Alexander Magidow

A relational database model and prototype for storing diverse discrete linguistic data

1 Introduction

This article describes a model for storing multiple forms of linguistic data within a relational database as developed and tested through a prototype database for storing data from Arabic dialects. A challenge that typically confronts linguistic documentation projects is the need for a flexible data model that can be adapted to the growing needs of a project (Dimitriadis, 2006). Contributors to linguistic databases typically cannot predict exactly which attributes of their data they will need to store, and therefore the initial design of the database may need to change over time. Many projects take advantage of the flexibility of XML and RDF to allow for continuing revisions to the data model. For some projects, there may be a compelling need to use a relational database system, though some approaches to relational database design may not flexible enough to allow for adaptation over time (Dimitriadis, 2006). The goal of this article is to describe a relational database model which can adapt easily to storing new data types as a project evolves. It both describes a general data model and shows its implementation within a working project. The model is primarily intended for storing discrete linguistic elements (phonemes, morphemes including general lexical data, sentences) as opposed to text corpora, and would be expected to store data on the order of thousands to hundreds of thousands of rows.[1]

The relational model described in this paper is centered around the linguistic datum, encoded as a string of characters, associated in a many-to-many relationship with 'tags,' and in many-to-many named relationships with other datums.[2] For this reason, the model will be referred to as the 'tag-and-relationship' model. The combination of tags and relationships allows the database to store a wide variety of linguistic data.

This data model was developed in tandem with a project to encode linguistic data from Arabic dialects (the "Database of Arabic Dialects", DAD).[3] Arabic is an extremely diverse language group, with a dialects stretching from Mauritania to Afghanistan,

[1] The author would like to thank Yonatan Belinkov for his assistance with the initial phases of this project, and the two anonymous reviewers who provided extremely helpful feedback both for this article and for the associated web project, as well as Nicholas Coulombe for his insightful comments on early drafts. All remaining errors are the author's own.

[2] I will use the plural 'datums' as a plural of 'datum' for referring to a countable, individuated set of pieces of data, as opposed to 'data' which refers to a general, unindividuated collection. For example, one could discuss 'hundreds of datums', but write in general about the 'data' that needs to be inputted into a system.

[3] `http://database-of-arabic-dialects.org/`. The acronym is meant to evoke the Arabic letter *ḍāḍ,* originally a pharyngealized voiced alveolar lateral fricative [ɮˤ], traditionally considered characteristic of Arabic, which is sometimes referred to as *luġat aḍ-ḍād,* 'language of *ḍāḍ.'*

many of which are not immediately mutually intelligible with one another. Much of the diversity is lexical or morpholexical, so that closed-class words such as pronouns and open-class words such as the verb ‘to go’ function as shibboleths between dialects. Individual phonemes (realization of /*q/) and even phonological rules (raising of /a/ word finally) can also act as shibboleths. Since this information is scattered in a variety of different publications, the goal of the project is to develop a website which can act as a hub for researchers to input legacy and novel Arabic dialect data and visualize that data in a variety of ways (as lists, maps, paradigms, etc). It is also intended to allow for multiple researchers to use it as a research tool for inputting and analyzing their own data. Data can be made publicly available, or access may be restricted only to a researcher and selected collaborators.

2 Similar projects

The basic desiderata of the Database of Arabic Dialects project were as follows:

- Input and search of actual language data (i.e. words in Arabic dialects, not just typological meta-analysis)
- Ability to handle a wide range of linguistic data, from phonemes to short phrases, with the ability to easily add additional data types
- Ability to consider multiple relevant classifications for a given datum (e.g. a single word could be an interrogative, a pronoun, and a relative marker)
- Permission control and contributor attribution
- Intuitive and efficient interface for data input and analysis
- Publication of all project prototype source code

The initial phase of the project was to conduct a survey of existing projects with an eye to making use of their database structure or their code if they were open-source, provided they could meet the requirements outlined above.

The most similar existing project was the FIELD (Field Input Environment For Linguistic Data) tool,[4] a branch of the larger E-MELD (Electronic Metastructure for Endagered Language Data) initiative.[5] This website provided input for primarily lexical data, as well as interlinear glossed texts. However, the project appears to have been moribund since at least 2010, the last copyright date listed on the E-MELD website. The website claims that an improved version of this tool is forthcoming, but this seems not to have happened. Indeed, the database behind the project (a PostgreSQL database) appears to be broken and one can no longer access the website as of August 2015. When

[4] `http://emeld.org/tools/fieldinput.cfm`. Unless otherwise noted, all websites mentioned were accessed on August 26, 2015.

[5] `http://emeld.org/`

the project was accessible, in 2012, it featured very simple HTML (not Javascript) based input forms which proved to be extremely slow. The FIELD project is not open source, and thus little useful information can be gleaned from what remains today.

The FIELD tool was also reliant on the incomplete GOLD (General Ontology for Linguistic Description) ontology. All data had to be related to the categories in the GOLD ontology, but GOLD simply is not detailed enough to properly describe the Arabic data. For example, the most recent 2010 version of GOLD only includes demonstratives as pronominals, and makes no distinction between demonstratives which act pronominally and those which can only be determiners.[6] Arabic dialects often make this distinction (especially those in North African) and it is not an uncommon distinction cross-linguistically (Diessel, 1999).

Another project is the Vienna Corpus of Arabic Varieties (VICAV), "an international project aiming at the collection of digital language resources documenting varieties of spoken Arabic."[7] The VICAV project focuses on curated presentations of linguistic information, with attractive manually written profiles of dialects, curated lists of dialect isoglosses, dictionaries of single dialects, bibliographies and complete texts in several dialects. The project operates largely on a document-level of organization, with XML documents based on Text Encoding Initiative (TEI) standards being served largely intact as single documents. For example, the project has several small dictionaries, one for Cairene Arabic, one for Syrian, one for Tunisian and one for Modern Standard Arabic. While each dictionary can be searched individually, it does not appear to be possible via the current interface to search across multiple dictionaries or otherwise collate data between dialects.[8]

The VICAV project is similar to the DAD project, but is based on a different design philosophy. Whereas the goal of DAD is to present only raw data from a large number of dialects in a highly structured and searcheable format, VICAV is designed for presenting general information about a small number of dialects. A relational database model is more appropriate for the DAD model of data storage since it is based on the individual piece of linguistic data as the smallest datum, whereas the VICAV project is based on a document paradigm, even if individual documents contain smaller divisions.

