Answers - CS 545, Fall 2012, Homework 2

Kumaresh Visakan Murugan
mvisakan@cs.wisc.edu

Note:

- **Individual Plots are placed in separate PDF files for clearer view. Information regarding it is clearly mentioned below.**
- **All log operations are consistently assumed with base 2.**

1. Answer to Question 1
    - For each Language, the lemma with maximum number of word types are listed below:

Language Name	**Language Code**	**Lemma with max # word types**	**# Word Types**
English	En	be	18
Bulgarian	Bg	мога	27
Serbian	Sr	biti	36
Slovak	Sk	veľký	36
Farsi	Fa	ﻦ ﻜ	37
Hungarian	Hu	maga	45
Romanian	Ro	fi	48
Estonian	Et	olema	51
Polish	Pl	Być	52
Slovene	Sl	Biti	55
Czech	Cs	Být	73

English has the least morphological complexity by this measure. The "**be**" is the lemma with maximum word types of 18.

- For each language, the average number of word forms that each lemma can take (ratio of unique word types to unique lemma types) are listed below:

Language Name	**Language Code**	**Average Word Forms per Lemma**
English	En	1.3803135150
Farsi	Fa	1.7126002118
Bulgarian	Bg	1.9180154821
Slovak	Sk	2.0088241126
Hungarian	Hu	2.0197773802
Estonian	Et	2.0431597023
Romanian	Ro	2.0941889287
Czech	Cs	2.0947633655
Slovene	Sl	2.1497474140
Serbian	Sr	2.1498754005
Polish	Pl	2.2247914687

With a rough analysis of comparing the ranks obtained by each of the 11 languages by both morphological complexity analysis, we obtain a graph as below.

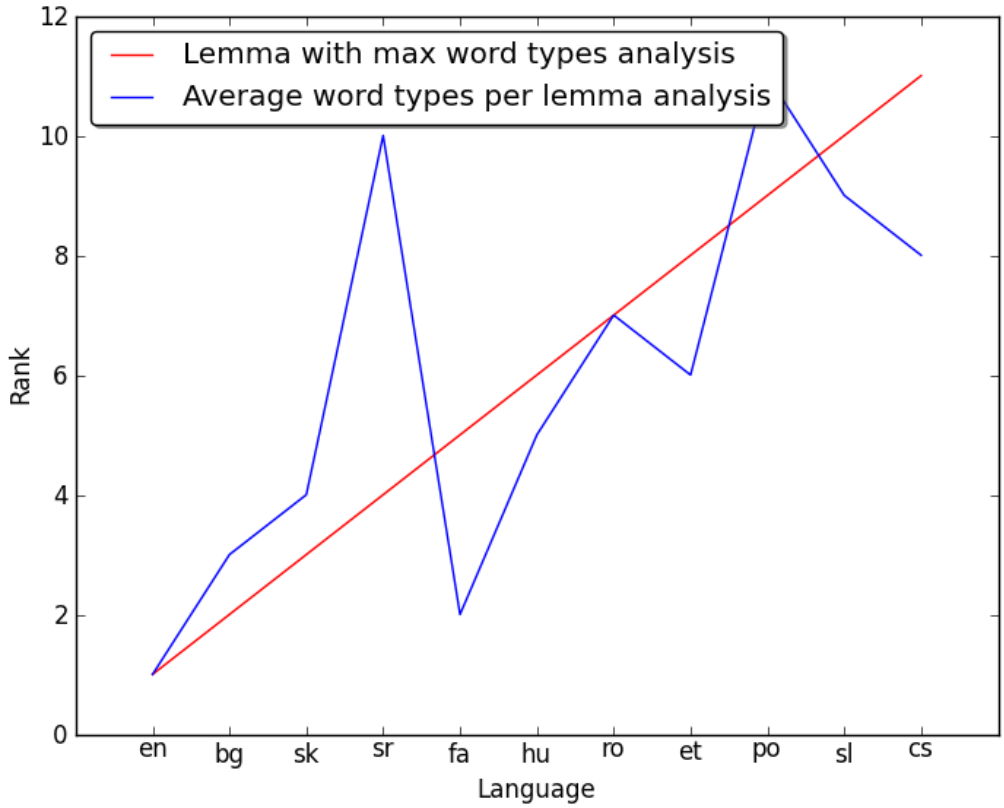
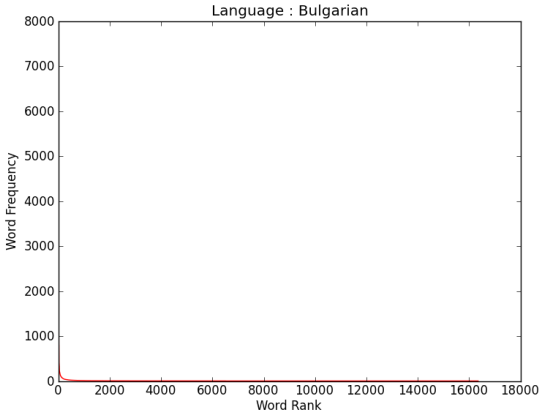


