MetaMap Lexicon Tables

January 10, 2013

François-Michel Lang

1 Background

MetaMap has for some time been working on converting the lexicon-access code from the legacy 'C' code to the new Java-based lexicon. Recently, thanks largely to Willie's help, we completed an initial implementation in which the Java code was called directly by MetaMap; unfortunately, the Java VM startup caused too great a performance slowdown, so Willie delivered the equivalent functionality as a standalone server, but the expected speedup did not materialize.

In order to increase the efficiency of lexical access, we would like to try encapsulating all needed lexicon information in Berkeley-DB tables to be called via Jim's 'C' code. The remainder of this document describes the tables that we expect to need.

2 Current Prolog Predicates Calling Java Code

The lexicon server handles calls from the following eleven Prolog predicates (the four starred (***) predicates are currently not used, but could be resurrected):

1. `lexAccess_is_root_form` ***
   input: any token
   output: 1/0 depending on success or failure

2. `lexAccess_is_root_form_with_cats` ***
   input1: any token
   input2: list of lexical categories
   output: 1/0 depending on success or failure

3. `lexAccess_is_form` ***
   input: any token
   output: 1/0 depending on success or failure

4. `lexAccess_get_lex_form_cats`
   input: any token
   output: list of lexical categories

5. `lexAccess_get_base_forms_for_form`
   input: any token
   output: list of base forms of the token or `[]` if no base form is found for the token

6. `lexAccess_get_base_forms_for_form_with_cats`
   input1: any token
   input1: list of lexical categories
   output: list of base forms of the token of one of specified lexical categories,
   or `[]` if no base form is found for the token

7. `lexAccess_get_varlist_for_form`
   input: any token
   output: list of variant, lexical category, and feature, e.g.,
   `[[[hearts,noun,plural],[heart,noun,base]]]`

8. `lexAccess_get_varlist_for_base_form` ***
   input: any token
   output: list of inflectional variants, lexical categories, and features, e.g.,
   `[[[hearts,noun,plural],[heart,noun,base]]]` if input is a base form, and `[]` otherwise

9. `lexAccess_get_dm_variants_by_category`
   input: any token
   output: list of derivational variants and lexical categories e.g.,
   `[[pseudoheart,noun],['pseudo-heart',noun],[intraheart,adj],`
   `['intra-heart',adj],[hearted,adj]]`

10. `lexAccess_get_lexical_records`
    input: an EUI
    output: list of lexical records, e.g.,
    `['{base=heart\nentry=E0030957\n\tcat=noun\n\tvariants=uncount\n\tvariants=reg\n}\n']`

11. `lexAccess_find_prefix` [not relevant here]

The last of these (`lexAccess_find_prefix`) is not directly amenable to a database treatment, so it will not be covered further. We believe, however, that the rest of the logic can be encapsulated in five tables, which we now present.

3 Citation Form Table

The first and simplest table will be used by

- `lexAccess_is_root_form` and
- `lexAccess_is_root_form_with_cats`

and should have the schema

Base Form | Lexical Category

e.g.,

```
heart|noun
attack|noun
attack|verb
```

A row is necessary for every citation form in the ASCII lexicon. This table could be constructed by running a Perl script over the ASCII lexicon flat file, but once we migrate to lexAccess, we will no longer need that file, so that approach will not work.

The table would be used to

- verify if a given token is a citation form (of any lexical category) and
- verify if a given token is a citation form of a specific lexical category.

4 Form Table

The next-simplest table will be used by

- `lexAccess_is_form`,
- `lexAccess_get_lex_form_cats`,
- `lexAccess_get_base_forms_for_form`, and
- `lexAccess_get_base_forms_for_form_with_cats`

and should have the schema
Inflectional Variant | Lexical Category | Base Form
e.g.,

```
heartache|noun|heartache
hearted|adj|hearted
heartened|verb|hearten
heartening|verb|hearten
heartens|verb|hearten
hearten|verb|hearten
hearten|verb|heartening
heartfelt|adj|heartfelt
heartful|adj|heartful
hearths|noun|hearth
hearth|noun|hearth
heartier|adj|hearty
heartiest|adj|hearty
heartily|adv|heartily
heartless|adj|heartless
heartsick|adj|heartsick
hearts|noun|heart
hearty|adj|hearty
heart|noun|heart
```

A row is necessary for every inflectional variant of every base form, and every inflectional variant must map to all its base forms. The table will be used to

- verify if a given token is a form (of any kind),
- obtain the lexical categories of a given form, and
- obtain the base forms of a given form.

5 Inflectional Variants Table

The inflectional variants table will be used by

- `lexAccess_get_varlist_for_form` and
- `lexAccess_get_varlist_for_base_form`

and should have the schema

Citation Form | Inflectional Variant | Infl Lexical Category | Feature

e.g.,

```
heart|hearts|noun|plural
heart|heart|noun|base
attack|attacks|verb|present
attack|attacks|noun|plural
attack|attacking|verb|ing
attack|attacked|verb|pastpart
attack|attacked|verb|past
attack|attack|verb|present
attack|attack|verb|base
attack|attack|noun|base
```

A row is necessary for every inflectional variant of every base form. The table will be used to generate the inflectional variants, lexical categories, and features of a given base form. The predicate `lexAccess_get_varlist_for_base_form` will use the table by first obtaining the base form of the specified form, and then using the base form to index into the table.

6 Derivational Variants Table

The derivational variants table will be used by `lexAccess_get_dm_variants_by_category` and should have the schema

Base Form | Base Lexical Category | Derivational Variant | Derivational Variant Lexical Category

e.g.,

```
heart|noun|pseudoheart|noun
heart|noun|pseudo-heart|noun
heart|noun|intraheart|adj
heart|noun|intra-heart|adj
heart|noun|hearted|adj
attack|noun|reattack|noun
attack|noun|re-attack|noun
attack|noun|preattack|noun
attack|noun|preattack|adj
attack|noun|pre-attack|noun
attack|noun|pre-attack|adj
attack|noun|postattack|adj
attack|noun|post-attack|adj
attack|noun|interattack|adj
attack|noun|inter-attack|adj
attack|noun|counterattack|verb
```

```
attack|noun|counterattack|noun
attack|noun|counter-attack|verb
attack|noun|counter-attack|noun
attack|noun|counter attack|noun
attack|noun|attacker|noun
attack|noun|attackable|adj
attack|noun|attack|verb
attack|noun|attack|noun
```

A row is necessary for every lexical category of every derivational variant of every base form. Moreover, only derivational variants that are themselves base forms or spelling variants should appear in this table.

This table will be used to obtain the derivational variants and their lexical categories for a given base form or spelling variant.

7 Lexical Record Table

The last table will be used by `lexAccess_get_lexical_records` and should have the schema

EUI | Lexical Record

e.g.,

```
E0431577|{base=heart attack\nentry=E0431577\n\tcat=noun\n\tvariants=reg\n}\n
E0011097|{base=attack\nentry=E0011097\n\tcat=verb\n\tvariants=reg\n\tintran\n\ttran=np\n}\n
```

Note the appearance of the newline (`\n`) and tab (`\t`) characters!

A row is necessary for every citation form in the ASCII lexicon. This table could also be constructed by running a Perl script over the ASCII lexicon flat file, but, as before, that approach is not a long-term solution.

This table will be used to obtain the lexical record(s) for a given EUI.

8 Implementation

With every release of the SPECIALIST lexicon, which corresponds to MetaMap's AA data migration, we will need automated and reproducible scripts to generate these five tables.