An example of the different between these projects is how they illustrate isoglosses between Arabic dialects. There is a relatively small set of areas in which dialects tend to vary significantly. For example, most dialects have very similar interrogative systems, though the actual wordforms differ. On the VICAV website, there is a section which shows the distinctive linguistic features of two dialects (Cairene and Damascene), including interrogatives, demonstratives, pronouns, etc. These pages appear to be manually written rather than automatically generated, and show only those linguistic isoglosses which have been chosen by the editor of the pages. In contrast, the DAD

[6] `http://linguistics-ontology.org/gold/2010/Demonstrative`

[7] `http://minerva.arz.oeaw.ac.at/vicav2/`. Note in spite of the term 'corpus' in the title of the project, searchable textual data only represents a small percentage of its current resources.

[8] The project serves data from a MySQL database that is derived directly from the XML documents (Karlheinz Mörth, p.c.), so in theory it should be possible to perform cross-dialectal queries.

project allows a user to search for items with any tags, so that one could search for all words tagged with `interrogative`, providing greater comparative coverage and flexibility in the choice of variables to investigate. With automatically generated results, it is easier to include more dialects and more data. In the DAD dataset there are nearly 80 dialects which currently have comparative data, whereas it is not clear how the VICAV approach of manual curation could easily scale up to a larger number of dialects.

Another intriguing project is the Oto-Manguean Inflectional Class Database, a comparative database of verbal inflection data from twenty Oto-Manguean languages spoken in Mexico.[9] The project is of interest because these languages differ significantly in their verbal inflectional classes. They are so diverse that comparative searches are difficult to carry out as the categories between languages are not always equivalent. The challenges of storing this data could indeed inform the database design here, but the project does not appear to have documented their database structure nor have they seem to have released their source code. This is an excellent illustration of the need to document project design and to make source code available for other projects.

During the initial survey of existing projects, it appeared that the World Atlas of Language Structure[10] and the related Atlas of Pidgin and Creole Language Structures Online[11] were based on a very simple, flat database structure that was capable only of storing meta-linguistic information about languages based on scholarly analyses, e.g. typological categories, not actual linguistic data, e.g. lexemes and their meaning. Only while revising this article did it become clear that the underlying software, CLLD, is significantly more sophisticated than it first appeared to be and can indeed store full linguistic data (Forkel, 2014). Given the late addition of this information, it is impossible to integrate it completely into this article. When appropriate, reference will be made to the design decisions made by the CLLD team and how they compare with those made in developing the DAD project.

2.1 General Structural Desiderata for Linguistic Data

Though the FIELD tool itself is no longer functional, the E-MELD project produced a number of articles discussing the efficient storage of linguistic data. Farrar (2006) describes a model for a fundamental data type for storing linguistic information. His model is based on the notion of a ‘linguistic sign,’ consisting of a “3-tuple” with a form component, a meaning component, and a grammatical component. The form component is “any annotation entity that represents the phonetic, phonological, orthographic or otherwise physical manifestation of the sign” (p. 7). The meaning component can refer to the semantic content in the sense of the meaning of an item, but this is considered distinct from a translation in another language, which itself would be a linguistic sign. The meaning component could also encompass semantic features such as [+animate]. Finally the grammatical component includes information such as part of speech and

[9] `http://www.oto-manguean.surrey.ac.uk/`

[10] `http://wals.info/`

[11] `http://apics-online.info/`

morphosyntactic features. Only these three components are considered essential to a linguistic sign. Information such as annotations or even translations are to be modeled as relations between linguistic signs, rather than stored as components of a linguistic sign (p. 8). He also emphasizes that this should be a content-based model, rather than a display-oriented model, with display handled at the software level (in the case of their XML model, via XSL transformations).

Penton et al. (2004) discusses how to store paradigmatic information, a category into which much descriptive linguistic data falls, from phonemes to open-class lexemes. In exploring paradigmatic elements, they find that paradigms “simply represent an association between linguistic forms and lingustic categories ” (p. 6). The tabular display of a paradigm is essentially an algorithm which places linguistic signs in the appropriate cell based on their linguistic properties, typically expressed in the labels above or beside it in the table. From the perspective of data storage, it is sufficient to store only the appropriate characteristics of the linguistic datum. A linguistic datum need not be ‘aware’ of its place within a paradigm. Only when it comes to displaying the data is there a need for the algorithm that will place that data into a table. In the terms of Farrar (2006), the meaning or grammatical components of the linguistic sign can store the necessary data for transforming multiple datums into any number of paradigms.

Good and Hendryx-Parker (2006) discuss a model for encoding the potentially contested relationships between the world’s languages. Their database model consists of nodes (corresponding to “langoids”) which have a primary key (a unique human readable identifier), basic metadata (human readable names) and a one-to-many relationship with digital documents and books. Each node is related to other nodes by way of relationships, which take the form of an RDF predicate, a relationship which links a subject to an object by way of a ‘predicate,’ an human readable expression of what kind of relationship exists between them. In this model, the subject and object would be nodes. One of their primary concerns is how to represent contested information (e.g. particular arguments about the structure of a language family) without compromising the integrity of the database. By using multiple RDF links between the same elements to encode competing relationship hypotheses, the nodes do not change (lingoids) and multiple hypotheses can be stored in the same database. For example, a hypothesis that groups languages B and C as daughters of language A would have RDF links between A (subject) and both B and C as objects, e.g. `A mother B`, and `A mother C`. On the other hand, if Language C is hypothesized to be a daughter of Language B, which in turn is a daughter of Language A, then there would be a RDF link of `A mother B` and `B mother C`, implying that A is the grandmother of language C. The database would store both sets of relationships. [12]

They use RDF links for linking to other contestable metadata. They link between a language and its ’language type’ (language area, language family, language, dialect, etc) with an `is of language type` predicate. This allows for encoding whether a

[12] The two hypotheses would be marked in such a way that they are distinguishable, presumably through the use of reification to allow the relationship itself to be treated as a subject or object in another relationship.

researcher considers Chinese, for example, to be a language family (with Chinese varieties considered languages in their own right) or a language (with Chinese varieties to be considered dialects). The database is implemented in an Object-Oriented Database (Zope Object Database) which supports RDF relationships natively. Thus, in this data model, contradictory pieces of information in the same category (i.e. classification) are stored as overlapping relationships between nodes.

