Description of Search Interface

Dennis Shasha and Chris Collins
June 1, 2009

The SSWL search functionality allows users to ask about the structure of particular languages or of sets of languages. The search interface allows one to take advantage of the relational database structure of the database in ways that are unusual for linguistics databases. Furthermore, the search interface is designed to be accessible to linguists who do not know SQL and who are unfamiliar with database technology.

The information in the database is stored in three main tables: a table for property definitions that is used for reference, a table for language information, and a table of examples. For those familiar with relational databases, we explain the semantics and pragmatics of these tables in the formal description section of the document.

I. Informal Description

Show

On the Show line, users ask to see some combination of Language, Property, Property_Value or Example. For example, clicking on Language and Example in the Show line will produce a listing of all the languages in the database and the examples that have been entered for each of those languages.

Constraints

Users may constrain the set of results from the Show Line using constraints defined in terms of languages, properties, property-values, and/or examples. All these constraint options are found in the boxes under the Show line. For instance, one might ask for all property-values of Dagaare, by clicking on Property_Value in the Show line (meaning: the results will be a list of property-value pairs), and then selecting Dagaare from the list of languages in the language constraint box (to the right of the words “Select Language”).

Combining Constraints across Boxes

It is possible to specify several different kinds of constraints (from different boxes) in one search. For example, clicking on Language, Property_Value and Example in the Show line, then clicking on Dutch and English in the language constraint box, and then on Adjective Degree, Adjective Demonstrative Noun, Adjective Noun, Adjective Noun Demonstrative in the property constraint box yields search results pertaining to Dutch or English having to do with at least one of the four properties mentioned as well as any examples associated with those results.

Intuitively, the semantics of this query are to take language, property-value, and examples triples pertaining to English or Dutch and intersect them with the language, property-value, and examples triples pertaining to any of the four properties specified.

Any and All

Two particularly useful functions for linguists are the Any and All functions found just under the property constraint box and just under the property-value constraint box. In logical terms, "Any" means disjunction (logical OR) within a constraint. By contrast, "All" means conjunction (logical AND) within a constraint. These functions are available both for properties and property-values. The All option is not available for the language constraint box. Selecting several languages is implicitly handled as an Any.

For example, suppose a user clicks on Language in the Show line and selects the following property-value pairs (in the constraint box): Attributive Adjective Agreement:Yes and Auxiliary Selection:Yes. If All is specified, this search will find the set of languages that have agreement with attributive adjectives AND for which the property of auxiliary selection holds. If one had clicked Any instead, then the search would yield the set of languages that have agreement with attributive adjectives OR for which the property of auxiliary selection holds (or both).

Cross

The "Cross" function allows a comparison among a pair of properties on all or a subset of languages. The essential function of Cross is to form tables that are similar to the tetrachoric tables of Greenberg 1963. For example, a cross among Adjective Degree and Adjective Demonstrative Noun, yields the counts and the languages for each combination of Adjective Degree:Yes/No/NA and Adjective Demonstrative Noun:Yes/No/NA. It is also possible to constrain Cross to a particular set of languages (using the language constraint box).

Examples

In the database, an example is represented as a sentence in a target language (with morpheme boundaries indicated), a gloss for that sentence, and a translation into English as well as the property-value pairs that the example illustrates. It is possible to constrain example searches based on the words and morphemes in the sentence, gloss or translation. For example, if the Show line has Language, Property and Example clicked where language is constrained to French and English and Gloss Contains is set to "1sg", then the search will return all properties having to do with French and English as well as examples of those properties whose gloss contains “1sg”.

If Example is clicked on the Show line, examples will be lined up with whatever else is clicked on the Show line (e.g., if Language and Property are also clicked on the Show line, then examples will be returned for every combination of language and property for which examples exist).

If the "prioritize example" box is clicked, then only information (in the above example, languages and properties) having a corresponding example will be shown.

Summary

A search yields results that are taken from the languages table as specified by the fields clicked on the Show line. These search results are then constrained independently depending on the constraints in the languages, properties, and/or property-value boxes. Lastly, the intersection is found of those independently constrained sets of results.

If Example is also clicked on the Show line, then examples are provided for each row produced by the above search (for which examples exist). If prioritize example is clicked at the bottom of the page, then the results that are retained are only those associated with an example.

II. Formal Description

In the above section, we gave an informal description of the system meant for linguists' eyes. The following description is a combined set theoretic and procedural explanation corresponding to our implementation.

