DBMS Assignment - 5

Text mining using apache spark

Problem statement

Implementation of various text mining paradigms on wiki pages.

The various paradigms which were implemented were:

1. Single word frequency
2. Two word frequency
3. Cosine similarity
4. POS tagging

1. Single word frequency

Simple word count was applied to the wiki corpus which was implemented using pyspark.

The graph of the most frequent words was plotted against their frequencies

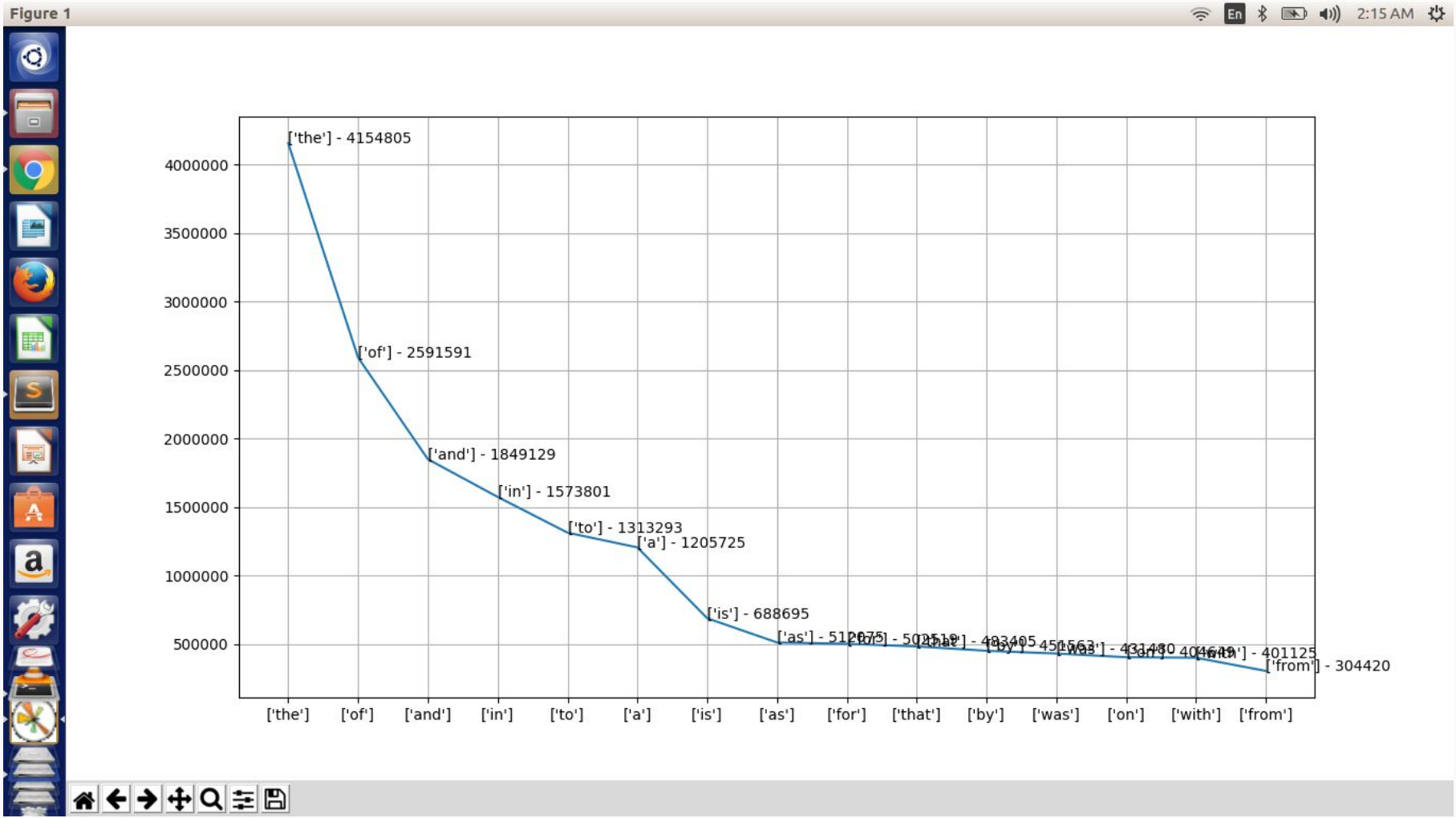


Figure 1
2:15 AM
['the'] - 4154805
['of'] - 2591591
['and'] - 1849129
['in'] - 1573801
['to'] - 1313293
['a'] - 1205725
['is'] - 688695
['as'] - 512075
['for'] - 502519
['that'] - 483405
['by'] - 451563
['was'] - 431480
['on'] - 404619
['with'] - 401125
['from'] - 304420
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
['the']
['of']
['and']
['in']
['to']
['a']
['is']
['as']
['for']
['that']
['by']
['was']
['on']
['with']
['from']