Lewis et al. (2006) present important considerations for citation, fair use and digital distribution of linguistics data. They suggest several important principles for data storage and attribution: full attribution (indication of the full citation for a datum), sheltering of data (providing tools to limit access to data) and acknowledgment of ownership (acknowledge additional ownership for data if it has changed from the original source). To follow these principles for a publicly accessible collection of descriptive linguistic data like DAD, the project should provide tools for fully citing data and controlling permissions for access to that data. There should also be acknowledgment for what they term "enrichment," adding to the data in a meaningful way. In the case of a crowd-sourced database, the very act of inputting the data (which often includes regularizing notation, adding annotations, geolocation, etc.) should be considered a form of enrichment and the person who performs this, even with legacy data found in published sources, should receive credit.

3 Tag-and-Relations Database Model

The data models from the E-MELD projects are mostly implemented in XML, which is common as a data storage and exchange format in digital humanities projects. XML is a very adaptable format which has the advantage of being relatively easy to change after the initial design phase of a project. However, there are a number of reasons why a project may prefer to use a relational database model, ranging from personnel expertise to software support to support within online communities.[13] For users of relational databases, a flexible database design is needed to adapt to changing requirements as a project evolves, preferably with a minimum of changes to the basic organization of the database's tables.

The models described by Farrar (2006) and Good and Hendryx-Parker (2006) both take a single, irreducible form as the central data element in the database. In the former, this is the linguistic sign, while in the latter it is the language node. Both models also allow for named relations between datums. These named relationships can be multiple and redundant — multiple relationships can express the same information with variations that represent different analytical interpretations of the data. Each of their models also allows for certain core metadata to be associated with each datum.

[13] In this particular project, which was unfunded, the sole developer's expertise was primarily in relational databases, while the software tools and hosting services that seemed best suited for the overall project worked with relational database software. The CLLD project made a similar decision to use relational database software rather than use an RDF graph database due to the latter's "non-standard requirements in terms of deployment, administration and maintenance" (Forkel, 2014, 3).

The data model describe here, therefore, builds on the central element of a transcribed linguistic datum, i.e. the actual linguistic signal, equivalent to the 'form component' of the linguistic sign in Farrar (2006). The linguistic datum, and its associated metadata, are stored as rows in a single table. Similar to those other models, every linguistic datum can also be related to other linguistic datums via a named relationship. In a relational database, this is easily operationalized as a through table with two foreign keys, both pointing to the table of linguistic datums, and with an additional field for naming the linguistic relationships.

An important concern is how to store the properties associated with each datum. All datums will have certain identical properties: a gloss or other indication of meaning, a bibliographical citation for where the information was actually contained, in order to give proper credit, the name of the language or dialect that the datum is drawn from, information about the transcription scheme or encoding used to transcribe the datum if that is relevant), a human readable annotation giving information about the datum for users, etc. Since this data is common to all datums, it can be included in the linguistic datum table as additional fields, some of which may constitute foreign keys to other tables (e.g. to a bibliographic reference table).

This contrasts with the approach recommended in Farrar (2006). In his model, much of the metadata is linked to a linguistic datum with a relational link. For example, a gloss is treated as a linguistic datum in and of itself, and a relational link connects the datum in the primary language of interest to the gloss datum in the language of translation. However, this is unnecessarily complex when a database is designed for storing data in a single language. For most purposes, it is much simpler to only store data in the language of interest, and to simply include the gloss (or glosses) within the same table as the linguistic datum, since they are functionally inseparable.

A greater difficulty comes in recording linguistic properties of datums. A single datum might belong to multiple classes of categories. For example, an interrogative may also function as a pronoun and a relativizer. Conversely, a single property may apply across many classes of datum. Pronouns, nouns and verbal agreement affixes all share properties of gender and number in many languages. A project may decide to add in classifications after the initial design and input of data that have not previously been encoded in the database. A synchronically oriented database may decide to include historical classifications, for example, and there might be multiple and contended classifications that need to be included. In an XML based approach, it is straightforward to add new attributes to each datum, but in a relational database it is difficult to organize the data in such a way that we can provide that level of flexibility.

A common approach to database design uses individual tables for each category of datum. In such a model, different parts of speech might each inhabit separate tables, with each table containing fields unique to that part of speech. A noun table could contain a field for gender, number, etc. For languages with unpredictable plurals, that too could be included in the same row as the singular as a separate field. However, it is difficult for such a model to handle datums that belong to different categories simultaneously, or to unite properties that are shared across tables. New categories require alteration of the basic structure of the database, and a small number of members in a given table may require a property that is not shared by the majority of members of that table.

The approach that is used in the model here is instead to store every linguistic datum in a single table, and to mark each datum with an unlimited number of tags. Tags are simply text fields, and as such they are flexible enough to store almost any kind of data. One datum could be marked with the tags `noun, feminine, dative`, another with `demonstrative, pronoun, adjective, deictic, article`, all while being stored in the same table. The textual tag system can easily be extended to include values by way of a delimiter, so one could mark `gender=masculine` or `gender=feminine`, in a manner very similar to XML attributes.[14] This also allows for searches on whether or not an item has a gender at all. In a small number of Arabic dialects, the interrogatives for 'who' and 'which' can be marked for gender and searching for they keyword `gender` would allow a researcher to find those dialects, rather than having to search for both `masculine` and `feminine`. The tag system makes it straightforward to mark these rare forms with the appropriate properties, which would be more difficult in a table-per-POS approach. Note that the database structure itself does not specify how structured these tags must be. Depending on the requirements of a project, they could even be in XML-like form, e.g. `attribute = 'gender' value = 'masculine'` or any other system that could easily be parsed by the underlying software.

