Encoding and Markup for the Oxford Pilot Corpus

Draft 1

Jeremy Clear

17th September 1990

1. Introduction

This paper considers the problems of encoding and marking-up electronic texts held as part of a growing English language corpus. It is assumed in this paper that one of the major problems is the feasibility of applying a rigorous, detailed, interpretive markup on each individual text, when the texts to be included in the corpus are numerous and represent a very broad range of text types. The Text Encoding Initiative has already published draft guidelines dealing with the markup of machine-readable texts for interchange, and many of the technical aspect of the use of SGML to record features of written texts have been thoroughly treated there. The TEI Guidelines are intended to form the basis of new standards, and they are written to encompass a very wide range of potential applications for marked-up text, including editing with wordprocessors, construction of hypertext systems, formatting and printing, loading into free-text or conventional databases, linguistic tagging and parsing, collation for critical editions and content analysis. Consequently, the TEI Guidelines propose a markup which achieves a high level of detail and descriptive generality. This paper is partly a response to the TEI Guidelines and makes reference to them throughout: whenever the TEI proposals seem suitable for adoption I have not bothered to repeat the details of the encoding.

The immediate problem for the compiler of a general language corpus is that the cost of ensuring that the corpus is fully marked up in conformance with TEI guidelines is high in terms of manual effort, and the cost/benefit balance may be badly upset unless it can be demonstrated that a high standard of markup will bring worthwhile gains in ease of processing, accuracy and re-usability. What is needed in order to build up a large and useful corpus is a level of markup which maximises the utility value of the text without incurring unacceptable penalties in the cost and time required to capture the data.

The need for compromise is clear. There has been over the past few years a tremendous growth in interest and activity in the area of corpus building and analysis. European, USA and Japanese efforts in the development of NLP and IT are converging on the recognition of the importance of some sort of corpus-based research as part of the infrastructure for the development of advanced language processing applications. Statistical processing of text corpora has been demonstrated as a viable approach to some of the traditional hard problems of computational linguistics, machine translation and knowledge engineering. The

drift is clearly towards gathering very large corpora (hundreds of millions of words of running text). The TEI offers the timely opportunity to create standards in text encoding without which the value of text corpora could be seriously limited. But the drive towards standards must not bear so heavily on the work of corpus linguistics that it acts as a brake on progress. Indeed at a very local level, within individual commercial or research organizations, the cost/benefit analysis of introducing a sophisticated level of markup into corpora will be evaluated in each specific case and, as so often in the matter of international standards, local pratice will tend to dictate emerging standards rather than the reverse.

This paper will consider primarily the issue of marking up texts which are converted into electronic form from written (usually printed) sources. A complication is introduced by the current proliferation of text which is created in electronic form but which may be disseminated either in print or electronically. In the last section I will consider the problems of markup in relation to spoken sources.

2. Presentational vs. Descriptive Mark-up

The TEI Guidelines[1] refer to two approaches which can be taken in marking-up texts. One may mark up the underlying structural features of a text (the sentences, paragraphs, sections, footnotes, etc.). These are usually signalled by spacing, punctuation, type font and size shifts and so on, but there is no one-to-one correspondence between features and realization. Or one may prefer to encode the typographical features themselves.

> The former approach, that of *descriptive* markup, allows for more sophisticated analysis and processing of the text, at the cost of requiring more time and effort and at the risk of introducing subjective or erroneous decisions. The latter approach, that of *presentational* markup has the advantage of making it simpler to tag texts acquired from typesetting tapes or optical scanners. Either approach may be used in tagging texts for TEI-conformant interchange.

The two types of markup are not mutually exclusive categories; they describe the two ends of a scale. All markup is descriptive to some extent. For example, it is conventional in print to adjust the amount of white space between words and characters to achieve a flush right margin on the page, but the text when converted to an ASCII file format will almost certainly *not* contain any encoding of the actual extent of physical white space between characters: we impose a simple descriptive markup which represents the variety of white space as either an inter-letter space (no encoding) or a inter-word space (one ASCII SPC (040) character). The descriptive markup can be taken to increasingly higher levels of abstraction, but at the lowest levels the markup can be treated as if it were merely a representational encoding of the physical features of the text.

In building a corpus as a basis for empirical linguistic study of a language variety, it will not be practical to adhere rigorously to either a descriptive or presentational principle. In some instances it will be important to "record what's there on the page", since the researcher will be concerned to discover patterns of usage relating to features of punctuation and spelling. A representational markup will be preferred for example for the following:

- use of fullstops in abbreviations

[1] *Guidelines for the Encoding and Interchange of Machine-Readable Texts*, edited by C M Sperberg-McQueen and L Burnard, Draft Version 1.0, July 1990 (hereinafter referred to as TEI-P1)

- italics or quote marks for mentioned words
- use of commas in numeric strings
- use of digits versus words for numbers

In other cases, the computational considerations relating to the processing of the text in indexing and free-text retrieval software are likely to take precedence. A descriptive approach will usually be more appropriate for

- numbers, letters, bullets, asterisks, etc. in lists
- opening and closing speech marks
- delimiting and referencing of footnotes

3. Computational vs. Manual Processing

3.1 The Information Source

The cost of collecting and processing a corpus lies substantially in the amount of human effort necessary to achieve the desired results. Computer hardware costs are falling steadily in real terms (such that it is quite normal today for the 1M words of the LOB or Brown corpora to be held and processed on a standalone desktop or laptop PC) but skilled human labour is available to most projects only at a high and rising real cost. These facts encourage corpus specialists to eliminate any redundancy in text encoding which would add to the costs of keyboarding (for data capture), correcting (after optical scanning) and software development.