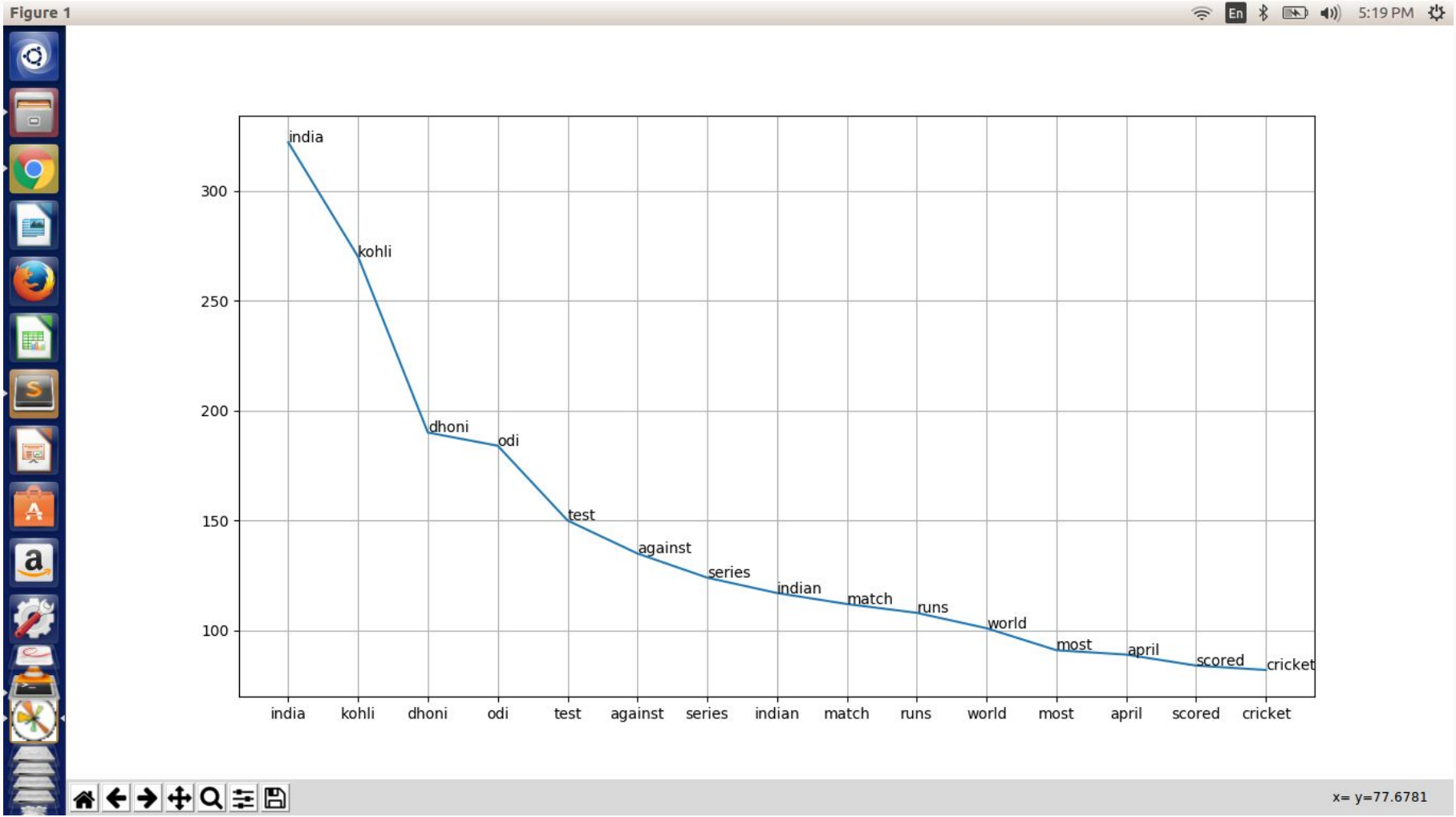


Figure 1
5:19 PM
300
250
200
150
100
india
kohli
dhoni
odi
test
against
series
indian
match
runs
world
most
april
scored
cricket
x= y=77.6781

2. Two word frequency

The most frequently used two word pairs were calculated using the RDDs available in pyspark. The words were split and necessary filters were applied to remove unnecessary words.