All of the models discussed in the previous section make use of named relationships between linguistic datums, and this model does so as well. A datum might be a variant of another datum, e.g. allophones, or a datum, e.g. a sentence, might exemplify another datum, e.g. a word. Named relationships between datums allow for the system to express an infinite variety of interrelations between the datums in the system.

The database system, then, is based on two simple mechanisms: Tags, which are applied to individual language datums, and relationships, which are named links between datums, so we can refer to this is a 'tag-and-relationship' database model. A schematic of the basic database structure is show in Figure 1. In the schematic, tags are used to mark both the predicates of relationships and the properties of datums — note that these are separate sets of tags. Only the linguistic datums are given relationships to one another in this current model, though a tag-and-relationship model could certainly be applied to other entities. For example, if storing hypothesized linguistic relationships was a priority, the tag-and-relationship model could easily be applied to a table that stores linguistic entities (i.e. languages or dialects). Both tags and relationships can be multiple, overlapping, and contradictory. This model therefore can support multiple analyses of the same material in a manner similar to the model described by Good and Hendryx-Parker (2006).

[14] For the DAD project, the period was chosen as a delimiter for different 'parts' of a tag. This proved to be problematic when performing searches with regular expressions, as the period is a metacharacter and must be escaped. The solution thus far has been to do searches as basic string searches instead, but the delimiter could easily be changed.

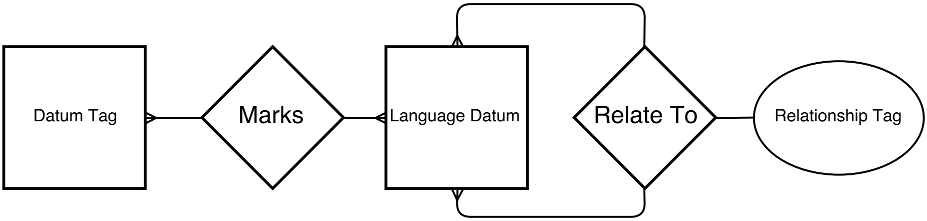


Abbildung 1: A diagram of the basic tag-and-relationship relational database model in entity-relation notation.

The content of tags is not limited to a particular area and this is the basis for the extremely flexibility of the tag-and-relationship model. Farrar (2006) argues for a separation of meaning and grammatical metadata, but there is no obvious reason to do so outside of a theoretical model of a linguistic sign. Tags can encode semantic data, such as `animacy=nonhuman`, and can also encode grammatical data such as `POS=noun`. If for some reason a separation between the two *is* necessary, it is simple to add that information into the tag along with a delimiter, for example `sem:animacy=nonhuman` or `gram:POS=noun`. Tags can mark other domains as well. A very inclusive project may contain complete datasets of a very specific nature. For Arabic, one dataset that may be included at a future date on the DAD website contains the babytalk (caregiver child-directed speech) equivalents of normal words in several Arabic dialects.[15] For someone who is not searching specifically for those items, that dataset may be of little use, so having all items from that dataset tagged in a particular way (e.g. `dataset=babytalk`) allows for users to easily include or exclude that data. Also, while we have earlier explored the issues with standardized ontologies such as GOLD, such ontologies can easily be used as the basis for the tag system while still being extensible beyond those standards if necessary.

The tag system also allows for inclusion of data which is underspecified, in contrast to a model which has a table-per-POS model. Manfred Woidich and Peter Behnstedt have granted the author access to the flat, spreadsheet style dataset that underlies their Wortatlases of Arabic dialects (Behnstedt and Woidich, 2010). This data is extremely messy, with almost no structured information on parts of speech. While part of speech can sometimes be inferred by data structure (e.g. some fields give the standard verbal citation forms separated by a comma), the majority of the data will necessarily be imported without any information on part of speech. This will make searching the data more difficult for scholars, but does not pose a serious problem for the database structure. Those datums which have POS information can be tagged accordingly, and those which do not simply will not be tagged.

With such an underspecified structure, the tag-and-relationship model requires each project to establish general principles of best practice. The primary criterion for how to

[15] See `http://babytalk.barefootlinguist.com/`

store data is whether it will be retrievable with a query. This is important both at the application level (computer-computer interaction) and the interface (human-computer interaction) level. For the data to be displayed, it must be sufficiently well tagged and interrelated that the frontend application can access it. For human interaction, the data must be accessible in a way that a human user can easily construct searches and read the results of those searches.

It is straightforward to store anything from a phoneme to affixes to open-class lexical items with this model, though a given project will have to decide exactly how to model some items. In essence, the tag-and-relationship model is fixed, but individual projects must design how they model their data within the system. In the DAD implementation, as we expect would be the case in most implementations, the gloss of a linguistic datum is a required field. For a phoneme, this gloss could be basically uninformative, e.g. `phoneme`, with the majority of the data stored in tags (e.g. `phoneme, bilabial, unvoiced, stop`). The gloss could be more elaborate, with the human readable and searchable, `unvoiced bilabial stop`. It would be best to encode those same properties as tags to allow for application level interactions with the data, as the application should be able to retrieve the entire class of stops without accidentally including lexemes glossed with, for example, `to stop`. Outside of phonemes, most items tend to be more straightforward in having a meaningful gloss.

