COR: Corpus Linquistics

Lecture 12
Review of Corpus Linguistics

Francis Bond
Department of Asian Studies
Palacký University
https://fcbond.github.io/
bond@ieee.org

https://github.com/bond-lab/Corpus-Linguistics

COR (2024)

Overview

- Markup and Annotation
- Using Corpora: Regular Expressions
- Multimodal and Multilingual Corpora
- Collocation, Frequency, Corpus Statistics
- DIY Corpora, Corpus Tools, Processing Raw Text
- Case studies: Lexical, Grammatical, Contrastive, Diachronic
- Corpora and Language Engineering
- Representativeness and Balance
- Copyright and Licensing

More SQL
Creating; Inserting; Updating and Deleting

How to create a Table

```
CREATE TABLE database_name.table_name(
   column1 datatype  PRIMARY KEY(one or more columns),
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype,
);
```

Each column should have a **datatype**

TEXT	A text string, stored using the database encoding
INTEGER	Signed integer (or INT)
REAL	Floating point number
CHAR(N)	String of N characters padded with spaces
VARCHAR(N)	String of N characters

`sqlite` is very forgiving, you can store any data type in any column.

There are more types, if you must know, google them

For example: thw word table

```
CREATE TABLE word (
-- store words, with POS and lemma
-- start and end in the corresponding sentence (cfrom, cto)
       sid INTEGER,     -- sentence ID
       wid INTEGER,     -- wid (should be consecutive)
       word TEXT,       -- surface form of the word
       pos TEXT,        -- part of speech
       lemma TEXT,      -- lemma (true-cased)
       cfrom INTEGER,   -- start position
       cto INTEGER,     -- end position
       comment TEXT,
       PRIMARY KEY (sid, wid),
       FOREIGN KEY(sid) REFERENCES sent(sid)
       );
```

PRIMARY KEYS

- The PRIMARY KEY constraint uniquely identifies each record in a database table.
- Primary keys must contain UNIQUE values.
- A primary key column cannot contain NULL values.
- Each table can have only ONE primary key.
- Most tables should have a primary key

If you don't define one, nothing bad happens

You can show it with `.tables` or `.schema`

```
sqlite>.tables
sent word concept ...

sqlite>.schema word
CREATE TABLE word (
sid INTEGER,
wid INTEGER,
word TEXT,
pos TEXT,
lemma TEXT,
cfrom INTEGER,
cto INTEGER,
comment TEXT,
        PRIMARY KEY (sid, wid),
        FOREIGN KEY(sid) REFERENCES sent(sid) );
```

Inserting Information

```
INSERT INTO word (sid, wid, word, pos, lemma)
  VALUES (1, 0, "The", "DT", "the");
INSERT INTO word (sid, wid, word, pos, lemma)
  VALUES (1, 1, "Adventure", "NNS", "ADVENTURE");
INSERT INTO word (sid, wid, word, pos, lemma)
  VALUES (1, 2, "of", "PP", "of");
```

Upating Information

```
UPDATE word SET lemma='adventure'
  WHERE sid=1 AND wid=1;
```

or

```
UPDATE word SET lemma='adventure'
  WHERE lemma='ADVENTURE';
```

Everything that matches the condition gets updated

Best to check with a SELECT first:

```
SELECT * FROM word
  WHERE lemma='ADVENTURE';
```

Deleting Information

Be very, very careful:

```
DELETE FROM table_name
  WHERE [condition];
```

Dates and times

Time String	Example
YYYY-MM-DD	2010-12-30
YYYY-MM-DD HH:MM	2010-12-30 12:10
YYYY-MM-DD HH:MM:SS.SSS	2010-12-30 12:10:04.100
MM-DD-YYYY HH:MM	30-12-2010 12:10
HH:MM	12:10
YYYY-MM-DDTHH:MM	2010-12-30 12:10
HH:MM:SS	12:10:01
now	2015-04-15

```
sqlite> SELECT date('now');
2015-04-15
sqlite> SELECT date('now', '+1 months');
2015-05-15
sqlite> SELECT date('now', 'start of month');
2015-05-01
```

And much more: http://www.tutorialspoint.com/sqlite/sqlite_date_time.htm

Task

- Create a new table in your database
- Add three entries
- Update two
- Delete one

Make a bigram Table

```
create TABLE bigram (sid INT, wid INT, bigram TEXT);

INSERT INTO bigram (sid, wid, bigram)
  SELECT a.sid, a.wid, a.lemma || ' ' || b.lemma
    FROM word AS a JOIN word AS b
    ON a.sid=b.sid AND a.wid = b.wid-1
  LIMIT 5;
```

The result:

```
sqlite> SELECT sid, wid, bigram FROM bigram;
60000 0 prime minister_tomiichi_murayama
60000 1 minister_tomiichi_murayama on
60000 2 on the
60000 3 the 28
60000 4 28 hold
```

Trading SPACE for TIME

- Storing bigrams makes the DB bigger
- But you can manipulate them quickly
- For large tables, you can also **INDEX** them

  ```
  CREATE INDEX word_idx
  on word (lemma, pos);
  ```

- This allows you to query `word` or `word+pos` much faster
- Use indexes for big tables you search often but don't update much
- Indexes can double the size of your database
  - But speed big searches up from hours to seconds

There are whole courses on this

Batch Import

- You can input well formatted data using `sqliteman` or similar
  - define the column separator ':' or '|' or TAB or ',' or ...
  - or load from spreadsheet
- Or through some program
  - Learn more in HG2051 *Language and the Computer*

Revision

The goal of this course

Master the uses of text corpora
in linguistics research and applications.

- Selecting text
- Marking up extra information
- The range of existing corpora
- How to build your own corpus
- Using corpora to test linguistic hypotheses
- Using corpora to train language tools

What did you learn?

You should be able to:

- Understand the uses of text corpora in language research
  Be able to manipulate them with simple tools
- Use a concordance program to extract data from a corpus
- Design and build a corpus for some task
  - considering representativeness, balance and legal issues
  - as well as usability and accuracy
- Understand how to analyse corpus data through basic statistical methods
- Understand the issues involved in using data for NLP

Reflection

- What was the most surprising thing in this class?
- What do you think is most likely wrong?
- What do you think is the coolest result/corpus?
- What do you think you're most likely to remember?
- How do you think this course will influence you as a linguist/specialist?
- What (if anything) did you hope to learn that you didn't?