The corpus can be viewed as as an information source—constituting a "database" of information about the language variety that it is intended to represent. Computational processing cannot *add* any new information to this source; it can merely restructure and re-present the information that is inherent in the encoding of the text. A human editor could either add new information to this source, or perhaps make the information units and their relationship more explicit and thereby facilitate the automatic retrieval of information by the corpus user. The former task (adding new information) simply cannot be done by computer. The latter, however, could be achieved automatically in theory. In practice some restructuring to make information more explicit can be easily achieved by program, whereas other units and relationships will be almost intractable given the curent state of NLP. For example, it will not be too difficult to develop a program which could delimit and mark sentence units in a text. On the other hand, it is very difficult to write a program which will distinguish the use of italics in a text for emphasis and the use of italics to signal technical words and phrases.[2]

It is not the business of corpus builders to *create* this information source; the task is to *convert* it from one format to another. Several years ago texts were rarely created on computer and in any event were usually disseminated on paper. This pattern is changing rapidly with the spread of word-processing and DTP, and it now seems (though I have little more evidence than personal observation) that a substantial proportion of texts are created on a computer, though the bottleneck created by restricted access to networking has prevented a commensurate growth in electronic transmission of texts. Currently, and for the foreseeable future, the compiler of a large general language corpus will acquire written material either by conversion from paper sources or directly from other computer systems.

[2]The text might possibly contain a glossary which could be parsed and used as a key to the technical terms which are italicised. In any case, considerable effort will be required.

3.2 Converting from Print

In the case of sources which need to be converted from printed form, markup will have to be introduced. This could be carried out during the processs of OCR scanning, during keyboarding, or as a post-editing operation once the plain text has been captured. The introduction of markup cannot be clearly dissociated from the basic data capture, since decisions concerning the encoding of certain printed symbols (diacritic marks, em dashes, international currency symbols, etc.) will be necessary at the first stage of scanning or keying. There is a cost penalty in introducing markup at the keyboarding or scanning stage, in that data processing agencies usually establish their charging rates according to the complexity of the task and the volume as measured in keystrokes. The additional bytes required for marking up a TEI conformant text could certainly double the cost of capturing the text in a more compact, implicit form. A plain copy-typing of a printed source will contain a certain amount of implicit signalling of text units and their relationship, and this can be made explicit by subsequent processing by software filters. For example, paragraph breaks are likely to be signalled either by the introduction of a blank line, or by indentation of the first line of the paragraph. Sections may be numbered, perhaps with a centred heading, and these features may be identified by program and re-coded in SGML form. Exploiting these implicit indicators of text structure will keep costs down, since keyboarding and scanning can be carried out more quickly and demand less training and skill if the special encoding instructions that are given to staff are kept to a minimum and are no more than common sense would suggest. One problem with this approach is that while some features may be consistently and accurately identified and encoded by this method, there will be some erroneous markup caused by ambiguities in the format of the source and some features which cannot be marked up because the information simply is not present. In order to minimise the initial data capture costs, one might devise a compressed encoding convention which can be easily followed by staff who will receive little training and who have no specialist linguistic expertise. This compressed encoding can be elaborated and expanded by software which can be written as a one-off task.

3.3 Converting Computer Files

In the case of data which comes from other computer systems, there is often some explicit descriptive encoding of text structure, often arbitrarily intermingled with representational codes. For example, many WP packages allow the user to create footnotes by means of a descriptive style of markup, such that the actual physical placing of the footnote is handled by the formatting phase of the WP software and the text of the footnote itself is placed in an explicitly marked block with a coded reference to the place in the text that the note is to be attached. On the other hand, italic and boldface are usually selected by the creator of the document with some sort of embedded codes bracketing the text to be highlighted. Since italics can be used for a wide range of different structural purposes in the document, this representational encoding requires some detailed programming or human editing if the underlying structural function is to be retrieved and marked up.

3.4 Problems with Automatic Encoding

In both cases there is a potentially large hidden cost involved in introducing, converting and standardising markup by program, even though this appears at first to be significantly more cost effective than tedious and costly human editing. For a large corpus, many hundreds of

text sources may be captured across a wide spectrum of text types. The programming work required to normalise the encoding and markup must often be repeated for each new text, because there is very little standardisation in printing format or in WP packages. Even if two documents are received which were both produced from the same WP package, the actual layout will be the idiosyncratic product of the author or publisher, so that some low-level encoding will be fixed for that WP program but higher level units will have to be identified ad hoc.