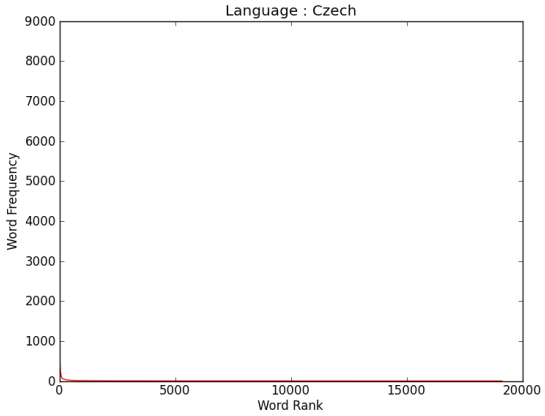
Fig 1.1 Plot explaining the results of both the morphological complexity analysis methods

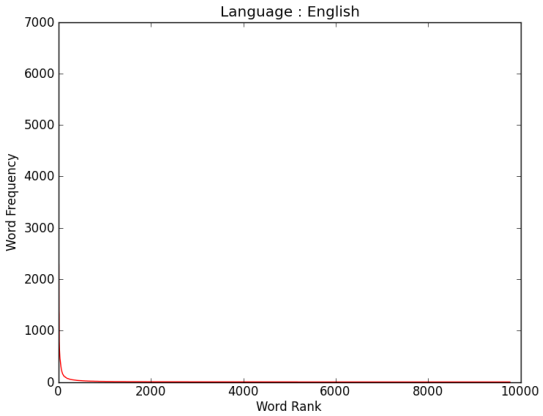
The ascending pattern in Fig1.1 suggests that the results of both the morphological complexity analysis methods are roughly consistent with each other.

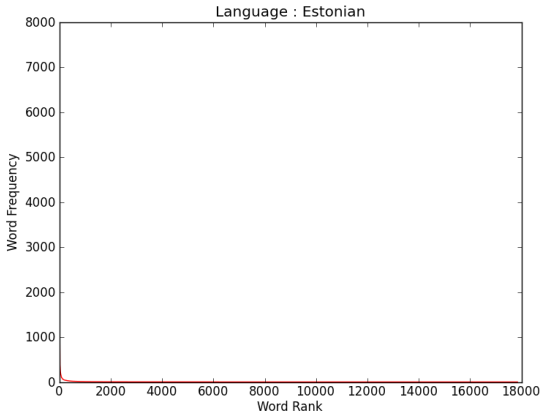
2. Answer to Question 2
    - For each language, a plot is created between the empirical rank of words (x-axis) and frequency of those words (y-axis)

