, making use of it (for instance) to assign an appropriate tag to a sequence of foreign words; for example, the tag 'Plural Common Noun' would be assigned to 'hoi polloi'.

11.2 The history of Claws

The first version of Claws (in retrospect called Claws1) was developed for tagging the LOB corpus over the period 1981–84. It incorporated what we have since referred to as the C1 tagset of 133 tags, a revision of the tagset used by Greene and Rubin in their work (Greene and Rubin 1971). Over the period 1983–86 parts of this suite of programs were rewritten (as Claws2), mainly to allow the program to make various decisions (such as the positioning of sentence breaks) which Claws1 did not have to make, since the information had already been explicitly encoded by hand in the LOB corpus. This development was part of a general attempt to avoid requiring manual pre-processing of the text before it could be processed by Claws, replacing it by optional manual post-editing to correct the automatic annotation if the required accuracy rate warrants it. The tagset used was a rationalization of the C1 tagset, the C2 tagset, subsequently modified to form the C2a tagset.

Preliminary work was carried out in 1987 on a further development of Claws to make use of verb subcategorization. This work was never developed further than a lexicon containing verb

subcategorization information, referred to as the Claws3 lexicon. Thus the next (and current) version of Claws is Claws4, first written in 1988, and incorporating the earlier suite of five programs into a single program. Claws4 has been further developed (we are currently at revision 19) to allow for the extensive changes required by the BNC project. One fundamental change has been to attempt to make the Claws program as independent as possible of the tagset, since the BNC uses two tagsets — a main tagset (C5) with 62 tags with which the whole of the corpus has been tagged, and a larger (C7) tagset with 152 tags, which has been used to make a selected 'core' sample corpus of two million words.

11.3 Issues concerning orthography

In early versions of Claws various different forms of annotation were used to mark special features of the text (such as new paragraphs, or a change of font) and for indicating special characters, such as the accented letters. The LOB corpus had a specially devised coding scheme (Johansson, Leech and Goodluck 1978) which was used for Claws1. Subsequent work used other coding schemes, largely based on the TeX text mark-up system (Knuth 1984). A major revision of the Claws system was entailed by the use of SGML (Standard Generalized Mark-up Language: see Goldfarb 1990) for the BNC, and the current version of Claws assumes that the input text is either in straight-forward ASCII with no special text coding conventions, or makes use of coding conventions based on SGML. Because other forms of annotation are commonly found in the more readily available collections of machine-readable text (such as in electronic mail and on the World Wide Web), we have written various pre-processing programs to convert these formats automatically into an SGML-based format which Claws can handle.

An attempt has been made to retain flexibility in the range of SGML annotation to be expected in a text, and its interpretation. Thus one of the resources read by the Claws program when it starts up is a list of SGML tags and entities, together with an indication of how each item is to be processed by Claws.

If accented letters are represented in the text it is assumed that the SGML form is used; for example the SGML entities 'À' and 'à' respectively represent upper and lower case 'a' with a grave accent. The SGML entity table used by Claws indicates what

would be the equivalent letter or letters without the accent, and words are held in the Claws lexicons without accents. Thus the word 'naïve' would be represented as 'naïve' in the input text, and would be transformed to 'naive' before it is looked up in the Claws lexicon. This mechanism allows Claws to handle more complicated cases of transliteration into a conventional character set before look-up in the lexicon, for example '&oelig'; representing the ligature 'œ', or even 'þ', representing 'þ' or 'ð' transliterated as 'th'. Since accents are represented erratically in English text, and are not usually significant in part-of-speech marking, we have chosen to use this rather straight-forward procedure; an alternative, suggested in Sampson (1989b), is to have the lexicon indicate the position of all possible accented letters in a word, and then perform a look-up with and without the accents. In principle the above transliteration scheme could be used for Greek characters as well, but these are marked as such in the SGML entity table, and are used to indicate that this is not an orthographic word in the conventional sense, but a foreign word or mathematical formula.

