**Problem Statement1**- How will we test the program to derive the profanity degree?

1. We have defined profanity as this (score\*100/float(len(words)-1)), whenever we are try to find len(words) in the denominator it also counts ‘.’ as a word. So, use (len(words)-1) to find the accurate length.

**Limitation-**

If sentence consists only one word, in that case the value of denominator will be zero. In order to overcome this problem we should add epsilon in the denominator.

1. Add some negative phrases in vocabulary and then check the working of function.

Ex- In my code I have defined a phrase (like-superiority complex) in the vocabulary, but the function fails to give the degree of profanity of the sentence containing the phrase because It only recognises the words in separate rather than recognising the full phrase as one.

For example, the function will read ‘superiority’ as one word and ‘complex’ as another word. That is why it fails to give the result.

1. If words present in the vocabulary are in lower case but words present in the sentence are in upper case, then also function fails to run.
2. We can also use stemming in place of lemmatization to check the working of function.
3. When word is not present in the vocabulary then except block will run.

**Problem Statement2**- How well versed are you on the Unix command line? Give us some examples to demonstrate that.

* The Unix operating system is a set of programs that act as a link between the computer and the user.
* Unix can be used by several people at the same time and a user can run multiple programs at the same time.
* I have knowledge of file management like how to create and remove files, copy and rename them, create links to them, etc.

For example- Copying Files

To make a copy of a file use the **cp** command. The basic syntax of the command is- **$ cp source\_file destination\_file.**

* I have knowledge of directories, file permission, basic utilities, pipes & filters, vi editor. **vi** editor is used to edit an existing file or to create a new file from scratch. We can also use this editor to just read a text file.

For example-(1) Changing Ownership

The **chown** command changes the ownership of a file. The basic syntax is as follows – **$ chown user filelist**

(2) Sending email

We use the Unix mail command to send and receive mail.

**$mail -s "Report 05/06/07" admin@yahoo.com < demo.txt**

**I am still learning Unix command line.**

**Problem Statement3**- We have a bunch of files in CSV format (with the same set of fields). Develop a web UI that has a dropdown to select one file and one or more fields in the file and extract the information? Write both the front-end code and back-end code.

**Assumptions-**

1. The name of the files available are hard coded.

**Explanation-**

* Defined a local variable (datapath in my code) which contains path of the csv file.
* Index function of the API is rendering the basic templates of demo.html.
* getColumn function of API reads a particular file and takes the columns selected by using checkbox in the form of list and sends the output to html page (generate\_table.html) and dynamically generates the table using javaScript.

**Limitations-**

1. Current version does not generate the dropdown list dynamically based on files available in the specified path.

**Approach to solve.**

We can get the list of files by using ‘glob’ and filtering based on allowed extension i.e., csv and passing the name of those files to the JavaScript to generate the list for the dropdown dynamically.

1. This code will fail to handle exception.

**Approach to solve.**

This can be handled by using try and except.