Text Processing using Jupyter notebook on Stampede

Abhishek Manoj Kumar, Chetan Kumar and Deepak Kumar
University of Massachusetts Dartmouth

December 20, 2016

Abstract

Text processing is used to extract knowledge and insights from any unstructured or raw data. We are going to classify text using python and NLTK (Natural language tool kit) library. We will be making use of Jupyter notebook which is a web application that allows to produce interactive code and output representations which will make the output reproducible and easy to share. We also configured Jupyter on stampede which is called as Jupyter Hub, so the browser and kernel(s) will be on separate machine and Jupyter Hub is a multi-user server for Jupyter Notebooks.

1 Introduction

Text processing also known as text analysis is the process of extracting useful information from any text document. The obtained knowledge can be useful for different purposes such as doing sentiment analysis, finding out users and products relation to enhance marketing strategies, security applications, academic applications etc. Most of the text processing work is already in existence and our key contribution is implementing text processing on Jupyter Notebook. Jupyter notebook makes it easy to code in upto 42 languages and its easy to share data, code and visualization with annotations. Also notebooks can be shared as as package by using email, Dropbox, GitHub, etc. We also installed Jupyter Notebook on Stampede and remotely accessed it with by local machine. This facilitates us to upload files or computational jobs from local machine directly to stampede using Jupyter notebook without using any code repository to upload the code by local machine and pull that on stampede. We can also use the Jupyter terminal to access stampede terminal which provides direct access to stampede. Further we used Natural Language Toolkit (NLTK) which makes use of python programs to process human language data. We are making use of text processing libraries provided by NLTK for sentence segmentation, word tokenization, parts of speech (POS) tagging and particular grammar junking. We will define each text processing term along with the implementation below. You can process text obtained from any blogs or online sources and find the information like nouns including names of people, places, relationship between them. This kind of information does have applications in security, for example by doing text processing on the social media data which have discussions about security i.e. by analyzing what people are talking about the current issues on the internet, can give us insights into the future and prevent any threats to the national security. Another possible application would be marketing, for example by finding the relationship between people and products, they are talking about over the internet and consumer targeted marketing can be achieved.

2 Background

In Text processing, usually documents are processed in a certain order. The documents containing texts are processed by the natural language processing techiques in order to extract the linguistic structure of the text. Futher information is extracted using the information extraction engine. The information extracted includes sentence boundaries, part-of-speech, sentiments etc[1]. Implementation of text mining on high performance computing to process exponentially growing text data is shown in [2]. In [2] research papers were mined to extract information from the entire paper which was previously limited to the abstract, this lead to better understanding of the research as only 20 percent of the information can be found from the abstract.

3 Setting up Jupyter Notebook on Stampede

Anaconda consists of packages such as Python, Jupyter Notebook and commonly used scientific and data science packages. Stampede is a super computer which has Linux based operating system. The steps to install Anaconda and setting its packages are given below:

- On terminal of linux type the following command.
  $>>> wgethttps://3230d63b5fc54e62148e-$
  $c95ac804525aac4b6dba79b00b39d1d3.ssl.cf1.$
  $rackcdn.com/Anaconda2-4.0.0-Linux-x86_64.sh$
  It will download Anaconda*.sh file which is in 64 bit format.
- Install Anaconda by using the following command
  $>>> bashAnaconda2-4.0.0-Linux-x86_64.sh$
  You need to enter many times $>> yes >>$ and also can specify the directory or it will get the default directory.
- Install Python3 because dependency of Jupyterhub is on Python3
  $>>> sudoapt-getinstallpython3-pip$
- Install nodejs/npm
  $>>> sudoapt-getinstallnpmnodejs-legacy$
- Install proxy with npm
  $>>> npminstall-gconfigurable-http-proxy$
- Install Jupyterhub
  $>>> pip3installjupyterhub$
- When we have installed the required dependencies, now we install Jupyter notebook
  $>>> pip3install--upgradenotebook$

Now you need to setup remote Jupyter server on stampede in order to access the Jupyter notebook by your local machine using that server. To do that we need to specify the login node because by default Stampede assigns random login node whenever someone logs in into stampede.

- >>> *ssh username@login*1*.stampede.tacc.u*
  texas.edu

Here at the place of 'username' you need to write the user name of TACC as "tg837687" and we have specified login node that is 'login1', likewise you need to define a login node. You need to enter the TACC password and TACC token. TACC token can be obtained by using "TACC token" application on a smart phone. To start getting the tokens, first you need to log into your TACC account on the web and then scan the QR code using smart phone application.

Afterwards we need to specify the port on which Jupyter server will be accessed.

- >>> *ipythonnotebook* $- -$*no* $-$ *browser* $- -$*port* $=$ 7000

We have specified port 7000 on which our Jupyter server is setup but we can assign any other port if the given port is not available.
The server will be running on the stampede as you can see on the terminal when you have setup the port. Now open a new terminal and create a ssh tunnel directly to stampede and connect to port 7000.

- >>> *ssh*$-$*N*$-$*f*$-$*Llocalhost* : 8888 : *localhost* : 7000*username@login*1*.stampede.tacc.utexas.edu*

Here we specified that '8888' port on local machine will be connected to the remote server on port '7000'.
Now open up a browser and point to

- http://localhost:8888

You will see screen of ipython notebook which is running on Stampede login node. You can also open terminal in the following interface which will show the directories of stampede.

