Setup Instruction

1. Create a directory C:\ir\_assignment
2. Download and Unzip the IR\_ASSIGNMENT\_AMITESH\_RANJAN\_SRIVASTAVA\_**2017HT12082.zip in that directory**
3. Open a Python IDE – such as spider (or any) and set the workspace to the following path –

“C:\ir\_assignment\IR\_ASSIGNMENT\_AMITESH\_RANJAN\_SRIVASTAVA\_2017HT12082\executable”

**WITH\_PRE\_PROCESSING**

1. Now in your python IDE , execute the following python scrips in following sequence:

- **Inverted\_Index\_WITH\_PRE\_PROCESSING.py** - it will create a following file in the same path –“ **inverted\_index\_file\_with\_pre\_processing.csv**”

1. Now run the script – “**QuerySearch\_WITH\_PRE\_PROCESSING.py**” – it will generate following files

–“ **output\_test\_file\_with\_pre\_processed.csv**”: This is the output file running against the preprocessed inverted index

-“**output\_test\_evaluation\_with\_pre\_processed.csv**” : This is evaluation file created from confusion matrix by using “input\_test\_file.csv” and “output\_test\_file\_with\_pre\_processed.csv”. “input\_test\_file.csv” contains column “WORD1” and “ WORD2” and since we know that if both the words are present in different documents therefore by taking there intersection they should return the matching documents . So the 3rd column “OUTPUT\_INTERSECTION” is 1 if intersection returns at least one common document otherwise it is 0. **“input\_test\_file.csv” is already known to us.**

**WITHOUT\_PRE\_PROCESSING**

1. Now in your python IDE , execute the following python scrips in following sequence:

- **Inverted\_Index\_WITHOUT\_PRE\_PROCESSING.py** - it will create a following file in the same path –“ **inverted\_index\_file\_without\_pre\_processing.csv**”

1. Now run the script – “**QuerySearch\_WITHOUT\_PRE\_PROCESSING.py**” – it will generate following files

–“ **output\_test\_file\_with\_pre\_processed.csv**”: This is the output file running against the preprocessed inverted index

-“**output\_test\_evaluation\_without\_pre\_processed.csv**” : This is evaluation file created from confusion matrix by using “input\_test\_file.csv” and “output\_test\_file\_without\_pre\_processed.csv”. “input\_test\_file.csv” contains column “WORD1” and “ WORD2” and since we know that if both the words are present in different documents therefore by taking there intersection they should return the matching documents . So the 3rd column “OUTPUT\_INTERSECTION” is 1 if intersection returns at least one common document otherwise it is 0. **“input\_test\_file.csv” is already known to us.**