Other possible interpretations of special characters recognized by Claws are 'noise symbols' (such as the copyright mark '©'), which are treated as if they are not present during lexical look-up; 'formula symbols' which indicate that the immediate stretch of surrounding text is likely to be a mathematical formula (as, for example, in 'a≤3'); 'currency symbols' and 'unit symbols' which indicate that the immediate stretch of text is probably a number and an indication of a unit of measurement (as in '£5' and '70°'); and 'punctuation symbols' (such as '¡') which are to be treated like the normal punctuation marks. It is also possible to indicate to Claws in this way alternative representations for different types of quotation marks, dashes, ellipsis marks, etc., all of which are to be treated in the same way as the normal forms of these characters. Thus this facility allows unusual characters to be represented in a text, and for Claws to be able to treat them in an appropriate way.

SGML 'tags' are used to mark the beginning and end of an element of text of a certain type (a paragraph, a chapter, a heading), perhaps together with one or more attributes with particular values for each instance of the element (identifying number, rendition information, and so on). On the whole Claws did not make use of these in any significant way during the processing of the BNC. Most SGML beginning and end tags were treated as items of text which were to be passed directly to the output and not assigned part-of-speech tags.

Furthermore, when Claws is disambiguating a sequence of text words, intervening SGML tags should normally be treated as if they were not there. Thus the sequence

<hi r=it>Don't</hi> call me that.

which represents italicized *Don't* for emphasis, is tagged and disambiguated as if it were

Don't call me that.

Some SGML tags, such as for the beginning and end of a paragraph (in written text) or of an utterance (in spoken text), indicate the position of a sentence break, even if the orthography of the text at this point does not warrant it. This information is specified in the list of SGML tags, and actioned by Claws.

A text in the BNC generally begins with an extensive set of header information, indicating the provenance of the text, its coding conventions, the editorial process, etc. There follows the text proper, and then possibly some trailer information. The text to be tagged could also consist of several separate segments of text, each with its own header information. In order for Claws to be able to distinguish sections of text to be tagged from header information to be transferred to the output unchanged, the list of SGML tags can indicate that certain beginning SGML tags mark the start of the text for part-of-speech tagging, with the corresponding end tag marking the end of the current stretch of tagging.

We have made an attempt in Claws to allow for normal variation in orthographic practice, particularly in the area of capitalization and the presence or absence of abbreviational full stops. All words are held in the lexicon in lower case, and each tag associated with a word includes a notation specifying the normal orthography associated with that tag. Thus a tag would be indicated as appropriate for a word which was all in upper case, or with an initial capital, or all in lower case; and as appearing with a final abbreviational full stop, with internal full stops, or with no full stops. The lexicon look-up procedure attempts to extract a suitable list of potential tags based on these markers and the orthography of the word in the original text. There has to be some flexibility in this extraction, because (for example) of changing patterns in the use of abbreviational full stops, and of course because of the presence of sentence-

initial capitalization. So the list of potential tags for a word will in appropriate cases include tags marked for a different orthography than that present in the text. This includes the case of a text appearing all in upper case, as for example in a heading. Since headings are explicitly marked in the BNC, it would have been possible to have the tag extraction procedure make use of this information. However we did not do this, since texts can also appear set all in upper case (particularly in electronic mail) and we wanted Claws to be able to handle this type of situation. Because Claws attempts to make use of normal conventions in orthography, it is less successful in handling text all in lower case, which also occasionally occurs in electronic mail.

Claws does not attempt to retain the exact word spacing of an original text, assuming that the 'normal' conventions of spacing between words and between preceding and following punctuation marks apply. In order to retain the convention of there being only a single word-tag associated with any individual unit of text, we have chosen to split into two or more separate textual units certain character strings which are conventionally written as a single orthographic word, such as 'cannot' and 'don't'. In cases like this Claws inserts a marker in the output text to show that the separate units were a single word in the original text. In the BNC project a much expanded range of words was treated in this way (for example, 'gonna', 'gorra', 'gotta'), and in the final version of the corpus the markers introduced by Claws were used to recover the spacing of the original text. There were a small number of mainly mathematical texts where the layout of word spacing was unconventional, and the system discussed above did not work. It would have been possible for Claws to insert in its output an explicit indication in all cases of the exact form of the spacing between units, but the frequency of this phenomenon did not warrant the change.