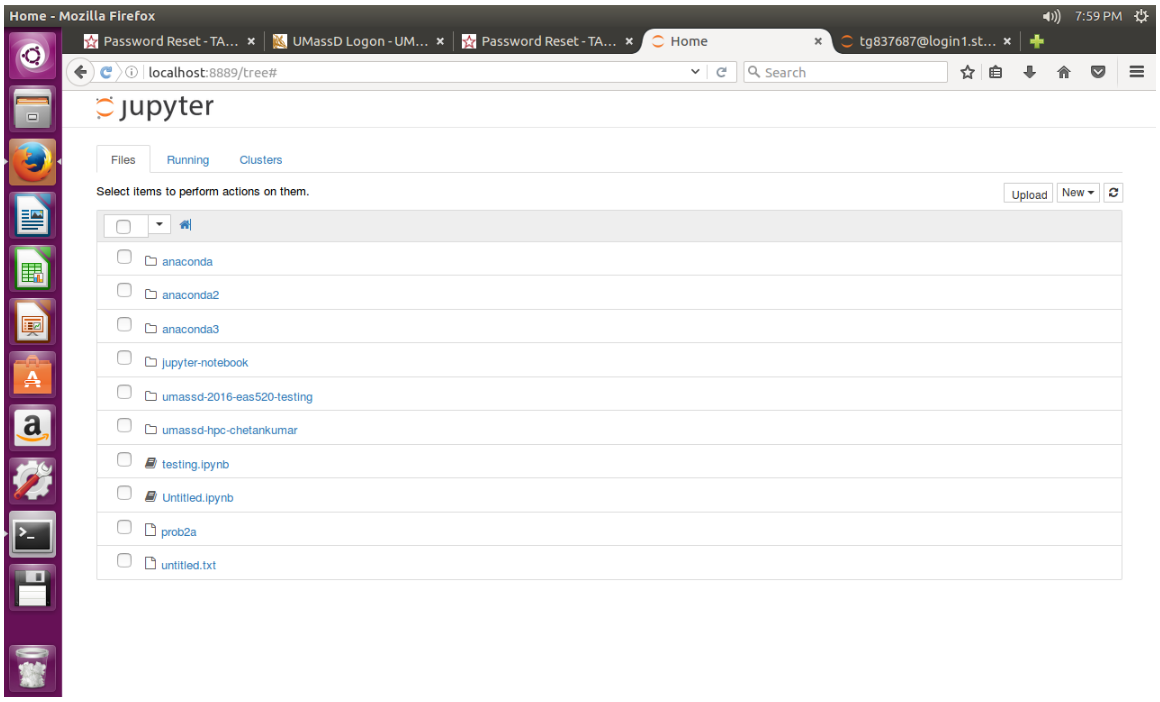
In the following figure **??** you can see the interface of Jupyter notebook running on stampede and remotely accessed by local machine. We can also upload the files using the Jupyter notebook interface directly to stampede as shown in figure **??**. We can access stampede terminal remotely and you can see the uploaded file 'prob2a' in figure **??**.

3.1 Installing NLTK on stampede

First on terminal of jupyter notebook run 'import NLTK' and if it is not available then follow the below steps to install NLTK library on jupyter notebook.
To install NLTK

- >>> pip install -U nltk

To test if installation was successful go to jupyter terminal and write 'python import nltk'.



Select items to perform actions on them.

anaconda
anaconda2
anaconda3
jupyter-notebook
umassd-2016-eas520-testing
umassd-hpc-chetankumar
testing.ipynb
Untitled.ipynb
prob2a
untitled.txt

Figure 1: Jupyter notebook interface on local machine

4 Experimental Work and Results

4.1 HTML Cleaning

We get the HTML code which contains the tags with the text data and we remove the HTML tags in order to get the cleaned text data for processing. The result after cleaning HTML tags can be seen in figure **??**.

4.2 Segmentation

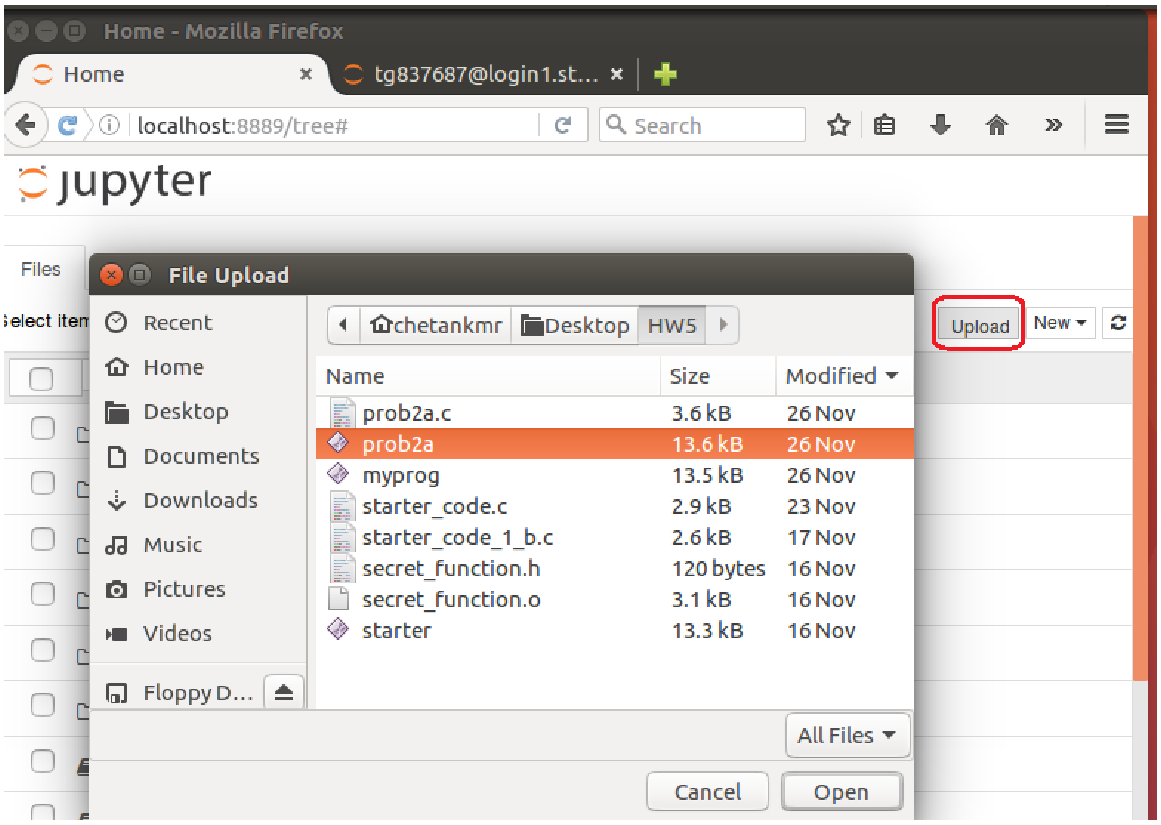
Sentence segmentation is the process of dividing a block of text into individual sentences and the strategy applied is based on punctuation mostly in English and some other languages. The output of the sentence segmentation can be seen in figure **??**.