Sentential or higher level data becomes more complex. An individual idiom or sentence can easily be stored, with the gloss operating as a free translation. More complex sentential data could simply be stored as XML, with tags aimed at the application to tell it that these datums need to displayed and searched in a manner different from data stored in plain text.[16] A more complex example would be interlinear glossing. Farrar (2006) discusses the issue of storing interlinear glosses in XML but it is not clear how they should be stored in this system. Interlinear glosses normally have three levels of information, the transcribed data from the language, a morpheme-by-morpheme gloss, and a free translation. Occasionally they have more levels depending on the complexity of the example. All three levels could be stored in the transcribed data field of an individual datum as XML. Alternatively, each level of the interlinear gloss could be its own datum, linked together with relationships such as `IL.morphemegloss` and `IL.freetrans`. The gloss fields could be empty or could have placeholder information.[17]

Relationships are best used when the application code must keep related datums together, though co-occurring linguistic forms such as paradigmatic data do not necessarily need to be explicitly linked. As Penton et al. (2004) have shown, paradigms are simply the intersection of different traits, which in this model would be stored as tags. The application would render the paradigm based on those tags, and there is no practical need for the members of that paradigm to be ‘aware’ of one another through

[16] It is not necessary that all data be stored in the same way, provided that the encoding is clearly marked. If the vast majority of transcribed data is in the form of a few characters of transcription from the language of research, it is better to not have redundant XML code used in every single entry. Instead, the exceptional entries could be marked as such.

[17] The CLLD project handles this issue by modeling sentential data as an entirely separate data type from morpholexical or typological data (Forkel, 2014).

relationships. On the other hand, if two members of a paradigm are stored as separate datums and co-vary, they should be linked so that they can be properly aligned during display. For example, in Arabic present-tense verbs, some conjugations of the verb take a circumfix, so in Syrian Arabic 'they write' is *yi-ktub-ū* 3M-write.present-PL. If a speaker is speaking more formally, they might say *ya-ktub-ūn,* with the same meaning but a change in the form of both the prefix and the suffix. In that case, the prefix and suffix datums should entered as separate datums, since a linguist may be interested in seeing only prefixes or suffixes, but they should be linked with relationships so they can be properly aligned upon display. While in principle it is best to provide too much marking for the data, rather than too little, there is a trade-off with programming complexity, as more complex queries are needed when both tags and relationships must be queried.

Relationships are useful for storing data such as allophones and allomorphs, since there is a clear base form and a clear variant form. It is also straightforward to mark the relationship between more and less common forms, since the more common form can act as the base form. Similarly, for the babytalk data, it is reasonable to consider the adult words to be the primary form, and to mark a relationship so that a babytalk form is the subject of a `baby talk variant of` predicate to an adult lexeme.

When there is a more equal relationship between datums, it is not always clear whether relationships should be marked in both directions (i.e. each datum has a relationship to the other for a total of two relationships), in only one direction (with queries tracing both sides of the relationship) or not at all. For example, many oblique pronoun suffixes in Arabic have allomorphs depending on the preceding phonological environment, usually with one form occurring in post-vocalic position, and one form after consonants. Neither form is clearly the base form. Ultimately in the DAD project it was found that since these allomorphs are inputted and displayed in paradigms, it was not necessary to link them with a relationship, as they are retrievable based on their tags alone.

4 Modeling Data in DAD

The model described in the previous section is a general model which can be instantiated in a variety of ways, depending on the needs of a project. In this section, we will use the Database of Arabic Dialects (DAD) website and database to illustrate the model and to discuss the details of some of the design decisions that are part of a large scale project of this nature. The goal of the DAD project is to provide a crowd-sourced website with comparative data for Arabic dialects. The web interface provides both the tools for data input and for data viewing and analysis. The website is implemented using the open-source, Python-based, model-view-controller Django web framework. Django provides the interface between a PostgreSQL database backend and the web interface, and specializes in allowing for database design and querying using Python code. Django is also valuable for an unfunded pilot project since it automatically provides a relatively efficient data entry interface for all tables present in the database. The DAD project is open source, and the code is hosted on GitHub.[18]

The central table in the database represents the linguistic datum. This table utilizes the tag-and-relationship model described above.A simplified schematic of the database model from the DAD project is included in Figure 2. In the DAD implementation this table includes the following fields:

normalizedEntry A text field of unlimited length for storing transcribed linguistic data

normalizationStyle This field indicates which of the standard transcription styles are used for the `normalizedEntry` field

gloss An HSTORE field in the current implementation which stores key–value pairs. This allows for storing glosses from multiple languages

entryTags A foreign key, allowing a many-to-many relationship to tags

relationships A foreign key to an intermediate table which enables a many-to-many relationship with other linguistic datums. The intermediate table has a field which is a foreign key to a table of ‘relationship tags’ used to mark the nature of the relationship

dialect A foreign key to a dialect table

sourceDoc A foreign key to a bibliography table

sourceLoc A text field for storing information about where an item is located in the source document

contributor A foreign key, allowing a one-to-one relationship to a contributor

permissions A field which stores a permission string, marking data as “Private” (only the contributor and collaborators can see it), “Public” (any visitor to the website can see it) or “Public No Export” (any visitor can see the data, but it cannot be exported from the website)

originalOrthography An optional text field for storing the item in the original transcription in the source

annotation An optional text field for storing human readable additional information or reflections on the data

This table is described in detail since it forms the heart of the database, but also because it represents some important design decisions. First, note that it conforms to several of the recommendations from the literature discussed previously: It makes the actual linguistic data the primary piece of information, it contains information

