Inf1B Data and Analysis
Tutorial 6 (week 8)

25 Feb 2009

- Please answer all questions on this worksheet in advance of the tutorial, and bring with you all work, including printouts of code and other results. Tutorials cannot function properly unless you do the work in advance.
- Data & Analysis tutorial exercises are not assessed, but are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Attendance at tutorials is obligatory; please let your tutor know if you cannot attend.
- *Recommended Reading: Chapter 2 of* `Corpus Linguistics` *by T. McEnery and A. Wilson until the end of section 2.2.2*

Introduction

This tutorial is largely based on the CQP tutorial developed by Stephan Evert available at
`http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQPTutorial/cqp-tutorial.pdf`

Setting up the working environment

This tutorial needs to be carried out on a DICE machine in the Informatics computer labs. The first thing you need to do is to tell your system where to locate the *corpus* you will be using today (which consists of a number of stories written by Charles Dickens). In order to do this, please type the following command on a terminal window:

```
> export CORPUS_REGISTRY=/group/corpora/public/corpus_workbench/registry
```

Now you can get `cqp` started by typing '`cqp -e`'. You will then see the following in your command prompt

```
[no corpus] >
```

This means that `cqp` is active but no specific *corpus* has been selected yet. In order to select the `DICKENS` corpus type

```
[no corpus] > DICKENS;
```

You will then see the following:

```
DICKENS >
```

From this point you can get some more specific information about this particular corpus by typing:

```
DICKENS > info;
```

Press the `space bar` to display the next page and `q` to get back to the command prompt. Furthermore, typing `exit;` will allow you to quit the `cqp`-program. Please take some time to read this information and make some notes of what you think is relevant and/or related to anything you have seen in the lectures so far.

1 Searching for words

Now that the working environment has been set, we can start performing some queries on the DICKENS corpus. The most basic type of token we may want to look for is a simple word. To search for a specific word (for example the word "dear") you need to type:

`DICKENS > "dear";` (again, press `space` for next page, `q` for coming back to command prompt)

Question 1. What would be the use of a query like this, providing the results (word in context) given by `cqp`? Think for example about what the use of this word (or other words) may help understand information in the corpus. Also think how this would help a writer when writing a story!

Sometimes we will want to search for words that are very similar, so that we can get a better sense of context, for instance think about a query like the previous one, but returning all the instances in context of some words. As an example, consider the following query:

`DICKENS > "regular(s|ly|ity)";` (Try it!)

The syntax of the query above is based on the use of a regular expression, like the ones you studied in INF1 - Computation and Logic.

Question 2. How can we interpret or paraphrase this query in plain English?

In some cases, to get more instances or a given pattern in a query, you may want to search for the given pattern ignoring if the letters are in lowercase or uppercase. In order to make your search case insensitive you only need to append a `%c` before your semicolon, like this:

`DICKENS > "dear" %c;` (Try it!)

Question 3. How do you retrieve sentences containing any of the different forms (singular and plural) of the nouns "book", "box", "gentleman".

2 Display Options

When working with `cqp` you will want to customise how the results of your queries are displayed to suit your needs. Perhaps the most important thing we may want to customise is for a query result to display (or not display) the annotations made on the corpus. This is easily achieved by running the following:

`DICKENS > show +pos +lemma;` (If you want query results to *show* annotations) or

`DICKENS > show -pos -lemma;` (If you want query results to *hide* annotations)

By default, this switch is off. Turn it on, and try one or two queries so you can see how this corpus has been annotated.

Question 4. Can you determine what kinds of tokens have been annotated? (Find at least two!)

3 Accessing linguistic annotations

One of the reasons we annotate text data is to enable us with the possibility of searching tokens annotated with some determined tag. For example, we might be interested in searching all *adjectives, verbs* or *nouns* in a given corpus. In order to do that, we need to know the actual set of tags used to annotate the corpus. In `cqp` you can see this by running the `show cd` command. To understand what some of these tags actually mean you can access the following website:

`http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html`

If you want to find out what *adjectives* have been annotated in the DICKENS corpus, you need to run the following query:

`[pos = "JJ"];` (JJ corresponds to adjectives, Try this!.)

Notice (in the webpage mentioned earlier) that different types of *noun* are annotated. You can use regular expressions in `cqp` to find a given pattern of tag.

Question 5. How can we retrieve all verb tokens in a `cqp` query? *Hint.* You can use the dot (‘`.`’) as a wild card. As such, it will match *any* character.

Question 6. Retrieve all tokens of any type except verb in a `cqp` query? *Hint:* Try using != instead of = in your query.

Question 7. Retrieve all sentences where the word `lock` is used only as a `noun`. *Hint:* It may be useful to combine two tests using `&` for conjunction.

4 Sequences of words

Question 8. Retrieve all occurences of the words `old gentleman` in the corpus.

Question 9. How can we retrieve all adjectives that appear before the word `gentleman` in the corpus?

Now we will recall a few other operators used in regular expressions (REs). Whenever a (*) is added to a regular expression, the preceding expression will be matched zero or more times. A (+) sign works in a similar way but will match the preceding expression once or more, and finally (?) will match zero or once only.

Question 10. Consider the following queries. Explain in words what each one retrieves. Write down some examples from the results retrieved.

- `[pos = "IN"] [pos = "DT"]`
- `[pos = "IN"]+ [pos = "DT"]`
- `[pos = "IN"]+ [pos = "DT"]+`
- `[pos = "IN"]* [pos = "DT"]?`

5 Frequencies, Bigrams and Collocations

It might be helpful to refer to slides from lecture notes (Part III, slides 17 onwards) on `Corpora` to solve the following problems.

Question 11. Formulate CQP queries to find the frequencies of the words ”the”, ”a”, ”in”, ”Oliver”, ”girl”, ”asked” occur in the corpus?

Question 12. Find the most frequent bigrams in the corpus starting with the words, `The, a, an, to`.

Question 13. `Darnay` is the last name of an important character in one of Charles Dickens' famous novel. What is the first name? How can you use n-grams to search the Dickens corpus for the first name?

Question 14. Find some collocations containing the word `Christmas`. Use your knowledge of English to determine which pairs of words involving `Christmas` are collocations.

Type `exit;` to quit the `cqp`-program.