4.3 Word Tokenization

Word tokenization refers to the process of dividing the sentence into separate words and which acts as a first step towards text processing. The output of the word tokenization is given in figure **??**.

4.4 Parts of Speech (POS) Tagging

In POS tagging, each token is read and part of speech(noun, pronoun, verb, adjective, adverb, preposition, conjunction and interjection) is assigned to it. The output of the POS tagging is given in figure **??**.



Name	Size	Modified
prob2a.c	3.6 kB	26 Nov
prob2a	13.6 kB	26 Nov
myprog	13.5 kB	26 Nov
starter_code.c	2.9 kB	23 Nov
starter_code_1_b.c	2.6 kB	17 Nov
secret_function.h	120 bytes	16 Nov
secret_function.o	3.1 kB	16 Nov
starter	13.3 kB	16 Nov

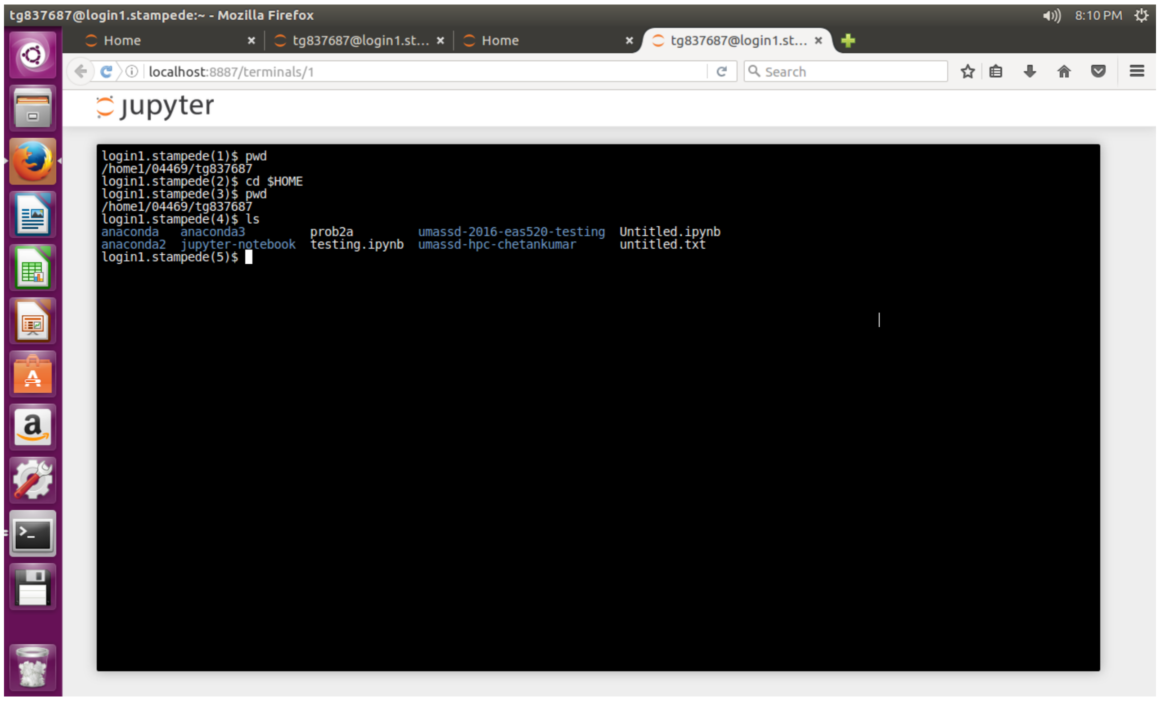
Figure 2: Jupyter notebook interface to upload files

4.5 Chunking

We define some expression such as finding out nouns phrase chunks by using output of POS tagging to find particular words. We can further make use of this information to find the relationship between entities. The result obtained after chunking can be seen in figure 1.In the figures obtained, we can see chunking done to find the simple chunking, proper noun and adjective chunking can be seen in 3 and noun phrase chunking can be seen in figure 2.
Chunking allows us to define expressions to obtain meaningful relationships. In noun phrase chunking we have defined the expression to obtain the relationship between noun and phrase. Similarly in proper noun and adjective chunking we get the relationship between noun and the adjective.

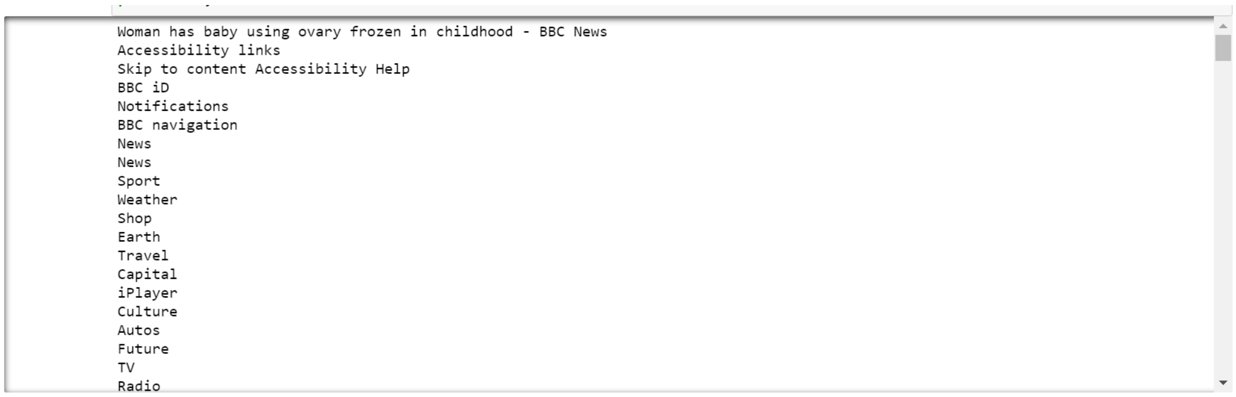
4.6 Extracting entities

From a given block of text we have extracted entities as seen in figure **??**.


