VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Tasks

Natural Language Processing

(CSE4022)

Name: Nachiket Talwar

Registration Number: 19BCE0840

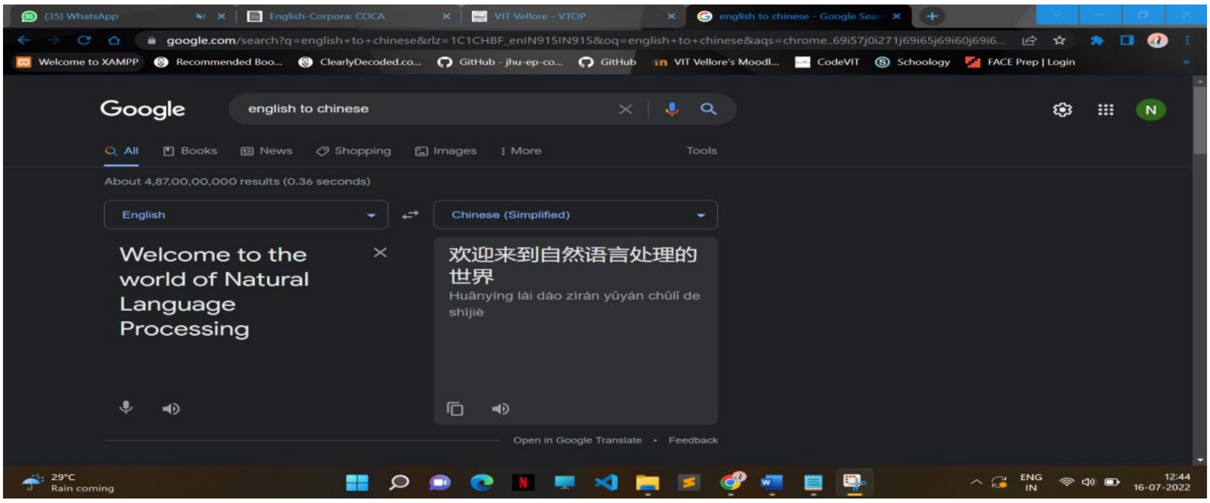
Slot: C1

Tasks assigned:

1. encoding demo
2. coca: story from data+ search string in text box: concirdance searching
3. accessing books : brown corpus, inaugral corpus,
4. experiening: frequency dist & conditional freq dist
5. suggest an application that u can build using these corporus: less than 20 words

Tasks solution:

1. We first get our Chinese letters which we will save to notepad



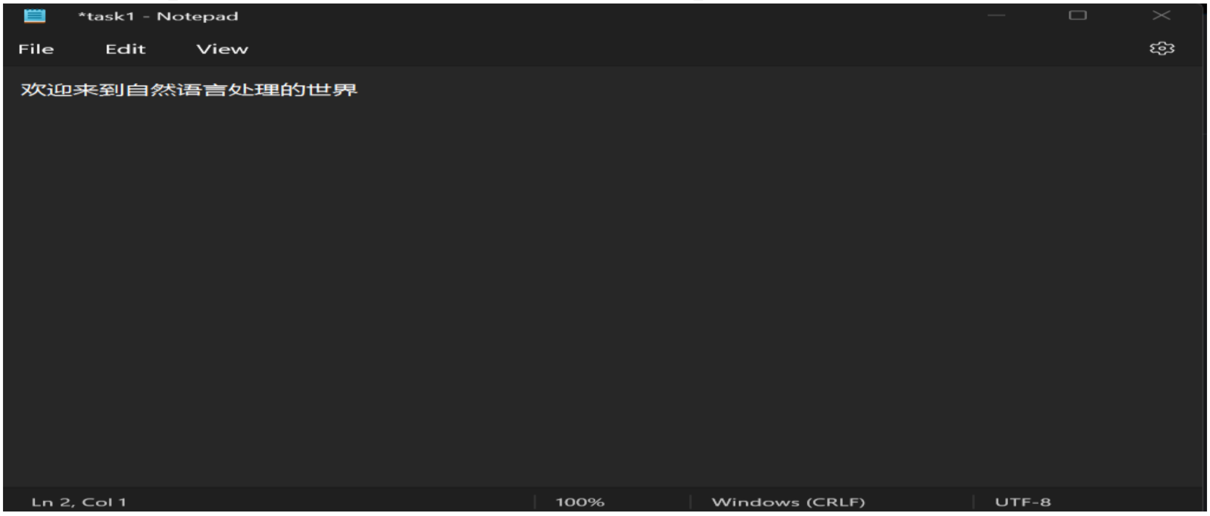
english to chinese

Welcome to the world of Natural Language Processing

欢迎来到自然语言处理的世界

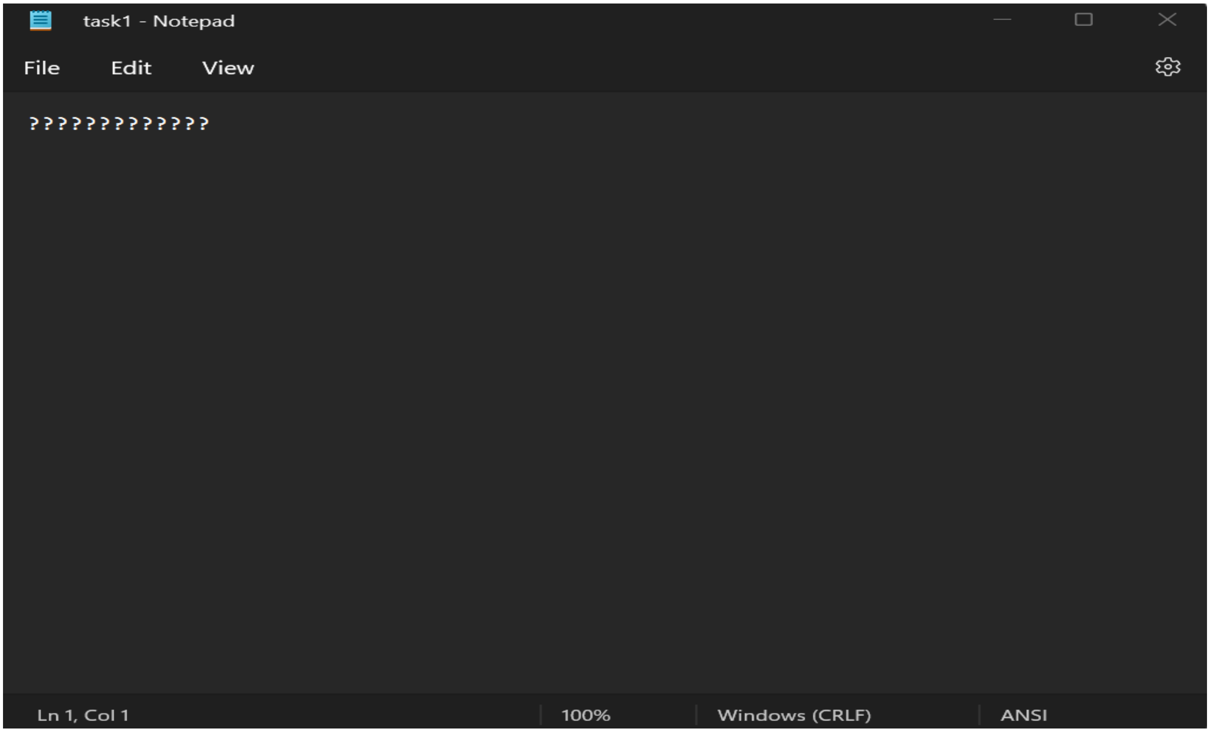
Huānyíng lái dào zìrán yǔyán chǔlǐ de shìjiè

