Big Data & Predictive Analytics

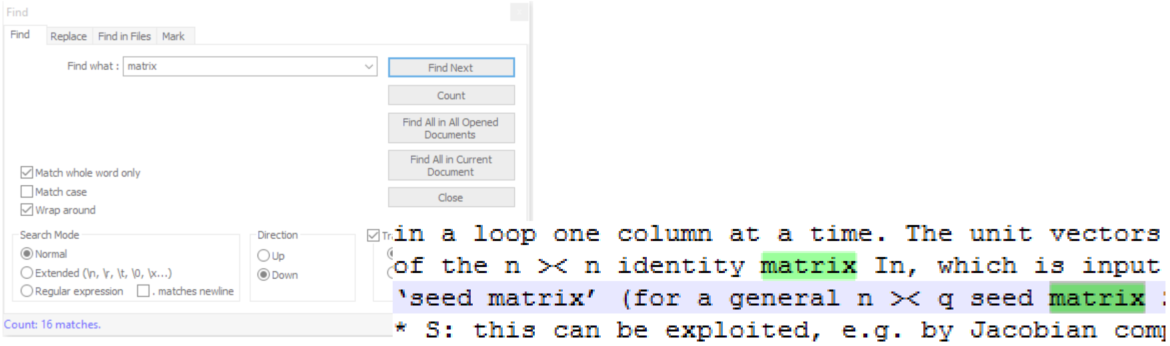
Coursework Assessment One

Short Summary by Will Peck

14th February 2017

Testing

The tests were conducted as described by the table below. An attempt was made to ensure the program returned the expected result for all inputs. This was achieved through comparing the results with those produced by the 'word count' feature in Microsoft Word. Microsoft Word was used in replacement for Notepad++ after it was discovered that the 'find' feature returned incorrect word count values where text files contained non-unicode characters. For example; Notepad++ only returned sixteen instances of the word "matrix" when searched over the text from "paper1.txt" and "paper2.txt" when there are seventeen:



Find what : matrix

☑ Match whole word only
☐ Match case
☑ Wrap around

Search Mode: ● Normal ○ Extended (\n, \r, \t, \0, \x...) ○ Regular expression ☐ . matches newline

Direction: ○ Up ● Down

Count: 16 matches.

```
in a loop one column at a time. The unit vectors
of the n >< n identity matrix In, which is input
'seed matrix' (for a general n >< q seed matrix :
* S: this can be exploited, e.g. by Jacobian com
```

			TextProcessing Program		Microsoft Word			
Filename	Word1	Word2	Word1 Qty	Word2 Qty	Word1 Qty	Word2 Qty	Minimum	Result
"paper1"	"this"	"the"	609	52	609	52	52	Passed
"paper1"	"algorithm"	"matrix"	31	14	31	14	14	Passed
"paper1"	"notinthetext"	"matrix"	0	14	0	14	0	Passed
"paper1"	"notinthetext"	"notinthetext"	0	0	0	0	0	Passed
"paper1"	"matrix"	"matrix"	14	14	14	14	14	Passed
"paper1"	"The Art of Computer Programming"	"computer code"	1	3	1	3	1	Passed
"paper1", "paper2"	"this"	"the"	78	934	78	934	78	Passed
"paper1", "paper2"	"algorithm"	"matrix"	61	17	61	17	17	Passed
"paper1", "paper2"	"notinthetext"	"matrix"	0	19	0	19	0	Passed
"paper1", "paper2"	"notinthetext"	"notinthetext"	0	0	0	0	0	Passed
"paper1", "paper2"	"The Art of Computer Programming"	"computer code"	1	3	1	3	1	Passed
"paper1", "paper2"	"matrix"	"matrix"	19	17	19	17	17	Passed

To verify each test result; the use of the "assert" command allowed for the value produced by the function call to be compared against the expected value. The resulting Boolean value then determined the success of the test.

During the testing process; the non-Unicode character issue caused some words to be missed from the count as described above. To fix this problem; a regular expression was used to extract and replace all illegal characters from each word as processed.

To solve the problem involving composite terms; after the number of words were counted and stored, the spaces were removed from the phrase. Then for each word in process; the next words in sequence were read ahead up to the length of phrase word count, concatenated and then compared against the phrase.

Computational Issues

The inefficient method used for finding words or phrases cross multiple files could be possibly be improved upon by avoiding the use of 3 nested for-loops. For large volumes of data this time complexity would likely cause higher than necessary processing times.

Furthermore; the performance of the program is effected by imported modules which were used to help solve some of the problems. Whilst it could be presumed that these modules were developed to the best possible time complexity; without measuring their performances their affect is uncertain.