4. A Guide to Basic Markup for a General Purpose Corpus

4.1 Features of Written Text

In TEI-P1 some features of text structure are considered under one section entitled "Features Common to Many Text Types". A later section deals with a few examples of specific text types. When adding a new text to a corpus under construction, one has to make the decision for each individual text as to whether it is an instance of a more general text type, or whether it deserves separate treatment with different markup. The corpus design will have established the range of text types that are to be sampled. These categories may be very broad or loosely defined—narrative fiction, newspapers, business correspondence—but a decision to make use of SGML markup will require the specification of a set of *document type definitions* (DTDs) which formally define the structure which is to be marked up for each document type. Since the corpus will be made up substantially of texts which already exist and which have to be converted to bring them into conformance with the markup standards for the corpus as a whole, there will be a tendency to define a large number of DTDs to deal with the variety of actual instances.

The advantage of creating more DTDs is that the markup will better reflect the structure and content of each text sample. For example, a news story from one newspaper received on disk includes captions relating to photographs placed at the end of the text of the story. It also contains short phrases or quotations (which would be placed typographically as eye-catching "titbits" on the page) embedded at points in the main text: let us call these "catchlines". A news story from another newspaper seems to have both catchlines and photograph captions sprinkled through the text of the story, but they are not differentiated. Two DTDs could be specified so that news stories from the first source have `<photo.capt>` and `<catchline>` units, while the second has only some more general tag, say, `<caption>`.

The disadvantage in creating more DTDs is that the marking up of a large number of different texts from many different sources becomes increasingly complex, and the generalising power of the markup is diminished. In the case of the news stories, it might be convenient to be able to produce a word frequency list from the corpus, omitting captions. Applications software would have to recognise three different tag labels in order to achieve this result. A more general DTD for the broad range of formats found in newspapers would allow captions to be simply identified. The SGML markup may, in this example, define in one DTD an attribute of the general `<caption>` unit which specifies `type=photo|catch|empty` which would preserve the more detailed information in the first source while generalising for both. The increased complexity of the markup may not be justified, however, if most of the processing on the corpus is to be done over large aggregates of text rather than individual samples. If several newspapers are captured which do not signal captions for photos and catchlines separately, then there is no point in attempting to omit catchlines (but include photo captions) in a word frequency listing. In such cases, manual checking

against the printed page will be required to add the `type` attribute for news stories which do not already distinguish, and this could be a time-consuming and expensive operation.

A practical approach is to define only such DTDs as are necessary to capture general features that will be of demonstrable value to those who will access the corpus. There is no need for the document type definitions to correspond to the text type categories that may have been identified for the design and sampling of the corpus, even though such categorisation may correspond fairly closely to the observed characteristics of different document formats. A corpus may sample from two categories, for example "novels" and "learned monographs", but there may be just one DTD for "Book" which will make provision for the descriptive markup of both.

4.1.1 Non-ascii Characters

The encoding of non-ascii characters is something which will be required at the earliest stages of data capture. In many cases texts which are received already in machine-readable form will include special codes for graphic shapes which are not specified as part of the ascii set. The number of such characters which may need encoding could be quite large and would include mathematical symbols, non-roman alphabets, bullets, arrows, the copyright symbol, diacritic marks, Old English ash, eth and thorn, fractions, and several other rare and recondite symbols.

These must be encoded at the time of data capture and for the sake of economy, might be recorded using any local convention which ensures that the codes are unambiguous yet easily keyed and checked. They can be expanded into standard SGML entity references by a search-and-replace operation carried out later.

Often it is possible for texts to be keyed on PCs which have an extended character set (often based on IBM's national language support codes) interpreted consistently by word-processors and other applications software. The commonly occurring accented characters (e-acute, e-grave, a-acute, a-grave, c-cedilla, u-umlaut, a-umlaut, etc.) are included, but many of the more esoteric symbols are missing. For a keyboarder, the use of these additional characters is convenient and intuitively appealing (the text on the screen is "clean" and readable) and popular and familiar word processing packages can be used to capture text. Caution is needed in transferring the captured text out of the WP package, and onto another system for post-processing, since it is possible for the special eight-bit characters to be truncated to seven bits with a consequent loss of information (and the introduction of spurious control-characters into the text).

4.1.2 Quotation

This is an important aspect of text encoding in a corpus. There are three types of quotation that need to be considered:

- direct speech
- block quotes
- other uses of quotation marks

Direct speech is probably the most difficult of these to deal with satisfactorily. For more advanced processing of a corpus, we might expect part-of-speech tagging and syntactic parsing to be carried out. Direct speech conventions in prose writing allow the *subsiduary* discourse (the quoted text) to be interwoven with the *primary* discourse (the quoting text). For example:

> 'In reality,' Surkov interposed, 'it was the Pope.'
>
> She looked up sharply. 'You're pulling my leg.'
>
> 'No, there's this beautiful Leningrad actress who went to Rome for several months. There's reasonable evidence that she was trying to "turn" the Pope.'
>
> 'I didn't dare to be truthful,' I said, nodding agreement. 'It weakened it; everyone knows cardinals have mistresses.... '

The first quoted sentence is interrupted by the insertion of "Surkov interposed" though the quoted utterance is a sentence unit. The last utterance is also split by "I said, nodding agreement", but the text before the interpolation is a complete sentence unit as is the text following. In order to tag or parse this fragment accurately, it would be necessary to distinguish the two levels of discourse, each with its own syntactic structure. This could be achieved if the subsiduary discourse were marked up so as to be clearly separable from the primary, in such a way as to preserve the syntactic integrity of both.

