Data and language documentation

JEFF GOOD
University at Buffalo

1. INTRODUCTION

The topic of this chapter is the relationship between data and language documentation. Unlike many fields of study, concerns regarding data collection and manipulation play a central role in our understanding of, and theorizing about, language documentation. The field to a large extent, in fact, owes its existence to a shift in focus in the goals of linguistic field work from concerns regarding outputs derived from primary data, like grammars and dictionaries, to the collection of the primary data itself.

When trying to understand the role of data in language documentation, the first question we must consider is what precisely do we mean by *data*? Beginning with the work of Himmelmann (1998), it has become customary in language documentation to distinguish between *primary data*—constituting recordings, notes on recordings, and transcriptions—and *analytical resources*—like descriptive grammars and dictionaries—constructed on the basis of, and via generalization over, primary data. While making this conceptual distinction is essential to the practice and theorizing of language documentation, most individuals or teams working on language documentation projects are ultimately interested in both collecting primary data and producing the kinds of analytical resources associated with traditional language description, most prominently grammars, dictionaries, and texts (whether oriented for community or academic use). Therefore, each will be considered here. That is, the discussion will cover topics both regarding the collection, storage, and manipulation of primary data as well as the mobilization (see Holton (this volume)) of that data to create analytical resources. While it is also important to keep in mind that *data* is not synonymous with *digital data*, for the most part, in this chapter, only digital data will be discussed. Generally, digital, rather than analog, data has been the focus of work in language documentation both because new data is typically captured solely in digital form at present and because analog data is increasingly being digitized so that it can be manipulated and disseminated with digital tools. Discussion of important aspects of digitization—i.e., the process through which a digital representation of a non-digital object is created—can be found in the E-MELD School of Best Practices in Digital Language Documentation[2] (Boynton et al. (2006)), and an exemplary case study of the digitization process can be found in Simons et al. (2007).

This chapter will focus on conceptual issues rather than specific technical recommendations, though such recommendations may be discussed to provide illustrative examples. This is because our understanding of the conceptual issues evolves at a much slower rate than the technical recommendations, which change as the technologies we use for capturing and analyzing data themselves change and, therefore, largely outpace the speed through which works like this one make their way into publication. At least for the time being, the best way to find answers to questions like *What audio recording device should I use?* or *What*

[2] http://e-meld.org/school/

software should I use for text annotation? will be to use online resources like the E-MELD School just mentioned above, electronic publications like *Language Documentation and Conservation*[4] or the *Transient Languages and Cultures* blog[5], and email lists like the one run by the Resource Network for Linguistic Diversity[6]. The role of a chapter like this one is, therefore, not so much to tell language documenters what to do but, rather, to put issues surrounding data in a broader context, to allow them to understand why recommendations take a particular shape, and to better equip them to evaluate new technologies as they become available. Readers looking to augment the discussion here with more specific recommendations will find Austin (2006) helpful, as it covers similar subject matter to this chapter but on a more concrete level. More advanced conceptual discussion can be found in Bird and Simons (2003) which overlaps partially with the discussion here but also goes beyond it in many respects.

This chapter divides the discussion into the following topics: Data types in section 2, data structures in section 3, data formats in section 4, metadata in section 5, a brief discussion of needs assessment in section 6, and a concluding section on the linguist's responsibilities for navigating the relationship between their data and new technologies in section 7.

2. DATA TYPES

The discussion in this section is subdivided here into the topics of recordings, transcriptions, and traditional descriptive resources, each of which is treated in turn, followed by discussion of community-oriented versus academic-oriented data. I do not treat written language, as opposed to transcription, specifically here both because of the general emphasis in language documentation on collecting instances of spoken language (though, see Woodbury (this volume)) and because, from a data management perspective, written representations do not generally differ significantly from transcription. I also do not discuss scanned images, though these can play a role in language documentation, as well, particularly for projects making use of paper-based materials. (See Simons et al. (2007) for a relevant case study using scanned images to create high-quality documentary resources.)

2.1 Recordings

In the present context, following Himmelmann (1998:162), primary data will be used to refer two very distinct classes of resources. Direct recordings of events on the one hand, and written representations of those events on the other. Direct recordings include, most prominently, audio recordings and, increasingly, video recordings as well as photographs, though they can also include more "exotic" resources like laryngographs or palatograms. These kinds of resources are sometimes referred to as *raw data* (see, for example, Schultze-Berndt 2006:215), to highlight the fact that they can be created without extensive linguistic analysis, unlike transcriptions.

However, one should not be complacent and assume that the "rawness" of this data implies that it represents a purely objective rendering of a given

[4] http://nflrc.hawaii.edu/ldc/

[5] http://blogs.usyd.edu.au/elac/

[6] http://rnld.org/

communicative event. All recording involves selection: what to record, when to record, how to record, etc. And these selections, made by a person, not a machine, can shape the record tremendously, not only influencing the perceived quality of the recording but also emphasizing and deemphasizing features of the recorded event and the language in possibly significant ways. For example, use of a unidirectional microphone in making an audio recording will result in a resource where one speaker is framed as more central to a speech event than any others, while use of an omnidirectional microphone will produce a resource where different participants' voices are recorded more equally. Analytical linguistic factors may influence which kind of microphone is chosen for a given recording. In a grammatical elicitation session with a single speaker, for example, a unidirectional microphone is more likely be chosen, while for a recording made of a story an omnidirectional microphone may be used even though only one participant has the special role of storyteller if the story is being told in a society where audience participation is the norm. Similar issues arise in making the choice to make video recordings in addition to audio ones. For certain kinds of events—or even languages, in the case of sign languages—use of video may be essential, but the question of what visual aspects of a scene to capture is a particularly clear kind of selection.