Since the BNC contains a section of spoken texts, Claws has to be able to handle textual representations of whatever speech phenomena are transcribed. This turned out not to be a major problem, since the transcription of the spoken parts of the BNC does not attempt to transcribe such speech phenomena as stuttering, etc., although such things do occur occasionally in representations of direct speech in some parts of the written part of the BNC. The coding conventions of the spoken part of the BNC incorporate a marker for truncated words, and these are simply marked with the 'unknown word' tag. Code was also incorporated into Claws to handle dropped

initial 'h' and final 'g', by modifications to the lexical look-up which would, for example, match ' 'avin' ' to 'having'. Procedures were considered to handle representations of other speech phenomena in the written texts, such as 'm-m-must' and 'zooooooom', but were not implemented because of their relative rarity in the BNC.

11.4 The structure of the Claws tagging system

The Claws tagging program is divided into six main sections, as follows:

1. the input running text is read in, divided into individual tokens, and sentence breaks are recognized.

2. a list of possible tags is then assigned to each word, the main source being a lexicon.

3. a number of words in any text will not be found in the lexicon, and for these there is a sequence of rules to be applied in an attempt to assign a suitable list of potential tags.

4. since the lists of potential tags from steps 2 and 3 are based solely on individual words, the next step uses several libraries of template patterns to allow modifications to be made to the lists of tags in the light of the immediate context in which a word occurs.

5. the next step is to calculate the probability of each potential sequence of tags, and to choose the sequence with the highest probability as the preferred one.

6. finally the text and associated information about tag choice is output.

In this section we briefly discuss each of these steps, and the modifications as a result of the BNC project. The output from Claws may be manually post-edited, and possibly re-formatted; this is discussed further in section 11.5.

For the BNC project we had a large quantity of texts of various kinds to be tagged over a relatively short period of time. The tagging

initial 'h' and final 'g', by modifications to the lexical look-up which would, for example, match ' 'avin' ' to 'having'. Procedures were considered to handle representations of other speech phenomena in the written texts, such as 'm-m-must' and 'zooooooom', but were not implemented because of their relative rarity in the BNC.

11.4 The structure of the Claws tagging system

The Claws tagging program is divided into six main sections, as follows:

1. the input running text is read in, divided into individual tokens, and sentence breaks are recognized.

2. a list of possible tags is then assigned to each word, the main source being a lexicon.

3. a number of words in any text will not be found in the lexicon, and for these there is a sequence of rules to be applied in an attempt to assign a suitable list of potential tags.

4. since the lists of potential tags from steps 2 and 3 are based solely on individual words, the next step uses several libraries of template patterns to allow modifications to be made to the lists of tags in the light of the immediate context in which a word occurs.

5. the next step is to calculate the probability of each potential sequence of tags, and to choose the sequence with the highest probability as the preferred one.

6. finally the text and associated information about tag choice is output.

In this section we briefly discuss each of these steps, and the modifications as a result of the BNC project. The output from Claws may be manually post-edited, and possibly re-formatted; this is discussed further in section 11.5.

For the BNC project we had a large quantity of texts of various kinds to be tagged over a relatively short period of time. The tagging

of the texts was carried out by a team of analysts, who, as well as running the tagging system over the texts, were also post-editing selected portions of the BNC, particularly the early sections of a new genre of texts, which might indicate places where the tagging system was not working as well as it might. The results of this post-editing would be a series of proposed changes to the resources used by the tagging system, such as the lexicon and libraries of patterns, and occasional changes to the code of the Claws program itself. For this reason we attempted to hold the resources used by Claws in as flexible form as possible, so as to ease the problem of updating them, particularly given the fact that two tagsets were in use.

The resources used by the Claws system are as follows:

1. the list of tags to be used for the current text.