tg837687@login1.stampede:~ - Mozilla Firefox
Home
tg837687@login1.st...
Home
tg837687@login1.st...
localhost:8887/terminals/1
Search
jupyter

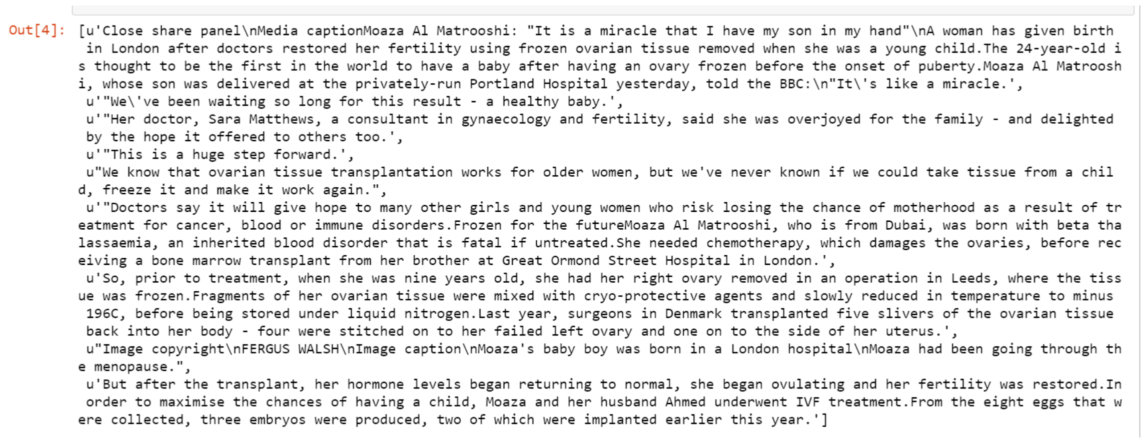
```
login1.stampede(1)$ pwd
/home1/04469/tg837687
login1.stampede(2)$ cd $HOME
login1.stampede(3)$ pwd
/home1/04469/tg837687
login1.stampede(4)$ ls
anaconda   anaconda3         prob2a         umassd-2016-eas520-testing  Untitled.ipynb
anaconda2  jupyter-notebook  testing.ipynb  umassd-hpc-chetankumar      untitled.txt
login1.stampede(5)$
```

Figure 3: Stampede terminal running remotely



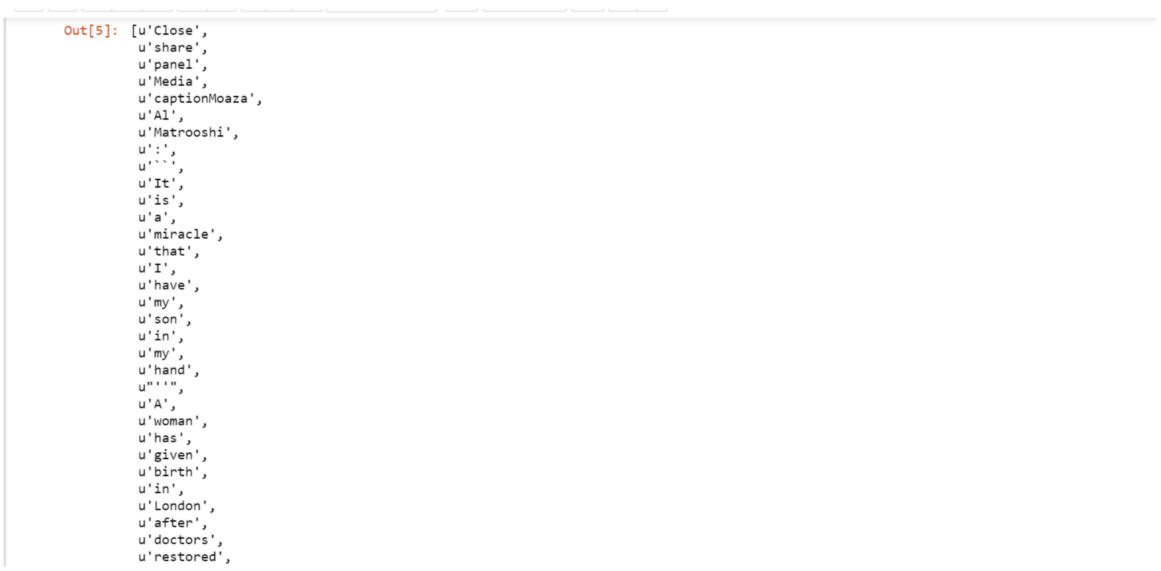
```
Woman has baby using ovary frozen in childhood - BBC News
Accessibility links
Skip to content Accessibility Help
BBC iD
Notifications
BBC navigation
News
News
Sport
Weather
Shop
Earth
Travel
Capital
iPlayer
Culture
Autos
Future
TV
Radio
```