It will not be feasible for the Oxford Corpus to introduce a fully elaborated descriptive markup in these cases, and the best that can be expected is that opening and closing quotation marks can be interpreted and converted so that any string can be identified as belonging either to the primary or the subsiduary discourse. In the example above, it should be possible to search for the idiomatic phrase "you're pulling my leg" excluding instances where it occurs in direct speech. The distinction between primary and subsiduary discourse is likely to be very significant for narrative fiction, in which a substantial proportion of the length of the sample may be made up of simulated speech—similar to drama texts, for example, with respect to the mode of composition but quite dissimilar to authentic spontaneous speech. This embedding of one discourse within another is one which cannot be simply resolved by elaborate encoding of direct speech. Reported speech, stream-of-consciousness, and free indirect speech are just some of the stylistic techniques which are often found in prose fiction and which blur the distinction between the two levels of discourse.

Block quotes are much more easily handled. Either in print or in machine readable form, most texts signal the start and end of block quotations (with indentation, typeface change, opening and closing quote marks, etc.). Such representational features need not be recorded, as long as the extent of the quotation is indicated. Block quotes may conclude with a note of the source of the quotation. This should be tagged, ideally nested within the quote unit so that the source information can be related directly to the text of the quotation.

Other uses of quotation marks, to signal ironic use, cited words, titles of books and films, etc. do not require descriptive markup in a general language corpus. Indeed, it may be very difficult for the reader of the text to determine precisely what the significance of the quotation marks might be. In the sample stretch of text above, the use of quotation marks around the word *turn* in the last sentence could be a quotation from the utterance of some other character in the novel, or just a vaguely jocular use of the word to refer to the "turning" of espionage agents. We simply cannot tell for sure, and there is little point in attempting to do more than markup the appearance of the punctuation marks.

4.1.3 Lists

Lists appear in many types of written text. If they are not marked up in any special way they give rise to a number of undesirable side-effects in text analysis. First, the *item labels* of a list may be roman or arabic numerals, letters or other printer's mark, and these can

be misinterpreted by text searching software as "real" words. The overall frequency of the words *a* and *I* are likely to be skewed to some extent if item labels are not specially dealt with. Second, lists are often not punctuated according to the normal conventions with respect to sentences. This is likely to confuse tagging and parsing software.

For a text corpus, there is no need to make explicit the numerical sequence of list items. It is sufficient simply to preserve the realisation of the item label and to mark it as such, so that processing software can be programmed not to treat item labels as if they were words of the text.

The boundaries of a list are often signalled in print and in machine readable texts. If there is no unambiguous clue in the source text as to the start and end of a list structure, then it will nevertheless be helpful to match conventional patterns and convert these to tags. The examples below show some of the typical formats for list labels.

```
a.  This is the first list item...

a)  This is the first list item...

i.  this is...

(a) this is...

1.  Numbers are not too difficult...

*   unordered list items using special symbols

i   This is more tricky....

ii  But this is easier...
```

The text of each item may or may not have an initial capital letter and since the punctuation at the end of each list item can be quite idiosyncratic, there is some benefit in identifying the text of the list item as a marked-up unit.

4.1.4 Headings

Most written texts contain headings of some sort. The problem is whether to mark up headings in a representational or a descriptive way. Headings can usually be interpreted as labels attaching to some structural unit: chapter, section, article, and so on. Or they could be treated as short interruptions in the main flow of the text. Unfortunately, real-life texts are much less tidy than our idealisations of them, and often texts are found in which apparent headings do not seem to be functioning as labels to any structural unit. Newspapers and magazines are particularly inconsistent in this respect. The canonical text considered as basically a stream of words organised into sentences, paragraphs and then higher units which are given headings turns out to be a weak model for much modern magazine production. Newspaper and magazine stories often have what appears to be a lengthy sub-heading which may be just the first paragraph of text set in boldface or in a slightly larger typesize. The use of "catchlines" is discussed above: these also appear at first to be something like headings which accompany sections of text. However, the discourse organisation of the article will often seem unrelated to the placing of the catchline, leading to

the conclusion that these are not headings at all but some extraneous visual keys, serving a function rather similar to printers' ornamentations, fingers and arrows placed in the margin, rules, and the like.