[18] `http://github.com/amagidow/dialects`

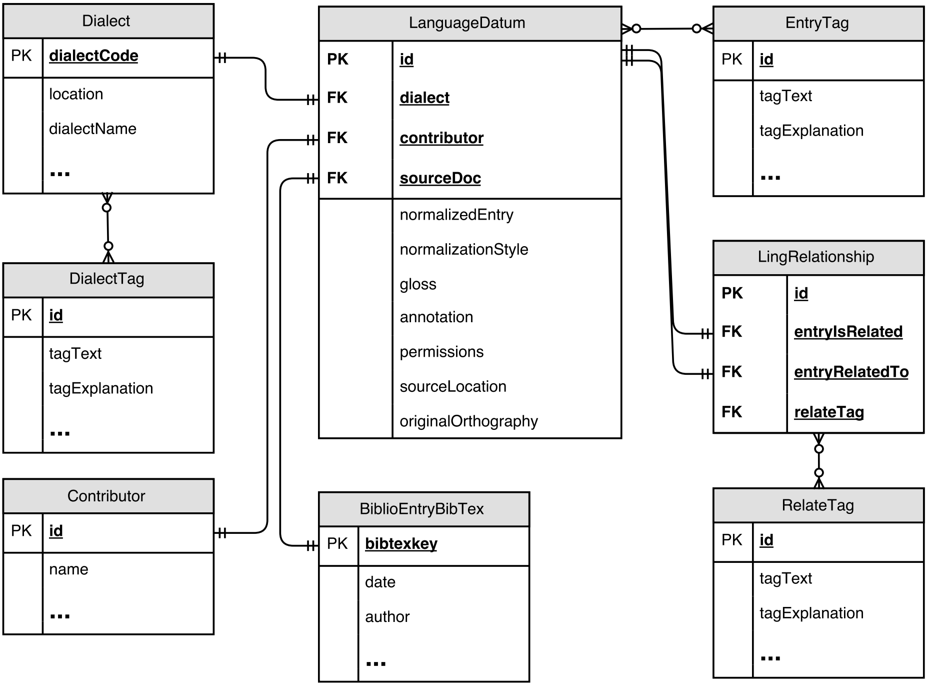


Abbildung 2: A simplified diagram of the DAD database structure. Ellipses indicate columns which exist in the database but are not shown explicitly on this chart.

about the original source and it assigns credit to the data contributor, who also has fine grained, changeable permission control. That is to say, it provides credit both to the original, published source of data, and to the contributor who provided what Lewis et al. (2006) termed the "enrichment" of digitizing that data on the website.

Second, it was necessary to make important decisions about normalization. In this table, the `sourceLoc` field represents a violation of third normal form, when "a non-key field is a fact about another non-key field." (Kent, 1983, 121). In theory and in practice, many datums come from the same source (a single page, or a single map in a volume) so normalizing this information into an intermediate table would be the most data-efficient way to store it. This would eliminate repetition and allow for more consistent update in case of errors. However, an intermediate table greatly increases the complexity of queries and of the database, and the citations are meant for human users, so some imprecision is acceptable. Leaving this unnormalized also does not represent a huge increase in duplicated data, since the `sourceLoc` field is rarely more than a few characters in length (e.g. `p. 15`) and the `sourceDoc` field is a very efficient integer foreign key field. As

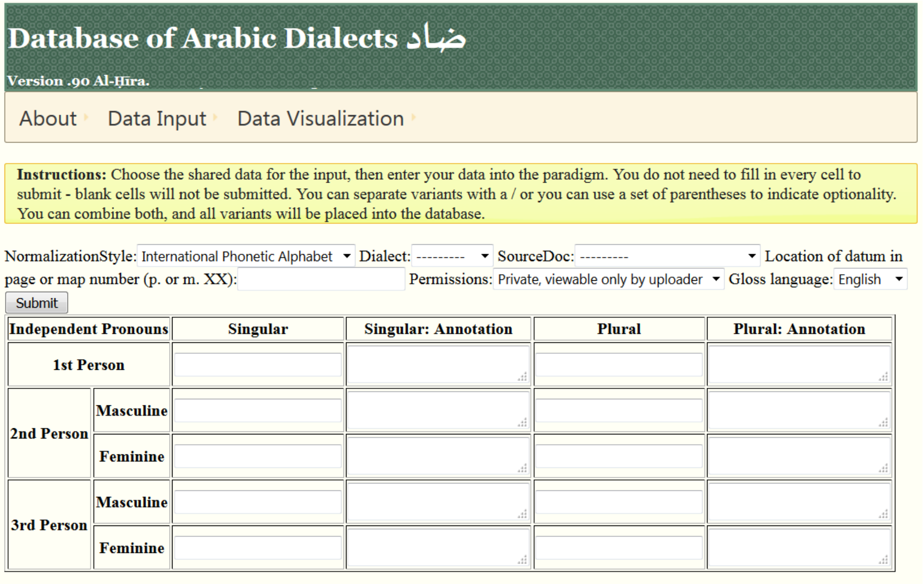
Dimitriadis (2006) notes, linguistic databases are tiny in comparison to most commercial databases, typically with under a million records, and therefore storage efficiency is not as important as it might be in a database with millions of rows.

The tags for linguistic datums are stored in a separate table from the linguistic datums, with an intermediary table to allow for a many-to-many relationship between tags and datums. Each tag has a both the text of the tag, as well as an explanation of the tag's use. This allows for more consistent application of tags. The same is true for the tags used to mark linguistic relationships. The `Dialect` table also links to tags with explanations, so that entire dialects can be tagged to indicate properties of the dialect, or hypothesized classifications. For example, Arabic dialects are traditionally classified as urban, rural or nomadic, so they are tagged accordingly, and a researcher could also tag dialects according to their own hypothesized classifications.

The bibliographic entry table is another area where some compromise was needed. Ideally we would be able to take advantage of existing software that has a clean interface for interacting with a database backend. We could link directly to a centrally stored bibliography database and avoid the need to implement our own. The widely used, open-source program Zotero seemed like a strong candidate, but its database structure has no persistent key analogous to a key in BibTex. Though each entry has an integer primary key in the database, if an item were accidentally deleted (a not uncommon occurrence when using the program's interface), any links to that primary key would be lost and the integer key would provide no clue as to the original source. Instead the built-in interface provided by Django (referred to as the "admin interface") is used for entering bibliographic items. All bibliographic entries must have a unique, human readable key. Should a bibliographic entry somehow be lost, the human readable key retained in the linguistic datum table would still provide information that would allow for reconstructing the bibliographical entry. A field in the `BiblioEntryBibTex` table also allows for entering a full BibTex entry, as this at least constitutes a defacto standard for storing bibliographical references. One limitation of this implementation is that the bibliographic entry table was designed around published works (with fields for author, title, publisher, etc.) and it is unclear how it should be modified to accommodate elicted field data, as the DAD project is intended to eventually accommodate field researchers. Going forward, the bibliographic entry table will probably need to be redesigned.

