Informatics 1B
Data and Analysis: Assignment 2
IMDB Corpus Analysis

Manuel Marques-Pita
Edited by Gail Sinclair

February 24, 2007

This assignment must be submitted by noon on Friday 9th of March 2007. Please follow instructions carefully or you may be penalised by losing marks on your assignment.

General Instructions

- Use your favourite text editor to create a file called a04-answers.txt and edit this file with your answers.
- Make sure you submit your file in .txt (ASCII) format.
- Add **Exam Number: `<your exam number>`**ONLY at the top of your answer file. This can be found on your matriculation card. Please do not add any other identifying information.
- Proof read your practical. Make sure it does not have spelling mistakes and that it is coherent.
- Submit your answers by using the command
  `submit inf1 inf1b da5 a04-answers.txt`
  and keep a back up copy of what you submitted.

1 Introduction

In this assignment you will have an opportunity to explore the Corpus Query Processor further and to work on some Information Retrieval exercises. In this case you will use `cqp` again which was introduced in the last lab session. You will need a copy of that lab session's handout for this assessed practical.

Before you start working on your answers, you will need to export an *environment variable* so that `cqp` can find the corpus you will be working with. On a terminal window type the following command:

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

Now start `cqp`

```
> cqp -e
```

And load the IMDB-CORPUS

```
[no corpus] > IMDB-CORPUS;
```

Your first task is to understand how this corpus has been annotated. Specific information about annotation of a loaded corpus can be determined by running the command `show cd;`

Each token that corresponds to a word is tagged as `word.` The `pos` tags correspond to grammatical attributes of the tokens as described in Lab 2 for Data and Analysis. It should be noted that regular expressions in quotes (“ ”) in cqp generally search for a match *within* individual words/tokens and not across white space and if we do not care about the content of a word, that word can be represented as `[]`, or `[]*` for multiple words.

2 The IMDB-CORPUS

The corpus you will work with contains *plot summaries* as the ones you explored in the Data and Analysis Lab 2. This corpus has been annotated on the structural level using the structural tags `movie, title, summary` and `author` (do `show cd` to see this). The first tag `(movie)` encompasses title, all the plot summaries (one or more) and their corresponding authors for one film. Movies in this corpus are structured as follows:

```
<movie>
  <title>The film title</title>
  <summary>The first summary available</summary>
  <author>The author of the previous summary</author>
  <summary>Another summary</summary>
  <author>The author for the next summary</author>
  ...
</movie>
<movie>
  <title>Another film title</title>
  <summary>The first summary available for this film</summary>
  <author>The author of the previous summary</author>
</movie>
```

On the grammatical level, this corpus is annotated as the DICKENS corpus you worked with in the lab session last week.

When running queries in `cqp` depending on your specific goals you can show or hide one or more of these tags in your query results. For example, if you want your query results to display where the tags `title` and

summary are you just need to run the command

```
[IMDB-CORPUS] > show +title +summary;
```

If you then want to switch one of them off (summary) for instance you only need to

```
[IMDB-CORPUS] > show -summary;
```

2.1 Searching within structural blocks

As seen in Lab 2, With the annotation scheme for this corpus, it is possible to design queries in which you can retrieve, for example, all the *nouns* starting with "Nation". But, what about searching for all the instances of tokens starting with "Nation" in the available *film titles*? For that you can write a query like the following:

```
[IMDB-CORPUS] > [pos="N.*" & word="Nation.*] within title;
```

Which is the same original query with the extra construction within title. This query specifies that nouns starting with "Nation" will be matched against tokens inside a <title> </title> tag. Make sure you have run ([IMDB-CORPUS] > show +title;) and execute the query above. To move down the page press the spacebar, to move up press 'b', to go back to the cqp prompt press "q".

2.2 Setting boundaries for retrieval of information using the corpus structure

Another interesting feature of cqp is that it allows you to expand the query results to the chosen boundaries within the structure imposed on your data (by the annotation scheme). For example, if you want to save all the *movie* information for a query that searches words within the title, you can, for example, do the following:

```
[IMDB-CORPUS] > [pos="N.*" & word="Nation.*] within title expand to movie;
```

This will retrieve all information within the <movie> </movie> tags for those movies in which words starting with "Nation" are found in their titles.