Despite the difficulties created by magazines and newspapers, it is useful to identify headings and mark them up. In the case of a straightforward linear text like a report or textbok, the larger structural units can usually be identified by program and the heading enclosed within tags. If it is not possible to identify the larger sectioning units automatically, then manual editing in this instance will not require much effort. The human editor can quickly and easily search for the chapter and section boundaries and add markup to delimit these units and their associated headings. If the interrelation of structural units and headings is not straightforward, then a simplified markup can be applied, such that headings are encoded as bracketed fragments of text and tagged as headings. It may be possible automatically to identify more than one level of heading: typesize and style indicators may correspond to major section and subsection headings.

For several reasons it is necessary to attempt to identify and mark up headings. Headings which are not distinguished in any way from the main body of running text are difficult to deal with in tagging and parsing. They are often composed without regard to the normal syntax of English, they may include words in block capitals or with initial capital letters, and they are often not punctuated in accordance with standard conventions.

4.1.5 Abbreviations, Initials and Acronyms

Past experience of processing large general English text corpora indicates that a surprisingly high proportion of the word tokens of a corpus will be accounted for by abbreviations, initials and acronyms. Personal names, organisations, titles of address, postcodes, units of measurement, days of the week, month names, chemical elements, conventional Latin-derived abbreviations: these are found in abundance in almost every type of written source. Of these a substantial proportion could be identified automatically by pattern matching and tagged as contracted forms. Acronyms will be most difficult to handle, as these are, often deliberately, written and used as if they were "normal" words. At any given time, English contains some acronyms which are usually written in uppercase (*NATO*), some which are in transition (*ASCII*, *ascii*) and some which are fully naturalised as ordinary words (*radar*, *yuppie*). The omission of full-stops after each letter in abbreviations (as *VDU*, *VIP*, *UK*, etc.) leaves only the fact that these appear in uppercase to signal their status as abbreviations and, since ordinary words appear in block capitals for other reasons, these abbreviations would have to be manually marked up.

The advantages of marking up these contracted forms are:

- full-stops used in abbreviations can be distinguished from sentence-terminating full-stops;
- forms which happen to be homographs of ordinary words (e.g. AM = ante meridiem, IT = information technology) will be treated separately.

The widespread and increasing use of acronyms in contemporary texts, however, makes it inevitable that the distinction between an acronym and an "ordinary" word cannot be clearly drawn. Automatic identification and tagging of a large percentage of abbreviations in a corpus may be supplemented with manual editing with the aim of eliminating most of the overlap between words and abbreviations.

It is preferable not to standardise the spacing and use of full-stops in abbreviations since many users of a corpus are likely to be interested to discover patterns of usage relating to these features, particularly for lexicography.

4.1.6 Front and Back Matter

Books will usually include a certain amount of front matter (e.g. preface, foreword, contents, list of figures, acknowledgements) and back matter (e.g. index, appendices, bibliography). Some of these may be captured and included in the corpus, while others may be omitted. Typically, the front and back matter which is discursive is likely to be of some value for linguistic study, whereas tables, lists and figures are not. It is very straightforward to mark the start and end of each unit of front and back matter or to add a note of the form suggested in TEI-P1:

```
<note source=ed>List Of Illustrations omitted</note>
```

TEI-P1 provides very acceptable encoding suggestions for front and back matter. If, for example, the bibliography section of a chapter or book is to be captured within the corpus, then TEI-P1 suggestions for markup can be followed.

4.1.7 Chapters and Sections

Chapters or sections can be easily encoded with little extra effort and keying. In accordance with TEI-P1 the `<div`*n*`>` tag should be used for nested structural divisions in the body of the text. In a novel which has only one level of structural division, chapters, the markup would be as follows:

```
<div0 n=1 type=chapter>
<head>Some Chapter Title</head>
This is the first line of Chapter 1 ...
...
and so to the last line of this chapter.
</div0>
```

4.1.8 Correspondence and Addresses

Addresses occur quite frequently in most written material, except for monographs and novels. They are composed primarily of proper names, numbers and codes and consequently they do not yield much in the way of valuable linguistic information. They should not be omitted if they participate in some way in the discourse structure of the containing text, fulfilling a syntactic role as a noun group, for example, as in:

```
Enquiries concerning payments should be addressed to Telecom House,
Newhall Street, Toytown, TT12 3AA.
```

Where addresses are not integral to the discourse structure, as for example when lists of contact addresses are given at the end of a chapter of a book, then they may be omitted altogether. For certain documents (business or personal correspondence, for example) it may be necessary to omit addresses in order to preserve the anonymity of the correspondents to some extent. The omission will not seriously impair the usefulness of the text corpus, though there is a small set of words such as *way, crescent, walk*, etc. which have a sense

roughly synonymous with *road* or *avenue* but which very rarely occur with this sense except in addresses.

Attempting to standardise the format of all addresses within the corpus through the markup is probably not worth the effort. What is important is that, say, the frequency of the word *road* in a corpus is sensitive to its use as part of an address. It may be useful for some processing of the corpus to treat addresses as "special" text which is to be ignored for the purpose of generating statistics about the language sample. To achieve this, it will be sufficient to delimit the start and end points of an address string with tags. Addresses are often formatted using a conventional layout and it may be possible to identify and tag addresses automatically, particularly when they occur in the front matter and end matter of formal correspondence.