Figure 4: HTML Cleaning



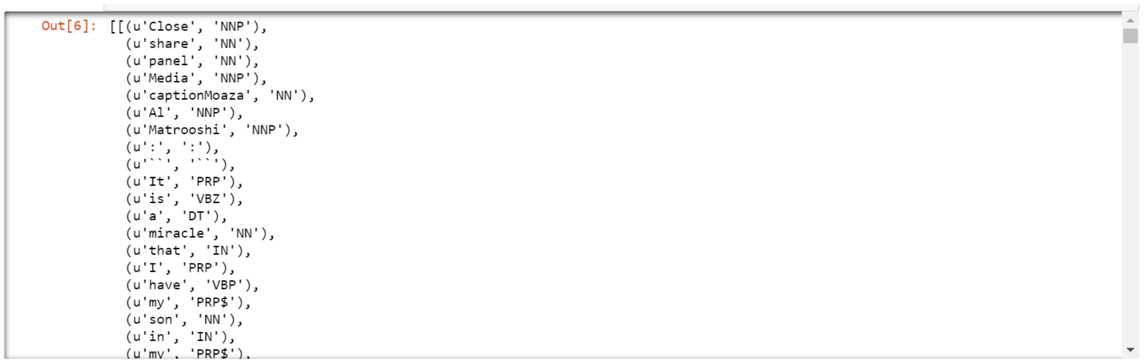
```
Out[4]: [u'Close share panel\nMedia captionMoaza Al Matrooshi: "It is a miracle that I have my son in my hand"\nA woman has given birth
         in London after doctors restored her fertility using frozen ovarian tissue removed when she was a young child.The 24-year-old i
        s thought to be the first in the world to have a baby after having an ovary frozen before the onset of puberty.Moaza Al Matroosh
        i, whose son was delivered at the privately-run Portland Hospital yesterday, told the BBC:\n"It\'s like a miracle.',
         u'"We\'ve been waiting so long for this result - a healthy baby.',
         u'"Her doctor, Sara Matthews, a consultant in gynaecology and fertility, said she was overjoyed for the family - and delighted
         by the hope it offered to others too.',
         u'"This is a huge step forward.',
         u"We know that ovarian tissue transplantation works for older women, but we've never known if we could take tissue from a chil
        d, freeze it and make it work again.",
         u'"Doctors say it will give hope to many other girls and young women who risk losing the chance of motherhood as a result of tr
        eatment for cancer, blood or immune disorders.Frozen for the futureMoaza Al Matrooshi, who is from Dubai, was born with beta tha
        lassaemia, an inherited blood disorder that is fatal if untreated.She needed chemotherapy, which damages the ovaries, before rec
        eiving a bone marrow transplant from her brother at Great Ormond Street Hospital in London.',
         u'So, prior to treatment, when she was nine years old, she had her right ovary removed in an operation in Leeds, where the tiss
        ue was frozen.Fragments of her ovarian tissue were mixed with cryo-protective agents and slowly reduced in temperature to minus
         196C, before being stored under liquid nitrogen.Last year, surgeons in Denmark transplanted five slivers of the ovarian tissue
         back into her body - four were stitched on to her failed left ovary and one on to the side of her uterus.',
         u"Image copyright\nFERGUS WALSH\nImage caption\nMoaza's baby boy was born in a London hospital\nMoaza had been going through th
        e menopause.",
         u'But after the transplant, her hormone levels began returning to normal, she began ovulating and her fertility was restored.In
         order to maximise the chances of having a child, Moaza and her husband Ahmed underwent IVF treatment.From the eight eggs that w
        ere collected, three embryos were produced, two of which were implanted earlier this year.']
```

Figure 5: Sentence Segmentation



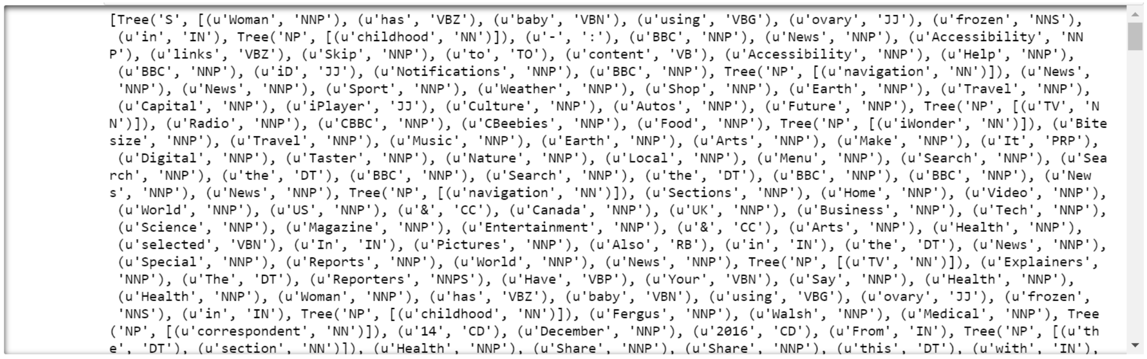
```
Out[5]: [u'Close',
         u'share',
         u'panel',
         u'Media',
         u'captionMoaza',
         u'Al',
         u'Matrooshi',
         u':',
         u'``',
         u'It',
         u'is',
         u'a',
         u'miracle',
         u'that',
         u'I',
         u'have',
         u'my',
         u'son',
         u'in',
         u'my',
         u'hand',
         u"''",
         u'A',
         u'woman',
         u'has',
         u'given',
         u'birth',
         u'in',
         u'London',
         u'after',
         u'doctors',
         u'restored',
```

Figure 6: Word Tokenization



```
Out[6]: [[(u'Close', 'NNP'),
          (u'share', 'NN'),
          (u'panel', 'NN'),
          (u'Media', 'NNP'),
          (u'captionMoaza', 'NN'),
          (u'Al', 'NNP'),
          (u'Matrooshi', 'NNP'),
          (u':', ':'),
          (u'``', '``'),
          (u'It', 'PRP'),
          (u'is', 'VBZ'),
          (u'a', 'DT'),
          (u'miracle', 'NN'),
          (u'that', 'IN'),
          (u'I', 'PRP'),
          (u'have', 'VBP'),
          (u'my', 'PRP$'),
          (u'son', 'NN'),
          (u'in', 'IN'),
          (u'my', 'PRP$'),
```