Relational Tables

The two tables most important to search have the following main fields:

languages(languagename, propertyname, value, contributorname, date, time)

In this table, languagname and propertyname together constitute a key. In the rest of this description, we often treat propertyname-value as a single concatenated field.

Because propertyname is a field, the set of properties is open-ended and the number of rows associated with a language may differ from language to language. Thus a single language is represented by several rows in this table. For example, French has rows corresponding to Adjective Degree:No, Adposition Noun_Phrase:Yes and so on. The fact that propertyname is a field is what we mean when we say that SSWL follows a property-as-value design philosophy.

examples(languagename, sentenceid, type, propertyname, value, contributorname, date, time)

In this table, languagename, sentenceid, and propertyname together constitute a key.

For each language, there may be several examples. The information about an example is held in all the records having a particular sentenceid. So for example, French sentenceid (sentenceid is called Example Number in the system interface) 6 has a translation value of "Jean arrived" and an Auxiliary Selection value of Yes, among other properties. The full set of properties used to characterize an example are: sentence, gloss, translation, comments, and the set of properties which represent the grammatical information that the example illustrates.

Definitions:

allclickedfields = fields clicked in Show line

clickedfields = allclickedfields except Example

anyconstraint(f, c) = a constraint that is produced by clicking an Any under the box corresponding to field f (in this case f is either Language, Property, or Property-Value) and a set of selected items c within the box. The result of this constraint, i.e., the semantics of the constraint, will be described below. As mentioned before, "any" is implicit for languages. Note also that if a given box has no item selected, then c consists of every item in the box (i.e. selecting nothing means every value is allowed) and "any" is implicit.

allconstraint(f, c) = a constraint that is produced by clicking an All under a box corresponding to field f and a set of selected items c within the box.

exampleconstraint(f, t) = a constraint that is produced by specifying field name f as either gloss, translation, or sentence (depending on one of the choices gloss contains, translation contains, or sentence contains), and text t.

Notation

In the sequel, the notation $\in$ refers to membership as in $5 \in \{4, 5, 6\}$. The symbol "&" is the Boolean and. The symbol "^" denotes set intersection. The notation $\{r.F \mid r \in R \ \& \ C(r)\}$ is a set former consisting of the set of rows of relation R projected onto fields F where each row r satisfies some constraint C(r). If r is a record in a table, and f is a field of that record, then r.f is the value of that field for that record.

Phase 1: Individual Constraints

anyconstraintresult(f,c)
= result of applying anyconstraint(f,c) to clickfields
= {values of clickedfields of the rows in the languages table whose f value $\in$ c}
= { <r.clickfields> | r $\in$ languages & r.f $\in$ c}

Example: if clickedfields = {language, property} (clicked in the Show line)
f = language (Select Language is the constraint box)
and c = {French, German} (two languages selected in the language constraint box)
with Any (implicitly) clicked then:

anyconstraintresult(language,c)
= {<r.language, r.property> | r ∈ languages & r.language ∈ {French, German}}
= the set of language, property pairs where the language is either French or German.

allconstraintresult(f,c)
= result of applying allconstraint(f, c) to clickfields
= intersection of the sets of clickfield tuple of the rows in the languages table that satisfy each member of c
= Denote the members of c as x1, …, xn. Construct Mi as {<r.clickfields> | r ∈ languages & r.f = xi}. The result is the intersection M1 ^ … ^ Mn.

Example: if clickedfields = {language} (clicked in the Show line)
f = property (Select Property is the constraint box)
and c = {Adjective Noun, Subject Verb} (two properties selected in box)
with All clicked then:
allconstraintresult(property,c)
= {<r. language> | r ∈ languages & r.property = Adjective Noun} ^
{<r. language> | r ∈ languages & r.property = Subject Verb}
= the set of languages for which the property Adjective Noun is specified (in this case as either Yes, No, or NA) and the property Subject Verb is also specified.

exampleconstraintresult(f,t)
= result of applying examplecontraint(f,t) to the examples table
= { language, sentenceid pairs in examples whose field f contains text t }
= { <e.language, e.sentenceid> | e ∈ examples & e.f contains text t}

Example: exampleconstraintresult(gloss, 1sg) finds all language, sentenceid pairs containing "1sg" in the value of gloss. That is, { <e.language, e.sentenceid> | e ∈ examples & e.gloss contains text "1sg"}

phase1resultbasic = intersection of the clickedfield values of all anyconstraintresults and allconstraintresults. Formally, if the query asks for anyconstraint(f1, c1), … anyconstraint(fn, cn) and allconstraint(h1, d1) ,,, allconstraint(hm, dm) then phase1resultbasic = anyconstraintresult(f1,c1) ^ … ^ anyconstraintresult(fn, cn) ^ allconstraintresult(h1, d1) ^ … ^ allconstraintresult(hm, dm)

In other words, if a search involves several boxes, the search is run for each constraint box independently, and then the intersection is found for the results for each box. So we have now formalized "combining constraints across boxes" from part I of this paper.