Therefore, while the production of raw recordings involves less intensive linguistic analysis than creating, say, a transcription, it should not be forgotten that it involves a series of choices, some of which may be mostly pragmatic in nature (e.g., not to use a video recorder for a given session to conserve scarce battery power) while others (e.g., not to use a video recorder because a session is deemed to be visually "uninteresting") may actually be informed by an underlying—if only implicit—theory of recording. This point bears special importance for researchers choosing to adopt collaborative modes of fieldwork with their communities (see, e.g., Mithun 2001, Grinevald 2003, Dwyer 2006 for relevant discussion) or who intend their work to assist in community language maintenance and revitalization projects (see, e.g., Mosel 2006, Nathan 2006 and Hinton, Penfield, McCarty and Coronel-Molina (this volume)) since community input may be required to ensure that the form of the recordings is not unduly skewed towards research needs.

2.2 Transcriptions

Transcriptions (often annotated—see Schultze-Berndt (2006) for detailed discussion of annotation) have generally been treated under the heading of primary data due to the fact that they are intended to be a representation of a particular speech event rather than serving as generalizations over distinct speech events. Unlike recordings, however, the creation of transcriptions implies extensive linguistic analysis (see, e.g., Himmelmann 2006), and they, therefore, occupy a territory between documentation and description. (The same could be said for written representations of language in general, though in some cases written examples of language serve as primary data not merely by convention but because they constitute the only available representation of a given use of language.)

A crucial difference between transcriptions and recordings, however, is that recording techniques and technologies tend to be general in nature while transcription is a specifically linguistic task. The devices used by linguists to make audio recordings are more or less the same as those used by musicians, oral

historians, journalists, etc. However, many of the transcription conventions used by linguists, e.g., the International Phonetic Alphabet or aligned glossing, are domain-specific and largely under the control of the linguistic community.

An important consequence of this is that while language documenters will generally be reactive in the domain of recording techniques, they will often need to be proactive in the domain of transcription techniques. Thus, language documentation work is at the forefront of the next generation of transcription and annotation tools, as evidenced, for example, by the ELAN annotation tool[7] (see Berez (2007) for a review) produced specifically in the context of the Dokumentation Bedrohter Sprachen (DoBeS) Programme.

2.3 Descriptive resources

Three kinds of resources have long been given a special place in descriptive linguistics: texts, dictionaries, and grammars. If the most important feature distinguishing descriptive resources from documentary resources is the fact that they attempt to arrive at generalizations about a language based on raw data, it is clear that texts are less prototypically descriptive than dictionaries and grammars. However, to the extent that they are normalized and edited for internal consistency, they shift from being records of a specific speech event, as with a transcription, to being representations of an idealized speech event and, therefore, begin to cross the boundary into description.

By contrast, dictionaries and grammars are unambiguously instances of description. A dictionary is an attempt to generalize over the known lexical items of a language to create a concise summary of their uses and meanings, while a grammar generalizes over textual and elicited data to create a summary of the phonological, morphological, and syntactic constructions of a language. Formal work making use of extensive language data is not generally construed as an essential part of the creation of an adequate description of a language. However, in the present context, it could, in principle, also be included under the broad heading of language description as well. In practice, however, the field of linguistics tends to reserve the term for informal description rather than formal description. (See Dryer (2006) for discussion of relevant issues.)

2.4 Community data versus academic data

It has become standard practice for linguists documenting under-resourced languages to consider ways in which their work can result in outputs not only for use in academic spheres, but also community ones. Accordingly, brief discussion of this issue is in order here.

It is important to be clear that trying to serve multiple communities will always require more work than serving only one community. At the same time, modern technology can significantly reduce the extra burden placed on language documenters who opt to do this. This is because, digital data, unlike data on paper, can be copied and transformed relatively easily. To take an example outside of the domain of language documentation, it has now become commonplace for individuals to transform text documents from whatever format they were originally composed in (e.g., in the native format of their word processing program) to Portable Document Format (PDF), a format specifically designed to create documents which are readable across a wide range of computer platforms.

[7] http://www.lat-mpi.eu/tools/elan/

This transformation process has been largely automated requiring only a trivial investment of time on the part of the user.

The kinds of data transformations required to allow a single language resource to serve speaker and research communities, of course, will never be as straightforwardly automated as conversion to PDF if for no other reason than the fact that groups interested in such functionality do not have the economic power to attract the interest of large software companies. However, as will be discussed in the following sections, if the data collected by a project is encoded in certain ways, allowing it to serve multiple audiences becomes more manageable. Furthermore, if non-proprietary, open formats are used and the way the data is encoded is well-documented (see section 4), anyone with sufficient technical expertise will be able transform the original data into new formats, substantially increasing the potential impact of a project and perhaps also decreasing the workload of the language documenter who would not, then, be required to perform such data transformations themselves.[9]

3. DATA STRUCTURE VERSUS IMPLEMENTATION

Often, when people talk about their data, they conflate the abstract structure of the various datatypes they collect with the ways those datatypes happen to be encoded in a particular *view*—that is, a way of representing the data in a human-readable form. Thus, for example linguists often speak of *interlinear glossed text* as a basic data type when, in reality, it is probably better understood as a specific way of expressing a data type we might refer to *morphologically-analyzed text*—that is, a text on which an exhaustive morphological analysis has been performed. Interlinear glossing has become widely adopted as an effective way of presenting such a morphological analysis, in particular on the printed page, but it is just one of many imaginable ways of doing this. For example, in early twentieth century texts one sometimes finds a convention where individual words are associated with endnotes giving analytical details well beyond what is possible with a short gloss (see, for example, the texts in Boas (1911)). And, of course, using modern hypertext methods, interactive forms of glossing have become possible as well.

Linguists tend to think of interlinear glossed text as a basic data type in and of itself because it represents a primary way they interact with texts, and, this is, of course, a perfectly natural conflation. However, when it comes to encoding data on a computer, it is important not to let one particular view unduly influence the way the data itself is coded. Each view is optimized for a particular use and encoding some data too closely to one particular view on a computer will make it hard for it to be reused to create other views. Instead, one should attempt an analysis of the underlying logical structure of the data being collected, encode it using that logical structure, and then allow existing software tools to create views of the data of use to the various interested communities and individuals.