The graph of the most frequent words was plotted against their frequencies

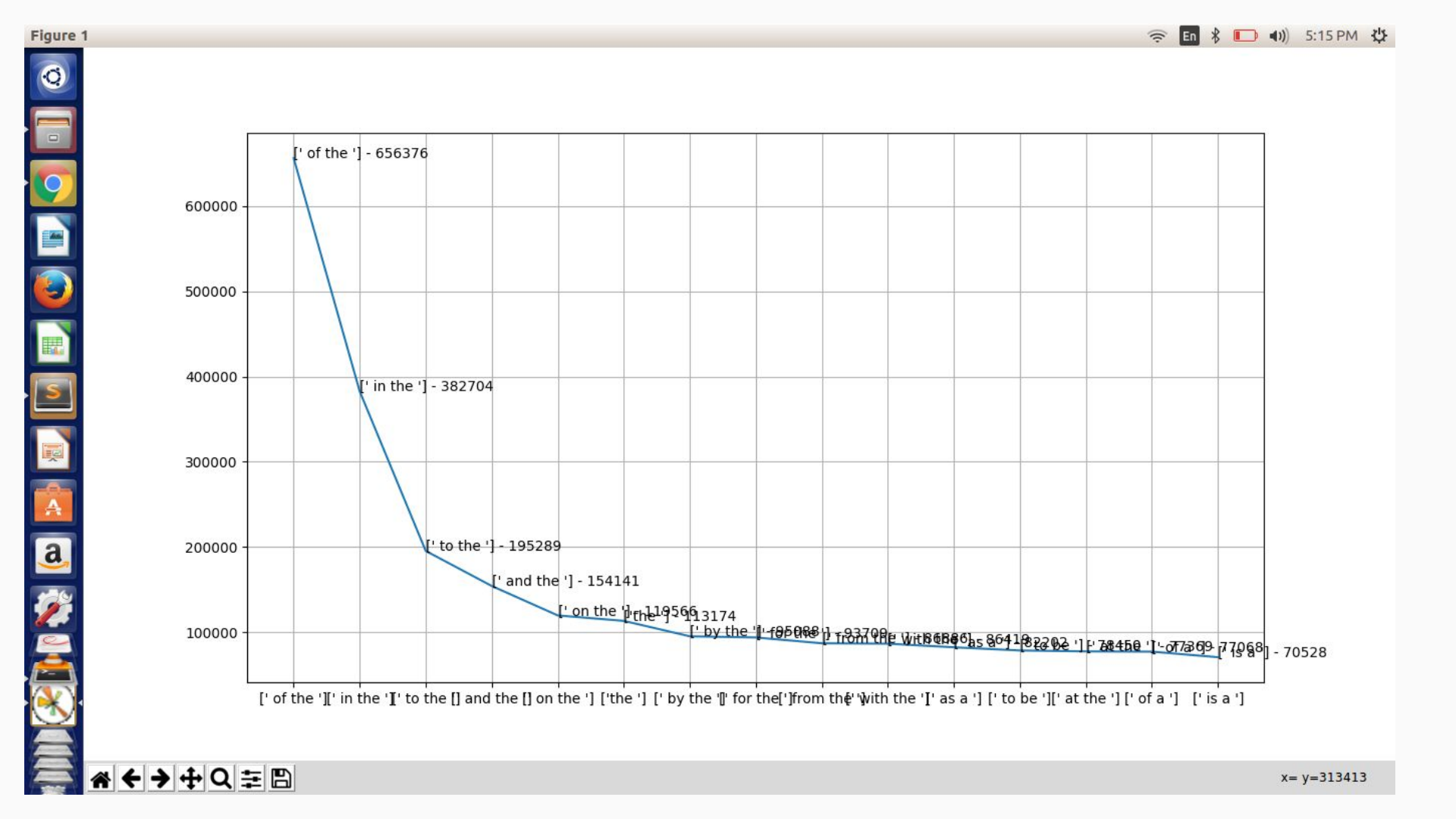
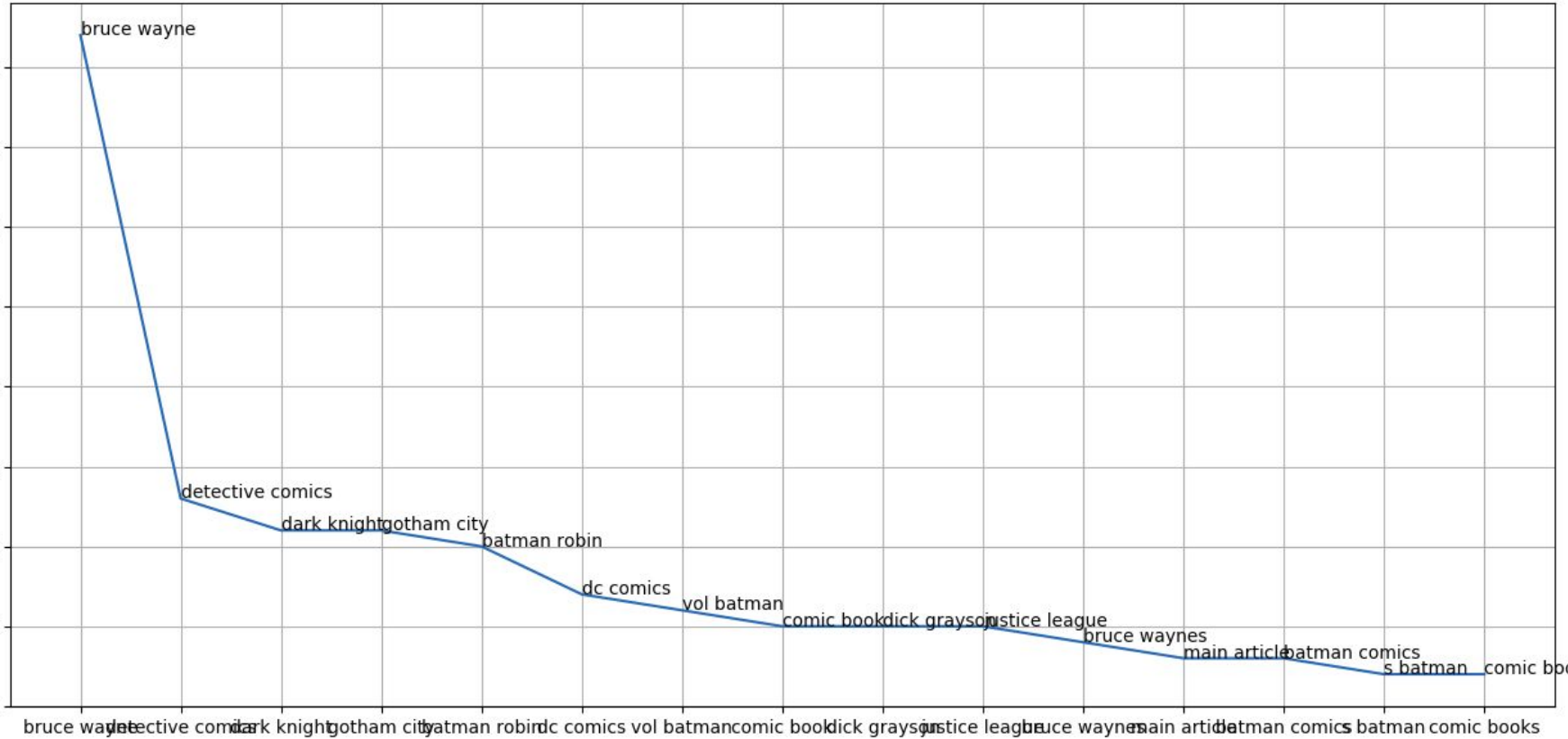
