Informatics 1B, 2008

School of Informatics, University of Edinburgh

Data and Analysis

Note 10

Querying a Corpus

Alex Simpson

Part II — Semistructured Data

XML

Corpora

Topics

- how to do something useful with corpus data and its annotation;
- how to extract statistics that are useful for linguistic questions or NLP applications;
- how to use regular expressions for queries, obtain concordances, extract collocations from corpora.

Concordances

Concordance: all occurrences of a given word, displayed in context.

More generally, one looks for all occurrences of matches for a given query expression.

- generated by concordance programs based on a user keyword;
- keyword (search query) can specify word, annotation (POS, etc.) or more complex information (e.g.,using regular expressions);
- output displayed as keyword in context: matched keyword in the middle of the line, predefined context to left and right.

Example

A concordance for all forms of the word *"remember"* in the Dickens corpus (used in tutorial 6).

```
 's cellar . Scrooge then <remembered> to have heard that ghost
, for your own sake , you <remember> what has passed between
e-quarters more , when he <remembered> , on a sudden , that the
corroborated everything , <remembered> everything , enjoyed eve
urned from them , that he <remembered> the Ghost , and became c
ht be pleasant to them to <remember> upon Christmas Day , who
its festivities ; and had <remembered> those he cared for at a
wn that they delighted to <remember> him . It was a great sur
ke ceased to vibrate , he <remembered> the prediction of old Ja
as present myself , and I <remember> to have felt quite uncom
```

Concordance programs

Concordances are generated automatically by concordance programs, such as the *Corpus Query Processor (CQP)* used in tutorial 6.

CQP s query engine searches corpora based on user queries over words, parts of speech, or other markup.

Regular expressions make the CQP's query language powerful.

N.B. This is the second time we have found an application for regular expressions in Data & Analysis.

CQP syntax for regular expressions

CQP makes use of the following format for regular expressions.

- **`exp1 exp2`** : first **`exp1`** then **`exp2`** in sequence.
- **`exp*`** : zero or more occurrences of exp.
- **`exp?`** : zero or one occurrences of exp.
- **`exp+`** : one or more occurrences of exp.
- **`exp1|exp2`** : either exp1 or exp2.

Question: What is the one difference here from the regular expression syntax used in DTD's (see slide 6.21)?

Example CQP query

The query:

- `[word="remember|remembers|remembered|remembering"];`

Returns all forms of the word “remember”, as on slide 10.5.

Here **`word`** is a *positional attribute* looking for tokens that have been marked up as words.

The value of the attribute is matched against the right-hand side of the query (here: all forms of remember).

N.B., In this case the right-hand side is a (very simple) regular expression.

Other operators

CQP offers additional regular expression operators.

The *dot operator* matches any character, e.g.

- `[word="s.ng"];`

matches **sing**, **sang**, **sung**, but also **song**, **szng**, **s6ng** etc.

The *list operator* `[...]` matches all characters in the list, e.g.

- `[word="s[iau]ng"];`

Abbreviations for subsets are allowed, e.g., `[a-d]` or `[1-6]`.

POS information and boolean expressions

The positional attribute **word** is available (in one form or other) in every corpus.

Most corpora contain additional annotation, e.g., part of speech information. In CQP this is given by the attribute **pos**.

- `[pos="NN.*"];`

This returns all nouns: **NN.*** matches **NN** for regular nouns, **NNP** and **NNPS** for singular and plural proper nouns, etc.

Regexes can be combined using Boolean operators **&** (and), **|** (or), and **!** (not):

- `[(word="like.*") & (pos!="NN.*")];`

returns all words starting with “like” not tagged as noun.