Section 4 will cover specific issues relating to the encoding of language data on a computer. In the remainder of this section notion of an underlying data structure will be explored in more detail (section 3.1) and general aspects of the

[9] We should clearly distinguish here between encoding data in non-proprietary, open formats which, in principle, allow it to be straightforwardly repurposed by outside parties and actually making the data available to them, for example by posting it on a website. Open access and open formats are distinct concepts, and neither implies the other.

problem of encoding that structure in machine-readable format will be introduced (section 3.2). For purposes of illustration, the discussion will focus on the structure of a simple entry in a wordlist.

3.1 Underlying data structures

In trying to determine what the basic underlying structure is for a given kind of data, the first point one must keep in mind is that this is a complex analytical task and developing a universal mechanistic algorithm to determine the underlying structure of language data is no easier than, say, developing such an algorithm for discovering the phoneme inventory of a language based on phonetic transcriptions. Each kind of data from each language will present its own conceptual difficulties, though just as with grammatical analysis, these will often be variations on a theme rather than completely unexplored problems.

To make the discussion more concrete, consider the very simple lexical entry in (1), associating a French word with a part of speech and an English translation. (See Austin 2006:97–98 for comparable discussion of the structure of a lexical entry.)

(1) *chat* **n.** cat

The example in (1) gives a particular view of a bilingual lexical entry consisting of a headword from the language being described in italics, an indication of its part of speech in bold, and a basic translation in plain text. The underlying structure of the data is largely implicit, though the view does at least imply that the data can be analyzed into three core pieces. We can give a first approximation of the underlying structure of the data in (1) as in (2), where the typological conventions of (1) are repeated in the interests of clarity.

(2) *headword* **pos** gloss

While (2), at first, may seem to be a reasonable representation of the logical structure of (1), it, in fact, still leaves many characteristics of the data itself implicit. This is because it only analyzes those features of the data explicitly represented in the view seen in (1), leaving out many important other features, which, while easily reconstructible from context by a human, will be unknown to a computer without explicit coding. Perhaps the most important of these implicit features is the most easily overlooked: the three logical pieces in (2) are part of a larger unit we might refer to as an *entry*, and represent as in (3).

(3) $[[\textit{headword}]\ [\textbf{pos}]\ [\text{gloss}]]_{\text{ENTRY}}$

There is at least one set of important additional characteristics associated with the entry in (1) not yet described by the analysis in (3)—that each of the parts of the entry is associated with a particular language. The headword is in French, the part of speech label is an abbreviation from English (though an abbreviation like *n* is, of course, potentially ambiguous as to what language it is drawn from), and the gloss is in English. We might, therefore, want to expand our analysis of the underlying structure of the word list entry in (1) as in (4).

(4) $[[\textit{headword}]_{\text{lang:french}}\ [\textbf{pos}]_{\text{lang:english}}\ [\text{gloss}]_{\text{lang:english}}]_{\text{ENTRY}}$

While (4) is significantly more complex than (2), it is still just a beginning. Nowhere is it explicitly indicated yet, for example, that that part of speech label applies to the headword and not to the gloss. Nor is there indication of the nature of the representation of the headword—that is, we do not know (without using outside knowledge) whether the sequence *chat* is a phonetic, phonemic, or orthographic representation.

Should we further refine the analysis given in (4), then? How one answers this depends on the details of the data being collected as well as what the data will be used for. For example, if one was working with a dataset wherein some of the headwords were given in an orthographic representation while others were given in phonetic transcription, then it would be important to include the possibility for specifying the nature of the headword's representation in an analysis of the entry's underlying structure. However, if all the headwords used an orthographic representation, this would be relatively less important.

3.2 Implementing a data structure

Analyzing some data in order to arrive at an understanding of its underlying structure could, in principle, be a purely theoretical enterprise. However, in language documentation, it is mostly a means to an end: What one wants to be able to do is store data on a computer in a form which will facilitate its being used to produce human-usable language resources. Therefore, there will generally be a point when some analysis of this structure, even one that may be known to be imperfect, must be chosen for *implementation* on a computer—that is, a method must be devised for it to be expressed in a machine-readable form which can be straightforwardly manipulated by the user.

Deciding on an implementation for a given data structure, ultimately, is largely dependent on practical considerations relating to the intended uses for the data and the range of data manipulation tools available to the language documenter. Nevertheless, it is still essential to devote some time to abstract data modeling of the sort described in section 3.1. Simply put, the better one understands the underlying structure of one's data, the easier it will be to arrive at an implementation which will be sustainable over the lifespan a given project.

An implementation of a data structure by definition will need to be done using some computational tool. From the present perspective, one of the most crucial factors in choosing a tool is that it will be able to straightforwardly create a reasonable implementation of the underlying data structure one chooses to work with. In that sense, one of the most ubiquitous kinds of application, the word processor, is usually insufficient since word processors are optimized to work with a kind of data—unannotated text documents—that plays a relatively minor role in language documentation. Thus, while one may be able to create reasonable presentations of data (see section 4.3), like what is seen in (1) using a word processor, the resulting resource will not actually code the structure of the data but, rather, aspects of formatting (e.g., bold and italics) that are only indirectly related to the structure. Another common office application, spreadsheet software, by contrast, can be used profitably to implement data structures which are well expressed in a table. The crucial issue here is not the fact that each of these products was designed for use in an office environment. Rather, it is that one kind of application (spreadsheet software) builds a basic kind of data structure (the table) directly into its design.

Software specifically designed for language documentation will be optimized to work with a particular linguistic data type (or set of data types)—e.g., time-aligned annotated texts in the case of Elan. But, such software will not be available for every kind of data and, depending on the needs of a project, may not always be the ideal choice, particularly when a documentary team consists of not only linguists but also non-linguists, who might not be familiar with the ways that linguists think about their data which inform the design of the linguistics-specific tools.

Returning to the example of a lexical entry discussed in section 3.1, how might we implement the data structure associated with it? In this case, the structure is relatively simple, and we could straightforwardly implement it in a spreadsheet where each row corresponds to an entry, and where each part of the entry occupies a single cell of the row, along the lines of what is depicted in table 1. (See section 4.2 for an alternative way of encoding the data.)

