A Brief Description of the Spinoza Typological Database

Simon Musgrave
ULCL / Spinoza Project: Lexicon & Syntax
University of Leiden
S.Musgrave@let.leidenuniv.nl

1. Introduction

The Spinoza Project: Lexicon and Syntax is a research project directed by Prof Pieter Muysken (Catholic University of Nijmegen). The project aims to investigate basic properties of human language through a detailed study of the phenomena observed in situations of intensive language contact, especially where the languages involved are genetically unrelated. Four geographical areas were selected for close study: the Balkans, Bolivia/Rondonia, Eastern Indonesia, and Suriname/Benin/Ghana. Data on approximately eighty languages from these four regions is available to the project, or is being collected as a part of the project. In order to assist comparison over such a large body of data, the project is developing a database to contain both primary and analytic data. In addition to the data generated by the areal studies of the project, the database will also contain parallel data from a balanced sample of the world's languages (Rijkhoff, Bakker, Hengeveld & Kahrel 1993). This data will serve as controlled comparative data for the areal sample, and will be assembled under the supervision of Prof Kees Hengeveld (University of Amsterdam).

2. Types of data

The Spinoza database contains six types of data for each language included.

i. General background data - This data includes information about the geographical location of the speech community which uses the language, the size of that community, the status of the language in the community (i.e. whether it is the only language, the primary language or a secondary language), whether it is used as a written language, and alternative names for the language.
ii. Data on sources - This data enables the end-user of the database to trace any piece of a primary linguistic data to a specific source. The possible types of source include published works, field notes, recordings, native speaker judgments etc.
iii. Data on analysts - This data enables the end-user to trace any piece of primary linguistic data to a specific analyst.
iv. Texts - This data is the core of the database. For each language, we aim to have a text of at least 50 clauses. Several representations of each clause are stored: orthographical, a Roman transliteration where required, phonemic, morphological analysis, morphological gloss, partial syntactic analysis and a free translation. Information about borrowed items (a very important matter in the context of the overall project) is also stored. Isolated sentences and paragraphs will also be included here, where they are necessary to clarify analytic points.
v. Vocabulary list - For each language, a basic vocabulary list will be collected. This list consists of the 200 word Swadesh list plus a small number of additional items from the Natural Semantic Metalanguage list of semantic primes (Wierzbicka 1996). The research team for each of the four target regions can also nominate additional items to be included for that region alone.
vi. Typological analysis – Data on a range of typologically interesting variables is stored for each language. The majority of this data consists of classical word order information, but information about word class systems and processes of derivational morphology is also included. As far as possible, this data is collected as

generalisations over the analysis of individual units, rather than as higher-level analytic statements inputted directly by an analyst (see below for further discussion).

3. Interrelations in the database

Rather obviously, all data pertaining to any particular language is interrelated. In addition, each item of primary data is tied to a specific analyst and a specific source. Where applicable, an item in the vocabulary list is cross-referenced to occurrences in the texts for that language. Finally, typological statements about a language are linked to specific examples in the primary data for that language.

4. Data input as analytic process

The Spinoza database was always intended to treat primary data, in particular text data, as being of great importance. However, in the process of developing the application, this orientation has assumed greater importance. The original intention was that the text data would be available to illustrate the typological analysis which the analyst provided, that is the data would have the kind of top-down structure commonly used in typological databases. Higher-level generalisations are entered directly, and a greater or lesser amount of supporting evidence is provided as the analyst sees fit. The current architecture of the Spinoza database is rather different. In this scheme, the analyst identifies relevant units in the primary data, and the application then responds by asking the analytic questions relevant to that linguistic unit. The typological generalisations which can be made about any particular language in the database are then summations of the individual analyses which have been entered.

This approach was initially used for morphological analysis. One feature that was considered highly desirable in the database was for morphological analysis and glossing to be represented in aligned interlinear text (as in the SIL Shoebox application). To do this, it is necessary to make the morpheme the basic unit of stored data; this move in turn opened up many other possibilities. Firstly, information about derivational morphology is naturally gathered during the process of inputting the morphological analysis. Whenever a morpheme is identified as an affix (simplifying slightly - there are other possibilities), this triggers a form which first asks whether the morpheme is derivational in effect, and then goes on to gather additional data about the derivational process, if relevant.

Further, one representation of a line of text in this system is as an ordered list of references to dictionary entries. But from this point of view, all linguistic units above the morpheme can be represented identically. Therefore it is straightforward to define such units (NPs, clauses etc.) in the same way, as ordered lists of dictionary references, and to tie analytic statements to the units so defined. Once the analyst has broken a text into clauses, they can then be asked to identify relevant units within the clause, and appropriate analytic questions can be asked about those units. The application can be constructed to repeat this process until individual words or morphemes are reached. In fact, the Spinoza application will not be exhaustive in this sense, but the architecture allows the possibility.