Next, we paste the characters in the notepad



欢迎来到自然语言处理的世界

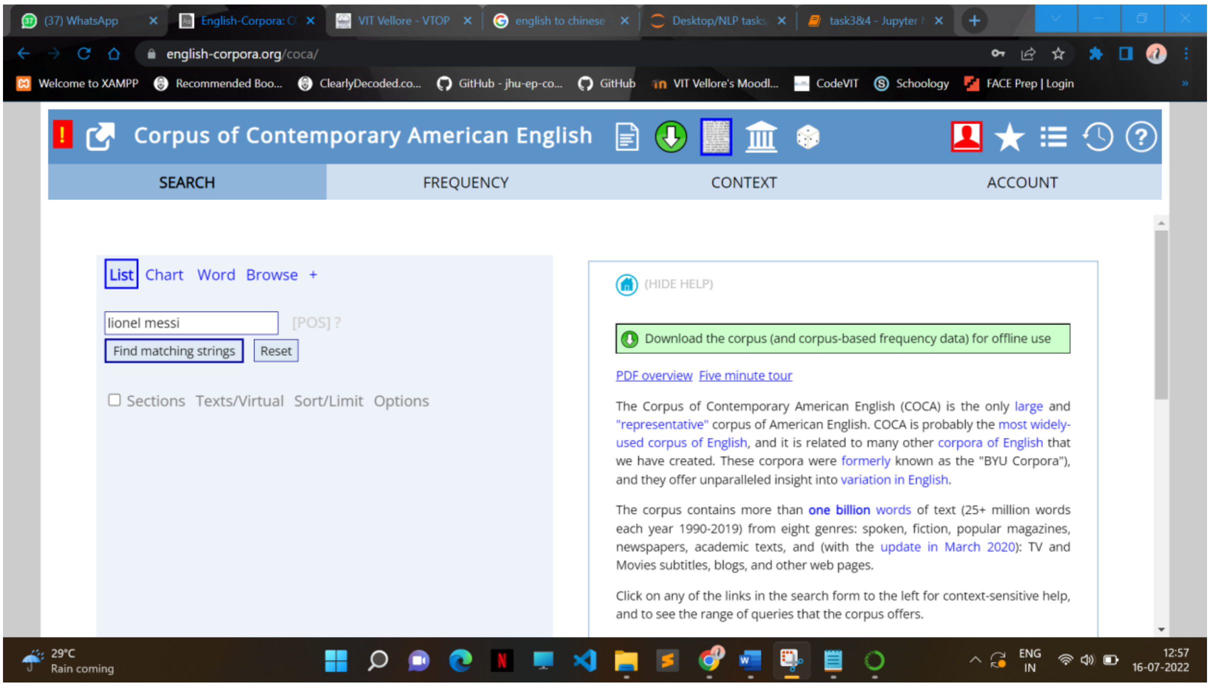
We lastly save the file as with ANSI encoding and get the following result



```
?????????????
```