Sequences

Queries can refer to *sequences of words*

- **`[pos="JJ.*"] [word="tea"];`**

matches all instances of the word "tea" preceded by an adjective (i.e. a word with pos value **`JJ`**).

```
now , notwithstanding the <hot tea> they had given me before
.' ' Shall I put a little <more tea> in the pot afore I go ,
o moisten a box-full with <cold tea> , stir it up on a piece
tween eating , drinking , <hot tea> , devilled grill , muffi
e , handed round a little <stronger tea> . The harp was there ; t
e so repentant over their <early tea> , at home , that by eigh
rs. Sparsit took a little <more tea> ; and , as she bent her
s illness ! Dry toast and <warm tea> offered him every night
of robing , after which , <strong tea> and brandy were administ
rsty . You may give him a <little tea> , ma'am , and some dry t
```

Collocations

Collocation: a sequence of words that occurs 'atypically often' in language usage

Examples:

- *run amok:* the verb "run" can occur on its own, but "amok" can't.
- *strong tea:* sounds much better than "powerful tea" although the literal meanings are much the same.
- Phrasal verbs such as *make up* or *make off* or *make out* (but not, for example, "make in").
- *rancid butter*, *bitter sweet*, *over and above*, etc.

N.B. The inverted commas around 'atypically often' are because we shall eventually need statistical ideas to make this precise.

Identifying collocations

Task: automatically identify collocations in a large corpus.

For example collocations with the word *tea* (see 10.11).

- *strong tea* occurs in the corpus.

  This is a collocation.

- *powerful tea*, in fact, does not.

- However, *more tea* and *little tea* also occur in the corpus.

  These are not collocations. These word sequences do not occur with an *atypically* common frequency.

Problem: How do we detect when a bigram (or n-gram) is a collocation?

Finding bigrams in CQP

Use CQP to compute *bigram frequencies* for all words that occur with *strong* and *powerful*.

- ```
  Q1 = [word="strong"] [];
  Q2 = [word="powerful"] [];
  ```

Use the **group** command to obtain frequencies:

- ```
  group Q1 matchend word by match word;
  group Q2 matchend word by match word;
  ```

This groups together the values of word at the position **matchend** (the end of the matching sequence) and sorts result by word at position **match** (number of matches).

strong	,	52	powerful	,	5
	and	31		effect	3
	enough	16		sight	3
	.	16		enough	3
	in	15		mind	3
	man	14		for	3
	emphasis	11		and	3
	desire	10		with	3
	upon	10		enchanter	2
	interest	8		displeasure	2
	a	8		motives	2
	as	8		impulse	2
	inclination	7		struggle	2
	tide	7		grasp	2
	beer	7		friends	2

Filtering collocations

The bigram table shows:

- Neither *strong tea* nor *powerful tea* are frequent enough to make it into the top 15.
- Potential collocations for *strong*: e.g., *strong desire*, *strong inclination*, and *strong beer*;
- Potential collocations for *powerful*: e.g., *powerful effect*, *powerful motives*, and *powerful struggle*;
- Problem: The bigrams *strong and*, *strong enough*, *powerful for*, are highly frequent. These are not collocations.
- To distinguish collocations from non-collocations, we need to filter out 'noise'.

The need for statistics

Problem: Words like *for* and *and* are highly frequent on their own: they occur with *tea* by chance.

Solution: use statistical testing to detect when the frequency of a bigram is atypically high given the frequencies of its constituant words.

In general, statistical tools offer powerful methods for the analysis of all types of data. In particular, they provide the principal approach to the quantitative (and qualitative) analysis of *unstructured data*.

This leads us into the final section of the course, where we shall return to statistics and to collocations.

Coursework assignment

Today (28th February), the Data & Analysis coursework assignment is available from the course homepage:

`http://www.inf.ed.ac.uk/teaching/courses/inf1/da/`

Because of the coursework assignment, there will be no Data & Analysis tutorial exercise for next week.

Instead, the purpose of the week 9 DA tutorials is to provide assistance with the coursework assignment, and discussion related to it.

You are strongly advised to begin work on the coursework assignment *as early as possible*, so that the tutorial can provide you with the maximal possible assistance.

The questions on the coursework assignment are very similar in nature to those on the exam.