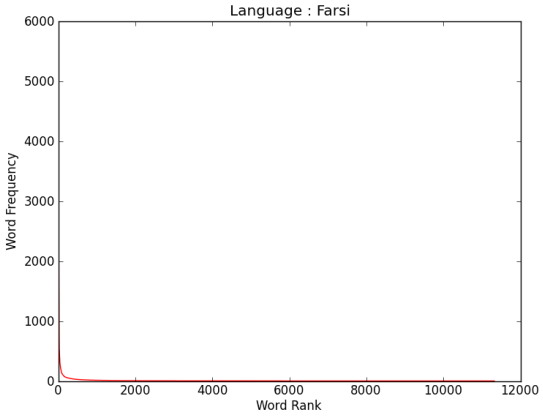
Fig 2.1 Plots based on word frequency to illustrate zipfian distributions/power law

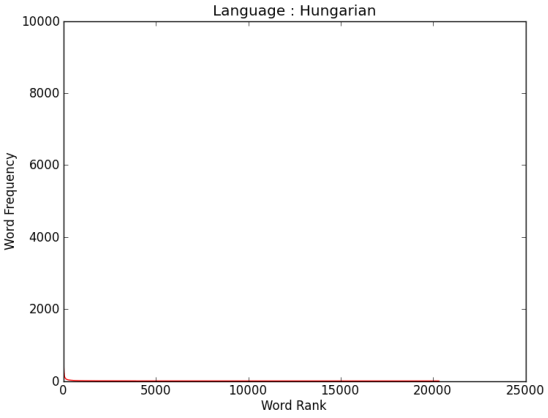


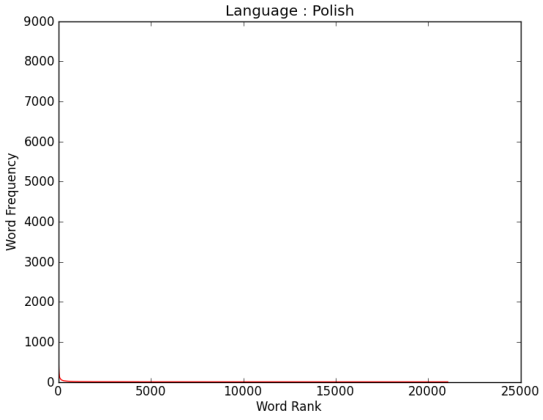


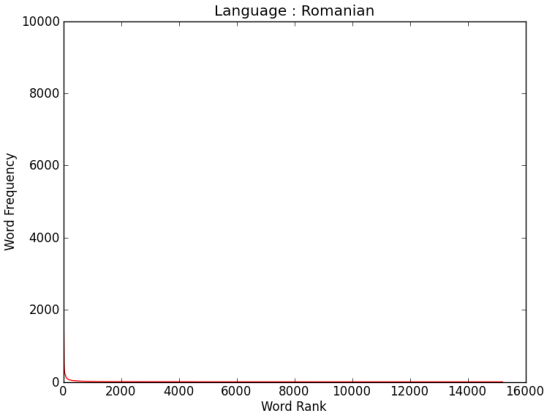

Language : English
Word Frequency
Word Rank
0
1000
2000
3000
4000
5000
6000
7000
0
2000
4000
6000
8000
10000

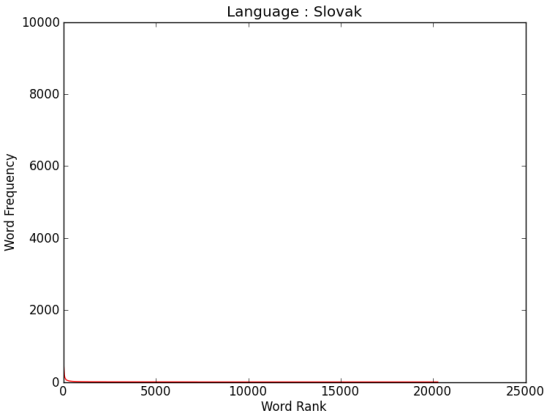

Language : Estonian
Word Frequency
Word Rank
0
1000
2000
3000
4000
5000
6000
7000
8000
0
2000
4000
6000
8000
10000
12000
14000
16000
18000