2. We used the corpus coca to find a string and create a story from that

   We have decided to use the string ‘lionel messi’ for our coca search



lionel messi

☐ Sections Texts/Virtual Sort/Limit Options

Download the corpus (and corpus-based frequency data) for offline use

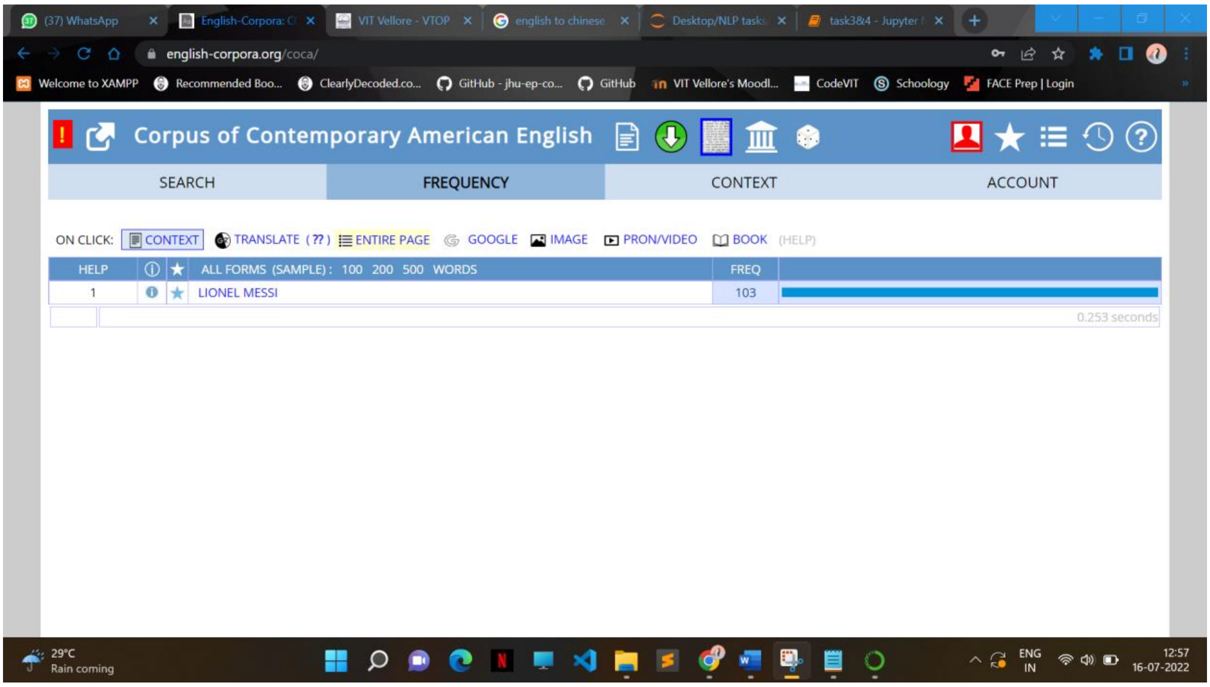
PDF overview Five minute tour

The Corpus of Contemporary American English (COCA) is the only large and "representative" corpus of American English. COCA is probably the most widely-used corpus of English, and it is related to many other corpora of English that we have created. These corpora were formerly known as the "BYU Corpora"), and they offer unparalleled insight into variation in English.

The corpus contains more than one billion words of text (25+ million words each year 1990-2019) from eight genres: spoken, fiction, popular magazines, newspapers, academic texts, and (with the update in March 2020): TV and Movies subtitles, blogs, and other web pages.

Click on any of the links in the search form to the left for context-sensitive help, and to see the range of queries that the corpus offers.

Below is the frequency of the appearance of our chosen string in the coca database