Table 1
Tabular implementation of word list entries

headword	**part of speech**	**gloss**
chat	n.	cat
chien	n.	dog

The implementation in table 1 does not contain all the information found in the underlying data analysis presented in section 3.1. For example, there is no specific indication that the headword is French and the glossing language is English. Some of the structure is explicitly indicated, however, in the header line which labels the uses of each column. In this case, the missing language information does not pose particular problems since it could be straightforwardly rectified with accompanying information documenting the nature of the data in the file, which could be as easy as giving the spreadsheet a title like "French wordlist with English glosses". In this case, we are dealing with data that has a relatively simple structure and which, therefore, can be a given a fairly simple implementation using a widely available kind of software.

Of course, this is just an illustrative example. In many—perhaps most—cases the data collected while documenting a language will be more complex than the example given in (1). Bell and Bird's (2000) survey, for example, of the structure of lexical entries across a wide range of published work gives a good indication of the level of complexity involved when one looks at real lexical data. A full dictionary entry—as opposed to word list entry—which might contain multiple senses of a given word, example sentences for each sense, and comparative notes, among other things, will require a tool allowing the definition of data structures with hierarchical relationships within an entry, for example linguistics-specific database software like SIL International's Shoebox/Toolbox or commercial database software like FileMaker Pro. Similarly, in a language documentation project, one will often want to create machine-readable representations of the relationship between textual data and audio or video recordings (e.g., in the form of time-aligned transcription). Doing this requires software which allows one to make direct associations between portions of distinct computer files—something

beyond the power of a spreadsheet program but which is made easy with a tool like Elan.

While the use of linguistics-specific software will generally facilitate the creation of implementations that are faithful to the underlying structure of the data, simply using such software does not guarantee that the data will come out "right". For instance, a lexicon tool may make it straightforward to specify morphosyntactic information like part of speech, but in a language where it is deemed valuable to list multiple paradigmatic forms of a word within a lexical entry, one may want to indicate not only a part of speech at the level of the lexeme but also associate each word form with additional grammatical categories (e.g., a case label). This requires a two-tiered model of grammatical specification, at the lexeme level. A given lexicon creation program may support this, but it cannot "know" to make use of such a feature unless the documenter is aware that it is needed in the first place. A "perfect" implementation of a flawed analysis of the structure of some data will be of little long-term value and, at least for now, arriving at good structural analyses of linguistic data is a task well beyond the skills of any machine.

It would be ideal, of course, if, in a chapter like this one, it would be possible to give explicit recommendations about what software is "best" for language data of a particular type. Unfortunately, the needs of every project are too particular for this to be possible, and there is a tradeoff between being able to implement a data model as faithfully as possible to its underlying logical structure, employing a tool that everyone on a project team can use comfortably, and ensuring that the tool that is used can produce resources which can be put to use by the audiences to be served by a project. The main advice one can give is to outline the overall goals of a project and data types to be collected in advance (see section 6) and then to solicit advice from experienced individuals when making choices of software. One important factor to consider when choosing software will be the kinds of formats it is able to work with (see section 4).

3.3 Audio and video resources and publications

It may seem like a gap in the discussion in this section that it has focused on "traditional" text-oriented resources rather than recordings. There is a reason for this: Many of the important components of the documentary record of a language employ data types which are of interest to communities well outside of the arena of language documentation and which, therefore, will be well-supported independent of language documentation efforts. Audio and video recordings are a prime example of this: Technologies for capturing, storing, and manipulating audio and video data have a large, stable market of which language documentation work is only a minute part. Therefore, efforts will be made to model the structure audiovisual information and implement those models regardless of the activities of language documenters.

Publication technologies are similar in this regard. The audience for old (e.g., print publications) and new (e.g., multimedia content) modes of information dissemination is vast and new models and technologies for producing publications—in a broad sense of the term—will emerge with or without language documentation work. Therefore, given limited resources, language documenters will need to devote more energy to issues relating to the modeling and implementation of data types specific to documenting languages, like annotated texts, lexicons, and grammars. Nordhoff's (2008) discussion of a possible set of

design principles and implementation decisions for the creation of "ideal" electronic grammars is a good recent example of the kind of work which is needed.

4. DATA FORMATS

Closely related to the notion of data model implementation is the notion of data *format*, that is, the way that information happens to be encoded in a digital resource. When using this term, we must first recognize that it is potentially quite vague and is better understood as a multidimensional concept referring to a number of distinct "layers" of data encoding rather than a single monolithic notion. In particular, in the present context it is useful to distinguish between *file format* and *markup format*. The former concept is likely the more familiar since it refers to the different file types associated with software applications. These include, for example, the DOC format created by Microsoft Word, PDF format, or WAV audio format. The details of the structure and digital composition of these formats are largely irrelevant to language documenters, though, as will be discussed in section 4.1, some are more suitable for language documentation than others. By contrast, markup format, in the present context, refers to the way the substantive content (at least from the documenter's perspective) of a resource is encoded on top of a particular file format. As such, it is directly relevant to language documenters and will be discussed in more detail in section 4.2. In section 4.3, a third way of categorizing formats, by their intended function, will be discussed.

This section will focus primarily on conceptual issues relating to data formats. For specific recommendations regarding appropriate formats to use for different kinds of data (e.g., text, audio, or video) and for different kinds of functions (e.g., archiving versus presentation), it is best to refer to up-to-date online resources (e.g., the E-MELD School) or to contact a digital archivist or other individual with the relevant expertise. Standards recommendation for digital formats tend to evolve rapidly, and periodic review of the state-of-the-art is required for successful language documentation. Video formats, in particular, have yet to see the same degree of stabilization as text and audio formats.

4.1 File formats: Open versus proprietary

The most important way in which file formats can differ from the perspective of language documentation is whether or not they are *open* or *proprietary*. Devising satisfactory definitions of these terms is not completely straightforward, but, practically speaking, the distinction centers around whether a given format is designed to be used in any application which may find that format a useful way to store data or whether it is intended to be used only by the format's owner or via licensing agreements with that owner.

