TAGGING AND TOPIC MODELLING OF MCQ QUESTIONS

Vishnu Praneeth

Implementation:

Given an MCQ question, we first extract the basic keywords from the text using a score that is generated for each word.
This is done as follows:

1.The text has to be cleaned and removed of all its stop words.This is done using a **SmartStoplist.txt** file which contains all the basic stop words.
2.Once the given text is removed from all the stop words,it is split into sentences and phrases using appropriate delimeters.
3. For each phrase in the phraseList,the words are split based on a **seperate_words()** function which removes some extra characters if any from words and splits them into seperate words.After seperation for each word in the word list,the word frequency is incremented by 1 and word degree is incremented by a value equal to the length of word list in which the word is present.Finally **dividing the degree upon frequency of the word gives the score of the word.**
4.So the keywords of the given text are considered to be the top 30% of all keywords when they are arranged in decreasing order of scores.*These keywords come in the final output.*

Training Algorithm:

1.A data set of a list of topics and their text corpus is scraped from online sites.These are used for training.
2.In the implementation we create a map of set of topics(as the key value) and set of keywords(extracted using the score technique as mentioned above) of the text document pertaining to those topics as the value.*So our key is a list and value is also a list in our map.*
3.Now when I'm given a new text document as my query,I initially calculate the keywords from the text(using score technique) and for each of these keywords,the frequency of their occurence in the test data are calculated

using the calculate_freq_kw_corpus() function.The test data as mentioned above in the map maintains the list of keywords belonging to a particular text corpus.So i consider a keyword from my text document and i calculate its frequency in a particular corpus using the list of keywords maintained.Now i keep doing this for all the keywords in my text(query) document. So I finally get a frequency array of frequencies of a particular keyword in all the corpuses as below.

Frequency array for a keyword : [fq1,fq2,fq3,……..fqn]
Where fqi is the frequency of occurence of the keyword in ith corpus.
I have multiple such frequency arrays now.
Now I add all the corresponding frequencies with same index and store it in a sum array.

Fw1[]=[fq1,fq2,fq3…….fqn] , Fw2= [fq'1,fq'2,fq'3…...fq'n]
…...Fwn=[fq'''''''1,fq'''''''2,........,fq'''''''n]

Sum array
Sof[]=[fq1+fq'1+....fq'''''''1,fq2+fq'2+....fq'''''''2,.........fqn+fq'n+.....fq'''''''n]
(basically im calculating the sum of frequencies of all keywords in a particular corpus)
Now it is obvious that i take the maximum of these values of this sum array,which gives me the corpus which my set of keywords belong to!! :)

Once i get that index where the maximum occurs,i can get the corresponding list of topics from the main map which stores the list to list mapping.

Another approach which I tried was using the standard tf-idf method.But the above method was observed to give better results. Also,the outcome and the accuracy of the algorithm is mainly dependent on training.If large corpus sets on diverse topics are used for training,the better the results will be.

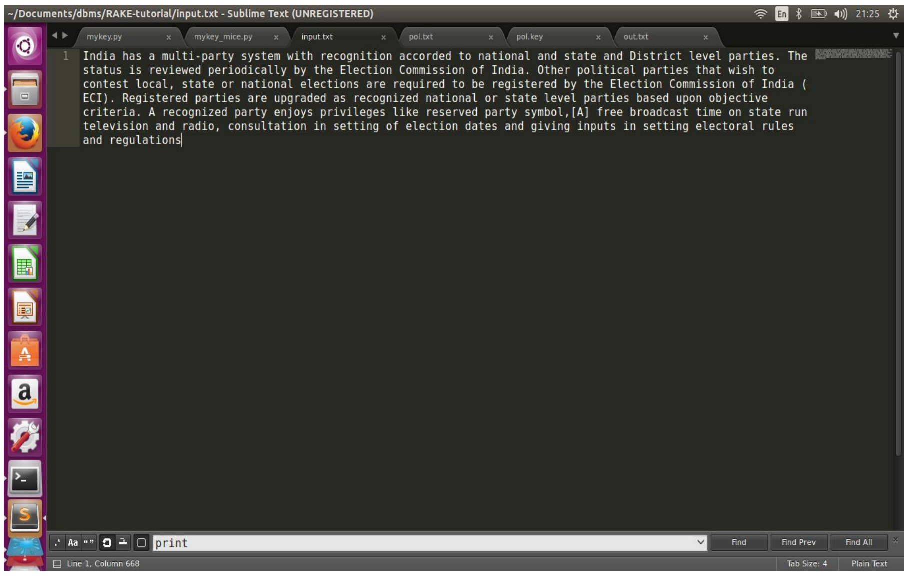
Other Insights:

Latent Dirichlet Allocation has also been tried for training,but it works very well when our input data set in huge.Since here it was mentioned that the input is of max 50 words, score computation(tf-idf) was the best technique for such small sized corpuses.