2. the list of SGML tags and entities and an indication of how each is to be processed.

3. the lexicon of words and potential tags.

4. a list of word-endings and potential tags.

5. a set of other miscellaneous lists, including such things as the list of words to be split (such as 'don't' and 'gotta'), how they are to be split, and the tag to be assigned to each portion of the word.

6. the libraries of template patterns.

7. the matrix of probabilities of tag combinations, from which the probability of a tag sequence can be calculated.

In order to make the lexicon easy to handle by the analysts it has been split into several sections, which are merged when Claws begins to run. One of these sections is arranged to allow it to overwrite entries in the other sections. Thus a normal run of Claws would make use of a set of standard sections, perhaps together with a supplementary section for the particular type of text being processed. A standard use of the supplementary section has been to hold words which have to be treated differently in the spoken part of the BNC. Because of the success of the rules for dealing with words not in the

lexicon, it has been possible to keep the lexicon fairly small. The main part of the lexicon contains some ten thousand words, and a separate section holds a list of some five thousand proper names of various types. The lexicon currently in use holds for each word a list of potential tags and associated orthographic information. It also holds information as to the frequency of use of a word-tag combination, and this can be in either of two forms; there is either a set of markers for the analyst to indicate a rough frequency (in the range 'common', 'rare', 'very rare'), or a set of frequency figures can be inserted from a tagged text. For the BNC tagging project the former hand-annotation of frequency information was used, but in a follow-up project we are experimenting with probability figures extracted from the 'core' part of the BNC for re-tagging other parts of the BNC.

Some 65–70% of words are assigned a set of potential tags from the lexicon. An important source of tags for other words is the list of word-endings. Claws attempts to match an unknown word against the word-ending list, working from longest to shortest, and assigns the tags indicated. A relatively recent addition to this has been the possibility of distinguishing word-endings with a particular significance for words with an initial capital from word-endings with a general significance. There are a number of further procedures for textual items which fail to match both the lexicon and the word-ending list. These include procedures for words which consist of digits, of a mixture of digits and letters, or contain special characters apart from accented letters — this third type is assumed to be either a combination of a number and an indication of the unit involved, or an arithmetic formula. If all else fails, a word will be assigned an appropriate list of the possible open-class tags, taking into account capitalization and whether or not the word ends with an 's'. These procedures are described in detail in Garside (1985).

Our early experience with the first version of Claws suggested that there were a number of simple patterns of words which the probabilistic disambiguation was getting wrong, or where it could be helped by eliminating one of the tags where we could be certain it was contextually inappropriate. Furthermore there was a requirement to assign a single tag to a sequence of orthographically distinct words, the 'dittotag' mentioned in the first section. We chose to handle this by having Claws match a library of templates against each sequence of words, a successful match requiring a modification to the list of potential tags already assigned to a word. The modification could involve deletion of some, or all but one, of the potential tags; or it

could involve the assigning of a tag distinct from all the tags previously assigned to this word, such as a dittotag. We originally referred to the library of templates (incorrectly) as an 'Idiomlist', and we have tended to continue to use this term. In Claws we have treated the dittotag as a special type of template matching; of course it could instead, and perhaps more logically, have been treated as a part of the lexicon, given an extended form of lexicon look-up.

This simple type of template matching has been extensively developed in more recent versions of Claws. One of the problems present in all versions of this mechanism is how to deal with overlapped patterns. On the one hand the matching of a shorter pattern might cause a tagging change which would allow a longer pattern to match; thus a decision to assign to the sequence 'as yet' a dittotag indicating that it is an adverb might allow the matching of a longer sequence containing 'as yet' to a template consisting of the various parts of a potential verb phrase. On the other hand, we will generally want to look for longer matches in preference to shorter matches. There are a number of ways of handling this, none of which is entirely satisfactory; in Claws it is handled by scanning from left to right through the text, searching simultaneously for all matching templates starting at each position in the text. If there are several matches at a certain point then one is chosen, using a value function which involves the length of match and also the type of match, since it could be on the word, on one of the potential tags, or simply an unspecified intervening word (and in later versions of Claws it could further be a part of a word, such as 'any word ending -ing', or a part of a tag, such as any of the noun tags). Then the tag changes associated with the preferred template are implemented, any unfinished template matches in the scope of this successful match being abandoned.