Among the most widely-used open file formats is the "raw" text file (sometimes referred to as a TXT file or by the file extension .txt), consisting of a sequence of unformatted characters—these days, ideally, of Unicode characters (see Anderson 2003 and Gippert 2006:337–361 for an overview of Unicode). Such files can be created and read by a wide array of programs on all widely used operating systems, and no one organization has any kind of ownership over the format. By contrast, a well-known proprietary format is the Microsoft DOC format. While this format is creatable and readable by programs not created by

Microsoft, it was not designed specifically for this, and the format has been subject to change under Microsoft's discretion regardless of how this may have impacted the ability for other software to create and read files in that format.[10]

For work on language documentation, one of the most important recommendations is to prefer the use open formats whenever possible, and always for the archival version of a resource (see section 4.3). There are two major reasons for this. First, open formats, by their nature, are more likely to be created and read by different computer programs, which means that resources encoded in open formats will generally be available to a wider audience than proprietary formats. Furthermore, open formats are much more likely to be supported by cost-free programs since, very often, the reason why a format is proprietary in the first place is so a company can profit from selling software which can work with files in that format. While the issue of cost may not be particularly relevant to linguists working at well-funded universities, one must keep in mind that the larger audience for a documentary resource will often consist of individuals or groups which are not particularly privileged financially.

The second reason to disprefer proprietary formats is that, by virtue of being largely under the control of a particular company, they are more likely to become obsolete—that is, resources encoded using them are more likely to become unreadable or uneditable—because the company controlling them may decide to change the format that its tools support over time, while discontinuing support for its earlier formats, or because the company itself may disappear, meaning that its formats will no longer be supported by any program. With open formats, even if one institution making a tool supporting that format should cease to exist, the nature of the format itself makes it relatively easy for a new group to create a tool supporting use of that format.[11]

4.2 Markup formats

Markup, in a digital context, refers to the means by which part of the content of a given document is explicitly "marked" as representing some type of information. Continuing the example of a wordlist entry discussed in section 3.1, markup could be used to indicate, among other things, that: (i) the data in question is a lexical entry, (ii) the first element of the lexical entry is the headword, (iii) the second element is an indication of part of speech, and (iv) the third element is a gloss.

An example of the data in (1) presented in a possible markup format is given in (5), where a markup language known as Extensible Markup Language (XML) is used. XML is a widely used open standard for marking up data using a system of start and end tags which surround data of the type specified by the tag. The distinction between a start and an end tag is maintained by the prefixation of a slash before the name of an end tag. Start tags can have complex structure wherein they include not only the tag but also specification of attributes of the

---

[10] In recent years, the DOC format has been replaced by the DOCX format which, in principle, is an open file format—though, in practice, it has not yet been widely adopted outside of Microsoft.

[11] It is important to distinguish between open source and open format. Open source refers to whether or not the computer code that forms the basis of a program is made freely available for inspection and modification. In practice, open source programs are more likely to use open formats for various reasons, some practical and some social. However, many closed-source programs also allow one to produce resources in open formats (e.g., Microsoft Word allows one to save documents into the open HTML format)

data using feature-value pairs indicated with equal signs. In (5) these are used to specify the language of the content of the tags. Readers familiar with HyperText Markup Language (HTML), the dominant markup format for web pages, should find the overall syntax of XML to be familiar since the two use the same basic conventions (see Gippert 2006:352–361 for additional relevant discussion).

(5)
```
<lexicalEntry>
    <headword lang="French">
    chat
    </headword>
    <pos>
    n.
    </pos>
    <gloss lang="English">
    cat
    </gloss>
</lexicalEntry>
```

The XML in (5) is somewhat simplified for purposes of exposition. Nevertheless, it gives a basic idea of data markup in general and XML specifically. While numerous markup languages have been developed, XML has been chosen here for illustration since, at present, it enjoys widespread popularity within the software development world as a format facilitating the exchange of data across individuals and computer programs and is considered an appropriate markup format for language data where markup is relevant.

XML has at least four attributes which make it especially well suited for language documentation. First, it can be expressed in plain text—i.e., the markup tags do not use any special characters or formatting not found in plain text files. This means that XML files can make use of a widely-adopted open format and facilitates archiving. Second, while XML is primarily designed to be a machine-readable markup format, the fact that the tags can make use of mnemonic text strings (e.g., "lexicalEntry" in (5)) means that it can be, secondarily, human-readable. Thus, even in the absence of materials documenting the specific markup conventions used in a given resource, it will still often be possible to discern the content of a document marked up with XML by inspecting it with a simple text editor. This self-documenting feature of XML markup is a desirable characteristic for the long-term preservation of the data in the document since it helps ensure its interpretability even if a document becomes detached from its metadata (see section 5). Third, XML is flexible enough to mark up a wide range of data types for diverse kinds of content—one simply needs to define a new kind of tag to mark up a new kind of data. Finally, XML has been widely-adopted in both commercial and non-commercial contexts. As a result, there is extensive tool support for processing and manipulating XML documents, going well beyond what would be possible to create with the resources solely devoted to language documentation.

While the XML example in (5) may make it appear to be a markup format of use only to specify the data contained in resources which would traditionally be printed (e.g., dictionaries or texts), it can also be used to annotate other kinds of resources, like audio and video recordings or images using so-called *stand-off* markup, wherein the markup itself is stored in a separate resource from the

resource it describes. Such stand-off markup can then specify which part of an external resource it refers to using some kind of "pointer", for example the specification of horizontal and vertical coordinates in a scanned image. A common use of such stand-off markup in language documentation is to create a time-aligned transcription of a recorded text where the text transcription is encoded in an XML file containing pointers to times in an audio file—as is done in the EAF files produced by the Elan annotation tool (while these files end in the extension .eaf rather than .xml, the data contained within them is expressed in XML).