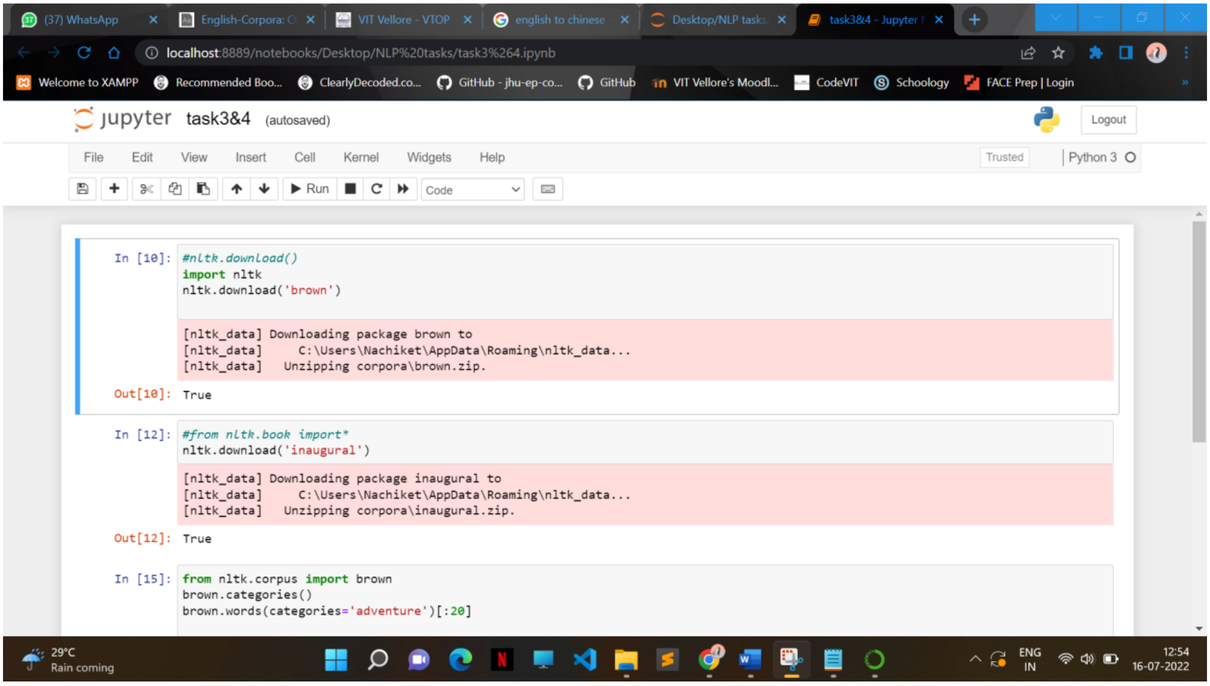


HELP	ⓘ	★	ALL FORMS (SAMPLE) : 100 200 500 WORDS	FREQ
1	ⓘ	★	LIONEL MESSI	103

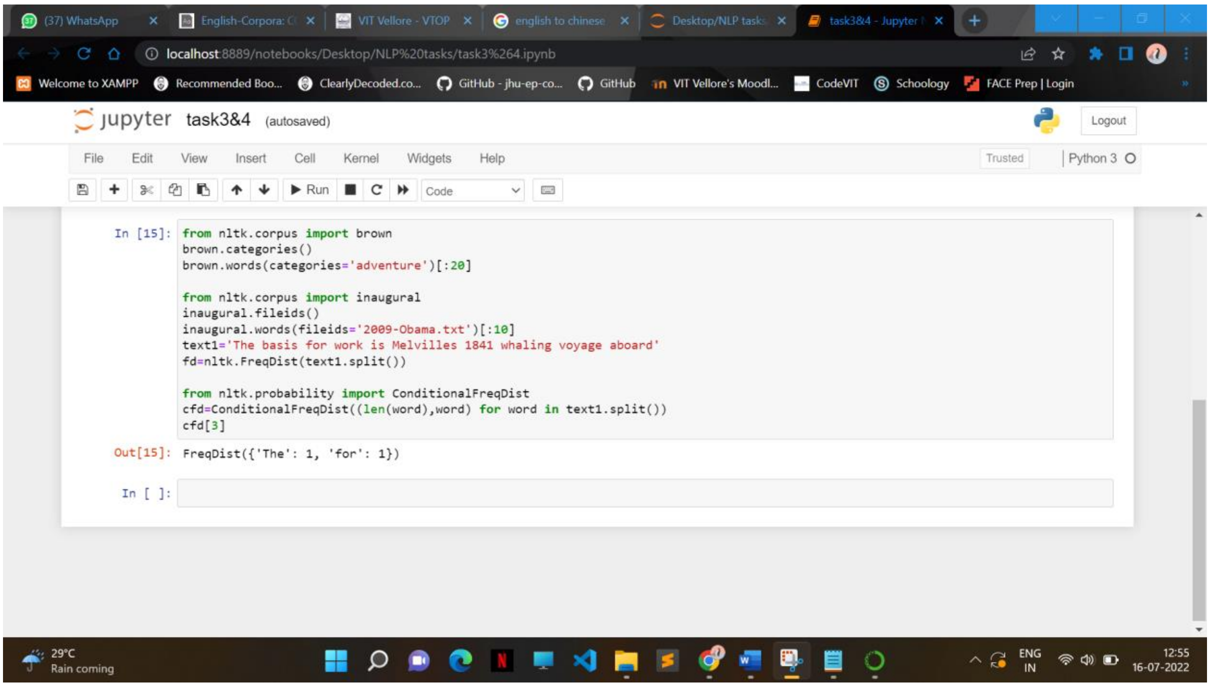
0.253 seconds

3. Tasks 3 and 4 have been plugged together as they work on the same piece of code

We use nltk toolkit to access brown and inaugural corpus and for task 4 we apply frequency distribution and conditional frequency distribution


localhost:8889/notebooks/Desktop/NLP%20tasks/task3%264.ipynb
jupyter task3&4 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Logout

```
In [10]: #nltk.download()
         import nltk
         nltk.download('brown')

         [nltk_data] Downloading package brown to
         [nltk_data]     C:\Users\Nachiket\AppData\Roaming\nltk_data...
         [nltk_data]   Unzipping corpora\brown.zip.

Out[10]: True

In [12]: #from nltk.book import*
         nltk.download('inaugural')

         [nltk_data] Downloading package inaugural to
         [nltk_data]     C:\Users\Nachiket\AppData\Roaming\nltk_data...
         [nltk_data]   Unzipping corpora\inaugural.zip.

Out[12]: True

In [15]: from nltk.corpus import brown
         brown.categories()
         brown.words(categories='adventure')[:20]
```


localhost:8889/notebooks/Desktop/NLP%20tasks/task3%264.ipynb
jupyter task3&4 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Logout

```
In [15]: from nltk.corpus import brown
         brown.categories()
         brown.words(categories='adventure')[:20]

         from nltk.corpus import inaugural
         inaugural.fileids()
         inaugural.words(fileids='2009-Obama.txt')[:10]
         text1='The basis for work is Melvilles 1841 whaling voyage aboard'
         fd=nltk.FreqDist(text1.split())

         from nltk.probability import ConditionalFreqDist
         cfd=ConditionalFreqDist((len(word),word) for word in text1.split())
         cfd[3]

Out[15]: FreqDist({'The': 1, 'for': 1})

In [ ]:
```

4. We can create a blog accessing app. It will take a string and give us all the blogs that access a particular string