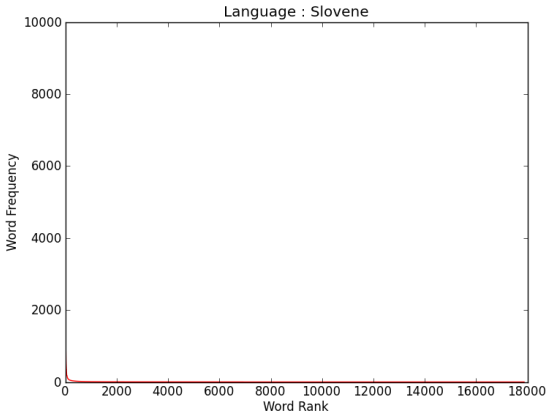

Language : Farsi
Word Frequency
Word Rank
0
1000
2000
3000
4000
5000
6000
0
2000
4000
6000
8000
10000
12000

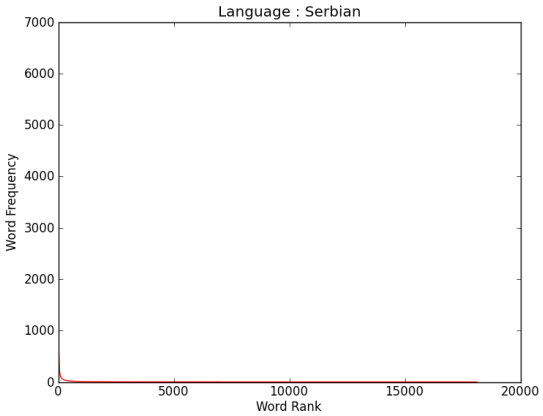

Language : Hungarian
Word Frequency
Word Rank
0
2000
4000
6000
8000
10000
0
5000
10000
15000
20000
25000


Language : Polish
Word Frequency
Word Rank
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
0
5000
10000
15000
20000
25000


Language : Romanian
Word Frequency
Word Rank
0
2000
4000
6000
8000
10000
0
2000
4000
6000
8000
10000
12000
14000
16000



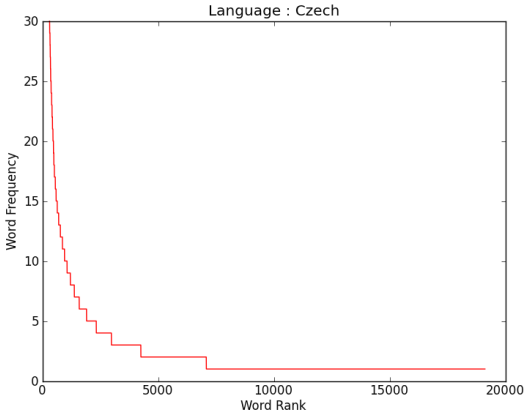


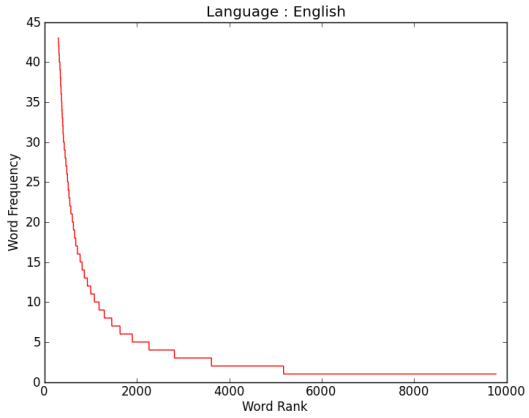


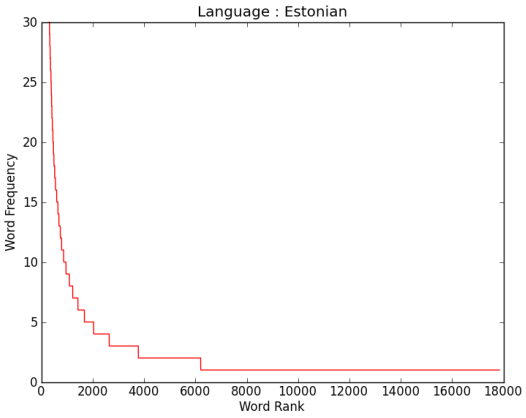
The plots for all the languages look similar. Though the plots are hard to see, **they appear to be power laws.**