While use of a markup language like XML solves many problems associated with describing the content of a language resource, it is important to understand that, on its own, it is merely a scheme for marking data with different kinds of tags—not, for example, a standardized way of encoding lexical data or an annotated text. Rather, one must, beforehand, develop an abstract model of a lexicon or a text, and then implement it in XML (see section 3 for discussion of modeling and implementation). XML—or any generalized markup language—serves merely as a kind of "skeleton" on which domain-specific markup schemes can be constructed. In the long run, the creation of long-lasting, repurposable language documentation will be greatly facilitated by the use of common markup conventions for basic linguistic data types, which will allow for the development of tools which can work with the data from diverse documentation projects making use of these conventions. At present, however, general consensus has yet to emerge for most aspects of the markup of linguistic data.[12] In the absence of such consensus, the best strategy is to employ markup conventions using mnemonic labels and to document how those labels are to be interpreted in the context of a given resource.

Finally, in general, one will not manipulate markup directly, for example by editing an XML document in a text editor. Rather, one will use software providing a graphical interface to the markup (as Elan does with its XML format, for example) or software which allows for the data it creates to be exported to an appropriate markup format—as is the case with, for example, FileMaker Pro's XML export. However, while one need not learn how to create or edit a suitable markup format directly, it is important to be able to determine whether a markup format is sufficiently open and transparent to be appropriate for a project's documentary needs, which requires some knowledge of the relevant issues.

4.3 Archival, working, and presentation formats

In addition to classifying formats by their various technical features, one can also classify a format by virtue of its possible or optimal functions. In the context of language documentation, three particular functions stand out: *archival*, *working*, and *presentation*. An archival format is one designed for longevity. In the ideal case, a resource stored in an archival format today would be readable in a hundred years or more (assuming it has not been lost on unreadable media). A working format is one manipulated by a given tool as the user creates or edits a resource—this is the format language documenters will spend most of their time with. A presentation format is a version of the resource optimized for use by a specific community. Presentation formats can range from a print dictionary to a

[12] To take one example, despite being fairly well-studied, consensus has yet to emerge on the ideal markup format for interlinear glossed text (see Palmer and Erk (2007) for recent discussion).

multimedia text presentation and are what those not involved in the language documentation process itself would generally consider to be the "normal" kind of language resource. For discussion of archival, working, and presentation formats for different data types referencing specific formats, consult the E-MELD School.

In an ideal world, a single format could function simultaneously as an archival, working, and presentation format for a given kind of resource. However, this is a practical impossibility. This is most clearly the case for presentation formats which are, by definition, audience specific (e.g., an ideal linguist's dictionary has a very different form from a community dictionary, even if they can be based on the same underlying lexical database) and also may require optimization for certain modes of dissemination (e.g., an audio file may need to be reduced in size, and therefore quality, in order to become suitable for distribution via the internet). Though such problems are not as acute when comparing archival formats and working formats, they do not disappear entirely. For example, archival formats often tend to be large and "verbose"—that is, they may express their content with lots of redundancy—since this helps ensure their long-term readability. Working formats, by contrast, are often more useful if expressed in ways that are concise, since this allows them to be manipulated more efficiently by a computer.

A language documentation project, therefore, needs to anticipate the use of formats with distinct functions over its lifespan, working formats for performing day-to-day tasks, archival formats for long-term storage, and a variety of presentation formats depending on the communities it wishes to serve and the ways it wishes to serve them. The need for such a variety will inevitably complicate the management of a documentation project, though such complications can be alleviated by forward planning (see section 6) and the use of tools either natively using open formats as working formats or allowing easy and reliable export of their working format to an open format since such formats tend to be more straightforwardly transformable to appropriate archival and presentation formats than proprietary formats.

5. METADATA

In order for the data collected by a project to be usable in the long-term, it not only needs to be well-structured internally but also must be associated with appropriate *metadata*—that is, information describing the constituent resources of a documentary corpus, including, for example, their content, creators, and access restrictions (see Good (2003) for introductory discussion in a linguistic context). Metadata is an essential part of any documentary corpus, and a metadata plan forms an integral part of a general data plan.

Since materials deposited in an archive will need to be associated with their metadata in order for them to be accessioned into an archive (see Conathan (this volume)), the best place to turn to for advice in terms of what metadata you should include with your resources is the archive where you will deposit your data, assuming it is clear what archive is best placed to protect the resources created by your project. While the metadata policies for language archives are all broadly similar, each archive will have its own specific expectations and, in some cases, an existing set of forms which can be used for metadata entry and which the archive will design to facilitate its own accessioning process.

In devising a metadata plan for a language documentation project, it is useful to think about your metadata needs across two broad parameters: the different kinds of items that will require metadata and the different users of your metadata. I will not consider here in detail the specific metadata "fields" one may want to record, since there are a number of complicated considerations involved relating to specific project requirements and resources (though see Conathan (this volume, section 3.2) for relevant suggestions). At a minimum, it is necessary to record basic "bibliographic" information like creators (a cover term encompassing anyone involved in a resource's creation), date of creation, place of creation, language being documented, access restrictions, and brief descriptive title or keyword (see Johnson 2004:250). At a maximum, one can consider the extensive IMDI[14] metadata set—most projects will fall somewhere in between. If you are starting a new project, it may be useful to look at the latest version of the IMDI set to get an idea for the range of information that, in principle, might be worth keeping track of.

5.1 What requires metadata

Most of the documentary objects requiring metadata can be arranged in a hierarchy from more general to more specific using the categories *project*, *corpus*, *session*, and *resource*.[15] An additional set of "objects" requiring metadata, but which do not fit directly into this hierarchy, are the various *people* involved, including most prominently speakers and documenters.

A *resource*, in this context, is a unique object, either a physical item or a computer file, comprising part of the documentation of a language. Often multiple resources are created as part of the record of a single event (e.g., an audio recording, a transcription, and an associated photograph). These would then be grouped into a *session* (following the terminology adopted by IMDI as discussed in Brugman et al. (2003), though the term *bundle* is also used for this concept). Sessions may then belong to some user-defined higher-level grouping which can be referred to as a *corpus*, which might, for example, consist of all sessions documenting a specific language in a multilingual documentation project. Finally, a set of corpora may be joined together into a larger *project*, for example all the materials collected by a given documentary team. While it is generally possible to apply the notions *resource* and *session* fairly consistently, *corpus* and *project* are somewhat more subjective and are more likely to be employed using conventions specific to a documentary team.