In the lab session, *named queries* were introduced. Named queries can be combined with queries using structural information to do a number of interesting things. For example, the last query explored could be saved in a variable called Q by doing

```
> Q = [pos="N.*" & word="Nation.*] within title expand to movie;
```

You can then query on this new variable, treating it as a subset of the original corpus by typing

```
[IMDB-CORPUS] > Q;
```
you should then see the following prompt

```
[IMDB-CORPUS:Q] >
```

All queries you type now will be performed within the domain consisting of the results retrieved by the query *Q* only. In order to go back to the original corpus, just type its name, followed by a semicolon:

```
[IMDB-CORPUS:Q] > IMDB-CORPUS;
```

2.3 Using structural tags in query-construction

Sometimes you will want to be able to find, for example, film titles that *start* or *end* in a certain word. There are cases in which using the standard *contains-word* strategy is not specific enough. Doing these more specific queries is very simple. The main idea is to think that the query is a *partial pattern* that will be matched against tokens in the corpus. Taking this into account, you can retrieve all films starting with the word *Purple* or *purple*:

```
[IMDB-CORPUS] > <title>"[Pp]urple";
```

2.4 Saving your queries

You will need to save some of your query results a file and then read these files to answer some questions. A named query (Qx, for example) can easily be saved (to a file called Qx_results, for instance) on your top level directory by doing the following

```
[IMDB-CORPUS] > cat Qx > "~/Qx_results";
```

This will override any existing `Qx_results` file, if you want to append your results to the existing content

```
[IMDB-CORPUS] > cat Qx >> "~/Qx_results";
```

3 Your tasks

Question 1.1

Write a query (or sequence of queries) to find out how many summaries in the corpus are written by someone whose name contains “Smith”.

Question 1.2

Write a query (or sequence of queries) to find out how many of the matches in Task 1 correspond to plot summaries written by “Dave Smith”. What is the proportion of summaries written by Dave Smith in the set of all summaries written by someone whose last name is Smith?

Question 1.3

Write a query (or sequence of queries) to find out how many plot summaries have been written by authors called "Dave" but whose last name is not "Smith". (Keep in mind that some authors names may contain only one word).

Question 1.4

Look at Question 1.1 and your solution again. Does your query/ies retrieve authors with the surname "Smithton" or only those authors with the full surname of "Smith"? (Do not change your answer to Question 1.1 - you will not losing any marks for only retrieving "Smith"s.) How would you now phrase your query/ies given the phrasing for Question 1.1?

Question 2.1

Write a query (or sequence of queries) to find out how many plot summaries have been written by *Marion*. After you have written your query, go to a web-browser and type the following:

`http://www.imdb.com/SearchPlotWriters?Marion`

This will display the films whose summaries have been written by authors whose name contains *Marion*. Did you get the same answer as IMDB? If not, explain what the differences are – not only in terms of the results, but also in terms of how you believe both queries, the `cqp` one and the IMDB one are interpreted.

Question 2.2

Try to answer the following question using the IMDB website (`www.imdb.com`)

How many film titles contain the words "Series" followed by zero or one (any) word followed by an *cardinal number* and preceded by a *noun*? Make your query case insensitive. If you find this difficult, please describe your attempts.

How would you answer this question using `cqp`? Which query system makes the work easier and why?

Question 3

Write a query (or sequence of queries) to retrieve the titles only for films whose corresponding summaries contain the words "round" and "table", with the constraints that there might be zero or one words between "round" and "table", but always "round" before "table" and both words are always in lowercase. There is an additional constraint that there is one or more co-ordinating conjunctions somewhere after "table".

In your answers file, list what is retrieved as well as your query/ies.

Question 4.1

What is the frequency distribution of the words in bigrams which contain "flight" (case insensitive) in film titles? Report the top eight only.

Question 4.2

What is the frequency distribution of the POS tags in bigrams which end in “flight” (case insensitive) in film titles?

4 Information Retrieval

Up to this point you have worked with `cqp`. For the next question you will calculate your results by hand (also using a calculator or a spreadsheet application). Figure 1 shows a table which contains information about the frequencies of a number of specific words within a number of plot summaries for five different movies. Also in this figure, on the right-most column and bottom row, you will find the sum of values for the corresponding rows/columns. We will treat each of the movies as a document (rows Doc1 to Doc5 are the documents).

Question 5

	Doc1	**Doc2**	**Doc3**	**Doc4**	**Doc5**	Total Instances
officer	14	17	4	1	3	39
fuzz	26	10	9	9	20	74
frost	31	14	12	2	15	74
justice	0	2	6	3	1	12
hot	12	29	20	10	17	88
All words	83	72	51	25	56	287

Figure 1

Calculate the similarity measures between these documents and the following queries and rank them from closest to farthest for each query.

1. Query1 = “hot” AND “fuzz”
2. Query2 = “frost” AND “justice” AND “officer”

Remember to show your working.