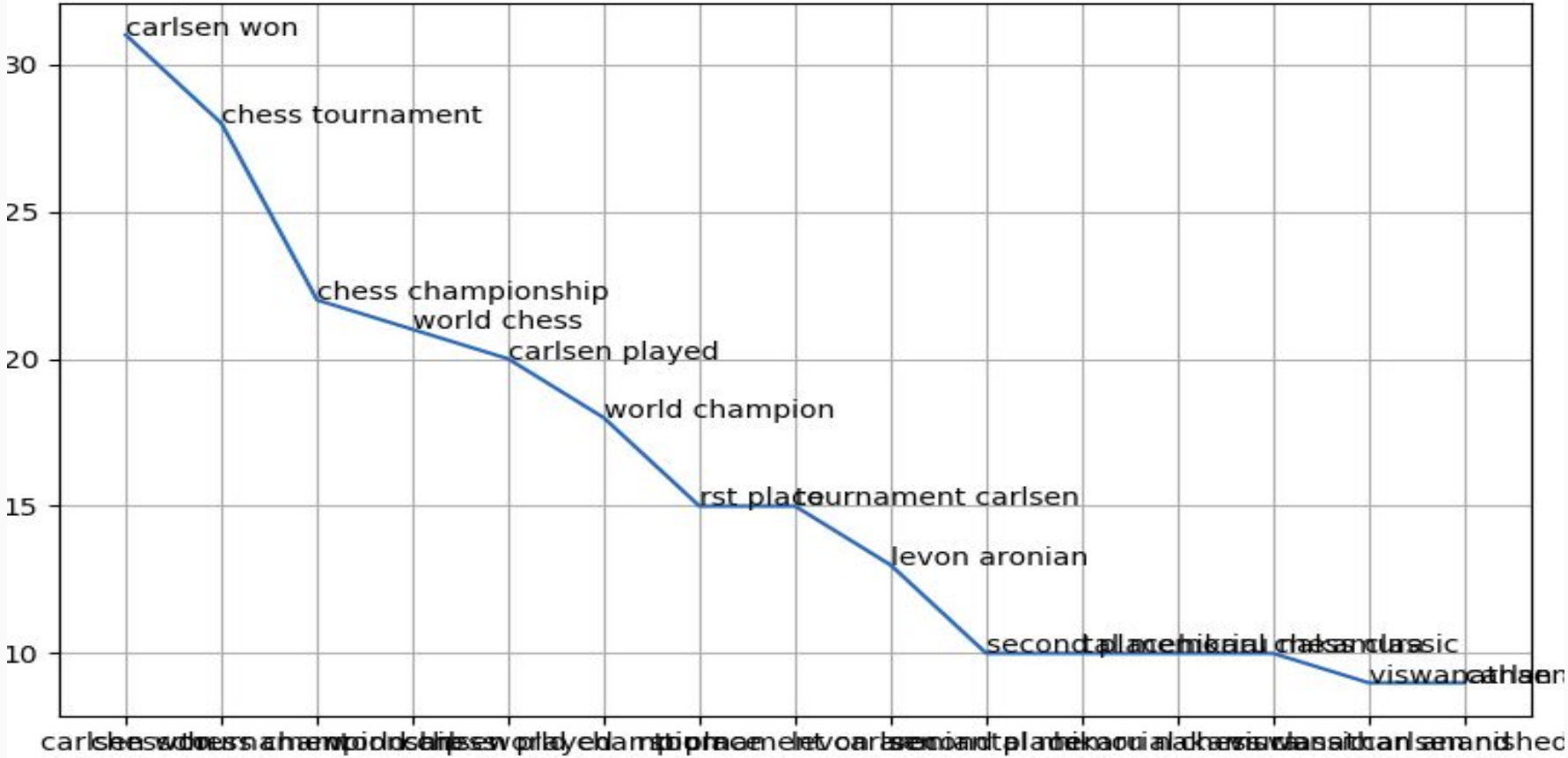
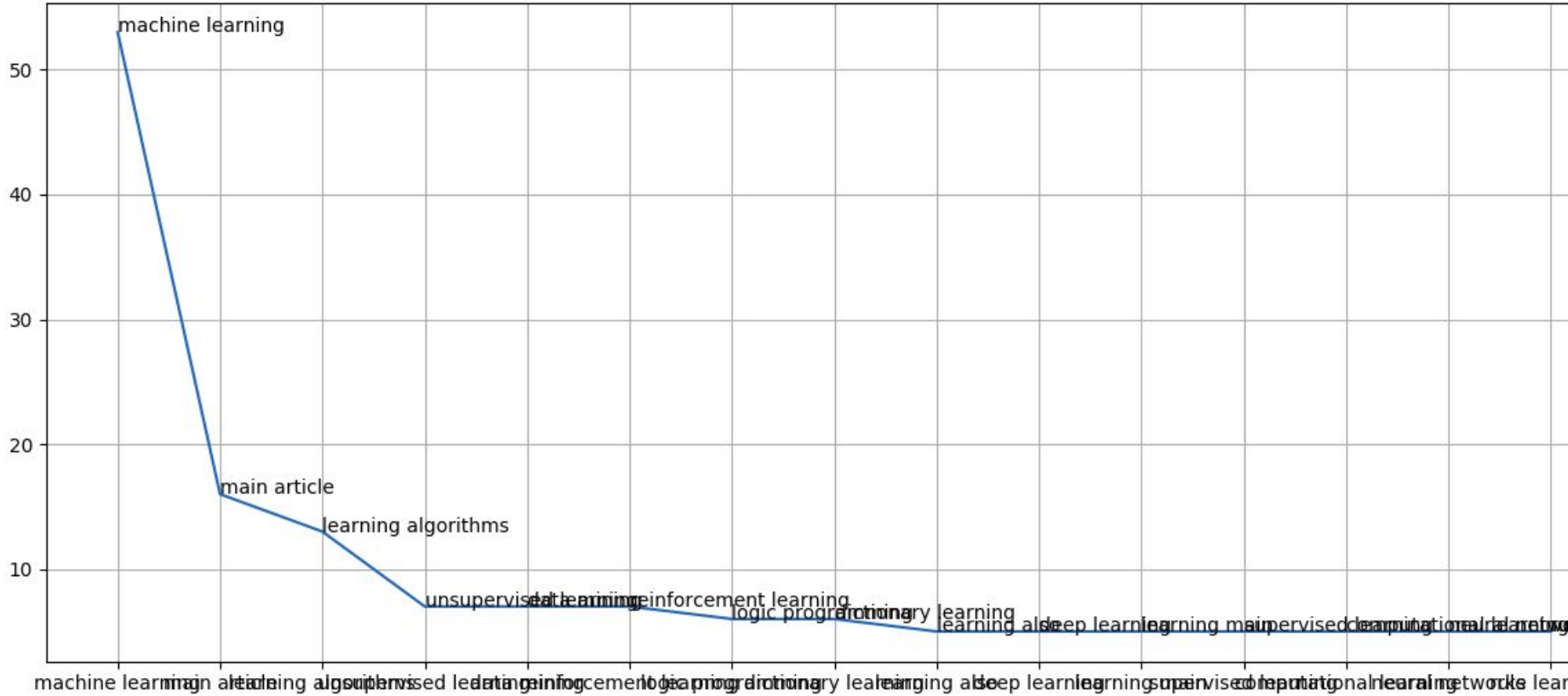