Conceiving of the items produced by a language documentation project as belonging to a hierarchy is useful insofar as it allows one to avoid repeating the same information in multiple places. For example, if documentary work is externally funded, it will often be necessary to acknowledge that funder somewhere in the metadata. This is most conveniently done at a high-level, like that of *project*, as opposed to specifying this for each individual resource. Similarly, resources documenting a single speech event will share information like *creators* and *date*, thus making it useful to employ the notion of *session*. Finally, since most information about people is independent of the actual resources they contributed to, person metadata constitutes a level on its own. Each

[14] http://www.mpi.nl/imdi/

[15] The conceptual metadata scheme discussed here is derived from work done in the context of IMDI. See Brugman et al. (2003).

person can be associated with a unique identifier (e.g., their name, if appropriate), which can then be referred to in session metadata.

5.2 Metadata users

When creating metadata, one should consider the range of users who are likely to make use of it, with the most important division being those directly involved in a project versus those outside of it. On the one hand, those involved in a project are unlikely to be, for example, interested in project-level metadata since they will already be aware of such information. By contrast, they are likely to be very interested in session-level metadata as a means to keep track of a project's progress. On the other hand, those outside of a project are likely to want to refer to project-level metadata as a first "entry point" into a set of documentary materials and will only be interested in session-level metadata for projects which they have determined are relevant to their interests.

A documentary team will presumably keep track of the metadata it needs for its own purposes without special consideration but may forget to record information that is shared among the team but will be unknown to outsiders. For example, the fact that a given speaker is an elder will be obvious to those working directly with that speaker but could be very difficult to determine from an audio recording. Therefore, the language documenter must try to keep in mind that the users of metadata are not privy to the same level of information that a documentary team will be. In fact, the concerns of one particular group of "outside" users should resonate particularly strongly with experienced documenters: Future versions of themselves who are likely to forget quite a bit about the context of their old recordings but will still be interested in using them.

This two-way distinction between project members and those outside of a project is, of course, quite simplistic and masks many internal divisions within those categories. With respect to outsiders, a further important division involves researchers versus community members. Existing metadata schemes for language resources, like IMDI (see above) and the Open Language Archives Community metadata set (OLAC; Simons and Bird (2008)) are oriented towards the research community, and speaker communities are likely to have distinct interests in terms of the information they find valuable. For example, linguists are typically more concerned with the languages a given speaker's parents may have spoken at home than they are with who that person's parents actually are, while speaker communities are quite likely to be interested in the genealogical relations of those who participated in the creation of a set of documentary resources—especially if they are close relations.

5.3 Practical considerations

While it is not possible here to go into details regarding metadata management techniques, two practical considerations are especially crucial. First, every resource created by a documentation project should be associated with a unique identifier. For computer files, this identifier should be the name of the file itself, which, therefore, needs to be created with uniqueness in mind. For physical resources, this identifier should be marked on the resource itself directly or with an adhesive label. (See Johnson 2004:149–151 for examples of possible schemes for creating unique identifiers relevant to a language documentation context.) In an ideal world, a given resource would be indelibly associated with its metadata so that its content would always be completely clear. However, in practice,

metadata tends to be stored separately from the resource itself. Therefore, it is also useful for a resource's identifier to give some minimal information about its content. Then, even if the resource cannot be straightforwardly associated with its metadata at a given time, some information about it can be gleaned from its label. For example, a recording of Angela Merkel in German made on 1 January 2009 might have a label like *deu-AM-20090101.wav*. This identifier contains a three-letter language code, followed by the initials of the speaker, a date, and, finally, a file extension indicating this is a WAV audio recording. Obviously, such an identifier does not substitute for a full metadata record, but it, at least, gives some information about a resource which will be quite valuable in case its metadata becomes lost.[16]

A second practical consideration regarding metadata is that, especially in field settings, it is essential that metadata entry be made as straightforward as possible. Ideally, metadata will be recorded for a resource on the same day it is created—while one's memory is still fresh. But, language documentation can often be a tiring task, leaving little energy at the end of the day to work with a complex metadata management system. Since metadata usually has a fairly simple structure almost any program one might use to create a table or a database, e.g., Microsoft Excel, FileMaker, or Shoebox/Toolbox, can be used for metadata entry and storage. Since one such tool is already likely be used for other aspects of documentation, the most straightforward route is to co-opt it for use as a metadata entry and storage tool as well—at least when in the field.[17]

6. NEEDS ASSESSMENT

Implicit in the discussion to this point has been that, either formally or informally, a given project has undertaken a technical needs assessment—that is, the overall goals of a project have been outlined, an enumeration of the different resources required to reach those goals has been formulated, and a workplan has been devised to ensure that those resources can be acquired or developed over the course of the project. Bowern (this volume) contains a general overview of issues relating to project planning, including some discussion of how to integrate a project's data needs into its overall design.

A useful notion to keep in mind while considering the data management aspect of a needs assessment is the *workflow* of the individuals involved in the project: That is, what will be the series of day-to-day tasks each project participant will work on at each phase of the project. Modeling a project's workflows will help ensure that the optimal technologies are chosen to accomplish its goals since it will clarify the specific technological needs of each member of the project team. So-called "lone wolf" research may only require an informal understanding of a project's workflow, while projects involving large and diverse teams may benefit from a more formalized depiction of workflow breaking down project work into a set of interconnected tasks. A very large project may even require a member of the documentary team to invest substantial (paid) time in managing its overall workflow.

---

[16] For similar reasons, it is often helpful to record some brief metadata at the beginning of an audio or video recording.

[17] The Archive of Indigenous Languages of Latin America (AILLA) has examples of Excel spreadsheets and Shoebox/Templates which can be used for metadata management.