4.1 Flexibility and Data Integrity

Since most of the information about a linguistic datum is stored by way of tags and relationships, the system cannot always take advantage of the data integrity tools that are characteristic of relational databases, such as constraints or triggers. This is a major trade-off of the flexibility of the tag-and-relationship model. There is nothing keeping contributors from accidentally adding duplicate or synonymous tags, or to enforce the use of relationships in a consistent manner. If a more rigid structure had been used, such as a design in which each part of speech has its own table, there



Instructions: Choose the shared data for the input, then enter your data into the paradigm. You do not need to fill in every cell to submit - blank cells will not be submitted. You can separate variants with a / or you can use a set of parentheses to indicate optionality. You can combine both, and all variants will be placed into the database.

NormalizationStyle: International Phonetic Alphabet Dialect: --------- SourceDoc: --------- Location of datum in page or map number (p. or m. XX): Permissions: Private, viewable only by uploader Gloss language: English

Independent Pronouns		Singular	Singular: Annotation	Plural	Plural: Annotation
1st Person					
2nd Person	Masculine				
	Feminine				
3rd Person	Masculine				
	Feminine				

Abbildung 3: Screenshot of DAD paradigm input.

would be stronger built-in control over data validity, but this would bring with it the disadvantages described above.[19]

The solution for this issue in the DAD project has been to enforce uniformity at the interface level. Data input can be performed in several ways, but since most of the data currently being entered into DAD is paradigmatic, the data entry is itself in the forms of paradigms. Figure 3 shows the input interface for the independent (nominative) personal pronouns. From the user perspective, they are simply entering data into a table-like entity, not unlike how they would enter data into a table while writing an article in a word processor. The web application adds the appropriate tags as it saves the data into the database. The user does not directly adds the tags themselves. If necessary, a user can later go in and edit individual datums to modify the tags. For example, an input page might be missing a category that is found only very rarely, and so the user can go and add in that category data after first taking advantage of the paradigmatic input page. The user can still be restricted to only using the existing tags, helping maintain the integrity of the tag system. The addition of new tags can be restricted to a trusted set of users, or added by administrators upon user request.

[19] The model described by Dimitriadis (2006) uses a table of features and lists of values for each feature to strike a balance between flexibility and automatic constraint, though it is still possible for a user to accidentally add a redundant feature, just as it is possible for users to add redundant tags in the tag-and-relationship model.

The paradigms input and display pages are themselves based on Python code which specifies the vertical and horizontal headers for the paradigm, and the tags and relationships that should be applied to the datum in each cell. This means that creating a bespoke paradigm based on user demand is a simple process that can be accomplished very quickly and with relatively little technical skill. Strategically it is better and easier to build an interface that itself enforces data integrity than it is to allow a user to input messy data that must later be cleaned.

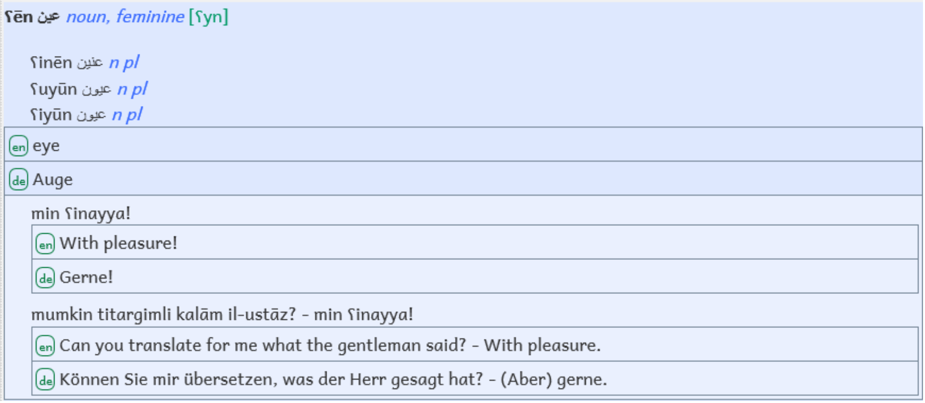
Other types of validation could be performed either by the SQL backend or the web interface. Duplicate entries could be disallowed by the web interfaced or restricted in the database using uniqeness constraints. Ultimately, of course, there is no foolproof method for ensuring clean data input, but if a given contributor often submits poor quality data, their contributions can be excluded from a search, or their user privileges revoked.

For data retrieval, the strategy has been to provide as much hinting as possible. There are a number of data views, but most of the views allow for searching against the Arabic datum itself, the gloss, the annotation, and the tags. For both the gloss and the tag fields, the website uses Javascript to provide auto-complete suggestions as the user types. This allows for the user to explore the system of tags, as well as common glosses, while they are in the midst of a search. Another page provides a complete list of tags and their explanations, but the auto-complete suggestions are intended to make it unnecessary for most users to visit that page.

4.2 Storing Diverse Data

It may be helpful to illustrate how a variety of different data types might be stored using the DAD implementation of the tags-and-relations model. The primary type of data that has been stored in DAD thus far is closed-class morpholexical items such as demonstratives, pronouns and interrogatives. In general, it has been straightforward to store this data within the tag-and-relationship model. For almost all of this data, only tags are necessary for input and retrieval, and so relationships have generally not been used.

The DAD project is flexible enough that it could also store data from other projects. The lexical data from the VICAV project is a good example. A sample dictionary entry is shown in Figure 4. The entry has a headword, basic grammatical information, the triconsonantal root of the word [ʕyn], multiple possible plural forms, multilingual definitions and two idiomatic phrases. In the DAD system, the singular and all three of its plural forms would each constitute a linguistic datum, as would each of the idioms, for a total of six linguistic datums. The appropriate properties of each of the forms would be marked with tags, e.g. `noun, feminine, root.`ʕyn The plural forms would be linked to the singular with the relationship `pluralOf`, and the idioms would be linked to the singular (and possibly to the plurals) with the relationship `idiomContaining`. The gloss fields are multilingual, and so could store the English and German glosses.