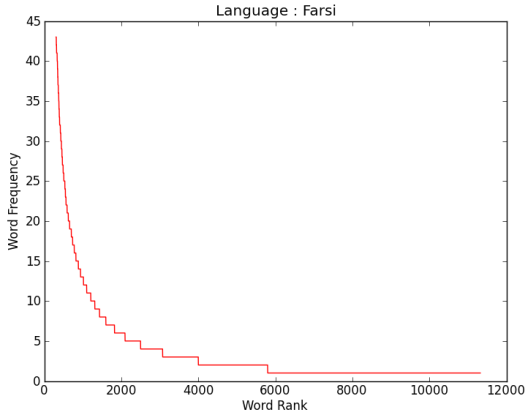
- In order to visualize the fat tail of the plots, let's begin at *rank*(*w*) = 300. Now the plots look like:

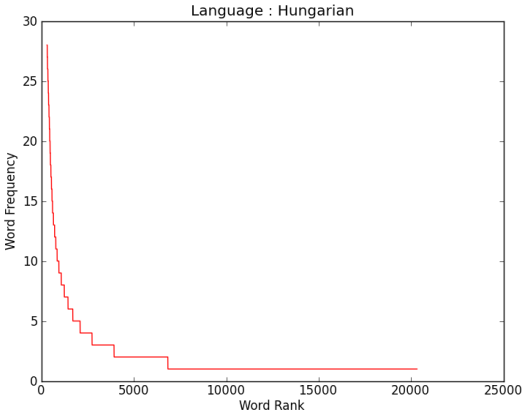
Fig 2.2 Plots based on word frequency to visualize the fat tail in zipfian distributions/power law