Figure 1
5:15 PM
[' of the '] - 656376
[' in the '] - 382704
[' to the '] - 195289
[' and the '] - 154141
[' on the '] - 119566
['the '] - 113174
[' by the '] - 95088
[' for the '] - 93709
[' from the '] - 86886
[' with the '] - 86418
[' as a '] - 82202
[' to be '] - 78450
[' at the '] - 77369
[' of a '] - 77068
[' is a '] - 70528
600000
500000
400000
300000
200000
100000
[' of the '][' in the '][' to the '] and the [] on the '] ['the '] [' by the '] [' for the '] from the '] [' with the '] as a '] [' to be '] [' at the '] [' of a '] [' is a ']
x= y=313413


bruce wayne
detective comics
dark knight
gotham city
batman robin
dc comics
vol batman
comic book
dick grayson
justice league
bruce waynes
main article
batman comics
s batman
comic books
50
45
40
35
30
25
20
15
10


carlsen won
chess tournament
chess championship
world chess
carlsen played
world champion
rst place
ournament carlsen
levon aronian
second
viswan
30
25
20
15
10


machine learning
main article
learning algorithms
50
40
30
20
10

3. Cosine similarity

Calculating the term frequency i.e number of times word occurs in a document.The frequency can be calculated by normalization(by dividing with the number of words in the document).

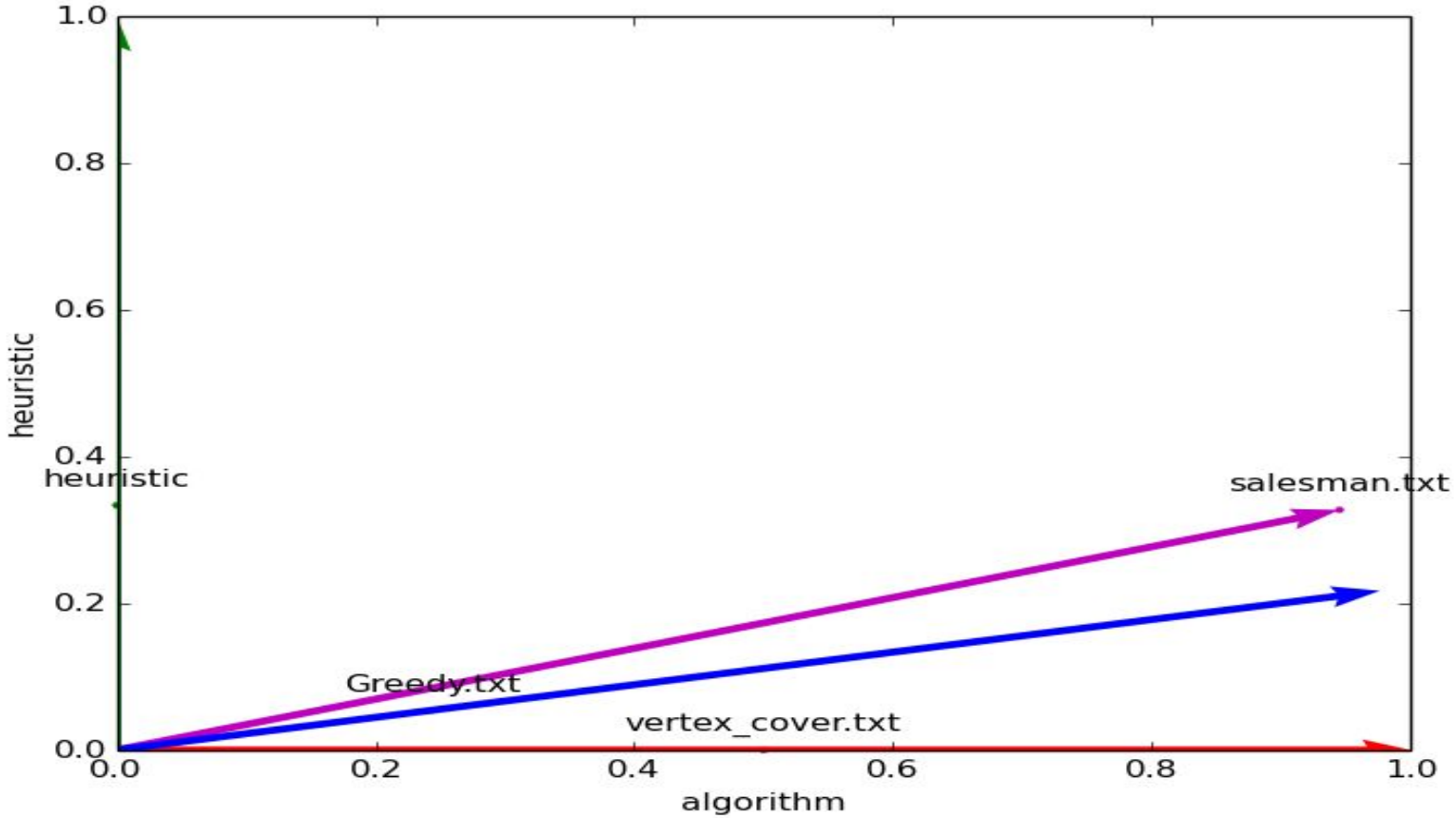
Then we find the Inverse Document frequency by:

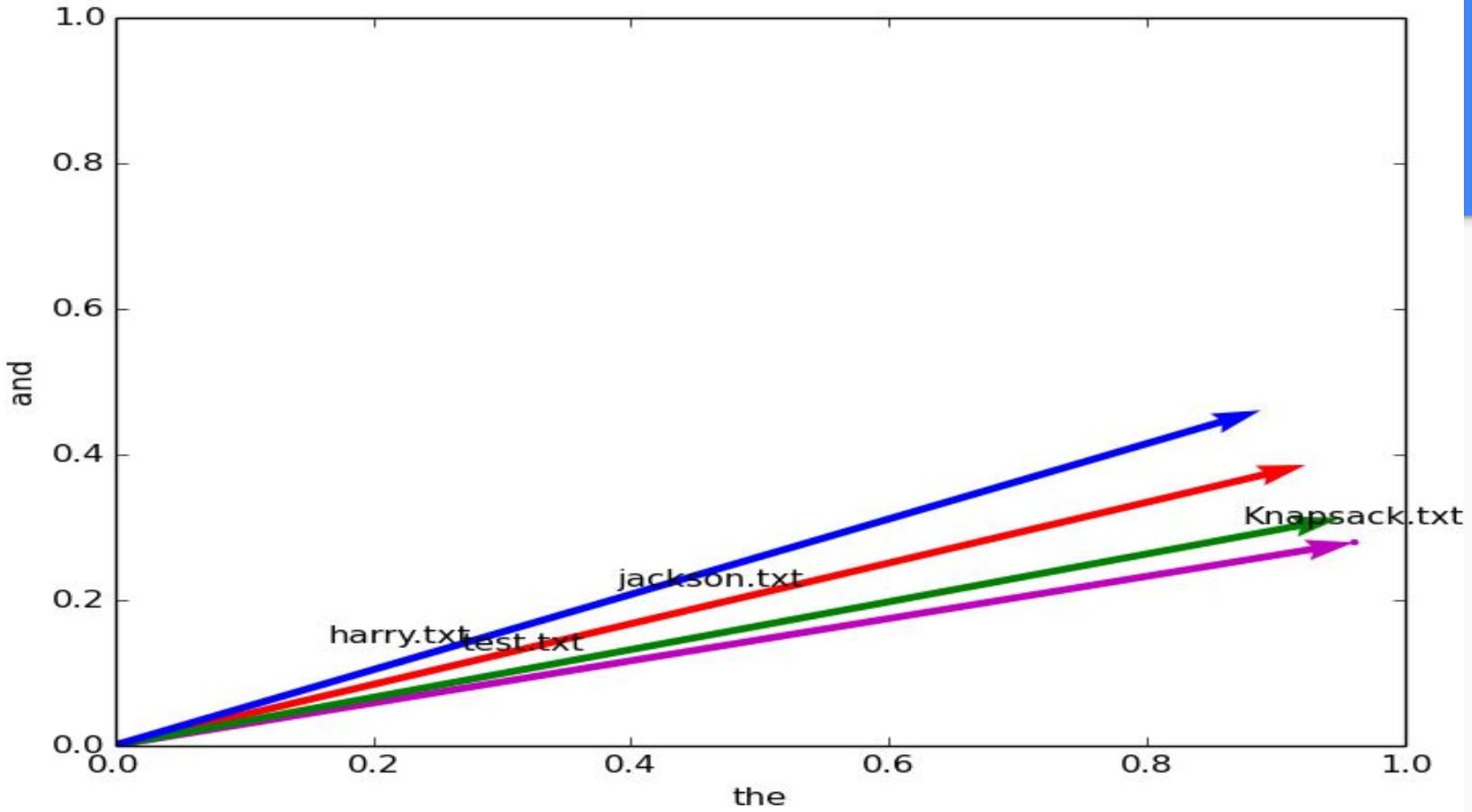
IDF(**word**) = 1 + $\log_e$(Total Number Of Documents / Number Of Documents with term **word** in it)

Since we are finding out the relevancy of each document with respect to the query of two words,we multiply and store the product of term frequency and Inverse document frequency of each word in query in each document.

Once the vectors are drawn,the angle between them can be computed by dot product,and the similarity between the lines is the measure of angle between them.