To allow more flexibility Claws now has three separate libraries of templates (although the same template can appear in more than one library if required). The first set of templates is matched to the input text, and the changes for any matched template are implemented. The second set of templates is then matched against the resulting text, and again the actions of any matched templates are implemented. The basic idea is that all the more-or-less fixed phrases, including dittotags, will be dealt with on the first pass, and the more general patterns will be searched for on the second pass, making use of the tags corrected by patterns on the first pass. A third pass is then made with a third library of templates, but this takes place *after* the probabilistic disambiguation phase. If a template specifies a tag to be

matched on the first or second pass, it is allowed to match against any of the list of potential tags associated with a word; but on the third pass the match must be against the tag preferred by the probabilistic disambiguation. Thus the third pass can be used for templates which would overgenerate matches if allowed to match against *any* potential tag. The above describes the rationale for the design of the template matching system, but because of the complex interactions possible within a text the decision as to how to write and place the individual templates depends on a degree of judgement and experimentation; this is discussed from a more linguistic point-of-view in chapter 12.

Within this multi-pass generalization of the template matching, other smaller changes have been made to allow more flexible forms of matching. We have already mentioned matching against part-words and part-tags; it is also possible to require the match against the word or tag to fail if a template match is to succeed, to require that an (otherwise unspecified) word has a word-initial capital, to specify alternative word or tag matches in the same template, and to allow repeated optional matches (to some specified maximum) against a portion of the template. The current version of Claws has some three thousand template patterns (eighteen hundred in the first pass, nine hundred in the second, and three hundred in the third). To allow the analysts to update the template libraries easily, template lists of different types are held separately and merged at run-time; for example, expressions involving compound nouns, foreign phrases, expressions involving proper names, and patterns involving verb sequences are all held as separate lists.

The probabilistic disambiguation phase is essentially the same technique that was used in earlier versions of Claws. The Viterbi alignment procedure is used to indicate the most likely sequence of tags where there is a sequence of one or more alternatives. The Viterbi calculation is now also used to estimate the probability of each alternative tag assigned to a word, since this value is required to implement the 'portmanteau tag' evaluation described in section 11.5. These probabilities are calculated from the word-tag probabilities in the lexicon, whether estimated by the analysts or extracted from tagged text, and from tag-tag probabilities, extracted from tagged data and read in as one of the Claws resources. The current version of Claws can make use of first- or second-order tag-tag probabilities; for the tagging of the BNC we have used first-order probabilities, but we are currently investigating the use of second-order probabilities (i.e. tag trigrams) (Leech, Garside and Bryant 1994a).

While performing the Viterbi alignment certain textual items can be treated as if they were not present in the sequence. Thus all SGML tags are ignored, as are all words in a list supplied as part of the Claws resources — generally this includes only 'fillers' in spoken text such as 'er' and 'erm'. This list of items to be ignored is also used during the template matching described earlier.

One other form of contextual disambiguation can be carried out by Claws; this is the recognition of repetitions in spoken texts. A repeated sequence of words (perhaps including one or more fillers) is tagged as if the repetition and the fillers were absent; this is discussed in more detail in Garside (1995).

11.5 The post-processing of tagged output

The output from the Claws program is in what we term 'vertical' format, with a line per textual item. Each line contains a reference number (linking the item back to the original input line), the textual item itself (word, punctuation mark, SGML tag, or whatever), the list of potential part-of-speech tags with the preferred tag listed first, some information about the tagging process (such as the source of the list of potential tags, whether lexicon, word-ending list, hyphenation procedures, etc.), and some subsidiary information, such as markers for textual items joined together in the input, and a marker for cases where internal checks during the tagging procedure indicate that the tagging of a word may be insecure (such as where the orthography of an input word does not match the possible orthographic forms of the word listed in the lexicon). This form of output is recognized by all the post-processing programs, including the editing software.