Figure 7: POS Tagging



```
[Tree('S', [(u'Woman', 'NNP'), (u'has', 'VBZ'), (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'), (u'frozen', 'NNS'),
 (u'in', 'IN'), Tree('NP', [(u'childhood', 'NN')]), (u'-', ':'), (u'BBC', 'NNP'), (u'News', 'NNP'), (u'Accessibility', 'NN
P'), (u'links', 'VBZ'), (u'Skip', 'NNP'), (u'to', 'TO'), (u'content', 'VB'), (u'Accessibility', 'NNP'), (u'Help', 'NNP'),
 (u'BBC', 'NNP'), (u'iD', 'JJ'), (u'Notifications', 'NNP'), (u'BBC', 'NNP'), Tree('NP', [(u'navigation', 'NN')]), (u'News',
 'NNP'), (u'News', 'NNP'), (u'Sport', 'NNP'), (u'Weather', 'NNP'), (u'Shop', 'NNP'), (u'Earth', 'NNP'), (u'Travel', 'NNP'),
 (u'Capital', 'NNP'), (u'iPlayer', 'JJ'), (u'Culture', 'NNP'), (u'Autos', 'NNP'), (u'Future', 'NNP'), Tree('NP', [(u'TV', 'N
N')]), (u'Radio', 'NNP'), (u'CBBC', 'NNP'), (u'CBeebies', 'NNP'), (u'Food', 'NNP'), Tree('NP', [(u'iWonder', 'NN')]), (u'Bite
size', 'NNP'), (u'Travel', 'NNP'), (u'Music', 'NNP'), (u'Earth', 'NNP'), (u'Arts', 'NNP'), (u'Make', 'NNP'), (u'It', 'PRP'),
 (u'Digital', 'NNP'), (u'Taster', 'NNP'), (u'Nature', 'NNP'), (u'Local', 'NNP'), (u'Menu', 'NNP'), (u'Search', 'NNP'), (u'Sea
rch', 'NNP'), (u'the', 'DT'), (u'BBC', 'NNP'), (u'Search', 'NNP'), (u'the', 'DT'), (u'BBC', 'NNP'), (u'BBC', 'NNP'), (u'New
s', 'NNP'), (u'News', 'NNP'), Tree('NP', [(u'navigation', 'NN')]), (u'Sections', 'NNP'), (u'Home', 'NNP'), (u'Video', 'NNP'),
 (u'World', 'NNP'), (u'US', 'NNP'), (u'&', 'CC'), (u'Canada', 'NNP'), (u'UK', 'NNP'), (u'Business', 'NNP'), (u'Tech', 'NNP'),
 (u'Science', 'NNP'), (u'Magazine', 'NNP'), (u'Entertainment', 'NNP'), (u'&', 'CC'), (u'Arts', 'NNP'), (u'Health', 'NNP'),
 (u'selected', 'VBN'), (u'In', 'IN'), (u'Pictures', 'NNP'), (u'Also', 'RB'), (u'in', 'IN'), (u'the', 'DT'), (u'News', 'NNP'),
 (u'Special', 'NNP'), (u'Reports', 'NNP'), (u'World', 'NNP'), (u'News', 'NNP'), Tree('NP', [(u'TV', 'NN')]), (u'Explainers',
 'NNP'), (u'The', 'DT'), (u'Reporters', 'NNPS'), (u'Have', 'VBP'), (u'Your', 'VBN'), (u'Say', 'NNP'), (u'Health', 'NNP'),
 (u'Health', 'NNP'), (u'Woman', 'NNP'), (u'has', 'VBZ'), (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'), (u'frozen',
 'NNS'), (u'in', 'IN'), Tree('NP', [(u'childhood', 'NN')]), (u'Fergus', 'NNP'), (u'Walsh', 'NNP'), (u'Medical', 'NNP'), Tree
('NP', [(u'correspondent', 'NN')]), (u'14', 'CD'), (u'December', 'NNP'), (u'2016', 'CD'), (u'From', 'IN'), Tree('NP', [(u'th
e', 'DT'), (u'section', 'NN')]), (u'Health', 'NNP'), (u'Share', 'NNP'), (u'Share', 'NNP'), (u'this', 'DT'), (u'with', 'IN'),
```

Figure 8: Chunking

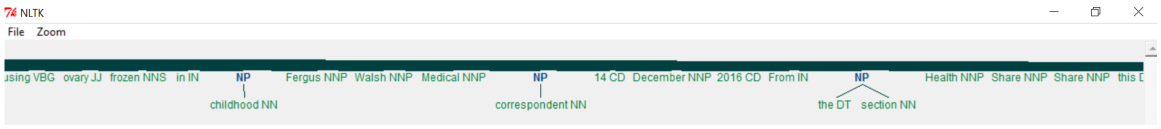
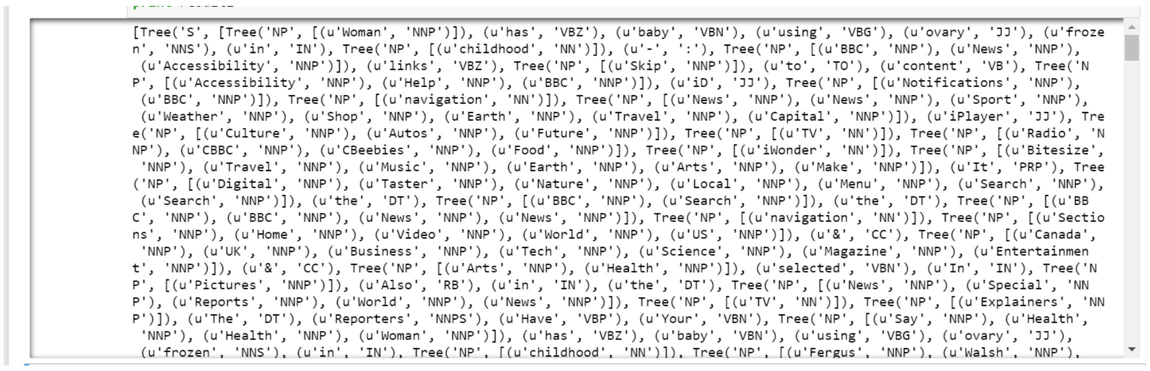
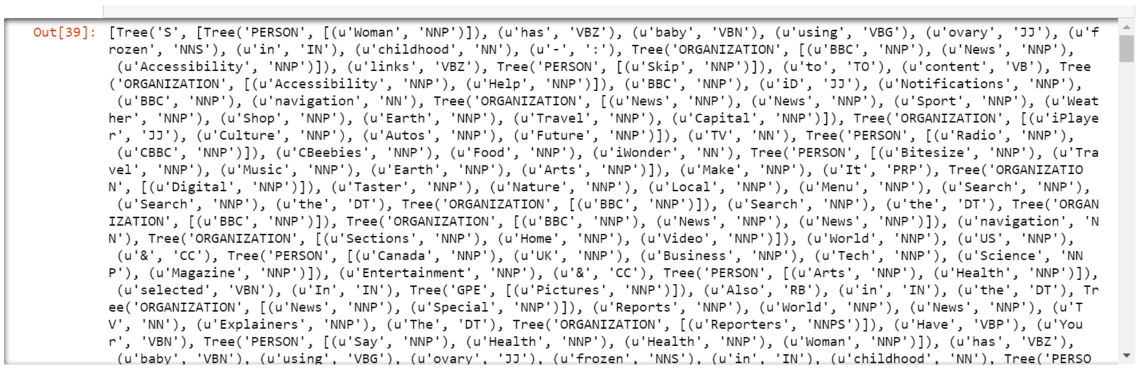