Corresponding graphs are drawn to represent this feature.


1.0
0.8
0.6
0.4
0.2
0.0
heuristic
heuristic
salesman.txt
Greedy.txt
vertex_cover.txt
0.0
0.2
0.4
0.6
0.8
1.0
algorithm

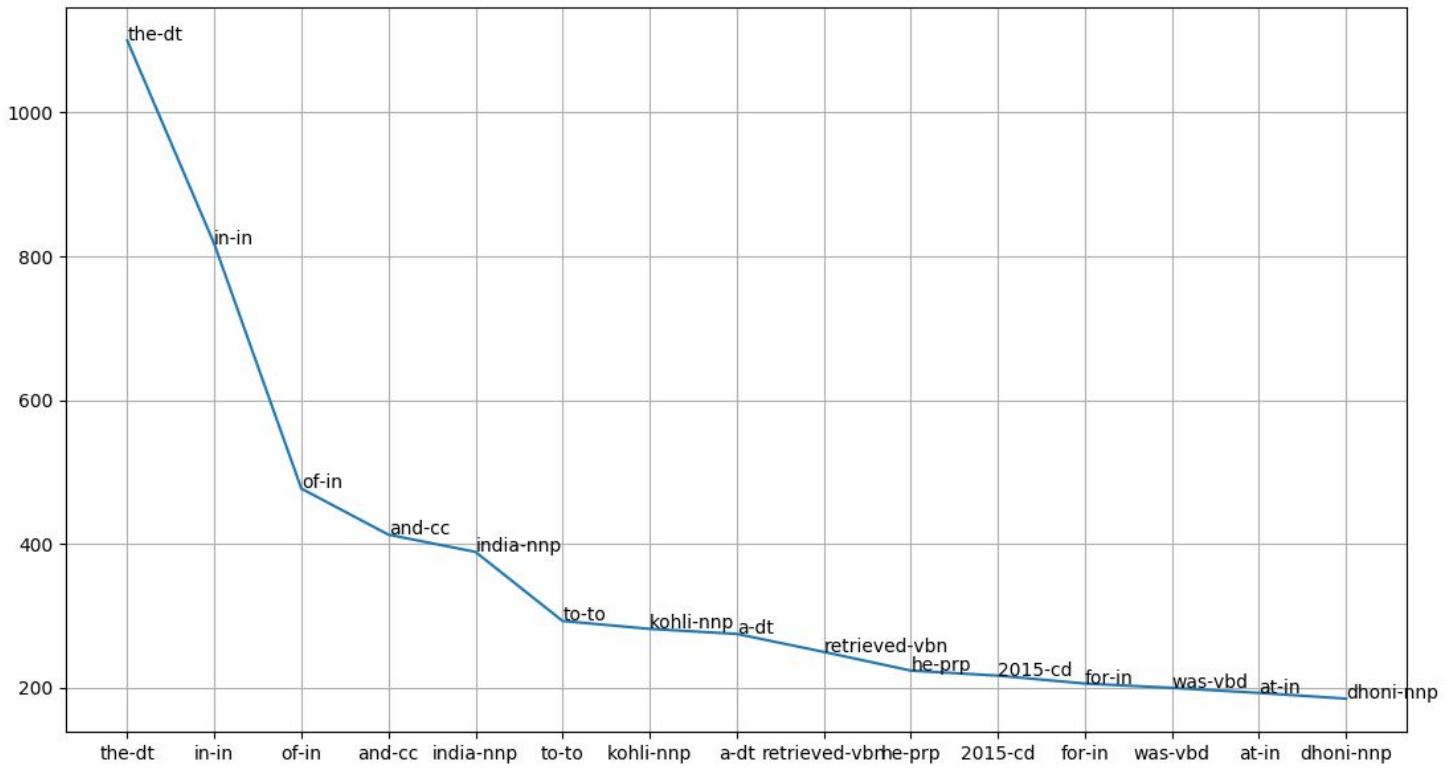

1.0
0.8
0.6
0.4
0.2
0.0
and
0.0
0.2
0.4
0.6
0.8
1.0
the
harry.txt
test.txt
jackson.txt
Knapsack.txt

4. POS tagging

Each word in a sentence is tagged corresponding to its part of speech, based on its definition and context.

NLTK library of python was used in implementing this paradigm.

The graph of the most frequent words along with their POS tags were plotted against their frequencies


the-dt
in-in
of-in
and-cc
india-nnp
to-to
kohli-nnp
a-dt
retrieved-vbn
he-prp
2015-cd
for-in
was-vbd
at-in
dhoni-nnp
1000
800
600
400
200
the-dt
in-in
of-in
and-cc
india-nnp
to-to
kohli-nnp
a-dt
retrieved-vbn
he-prp
2015-cd
for-in
was-vbd
at-in
dhoni-nnp

THANK YOU