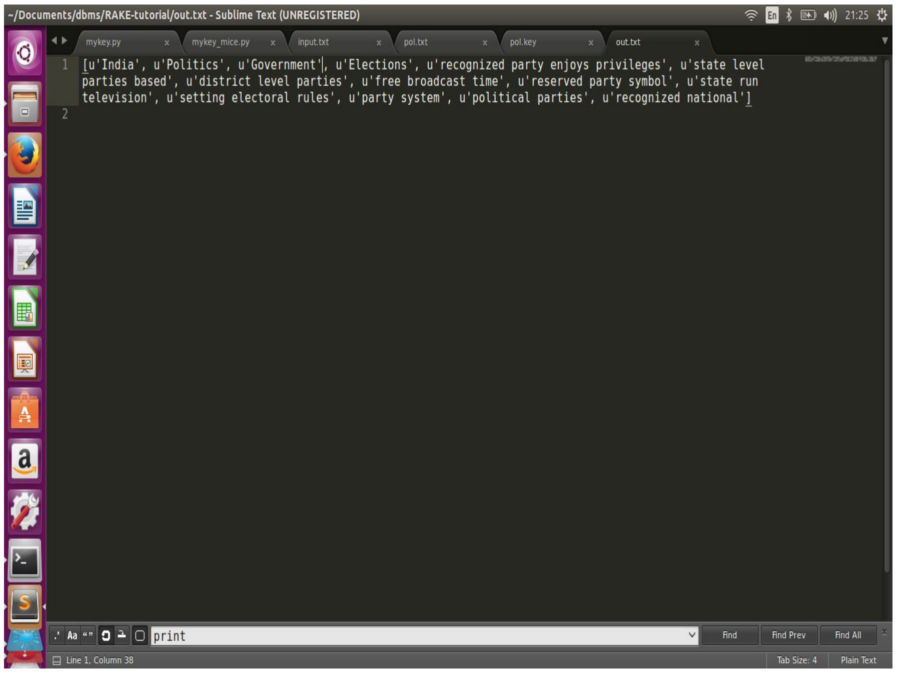
Running the program:

Please give a single MCQ question which is supposed to be tagged in the input.txt file and observe the output in the out.txt file.
Command to be run: **python mykey_mice.py > out.txt**

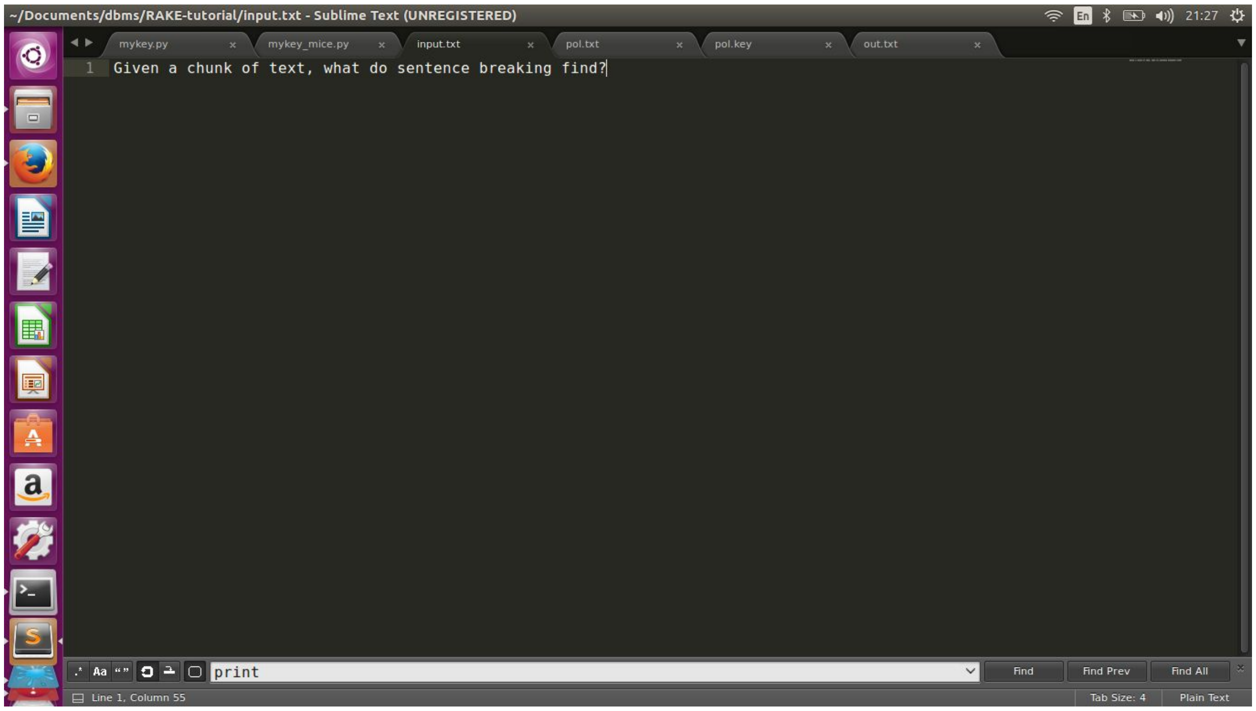
Sample outputs:


~/Documents/dbms/RAKE-tutorial/input.txt - Sublime Text (UNREGISTERED)

```
India has a multi-party system with recognition accorded to national and state and District level parties. The
status is reviewed periodically by the Election Commission of India. Other political parties that wish to
contest local, state or national elections are required to be registered by the Election Commission of India (
ECI). Registered parties are upgraded as recognized national or state level parties based upon objective
criteria. A recognized party enjoys privileges like reserved party symbol,[A] free broadcast time on state run
television and radio, consultation in setting of election dates and giving inputs in setting electoral rules
and regulations
```

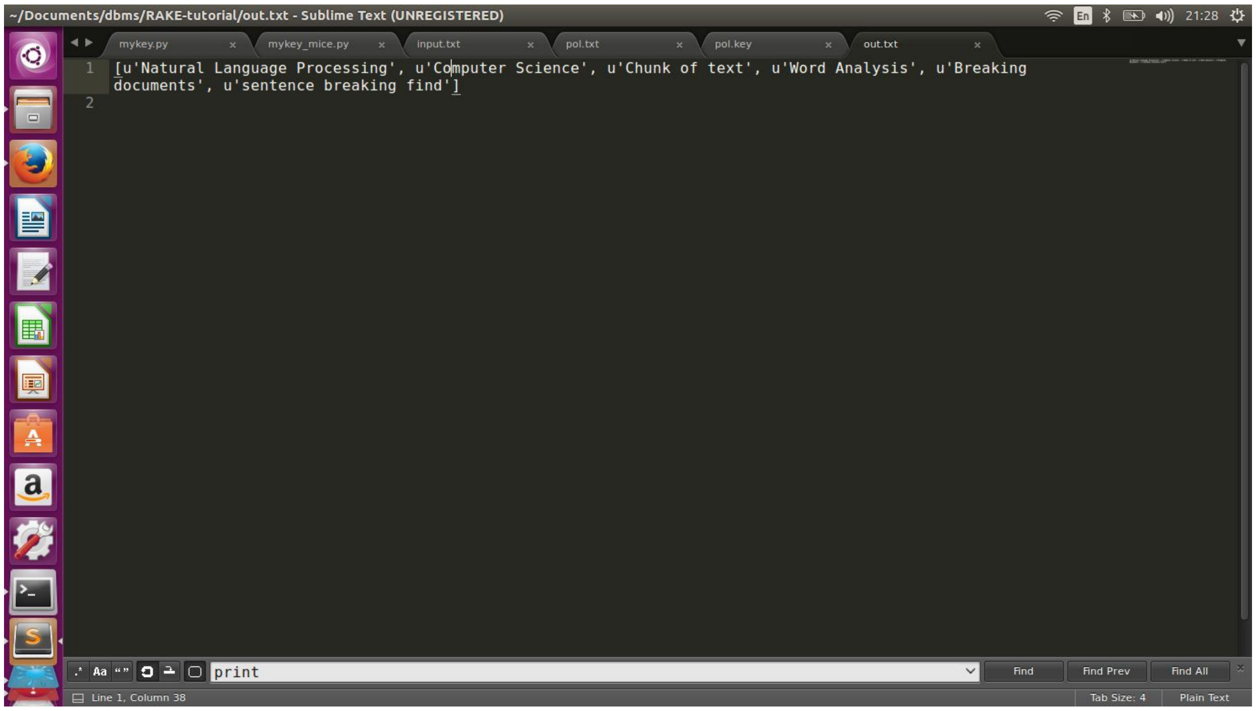

~/Documents/dbms/RAKE-tutorial/out.txt - Sublime Text (UNREGISTERED)

```
[u'India', u'Politics', u'Government', u'Elections', u'recognized party enjoys privileges', u'state level
parties based', u'district level parties', u'free broadcast time', u'reserved party symbol', u'state run
television', u'setting electoral rules', u'party system', u'political parties', u'recognized national']
```



~/Documents/dbms/RAKE-tutorial/input.txt - Sublime Text (UNREGISTERED)

```
Given a chunk of text, what do sentence breaking find?
```



~/Documents/dbms/RAKE-tutorial/out.txt - Sublime Text (UNREGISTERED)

```
[u'Natural Language Processing', u'Computer Science', u'Chunk of text', u'Word Analysis', u'Breaking
documents', u'sentence breaking find']
```

Hence more data of huge corpus has to be loaded in data/docs/fao_test in directory with diverse topics to ensure accuracy.I've taken only a few topics with less data corpus set.