The idea is perhaps made clearer by considering the flow of work required to input data into the application:

1. enter general data on language
2. enter data on analyst and source(s)
3. enter primary text in clause complexes
4. specify units within clause complexes and relation between them
5. enter individual clause
6. specify units within clause and clause-level typological analysis

7. specify units within units-already identified and give relevant typological analysis (if required)
8. loop back to 5, unless at end of text
9. generate typological profile of language on basis of analysed text
10. provide additional example clauses as required to complete typological profile
11. enter vocabulary data

This process looks rather onerous as set out here. However, it should be borne in mind that analysis is only required once relevant linguistic units have been identified. The primary data is intended to be natural text, and complex structures are not common in such data. In many cases, the only analysis required for a clause will be to split it into morphemes, and to specify the word order of the predicate and its arguments. The time required will, we hope, be amply justified by the close relationship between higher-level generalisations and specific analyses in the final data.

Although the architecture described here is intended for use in assembling comparative data, it could equally be applied to handling a large corpus of data from a single language. The analytic part of the application will be essentially modular: the code which handles the specification of units will be absolutely general, and the analytic questions which are posed will be the result of a single variable (the head of the unit specified). Therefore it will be straightforward to extend the analytic capacity of the application to include additional linguistic units (the database is a Microsoft Access application, the code is in Visual Basic for Applications).

5. Some Problems

a) Fonts – the standard font throughout the Spinoza database is Lucida Sans Unicode. This offers full Unicode compatibility and a character set which includes IPA extensions. However, the aligned interlinear text used for morphological analysis currently uses Courier New. This is the only Unicode compatible font which is fixed pitch, and therefore the only possibility for aligning text by counting characters. But it does not include IPA extensions, and is a more limited character set than Lucida Sans in other respects also. The solution to this problem would seem to be to use Lucida for interlinear also, and to handle alignment by counting pixels or twips. This will mean interfacing Access with a text editor giving the required degree of control over character placement. This is a problem which remains to be addressed at present.
b) Incomplete data – the initial design of the application treats all typological data as Boolean variables. During the development process, it has become clear that this will not be satisfactory. There are inevitably cases where the data does not allow for a definite answer to an analytic question, and where additional data is hard or impossible to obtain. The only honest response in these situations is “don’t know”, and we believe that this is the response that the end-user should see. (Note that simply not responding in such a situation is not sufficient: then a NEG value is ambiguous between a real “no” and “don’t know”.) This issue does not arise in the course of analysis, as the questions to be answered are dependent on the units identified by the analyst. (If a numeral and a noun head are identified, they must have a relative order.) But at the point where the application produces generalisations over the analysis, this is a problem Using Boolean variables has the advantage of giving a transparent data structure, an important consideration for long-term maintenance and for the possibility of the Spinoza database being part of an integrated group of typological databases (see contribution to this meeting by Monachesi et al). But in this case, the limited use of a multi-valued data type in conjunction with Boolean variables seems appropriate. For each language-level generalisation, if none of the relevant Boolean variables has a positive value or if more than one positive value occurs in a mutually-

exclusive set of variables, then a single multi-valued, dependent field will give information as to why no answer is possible (e.g. not enough data, inconsistent data).

c) Degree of normalisation – the Spinoza database links a large amount of data of very different types in one application. This is probably a feature of any database which aims to handle primary linguistic data in a useful way. In constructing such a database, the developers have to make many decisions about when it is useful to aim for a normalized set of tables, and to what extent pursuing that aim actually aids the design process. The remarks here are intended only as comments on this issue, based on one group's experience, and which may have some heuristic value for others. Some problems are not hard to solve, for example deciding where dependent data should be stored in separate tables and where this is not necessary. But other issues are more complex and more specific to linguistic databases. One important question is what is usefully regarded as a unit of information; this may not be clear in dealing with linguistic data. For example, one table in the Spinoza database contains information associated with each line of text (roughly, a sentence or a clause). A line of text is a complex (and sometimes rather large) piece of data. In terms of the architecture described above, it is clearly necessary to divide these units into smaller ones, but it is not immediately obvious what the best strategy for doing this might be. The solution we have adopted is to use the morpheme as a more basic unit, but it would be equally possible to use the orthographic word (although the result would be, we think, a less powerful and flexible application). Even having made the decision as to the correct units, the most useful way of storing them is still an open question. To continue with the same example, the morphemes of a text line are not entries in a separate table in the Spinoza database. Instead, the information is distributed between a dictionary table, a table of occurrences of dictionary items, and a text string listing the morphemes of a line and their order. The relation amongst these data is normalised in some cases (e.g. between the dictionary and the occurrences table) but not in others (e.g. between the morpheme list and the dictionary). The crucial factor in making such decisions is (unsurprisingly) the output capability which is sought. Thus, in our example, it is highly desirable to be able to access information about the dictionary entry of a morpheme, and it is highly desirable to be able to access information about where a morpheme occurs. But the order of morphemes in a line of text has been assigned lesser importance. The information is used in constructing interlinear glosses, but it will be hard for the end-user to search for the specific orderings of morphemes in the data. This may very well turn out to be a design defect in the long-term; another solution would certainly have traded added capability on this point against reduced capability elsewhere.

References:

Rijkhoff, Jan, Dik Bakker, Kees Hengeveld & Peter Kahrel (1993) A method of language sampling *Studies in Language* 17: 169-203

Wierzbicka, Anna (1996) *Semantics: Primes and Universals* Cambridge UK: Cambridge University Press