TEI-P1 contains specific recommendations relating to office documentation and proposes detailed coding for the front matter of letters, memoranda, minutes, etc. which is peculiar to this type of document.

4.2 Transcription of Speech

4.2.1 Treating Speech as Text

The TEI has not yet published any guidelines relating to SGML markup for speech in transcription. One reason for this may be that gathering and transcribing authentic speech is quite a different operation from handling written documents and is clearly a much more specialist area of activity, for linguists, lexicographers and speech technologists. The encoding and interchange of written documents will have much wider relevance and impact than the collection of speech corpora.

There are widely differing expectations among corpus linguists and speech and natural language researchers as to what is meant by a corpus of speech. For some purposes a speech corpus might mean 100 specially composed sentences, spoken and recorded in laboratory conditions by five different speakers, transcribed into an elaborate encoding of the intonation contours, pitch, volume, and other technical aspects of speech in performance. For the study of lexis, grammar, semantics or pragmatics, the spoken language included in a corpus can be collected with far less stringent constraints. The Oxford Corpus and the planned British National Corpus are to include a significant proportion of transcribed spoken language—millions of words rather than thousands—which will have to be gathered from the widest range of available sources and transcribed (if it is not already) using a simple system which can be applied by staff who are not linguistic specialists and which will not significantly lengthen the time required for transcription.

Speech recordings should be acquired by whatever methods are available, and rejected only in cases where the recording quality is so poor as to make transcription difficult. Many media organisations or research establishments will be able to supply paper or machine-readable transcriptions, which can be processed to bring them into conformance, as far as possible, with the basic markup for the corpus project. Transcriptions made by non-specialists will typically be in the form of quasi-written text. That is, there will be recognisable sentences and punctuation, with a high degree of normalisation of false starts, hesitation, non-verbal signals and other speech phenomena. This type of transcription converts spoken language into a form of idealised "script" (like a screenplay or drama script) which conforms to many of the establish conventions of written English. The advantages of transcribing in this way are:

- the cost and time of transcription are minimised
- the transcription is easily readable without any special training
- the transcription can be processed using established and widely available text processing software without substantial pre-editing

The following sections deal with specific aspects of speech transcription and markup which are likely to need attention. There is almost limitless scope for marking up an electronic encoding of speech: after all, there is no reason to use writing conventions like the alphabet—the conversion from audio signal to transcription is fundamentally different from the conversion of manuscript or print into an ascii text file. Unless the corpus is intended to serve the needs of speech specialists, then the usefulness of a "script" transcription is sufficient for a wide variety of linguistic studies.

4.2.2 Transcription into "script"

The majority of speech recordings can be represented in transcription as if they were the performance of a drama script. The following suggestions do not fit comfortably with the foregoing discussion of SGML markup of written texts, since in the case of audio recordings the concept of a basic text with its superimposed markup is not really appropriate. I propose that conventional punctuation (which in written texts is generally considered to be part of the "content" rather than "markup") be used to function rather like SGML markup. SGML can provide a formalism for encoding features which have no established conventional symbols.

The basic structure of speech transcription should be a sequence of speaker *turns*. Each turn should begin with an encoding identifying the speaker wherever possible. The identification of the speaker should be encoded during transcription using a minimal system of letter or digit identifiers. These identifiers can be elaborated elsewhere (perhaps in a header block for each spoken language unit) to supply information about each participant in the discourse. The use of a minimal encoding will reduce the amount of keyboarding required at the data capture stage. Sometimes it will be impossible for the transcriber to identify which speaker is speaking, and an appropriate code should be used in such cases.

Transcription should reflect the syntactic units of the language as far as possible. A pre-defined set of punctuation marks can be used in transcription, based on the conventions of the written language.

full-stop when the falling cadence and syntax suggests a written sentence termination;

comma for clause boundaries and lists, according to written conventions;

exclamation mark as conventionally used in writing, when the voice pitch and volume suggest its use;

question mark for question intonation, placed according to written convention;

double hyphen to indicate a re-start or fragmentary syntactic unit as e.g. `I was -- I didn't -- didn't really get cross.`

The use of punctuation cannot be precisely defined, since it is exactly to avoid the need for precise (and inevitably detailed and technical) definition that I am advocating the use of written punctuation conventions. Clearly, any transcription of an audio recording which

uses this quasi-prose method is an *interpretation* of the speech, and decisions concerning the placement of commas, full-stops and other marks may be challenged. The point is that since this is always true for all transcription using whatever encoding formalism, it is preferable, for the sake of reducing costs and rendering the collection of a large corpus feasible, to adopt an encoding which is familiar to most native English speakers. Experience has demonstrated that educated non-specialists, given no more than an hour's instruction, will transcribe audio recordings of a variety of speech situations in a way which yields a useful written record of the words that were uttered by the participants of the discourse. The resulting transcription will bear a similar relation to the recording as a playscript will to its performance on stage. Just as a dramatist can exploit the conventions of the English writing system to indicate that a character is to speak in a certain way, so the transcriber can use the same conventions to record in writing a certain way of speaking.

