# ASSIGNMENT 4

1. Read all the words from text5 and text7 in list L5 and L7 respectively. Stem all the words using Porter Stemmer and Lancaster Stemmer.
2. Extract the text5 as a string *strA.* Use *strA* to check the Part Of Speech (POS) tagging and Calculate the maximum likelihood estimation?
3. From the list of files given as *LstFile,* print the filename of MS word documents only.
4. Display only the URLs from the given tweet T.
5. Find hashtags and usernames in the given short-text T.
6. Implement the Viterbi algorithm for POS tagging for the following language
    - Group A: Hindi
    - Group B: Punjabi
    - Group C: Marathi
    - Group D: Gujarati
    - Group E: Punjabi
    - Group F: Bengali
    - Group G: Marathi

    You can use the Google scholar for existing literature, FIRE (Forum for Information Retrieval and Evaluation) for dataset and other resources, IIT Mumbai website for the reference of other Indian language resources.

## APPENDIX

### Basic Patterns

The power of regular expressions is that they can specify patterns, not just fixed characters. Here are the most basic patterns which match single chars:

- a, X, 9, < -- ordinary characters just match themselves exactly. The meta-characters which do not match themselves because they have special meanings are: . ^ $ * + ? { [ ] \ | ( ) (details below)
- . (a period) -- matches any single character except newline '\n'
- \w -- (lowercase w) matches a "word" character: a letter or digit or underbar [a-zA-Z0-9_]. Note that although "word" is the mnemonic for this, it only matches a single word char, not a whole word. \W (upper case W) matches any non-word character.
- \b -- boundary between word and non-word
- \s -- (lowercase s) matches a single whitespace character -- space, newline, return, tab, form [ \n\r\t\f]. \S (upper case S) matches any non-whitespace character.
- \t, \n, \r -- tab, newline, return
- \d -- decimal digit [0-9] (some older regex utilities do not support \d, but they all support \w and \s)
- ^ = start, $ = end -- match the start or end of the string
- \ -- inhibit the "specialness" of a character. So, for example, use \. to match a period or \\ to match a slash. If you are unsure if a character has special meaning, such as '@', you can put a slash in front of it, \@, to make sure it is treated just as a character.