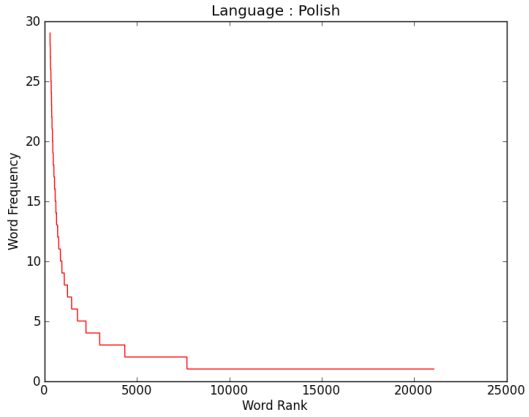


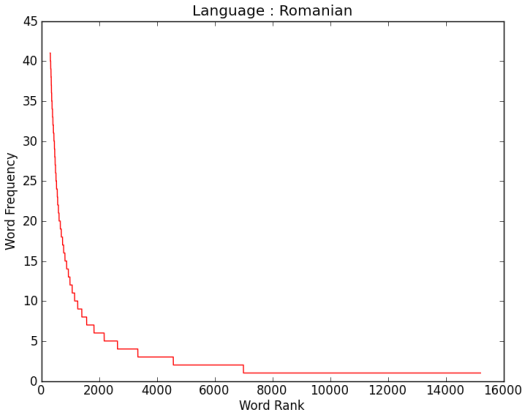

Language : English
Word Frequency
Word Rank

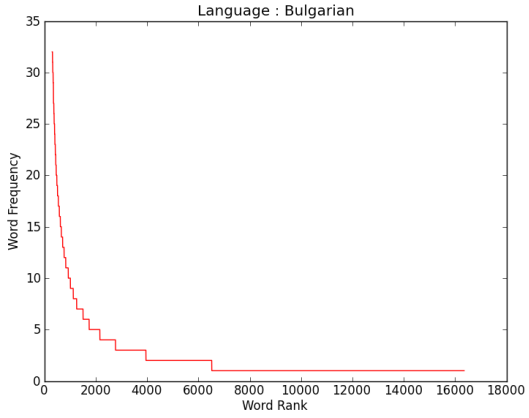

Language : Estonian
Word Frequency
Word Rank

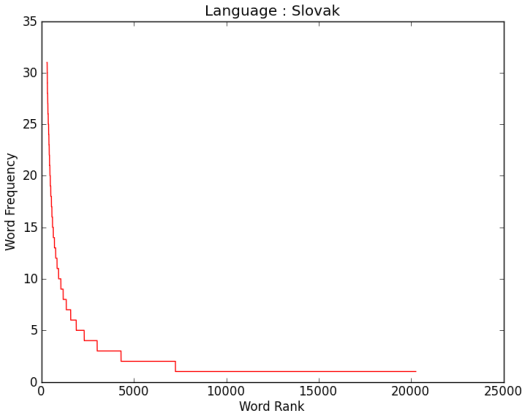

Language : Farsi
Word Frequency
Word Rank

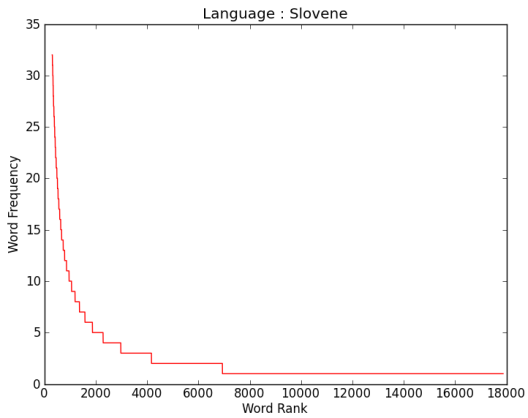

Language : Hungarian
Word Frequency
Word Rank

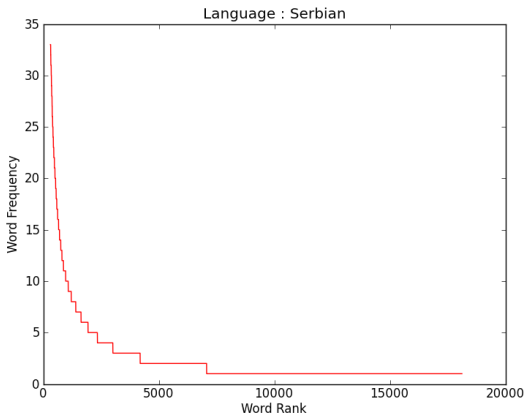

Language : Polish
Word Frequency
Word Rank


Language : Romanian
Word Frequency
Word Rank


Language : Bulgarian
Word Frequency