4.2.3 Spelling, Accent and Dialect

Orthographic irregularities should be avoided and clear specifications given for the use of enclitics such as em don't, *can't*, *he'd*. The enclitic forms found in writing are not a closed set, since it is acceptable in writing and print to use the apostrophe quite freely to represent in a stylised way the elisions and contractions of actual speech. *dunno*, *gonna*, *'orrible!*, *fish 'n' chips*, *whaddya mean?* are forms that are sometimes used in writing but which would introduce an undesirable irregularity into the transcription of spoken text in a corpus. Since it is clear that these quasi-speech forms are highly conventionalised in writing and that the writing system has no systematic conventions for recording the sounds of actual speech, it is preferable to keep the use of these non-standard forms to a minimum. A closed set of permissible forms can be given to the transcribers for guidance, or else full forms are to be used throughout the transcription. There will be instances where the enclitic form cannot be expanded to a fully explicit form because of ambiguity (*He said he'd hit him. he'd* = he would/had) and enclitic forms can be used in these cases.

Unless the corpus is to serve as a basis for detailed study of regional varieties of English, non-standard spellings and other orthographic tricks which transcribers might be tempted to use to represent a marked regional accent, for example, should also be proscribed. However, non-standard dialect, manifest in the use of syntax and lexis should be preserved. Unfortunately the distinction between non-standard and standard lexis is not clear-cut. Some words used by Scottish speakers (e.g. *och*, *aye*, *ye*, *auld*) have a recognised orthography in Scottish English even though they have similar Standard English equivalents (*oh*, *yes*, *you*, *old*). Are these dialect words? Or are they simulated phonetic representations of accent? Some decision must be taken as to whether forms such as these are transcribed in their regional orthography, or standardised. The following examples show some of the devices which writers use to represent speech characteristics. All such non-standard forms should be converted if they are found in transcriptions for inclusion in the spoken component of a general language corpus.

SE England "working class": *Yeah 'e said I never gave 'em nuffink.* (Yes he said I never gave them nothing)

Irish: *Oi've nivver been wi' Paddy* (I have never been with Paddy)

Yorkshire: *'appen if I did gi' 'im a kick up t'backside* (Happen if I did give him a kick up the backside)

Liverpudlian: *a lorra lorra laffs* (lots of lots of laughs)

In the example above, the form *lorra* could either be treated as standard *lot of* or *lots of*. Neither seems to produce a fully acceptable standardised form. There are a number of such speech characteristics which do not have an immediately obvious written representation. If the corpus will contain a large amount of informal spoken language it will be necessary to prescribe a representation for these cases. The non-standard forms *lorra* and *nobbut* may perhaps remain in the corpus as acceptable word forms in their own right.

4.2.4 Interruption and Overlapping Speech

These are phenomena which will occur frequently in informal speech situations. In prepared radio and TV broadcasts and in structured and directed discussions, overlapping speech is much less common, though it will occur occasionally. The writing system does not have very well-established conventions for representing this feature of speech. Interruption is a feature which can without difficulty be represented in the linear stream of writing. It merely requires the insertion of a code or tag which indicates that the preceding turn is incomplete because of the intrusion of the following turn. Interruptions are sometimes more messy, however, and the interrupted speaker may choose to continue regardless of the rival turn: the result will be overlapping speech, which cannot be so easily encoded in linear written form. The reason for marking up such features, rather than normalising the data to the extent that the overlapping speech is recorded in some arbitrary linear sequence, is that syntax and lexis could be significantly sensitive to the cut and thrust of turn-taking. A speaker may become incoherent for some seconds while an interruption is in progress, and unless the interruption is recorded as such, this valuable information concerning the possible cause of the incoherence will not be available to the analyst.

A simple markup would be as follows:

```
<s id=A> So I went over to him
<overlap>
  <os id=A> and I said -- I said "What do you think
  <os id=B> Great! yes!
  <os id=C> You didn't! Oh no.
</overlap>
you're doing?" and just looked, you know.
```

This method is straightforward to encode and quite readable in plain text form. It encodes only segments of overlapping speech without specifying precisely where each segment begins and ends in relation to other speakers' turns. This is likely to be adequate for most users of the corpus. A more detailed markup would encode for each overlapping segment of speech

- the start and end points
- an identifier
- the speaker
- the point at which this segment begins overlapping with another
- the point at which this segment ends overlapping with another
- the continuity of each speaker's turn

The resulting markup becomes dauntingly complex.

```
<s sp=A> So I went over to him <ov.seg id=2> and I said <ov.seg id=3> I said
</ov.seg id=2>"What do you think </ov.seg id=3> you're doing?" and
just looked, you know.
<overlap>
  <ov.text id=2,sp=B> Great! <ov.seg id=3> yes! </ov.seg id=3></ov.text>
  <ov.text id=3,sp=C> You didn't! Oh no. </ov.text>
</overlap> </s>
```