7. THE DOCUMENTER'S RESPONSIBILITY

This chapter can only give a brief outlines of the relationship between data and language documentation. Furthermore, because the technologies for capturing and storing data are continually evolving, our understanding of data in the context of language documentation will also continually evolve, and the language documenter will have to periodically reconsider their technological practices and keep abreast of new developments by consulting up-to-date resources.

Unlike, say, learning how to transcribe using the IPA, working with the data produced by language documentation is not something you can simply "learn once". Rather, it will be an ongoing, career-long process. Furthermore, since, in many cases, the access that many individuals leading language documentation projects have to new technologies greatly exceeds that of the communities they work with, it is, to some extent, their responsibility to serve as the conduit through which information about these technologies reaches these communities (see Jukes (this volume) for relevant discussion).

The most succinct way to summarize these points is: understanding how data collection and management fits into a documentation project is a kind of *research*. It, therefore, submits to all the requirements of research: keeping up with the field, knowing the limits of one's expertise, tracking down outside sources, constantly evaluating and reevaluating one's conceptual understanding and methodological practices, and instructing collaborators on appropriate practices. Just as analyzing your data requires research, so does working with the data itself.

REFERENCES

Anderson, Deborah. 2003. Using the Unicode standard for linguistic data: Preliminary guidelines. *Proceedings of the E-MELD Workshop 2003: Digitizing and Annotating Texts and Field Recordings*, 10pp. <http://www.emeld.org/workshop/2003/anderson-paper.pdf>

Austin, Peter K. 2006. Data and language documentation. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 87–112. Berlin: Mouton de Gruyter.

Bell, John & Steven Bird. 2000. A preliminary study of the structure of lexicon entries. *Proceedings from the Workshop on Web-Based Language Documentation and Description*. Philadelphia, December 12–15, 2000. <http://www.ldc.upenn.edu/exploration/expl2000/papers/bell/bell.html>

Berez, Andrea. 2007. Technology review: EUDICO Linguistic Annotator (ELAN). *Language Documentation and Conservation* 1:283–289. <http://hdl.handle.net/10125/1718>

Bird, Steven & Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language* 79:557–582,

Boas, Franz (ed.). 1911. *Handbook of American Indian languages*. Washington: Government Printing Office. (Smithsonian Institution Bureau of American Ethnology Bulletin 40.)

Boynton, Jessica, Steven Moran, Anthony Aristar & Helen Aristar-Dry. 2006. E-MELD and the School of Best Practices: An ongoing community effort. In:

Linda Barwick and Nicholas Thieberger (eds.), *Sustainable data from digital sources: From creation to archive and back*. Sydney: Sydney University Press. <http://hdl.handle.net/2123/1296>

Brugman, Hennie, Daan Broeder & Gunter Senft. 2003. Documentation of Languages and Archiving of Language Data at the Max Planck Institute for Psycholinguistics in Nijmegen. Paper presented at the "Ringvorlesung Bedrohte Sprachen" Sprachenwert - Dokumentation - Revitalisierung, Fakultät für Linguistik und Literaturwissenschaft - Universität Bielefeld, 17pp. <http://www.mpi.nl/IMDI/documents/articles/BI-EL-PaperA2.pdf>

Dryer, Matthew. 2006. Descriptive theories, explanatory theories, and basic linguistic theory. In Felix Ameka, Alan Dench & Nicholas Evans (eds.), *Catching language: The standing challenge of grammar writing*, 207–234. Berlin: Mouton de Gruyter.

Dwyer, Arienne M. 2006. Ethics and practicalities of cooperative fieldwork and analysis. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 31–66. Berlin: Mouton de Gruyter.

Good, Jeff. 2003. A gentle introduction to metadata. Open Language Archives Community Note. <http://www.language-archives.org/documents/gentle-intro.html>

Grinevald, Colette. 2003. Speakers and documentation of endangered languages. In Peter Austin (ed.), *Language documentation and description, volume 1*, 52–72. London: Hans Rausing Endangered Languages Project.

Himmelmann, Nikolaus P. 1998. Documentary and descriptive linguistics. *Linguistics* 36:161–195.

Himmelmann, Nikolaus P. 2006. The challenges of segmenting spoken language. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 253–274. Berlin: Mouton de Gruyter.

Johnson, Heidi. 2004. Language documentation and archiving, or how to build a better corpus. In Peter Austin (ed.), *Language documentation and description, volume 2*, 140–143. London: Hans Rausing Endangered Languages Project.

Mithun, Marianne. 2001. Who shapes the record: The speaker and the linguist. In Paul Newman & Martha Ratliff (eds.), *Linguistic fieldwork*. Cambridge: CUP.

Mosel, Ulrike. 2006. Sketch grammar. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 301–309. Berlin: Mouton de Gruyter.

Nathan, David. 2006. Thick interfaces: Mobilizing language documentation with multimedia. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 363–379. Berlin: Mouton de Gruyter.

Nordhoff, Sebastian. Electronic reference grammars for typology: Challenges and solutions. *Language Documentation and Conservation* 2:296–324. <http://hdl.handle.net/10125/4352>

Palmer, Alexis and Katrin Erk. 2007. IGT-XML: An XML format for interlinearized glossed texts. *Proceedings of the Linguistic Annotation Workshop*. Prague: Association for Computational Linguistics. 176–183. <http://www.aclweb.org/anthology/W/W07/W07-1528>

Schultze-Berndt, Eva. 2006. Linguistic Annotation. In Jost Gippert, Nikolaus P. Himmelmann & Ulrike Mosel (eds.), *Essentials of language documentation*, 213–251. Berlin: Mouton de Gruyter.

Simons, Gary & Steven Bird (eds.). 2008. OLAC Metadata. Open Language Archives Community Standard. <http://www.language-archives.org/OLAC/metadata.html>

Simons, Gary, Kenneth S. Olson & Paul S. Frank. 2007. Ngbugu digital wordlist: A test case for best practices in archiving and presenting language documentation. *Linguistic Discovery* 5:28–39. <http://journals.dartmouth.edu/cgi-bin/WebObjects/Journals.woa/2/xmlpage/1/article/314>