ʕēn عين *noun, feminine* [ʕyn]

ʕinēn عنين *n pl*
ʕuyūn عيون *n pl*
ʕiyūn عيون *n pl*

en eye

de Auge

min ʕinayya!

en With pleasure!

de Gerne!

mumkin titargimli kalām il-ustāz? - min ʕinayya!

en Can you translate for me what the gentleman said? - With pleasure.

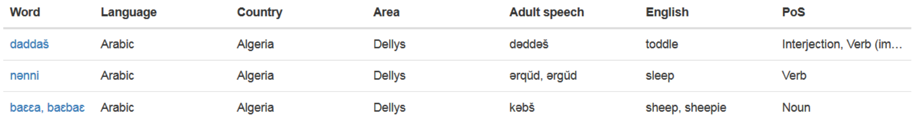
de Können Sie mir übersetzen, was der Herr gesagt hat? - (Aber) gerne.

Abbildung 4: A screenshot of a typical entry from the VICAV website.

Note that the singular and plural *must* be linked with relationships since Arabic plurals are unpredictable from the singular.

The lexical data from the Babytalk project could also be stored in the DAD database. Figure 5 shows sample data from this website. The actual babytalk word would be stored as a datum, as would the adult equivalent, with part of speech data marked with tags and both would share similar glosses (though 'sheepie' would not be an appropriate gloss for the general adult term). The location would be linked in the same way as other datums. Each babytalk item would be linked to its adult equivalent with a relationship marked with e.g. `babytalkOf`. Entering this data would also have the positive side-effect of increasing the general use lexical data present in the database.

Finally, this system can store phonological data as well. One dataset available to the author is a listing of the realization of different proto-consonants in various Syrian dialects. For example, the proto-phoneme /*k/ is variously realized in Syria as [k], [tʃ], [ʃ], sometimes with phonological variation between a base variant (usually but not always [k]) and conditioned variant. Neither this database, nor the source that it is based upon (Behnstedt, 1997, map 15) specify the exact conditioning environment. To encode



Word	Language	Country	Area	Adult speech	English	PoS
daddaš	Arabic	Algeria	Dellys	dəddəš	toddle	Interjection, Verb (im...
nənni	Arabic	Algeria	Dellys	ərqŭd, ərgŭd	sleep	Verb
baεεa, baεbaε	Arabic	Algeria	Dellys	kəbš	sheep, sheepie	Noun

Abbildung 5: A screenshot of a typical entry from the babytalk website (http://babytalk.barefootlinguist.com/)

this data, each modern realization of the proto-phoneme /*k/ (i.e. `k`, tʃ, etc) would be stored in the `normalizedEntry` field. The gloss field would simply read `phoneme` for the sake of this example. Each phoneme would be tagged with `reflexof.*k`. Conditioned variants would also be coded as datums, glossed as `allophones`, and linked to the base phoneme with a `variant.conditioned` relationship. The standard in DAD has been to express the conditioning environment of a datum with a tag, so the allophones would be tagged with `conditioning.unknown`. With this model, it should be straightforward to search for the realizations of this proto-phoneme.

5 Limitations of the model

The primary limitation of this database model is related to its flexibility. It has no inherent controls on the tagging or relational systems, and much of the validation of integrity and consistency must be performed at the level of the software. Prior to data entry, the project designers and stakeholders should design their ontology of tags and relations, and the software can be developed accordingly. The system does easily allow for growth or new requirements, since tags can easily be added and modified.

Unlike the RDF model used by Good and Hendryx-Parker (2006), the relational model here cannot easily treat relationships as objects that can themselves be parts of relationships, i.e. 'reification' (Good and Hendryx-Parker, 2006, pp. 18–20, illustrate this shortcoming of relational databases at length). For the purposes of their project, this could be a fatal flaw, but for projects such as DAD, there is no real need for such complex relational information. Any amount of potentially contradictory information can be stored both in the tags for datums and in relationships and their tags, but as that information gets more complex it would be better to modify the database structure.

Literatur

Behnstedt, P. (1997). *Sprachatlas von Syrien.* Harrassowitz, Weisbaden.

Behnstedt, P. and Woidich, M. (2010). *Wortatlas der arabischen Dialekte: Mensch, Natur, Fauna und Flora*, volume 1. Brill, Leiden.

Diessel, H. (1999). *Demonstratives : form, function, and grammaticalization.* John Benjamins Publishing Company, Amsterdam.

Dimitriadis, A. (2006). An extensible database design for cross-linguistic research. http://languagelink.let.uu.nl/burs/docs/burs-design.pdf.

Farrar, S. (2006). A universal data model for linguistic annotation tools. In *Proceedings of the EMELD'06 Workshop on Digital Language Documentation: Tools and Standards: The State of the Art*, Lansing, Michigan.

Forkel, R. (2014). The cross-linguistic linked data project. In *LREC 2014 (The International Conference on Language Resources and Evaluation)*, pages 60–66.

Good, J. and Hendryx-Parker, C. (2006). Modeling contested categorization in linguistic databases. In *Proceedings of the EMELD '06 Workshop on Digital Language Documentation: Tools and Standards: The State of the Art*, Lansing, Michigan.

Kent, W. (1983). A simple guide to five normal forms in relational database theory. *Communications of the ACM*, 26(2):120–125.

Lewis, W. D., Farrar, S., and Langendoen, D. T. (2006). Linguistics in the internet age: Tools and fair use. In *Proceedings of the EMELD'06 Workshop on Digital Language Documentation: Tools and Standards: The State of the Art*, Lansing, Michigan.

Penton, D., Bow, C., Bird, S., and Hughes, B. (2004). Towards a general model for linguistic paradigms. In *Proceedings of the EMELD'04 workshop on databases and best practice.*