Figure 9: Noun phrase chunking



```
[Tree('S', [Tree('NP', [(u'Woman', 'NNP')]), (u'has', 'VBZ'), (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'), (u'froze
n', 'NNS'), (u'in', 'IN'), Tree('NP', [(u'childhood', 'NN')]), (u'-', ':'), Tree('NP', [(u'BBC', 'NNP'), (u'News', 'NNP'),
 (u'Accessibility', 'NNP')]), (u'links', 'VBZ'), Tree('NP', [(u'Skip', 'NNP')]), (u'to', 'TO'), (u'content', 'VB'), Tree('N
P', [(u'Accessibility', 'NNP'), (u'Help', 'NNP'), (u'BBC', 'NNP')]), (u'iD', 'JJ'), Tree('NP', [(u'Notifications', 'NNP'),
 (u'BBC', 'NNP')]), Tree('NP', [(u'navigation', 'NN')]), Tree('NP', [(u'News', 'NNP'), (u'News', 'NNP'), (u'Sport', 'NNP'),
 (u'Weather', 'NNP'), (u'Shop', 'NNP'), (u'Earth', 'NNP'), (u'Travel', 'NNP'), (u'Capital', 'NNP')]), (u'iPlayer', 'JJ'), Tre
e('NP', [(u'Culture', 'NNP'), (u'Autos', 'NNP'), (u'Future', 'NNP')]), Tree('NP', [(u'TV', 'NN')]), Tree('NP', [(u'Radio', 'N
NP'), (u'CBBC', 'NNP'), (u'CBeebies', 'NNP'), (u'Food', 'NNP')]), Tree('NP', [(u'iWonder', 'NN')]), Tree('NP', [(u'Bitesize',
 'NNP'), (u'Travel', 'NNP'), (u'Music', 'NNP'), (u'Earth', 'NNP'), (u'Arts', 'NNP'), (u'Make', 'NNP')]), (u'It', 'PRP'), Tree
('NP', [(u'Digital', 'NNP'), (u'Taster', 'NNP'), (u'Nature', 'NNP'), (u'Local', 'NNP'), (u'Menu', 'NNP'), (u'Search', 'NNP'),
 (u'Search', 'NNP')]), (u'the', 'DT'), Tree('NP', [(u'BBC', 'NNP'), (u'Search', 'NNP')]), (u'the', 'DT'), Tree('NP', [(u'BB
C', 'NNP'), (u'BBC', 'NNP'), (u'News', 'NNP'), (u'News', 'NNP')]), Tree('NP', [(u'navigation', 'NN')]), Tree('NP', [(u'Sectio
ns', 'NNP'), (u'Home', 'NNP'), (u'Video', 'NNP'), (u'World', 'NNP'), (u'US', 'NNP')]), (u'&', 'CC'), Tree('NP', [(u'Canada',
 'NNP'), (u'UK', 'NNP'), (u'Business', 'NNP'), (u'Tech', 'NNP'), (u'Science', 'NNP'), (u'Magazine', 'NNP'), (u'Entertainmen
t', 'NNP')]), (u'&', 'CC'), Tree('NP', [(u'Arts', 'NNP'), (u'Health', 'NNP')]), (u'selected', 'VBN'), (u'In', 'IN'), Tree('N
P', [(u'Pictures', 'NNP')]), (u'Also', 'RB'), (u'in', 'IN'), (u'the', 'DT'), Tree('NP', [(u'News', 'NNP'), (u'Special', 'NN
P'), (u'Reports', 'NNP'), (u'World', 'NNP'), (u'News', 'NNP')]), Tree('NP', [(u'TV', 'NN')]), Tree('NP', [(u'Explainers', 'NN
P')]), (u'The', 'DT'), (u'Reporters', 'NNPS'), (u'Have', 'VBP'), (u'Your', 'VBN'), Tree('NP', [(u'Say', 'NNP'), (u'Health',
 'NNP'), (u'Health', 'NNP'), (u'Woman', 'NNP')]), (u'has', 'VBZ'), (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'),
 (u'frozen', 'NNS'), (u'in', 'IN'), Tree('NP', [(u'childhood', 'NN')]), Tree('NP', [(u'Fergus', 'NNP'), (u'Walsh', 'NNP'),
```