Example: if the only clicked field is language, and the language constraint box has French and German selected (with Any implicitly clicked), and the property constraint box has Adjective Noun and Subject Verb selected with All clicked, then:

Anyconstraintresult(language, c1)
= {<r.language> | r ∈ languages & r.language ∈ {French, German}}

Allconstraintresult(property,c2)
= {<r.language> | r ∈ languages & r.property = Adjective Noun} ^
{<r.language> | r ∈ languages & r.property = Subject Verb}

For this search, phase1resultbasic is the intersection of these two sets of languages.
That is:
phase1resultbasic = Anyconstraintresult(language,c1) ^ Allconstraintresult(property, c2).

If "prioritize examples" is not clicked then phase1result = phase1resultbasic. Otherwise, (if "prioritize examples" is clicked) then phase1result = phase1resultbasic intersected with the language-property-value results corresponding to the example rows whose language-sentenceid pairs are contained in exampleconstraintresult(f, t).

That is:
phase1result = phase1resultbasic ^ {<e.clickedfields> | e ∈ examples & <e.language, e.sentenceid> ∈ exampleconstraintresult(f,t)} }

In other words, if prioritize example is clicked, then only those rows that correspond to an existing example are displayed.

Phase 2: Example Constraints

Phase 2 applies only if Example is clicked in the Show line or if the text box near "gloss contains" is filled in or both.

There are three sources of information constraining examples found in a search. First, the user selects constraints from the constraint boxes, those constraints apply to the examples. Second, the only examples shown are ones corresponding to results from Phase 1. The third constraint is the Text Search at the bottom of the query interface. Therefore, the set of examples found will be the intersection of these three sets. Formally, this is described below.

A1: A1 is the set of examples in the database satisfying the constraints in the three constraint boxes, but using an "any" semantics (this "any" decision was taken for pragmatic reasons: whereas a language might have many properties defined, an example usually has only a few). That is, A1 consists of those language, sentenceid pairs correspond to languages selected in the language constraint box (as above, if none is selected then the semantics are that every one is selected) intersected with language, sentenceid pairs having a property among those selected in the property constraint box intersected with language, sentenceid pairs having a property-value pair among those selected in the property-value constraint box. Formally, if c1 is the set of selected languages, c2 is the set of selected properties, and c3 is the set of selected property-values, then A1 = {<e.language, e.sentenceid> | e ∈ examples & e.language ∈ c1} ^ {<e.language, e.sentenceid> | e ∈ examples & e.property ∈ c2} ^ {<e.language, e.sentenceid> | e ∈ examples & e.property-value ∈ c3}.

A2: A2 is the set of examples in the database corresponding to the contents of phase1resultbasic. For example, if only language is clicked, and phase1resultbasic contains French and German, then this would fetch all language, sentenceid pairs corresponding to French and German. Formally, A2 = {<e.language, e.sentenceid> | e ∈ examples & e.clickedfields ∈ phase1resultbasic}

A3: If the text box next to gloss contained is filled in, then find the set of language, sentenceid pairs that satisfy the gloss contains value. That is, if we have exampleconstraint(f,t), then A3 = examplecontraintresults(f,t). Otherwise, A3 = {<e.language, e.sentenceid> | e ∈ examples}

A4: constrainedexamples = A1 ^ A2 ^ A3.

A5: Show the sentence, gloss, and translations corresponding to constrained examples for each member of constrainedexamples.

Example: Suppose language, property and example are clicked in the Show line, Adjective Agreement is selected in the property constraint box and gloss contains “1sg” is filled in, and “prioritize example” is clicked.

Since there is a constraint only on property in the languages table, phase1resultbasic = { <r.language, r.property> | r ∈ languages & r.property = Adjective Agreement}. Further exampleconstraintresults(gloss, “1sg”) = {<e.language, e.sentenceid> | e.gloss contains the substring “1sg” }. Since “prioritize examples” is clicked, phase1result = phase1resultbasic ^ {<e.language, e.property> | <e.language, e.sentenceid> ∈ exampleconstraintresults(gloss, “1sg”) }