Unit 3 Corpus markup

3.1 Introduction

Data collected using a sampling frame as discussed in unit 2 forms a raw corpus. Yet such data typically needs to be processed before use. For example, spoken data needs to be transcribed from audio recordings. Written texts may need to be rendered machine readable, if they are not already, by keyboarding or OCR (optical character recognition) scanning. Beyond this basic processing, however, lies another form of preparatory work – corpus markup.

Corpus markup is a system of standard codes inserted into a document stored in electronic form to provide information *about* the text itself and govern formatting, printing or other processing. This is an area which often causes confusion for neophytes in corpus linguistics. This unit first explains the rationale for corpus markup. Following this, widely used markup schemes such as TEI (the Text Encoding Initiative) and CES (the Corpus Encoding Standard) are introduced. Finally we will discuss a related issue, character encoding, which may be a particularly important issue when corpora including a range of writing systems are being constructed.

3.2 The rationale for corpus markup

Corpus markup is important for at least three reasons. First, as noted in unit 2, the corpus data basically consists of samples of used language. This means that these examples of linguistic usage are taken out of the context in which they originally occurred and their contextual information is lost. Burnard (2002) compares such out-of-context examples to a laboratory specimen and argues that contextual information (i.e. metadata, or 'data about data') is needed to restore the context and to enable us to relate the specimen to its original habitat. In corpus building, therefore, it is important to recover as much contextual information as practically possible to alleviate or compensate for such a loss (see unit 10.8 for further discussion). Second, while it is possible to group texts and/or transcripts of similar quality together and name these files consistently (e.g. as happens with the LOB and Brown corpora, see unit 7.4), filenames can provide only a tiny amount of extra-textual information (e.g. text types for written data and sociolinguistic variables of speakers for spoken data) and no textual information (paragraph/sentence boundaries and speech turns) at all. Yet such data is of great interest to linguists and thus should be encoded, separately from the corpus data *per se*, in a corpus (see unit 3.3). Markup adds value to a corpus and allows for a broader range of research questions to be addressed as a result. Finally, pre-processing written texts, and particularly transcribing spoken data, also involves markup. For example in written data, when graphics/tables are removed from the original texts, placeholders must be inserted to indicate the locations and types of omissions; quotations in foreign languages should also be marked up. In spoken data, pausing and para-linguistic features such as laughter need to be marked up. Corpus markup is also needed to insert editorial comments, which are sometimes necessary in pre-processing written texts and transcribing spoken data. What is done in corpus markup has a clear parallel in existing linguistic transcription practices. Markup is essential in corpus building.

3.3 Corpus markup schemes

Having established that markup is important in corpus construction, we can now move on to discuss markup schemes. It goes without saying that extra-textual and textual information should be kept separate from the corpus data (texts or transcripts) proper. Yet there are different schemes one may use to achieve this goal. One of the earliest markup schemes was COCOA. COCOA references consist of a set of attribute names and values enclosed in angled brackets, as in <A WILLIAM SHAKESPEAR>, where A (author) is the *attribute name* and WILLIAM SHAKESPEAR is the *attribute value*. COCOA references, however, only encode a limited set of features such as authors, titles and dates (cf. McEnery and Wilson 2001: 35). Recently, a number of more ambitious metadata markup schemes have been proposed, including for example, the Dublin Core Metadata Initiative (DCMI, see Dekkers and Weibel 2003), the Open Language Archives Community (OLAC, see Bird and Simons 2000), the ISLE Metadata Initiative (IMDI, see Wittenburg, Peters and Broeder 2002), the Text Encoding Initiative (TEI, see Sperberg-McQueen and Burnard 2002) and the Corpus Encoding Standard (CES, see Ide and Priest-Dorman 2000). DCMI provides 15 elements used primarily to describe authored web resources. OLAC is an extension of DCMI, which introduces refinements to narrow down the semantic scope of DCMI elements and adds an extra element to describe the language(s) covered by the resource. IMDI applies to multimedia corpora and lexical resources as well. From even this brief review it should be clear that there is currently no widely agreed standard way of representing metadata, though all of the current schemes do share many features and similarities. Possibly the most influential schemes in corpus building are TEI and CES, hence we will discuss both of these in some detail here.

The Text Encoding Initiative (TEI) was sponsored by three major academic associations concerned with humanities computing: the Association for Computational Linguistics (ACL), the Association for Literary and Linguistic Computing (ALLC), and the Association for Computers and the Humanities (ACH). The aim of the TEI guidelines is to facilitate data exchange by standardizing the markup or encoding of information stored in electronic form.

In TEI, each individual text (referred to as *document*) consists of two parts: header and body (i.e. the text itself), which are in turn composed of different *elements*. In a TEI header (tagged <teiHeader>), for example, there are four principal elements (see Burnard 2002):

- A *file description* (tagged <fileDesc>) containing a full bibliographic description of an electronic file;
- An *encoding description* (tagged <encodingDesc>), which describes the relationship between an electronic text and the source or sources from which it was derived;
- A *text profile* (tagged <profileDesc>), containing a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting;
- A *revision history* (tagged <revisionDesc>), which records the changes that have been made to a file.