Word Rank


Language : Slovak
Word Frequency
Word Rank

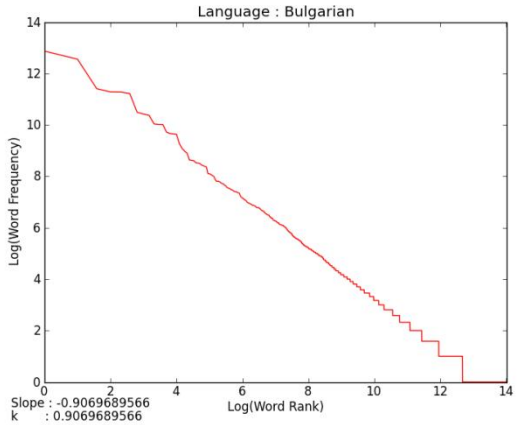


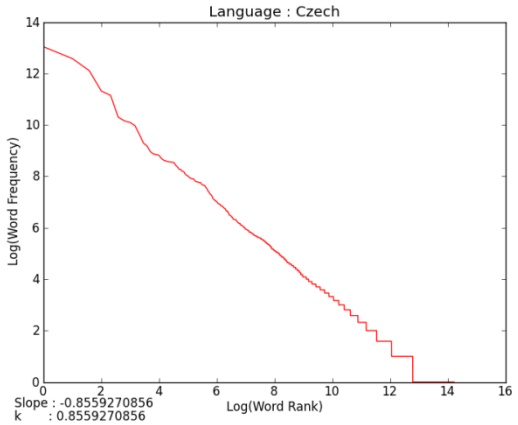


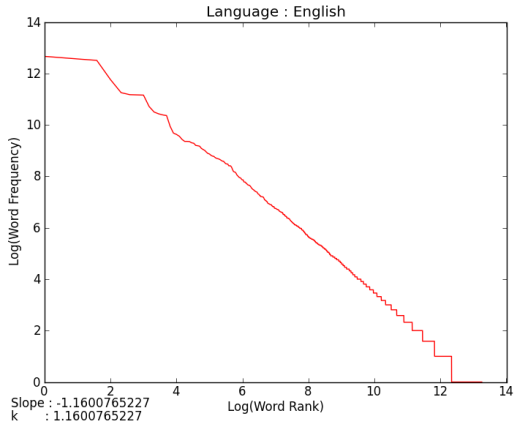
Now the fat tail is clearly visible after beginning with rank(w) = 300.

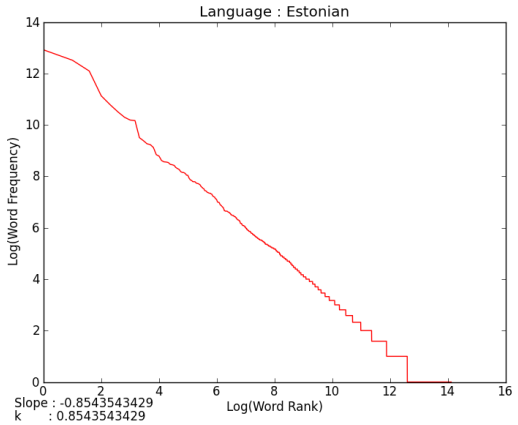
- Let's also do a log-log plot where the x-axis has log(rank(w)) and y-axis has log(freq(w)).

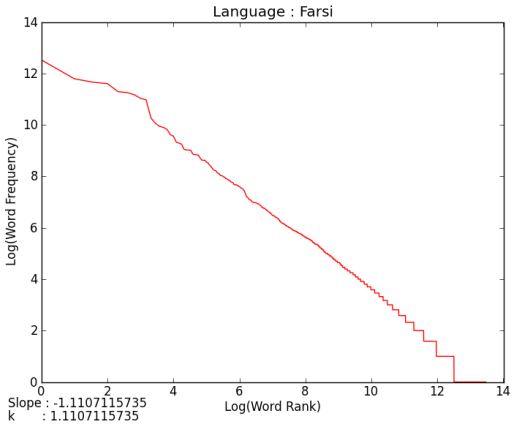
Fig 2.3 Log-Log Plots based on word frequency to illustrate the linear property of zipfian distributions/power law