Figure 10: Proper noun and Adjective Chunking



```
Out[39]: [Tree('S', [Tree('PERSON', [(u'Woman', 'NNP')]), (u'has', 'VBZ'), (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'), (u'f
         rozen', 'NNS'), (u'in', 'IN'), (u'childhood', 'NN'), (u'-', ':'), Tree('ORGANIZATION', [(u'BBC', 'NNP'), (u'News', 'NNP'),
          (u'Accessibility', 'NNP')]), (u'links', 'VBZ'), Tree('PERSON', [(u'Skip', 'NNP')]), (u'to', 'TO'), (u'content', 'VB'), Tree
         ('ORGANIZATION', [(u'Accessibility', 'NNP'), (u'Help', 'NNP')]), (u'BBC', 'NNP'), (u'iD', 'JJ'), (u'Notifications', 'NNP'),
          (u'BBC', 'NNP'), (u'navigation', 'NN'), Tree('ORGANIZATION', [(u'News', 'NNP'), (u'News', 'NNP'), (u'Sport', 'NNP'), (u'Weat
         her', 'NNP'), (u'Shop', 'NNP'), (u'Earth', 'NNP'), (u'Travel', 'NNP'), (u'Capital', 'NNP')]), Tree('ORGANIZATION', [(u'iPlaye
         r', 'JJ'), (u'Culture', 'NNP'), (u'Autos', 'NNP'), (u'Future', 'NNP')]), (u'TV', 'NN'), Tree('PERSON', [(u'Radio', 'NNP'),
          (u'CBBC', 'NNP')]), (u'CBeebies', 'NNP'), (u'Food', 'NNP'), (u'iWonder', 'NN'), Tree('PERSON', [(u'Bitesize', 'NNP'), (u'Tra
         vel', 'NNP'), (u'Music', 'NNP'), (u'Earth', 'NNP'), (u'Arts', 'NNP')]), (u'Make', 'NNP'), (u'It', 'PRP'), Tree('ORGANIZATIO
         N', [(u'Digital', 'NNP')]), (u'Taster', 'NNP'), (u'Nature', 'NNP'), (u'Local', 'NNP'), (u'Menu', 'NNP'), (u'Search', 'NNP'),
          (u'Search', 'NNP'), (u'the', 'DT'), Tree('ORGANIZATION', [(u'BBC', 'NNP')]), (u'Search', 'NNP'), (u'the', 'DT'), Tree('ORGAN
         IZATION', [(u'BBC', 'NNP')]), Tree('ORGANIZATION', [(u'BBC', 'NNP'), (u'News', 'NNP'), (u'News', 'NNP')]), (u'navigation', 'N
         N'), Tree('ORGANIZATION', [(u'Sections', 'NNP'), (u'Home', 'NNP'), (u'Video', 'NNP')]), (u'World', 'NNP'), (u'US', 'NNP'),
          (u'&', 'CC'), Tree('PERSON', [(u'Canada', 'NNP'), (u'UK', 'NNP'), (u'Business', 'NNP'), (u'Tech', 'NNP'), (u'Science', 'NN
         P'), (u'Magazine', 'NNP')]), (u'Entertainment', 'NNP'), (u'&', 'CC'), Tree('PERSON', [(u'Arts', 'NNP'), (u'Health', 'NNP')]),
          (u'selected', 'VBN'), (u'In', 'IN'), Tree('GPE', [(u'Pictures', 'NNP')]), (u'Also', 'RB'), (u'in', 'IN'), (u'the', 'DT'), Tr
         ee('ORGANIZATION', [(u'News', 'NNP'), (u'Special', 'NNP')]), (u'Reports', 'NNP'), (u'World', 'NNP'), (u'News', 'NNP'), (u'T
         V', 'NN'), (u'Explainers', 'NNP'), (u'The', 'DT'), Tree('ORGANIZATION', [(u'Reporters', 'NNPS')]), (u'Have', 'VBP'), (u'You
         r', 'VBN'), Tree('PERSON', [(u'Say', 'NNP'), (u'Health', 'NNP'), (u'Health', 'NNP'), (u'Woman', 'NNP')]), (u'has', 'VBZ'),
          (u'baby', 'VBN'), (u'using', 'VBG'), (u'ovary', 'JJ'), (u'frozen', 'NNS'), (u'in', 'IN'), (u'childhood', 'NN'), Tree('PERSO
```

Figure 11: Extracting Entities

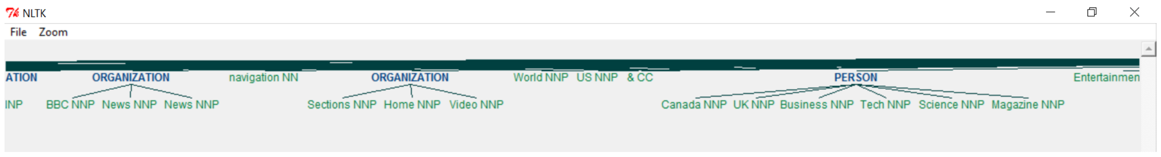

NLTK
File Zoom
ATION
ORGANIZATION
navigation NN
ORGANIZATION
World NNP US NNP & CC
PERSON
Entertainmen
NNP
BBC NNP News NNP News NNP
Sections NNP Home NNP Video NNP
Canada NNP UK NNP Business NNP Tech NNP Science NNP Magazine NNP

Figure 12: Entity Extraction Tree

5 Conclusion and Future Work

We have managed to successfully process text on Jupyter notebook running on Stampede, but all the computations were carried out on the login node. Looking ahead we will be trying to connect to the compute node of stampede and process even larger text documents.

6 Bitbucket Link

The code has been uploaded on the bitbucket, you can access the code on the given link
https://bitbucket.org/dkumar2/dsc520hpc.git

References

[1] Miyao, Y., T. Ohta, et al. *Semantic Retrieval for the Accurate Identification of Relational Concepts in Massive Textbases. Coling/ACL, Sydney, Australia, Association for Computational Linguistics.*. 2006.

[2] Firat Tekiner, Yoshimasa Tsuruoka, Junichi Tsujii *Highly scalable Text Mining - parallel tagging application.* 2006.

[3] Matthew A.Russell *Mining the Social Web.* 2006.

[4] Steven Bird, Ewan Klein, Edward Loper *Natural Language Processing with Python.* 2006.