Each element may contain embedded sub-elements at different levels. Of these, however, only <fileDesc> is required to be TEI-compliant; all of the others are optional. Hence, a TEI header can be very complex, or it can be very simple, depending upon the document and the degree of bibliographic control sought. Fig. 3.1 shows the corpus header of the British National Corpus (Word Edition) expanded to

the second level. The plus symbol (+) preceding an element indicates that the element can be expanded while the minus symbol (–) means that an element has already been expanded.

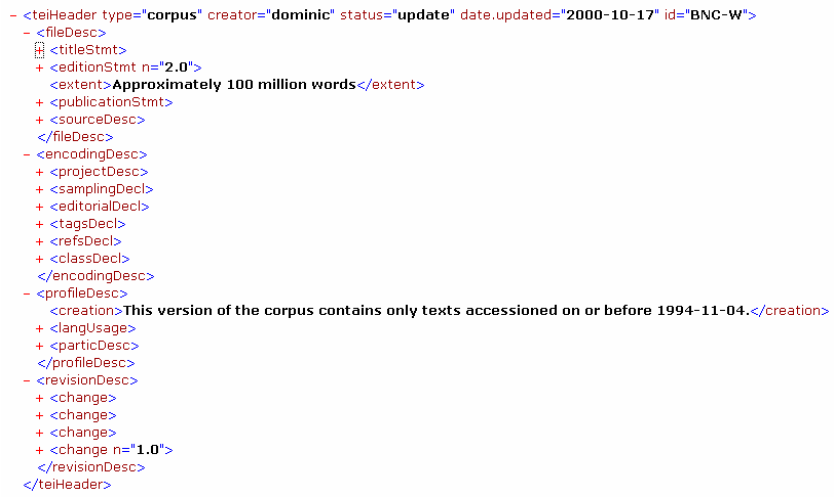

Fig. 3.1 The corpus header of the BNC World Edition

In the figure, a sequence of the form <XXX> is referred to as a start tag of an element while a corresponding sequence of the form </XXX> is an end tag. It is clear that these tags appear in pairs and can be nested within other elements. For example:

<extent> Approximately 100 million words </extent>

is embedded in the <editionStmt> element, which is in turn nested in <fileDesc>. Note that the start tag of an element may also contain an *attribute-value* pair, e.g. <editionStmt n = “2.0”>.

The body part of a TEI document is also conceived as being composed of elements. In this case, an element can be any unit of text, for example, chapter, paragraph, sentence or word. Formal markup in the body is by far rarer than in the header. It is primarily used to encode textual structures like paragraphs and sentences. Note that the TEI scheme applies to both the markup of metadata and the annotation of interpretative linguistic analysis (see unit 4). For example, the article *the* can be tagged thus:

<w POS=AT0>the</w>

This indicates that the part of speech (POS) of *the* is an article (AT0). In the BNC, the POS tag of *the* looks like this:

<w AT0>the

This is because end tags are omitted for the elements <s>, <w> and <c> (i.e. sentences, words and punctuation) in the BNC, the end of each being implied by the following <s>, <w> or <c>. In addition, attribute names (POS), together with the equal sign, are left out for the elements <w> and <c> to save space (cf. Aston and Burnard 1998: 33).

The TEI scheme can be expressed using a number of different formal languages. The first editions used the Standard Generalized Markup Language (SGML); the most

recent edition (i.e. TEI P4, 2002) can be expressed in the Extensible Markup Language (XML) (Sperberg-McQueen and Burnard 2002). SGML and XML are very similar, both defining a representation scheme for texts in electronic form which is device and system independent. SGML is a very powerful markup language, but associated with this power is complexity. XML is a simplified subset of SGML intended to make SGML easy enough for use on the Web. Hence while all XML documents are valid SGML documents, the reverse is not true. Nevertheless, there are some important surface differences between the two markup languages. End tags in SGML, as noted, can optionally be left out. They cannot in XML. An attribute name (i.e. generic identifier) in SGML may or may not be case sensitive. It is always case sensitive in XML; unless it contains spaces or digits, an attribute value in SGML may be given without double (or single) quotes. Quotes are mandatory in XML.

As the TEI guidelines are expressly designed to be applicable across a broad range of applications and disciplines, treating not only textual phenomena, they are designed for maximum generality and flexibility (cf. Ide 1998). As such, up to 450 elements are pre-defined in the TEI guidelines. While these elements make TEI very powerful and suitable for the general purpose encoding of electronic texts, they also add complexity to the scheme. In contrast, the Corpus Encoding Standard (CES) is designed specifically for the encoding of language corpora. CES is described as 'simplified' TEI in that it includes only the subset of the TEI tagset relevant to corpus-based work. It also simplifies the TEI specifications. Yet CES also extends the TEI guidelines by adding new elements not covered in TEI, specifying the precise values for some attributes, marking required/recommended/optional elements, and explicating detailed semantics for elements relevant to language engineering (e.g. sentence, word, etc.) (cf. Ide 1998).