In this example, speaker A holds the turn and continues. The `<s>` tag delimits each turn. If one of the overlapping segments continues to become the next turn, then further complexity is introduced, since the overlap cannot now be treated as wholly contained within A's turn.

```
<s sp=A> So I went over to him <ov.seg id=2> and I said <ov.turn sp=C> I said
</ov.seg id=2>"What do you think...
<overlap>
  <ov.text id=2,sp=B> Great! <ov.turn sp=C> yes! </ov.text>
</overlap> </s>
<s sp=C> That's -- that's just what happened to me, yes, I had the
same thing
```

Just as A's turn ends with an overlap, C's turn begins with an overlap. The third overlapping segment is now not a subsiduary feature of A's turn, but must be attached to the remainder of the following turn, and the tag `<ov.turn>` indicates at which point in A and B's speech the overlapping turn begins.

These examples do not cover the full range of possible phenomena. The markup could become very dense and specialist training and skill could be required to carry out transcription if precise details of overlapping speech are to be recorded faithfully. A level of markup similar to my last example above would require substantially more time and effort than the first example of a simplified encoding, and the cost of staff training, quality control and the additional time required for analysis and keying should be carefully estimated before final decisions are made.

4.2.5 Pauses

The encoding of pauses should recognise two features: voicing and duration. Pauses may be voiced or silent, long or short. It is very simple for a transcriber to record the voiced pauses and they can be encoded in several ways. SGML entity references might serve this purpose quite well. Silent pauses are more problematic, because a period of silence on a recording cannot simply be assumed to be a silent pause. A transcriber who has only an audio recording of the speech event cannot be sure what other activities or interference might be the cause of a silence on the tape. Unless the recordings are analysed in detail in relation to the physical action of the speech event, the encoding of silent pauses is likely to be misleading and unhelpful. Voiced pauses can be encoded using two codes; one for a short *um* and another for a long *uumm*. These codes should stand for a wide range of actual speech sounds: *um*, *er*, *ah*, *mmm* and so on. The definition of long and short will probably have to be loose and founded on the intuitive estimate of the transcriber.

4.2.6 Other Functional Speech Sounds

Some speech sounds have a clear discourse function. The recognised functions can be simplified to a small set:

- acknowledgement (e.g. *mmm*)
- affirmation (e.g. *uh.hu* rising intonation)
- rejection (e.g. *uh.uh* falling intonation)
- prompt (e.g. *eh* rising intonation)

Each function can be represented as a standard code. This places responsibility on the transcriber to interpret speech sounds and assign them to appropriate functions, as with descriptive markup discussed in section 2 above. An alternative, presentational, approach would be to devise a large set of orthographic representations to cover a full range of speech sounds and encode the sound rather than its discourse function.

If the corpus is primarily for grammatical and lexical studies, then the precise recording of functional speech sounds will not greatly enhance its value, and it will be sufficient merely to allow the transcriber to improvise orthographic transcriptions for speech sounds and to delimit them with SGML codes so that whatever strings are keyed can be isolated and omitted from word frequency counts, indexes, etc.

4.2.6 Inaudible Segments

Often, especially if recordings are made in real-life situations, extraneous noise or poor recording conditions will make it impossible for the transcriber to hear exactly what is said. Lacunae should be marked with an indication of the extent of the inaudible segment (measured roughly in, say, seconds, syllables, or "beats" of speech rhythm). If the transcriber can record an approximation to what is being said, then this can be transcribed within suitable "query" markers (see the next subsection) either so that an attempt can be made later to pick up the exact words, or else merely to indicate to the corpus user that the exact words spoken at this point are in doubt.

4.2.7 Spelling

Some words may not be familiar to the transcriber and cannot be spelled with certainty. Proper names and technical terminology are likely to be especially difficult for a transcriber. An attempted spelling can be made at the time of transcription and a marker inserted to allow review and correction during post-editing, or at least to indicate to the corpus user that the spelling is doubtful. E.g.

```
It is a single-sided <?>skuzzy</?> drive with six hundred ...
```

4.2.8 Numbers

TEI-P1 recommends standardising the format of numbers in written texts by the use of the `<num>` tag, which has a `value` attribute set to the normalised value of a number expressed in digits. This level of markup for numbers will be a significant burden on the transcriber. The linguistic interest in numbers is likely to be limited to the study of the distribution of the variant spoken forms of numbers: *one oh two, a hundred and two, one hundred two, one hundred and two*, for example. For this purpose it would be useful for the transcription to

record the words spoken, rather than the conventional notation using arabic numerals. This, too, might add significantly to the number of keystrokes that are required in transcription, since telephone numbers, dates, prices and quantities are referred to very frequently in normal speech. The use of arabic numerals will simplify the keyboarding at the expense of the accurate representation of the words uttered. Ordinals, fractions and other numerical expressions as well as cardinal numbers should be dealt with consistently, takin either a representational or a descriptive approach.