A problem for the BNC project was that an input file might contain a certain amount of data which cannot be fitted into this format; there will be an extensive set of header information which is not to be tagged, there may be SGML tags which contain extended lists of attribute and value pairs, etc. Because of this the output from Claws was redesigned so that the normal tagged output was still produced in the above format, so that the post-processing programs did not have to be changed. But the data which cannot be put in this format are output to a supplementary file, and a cross-reference to it is inserted in the primary output file. This is also how long textual items are now processed. Earlier versions of Claws simply truncated long textual items at the twenty-fifth character with an error message;

now the long word is inserted in the supplementary file, with a cross-reference in the primary file so that it can be extracted by the post-processing programs as required.

One form of post-processing performed to varying degrees on different output texts is post-editing by human analysts, to correct erroneous tagging and other decisions by Claws. The 'core' part of the BNC has been post-edited in full, while only selected portions of the rest of the corpus have been post-edited, to establish the types of error occurring in different genres. During most of the BNC project a special purpose text editor, called the LB editor and written by Tony McEnery, was used to perform this task.

More recently the LB editor has been replaced by an X-Windows-based editor called Xanthippe, which carries out essentially the same tasks. A section of the text being post-edited is presented on the screen, together with the tag preferred by Claws and a list of the other potential tags considered but rejected. Because the information per word is fairly large, only a small number of words can be displayed in this format. So a separate window displays the same words and surrounding text, but without the tagging information so that a larger context can be seen; the two windows are linked, so that they scroll together.

The most common type of tagging error is one where the list of potential tags is correct, but Claws has selected the wrong one. In this case the analyst selects the correct tag, and it is promoted to the preferred tag position. If the correct tag is not among the list of potential tags for this word, Xanthippe displays a menu of all possible tags from which the correct tag can be selected; a supplementary list allows a sequence of words to be tagged with a selected dittotag. These tag-correcting operations are the most frequent tasks carried out using Xanthippe, but there are facilities for correcting a textual item (for example, if it is a typographic error), for splitting or joining words (and adjusting the link markers, if necessary) if the Claws tokenization is in error, for inserting and deleting sentence breaks, and for inserting notes of various types to comment on the text or the tagging decisions.

For the BNC project a suite of programs was run after Claws to check various features of the SGML structure of the document, to match the spacing of the original to the final text, and so on. A further program converts the text into the final form of the corpus at the completion of processing at Lancaster, with the words back in 'horizontal' running text format, and with the tags represented as

SGML entities. A number of further reformating procedures were required at this stage. One example was that, with the insertion of an SGML tag to indicate a sentence break, certain parts of the SGML structure of the original document might be invalidated; for instance, a <hi> tag indicating a font change (perhaps for emphasis) which crossed a sentence boundary. In situations like this the post-processing program has to replicate the <hi> tag on either side of the sentence break. Another issue was the use of 'portmanteau tags'. This is a tag indicating that the Claws choice between two tags is particularly uncertain, so that the tag allocated to the word is a combination of the two possibilities. The post-processing program recognizes such situations (of a pair of tags with probability figures from Claws lying within a certain range of each other) and adjusts the tag accordingly (for more details see Leech, Garside and Bryant 1994a).

11.6 Conclusion

An error analysis has been performed on the automatic tagging of a portion of the BNC, using the smaller (C5) tagset (Leech and Smith 1995). This shows that there is an error rate of about 1.7%, and about 4.7% of the tags are portmanteau tags. The report gives further details, such as error rates for particular tags and for particular tag pairs.

We are currently engaged in an EPSRC-funded project (GR/K14223) to improve the tagging of the BNC, making use of the manually post-edited 'core' corpus. This is being used as a source of revised Claws resources which will be used to re-tag the BNC; it is also being used as a source of information about error patterns, from which we are developing 'patching' techniques for correcting the main corpus.