CES covers three principal types of markup: 1) document-wide markup, which provides a bibliographic description of the document, encoding description, etc.; 2) gross structural markup, which encodes structural units of text (such as volume, chapter, etc.) down to the level of paragraph (but also including footnotes, titles, headings, tables, figures, etc.) and specifies normalization to recommended character sets and entities; 3) markup for sub-paragraph structures, including sentences, quotations, words abbreviations, names, dates, terms and cited words, etc. (see Ide 1998).

CES specifies a minimal encoding level that corpora must achieve to be considered standardized in terms of descriptive representation as well as general architecture. Three levels of text standardization are specified in CES: 1) the metalanguage level, 2) the syntactic level and 3) the semantic level. Standardization at the metalanguage level regulates the form of the syntactic rules and the basic mechanisms of markup schemes. Users can use a TEI-compliant *Document Type Definition* (DTD) to define tag names as well as 'document models' which specify the relations among tags. As texts may still have different document structures and markups even with the same metalanguage specifications, standardization at the syntactic level specifies precise tag names and syntactic rules for using the tags. It also provides constraints on content. However, even the same tag names can be interpreted differently by the data sender and receiver. For example, a <title> element may be intended by the data sender to indicate the name of a book while the data receiver is under no obligation to interpret it as such, because the element can also show a person's rank, honour and occupation, etc. This is why standardization at the semantic level is useful. In CES, the <h.title> element only refers to the name of a document. CES seeks to standardize at the semantic level for those elements most

relevant to language engineering applications, in particular, linguistic elements. The three levels of standardization are designed to achieve the goal of universal document interchange. Like the TEI scheme, CES not only applies to corpus markup, it also covers encoding conventions for the linguistic annotation of text and speech, currently including morpho-syntactic tagging (i.e. POS tagging, see unit 4.4.1) and parallel text alignment in parallel corpora (see unit 5.3).

CES was developed and recommended by the Expert Advisory Groups on Language Engineering Standards (EAGLES) as a TEI-compliant application of SGML that could serve as a widely accepted set of encoding standards for corpus-based work. CES is available in both SGML and XML versions. The XML version, referred to as XCES, has also developed support for additional types of annotation and resources, including discourse/dialogue, lexicons, and speech (Ide, Patrice and Laurent 2000).

3.4 Character encoding

Another issue related to corpus markup is character encoding. In the earlier SGML versions of TEI and CES, special characters in English (e.g. the pound sign £), accented characters in European languages (e.g. those marked with an acute in French, such as é) and other non-ASCII (American Standard Code for Information Interchange) characters were replaced by *entity references* which are delimited by & and ; (e.g. £ refers to £ and é stands for é). While declaring entity references for characters can solve the problem of representing special characters for languages which have only a small number of such characters, it is not a feasible solution for languages such as Chinese, where tens of thousands of entities would have to be declared to cover the writing system as the SGML versions of TEI and CES are primarily geared toward English and European languages.

There are many complementary standardized characters codes (e.g. the ISO-8859 family of 15 members) and competing native character sets (e.g. GB2312 and Big5 for Chinese) that corpus builders can use to avoid this problem. As noted elsewhere (McEnery and Xiao 2005a), while these legacy encodings are efficient in handling the language(s) they are designed for, they are inadequate for the purpose of electronic data interchange in a rapidly globalizing environment. Unicode is an attempt to solve this problem. Unicode is truly multilingual in that it can display characters from a very large number of writing systems and hence it holds out the promise of providing a standard for multilingual corpus character encoding.

Unicode has also been adopted by XML as the required character set for all XML documents. With the advent of XML/Unicode, most problems previously associated with character representation in corpus building will be greatly reduced. An XML editor enables you to input characters in most writing systems of the world directly and stores these characters in a way that is directly transferable between different computer systems, whether that be Unicode characters or as character entity references (Sperberg-McQueen and Burnard 2002). The combined use of Unicode and XML is a growing and useful trend in corpus development. As noted in the previous section, TEI and CES are both available in XML versions. XML support has made it possible for these markup schemes to be adopted more widely in a multilingual context.

3.5 Unit summary and looking ahead

Markup is an essential step in corpus building. An understanding of corpus markup is also of importance to corpus users. This unit has presented the rationale for corpus markup. It has also reviewed a number of markup schemes, focusing on the TEI guidelines and CES, and reviewed briefly the issue of character encoding in corpus construction. This basic overview should provide a sound basis on which to begin corpus building and use.

The discussions in this unit show that markup – especially information traditionally stored in a corpus header – often holds important information relating to context in spoken corpora and genres in written corpora. This information is crucial to the process of interpreting the data extracted from a corpus. The markup of the text in the body (e.g. paragraph and sentence markers) is also often very useful for linguists.

Markup schemes vary in complexity, as does the markup of individual corpora. This complexity depends on the level of detail required for particular research questions. This unit only provided a brief introduction to two major markup schemes, TEI and CES/XCES. Readers who wish to find further details of these schemes, or details of other schemes, should refer to their individual websites. The next unit will explore a related area of corpus construction – the annotation of interpretative linguistic analysis.