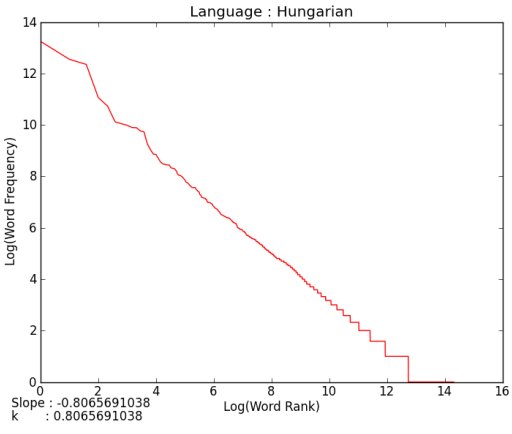


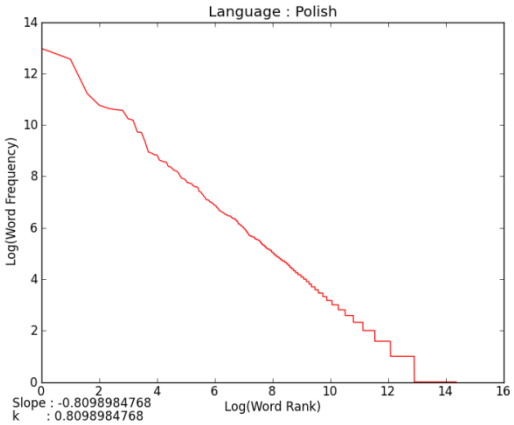


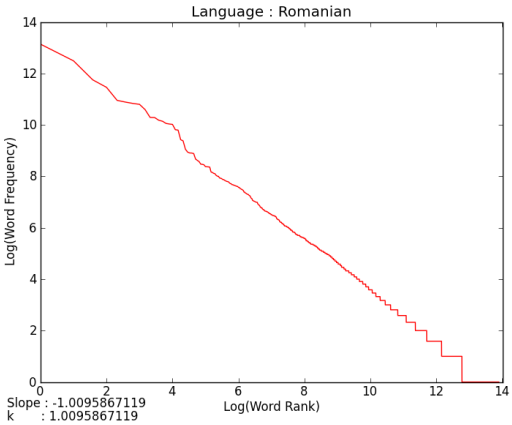


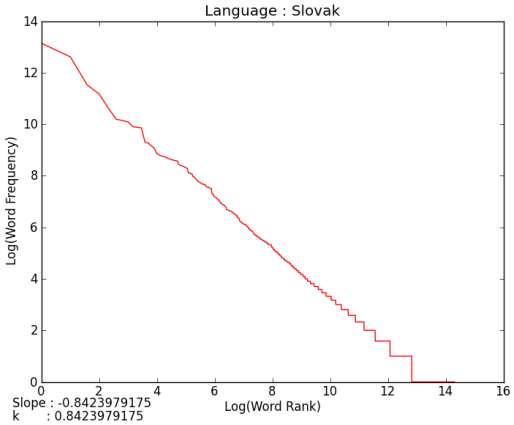


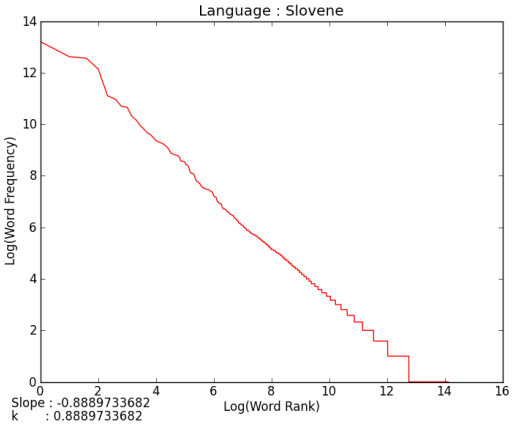

Language : Farsi
Log(Word Frequency)
Log(Word Rank)
Slope : -1.1107115735
k : 1.1107115735

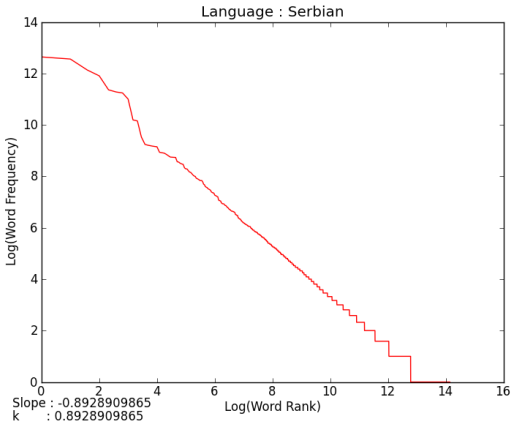

Language : Hungarian
Log(Word Frequency)
Log(Word Rank)
Slope : -0.8065691038
k : 0.8065691038


Language : Polish
Log(Word Frequency)
Log(Word Rank)
Slope : -0.8098984768
k : 0.8098984768


Language : Romanian
Log(Word Frequency)
Log(Word Rank)
Slope : -1.0095867119
k : 1.0095867119


Language : Slovak
Log(Word Frequency)
Log(Word Rank)
Slope : -0.8423979175
k : 0.8423979175

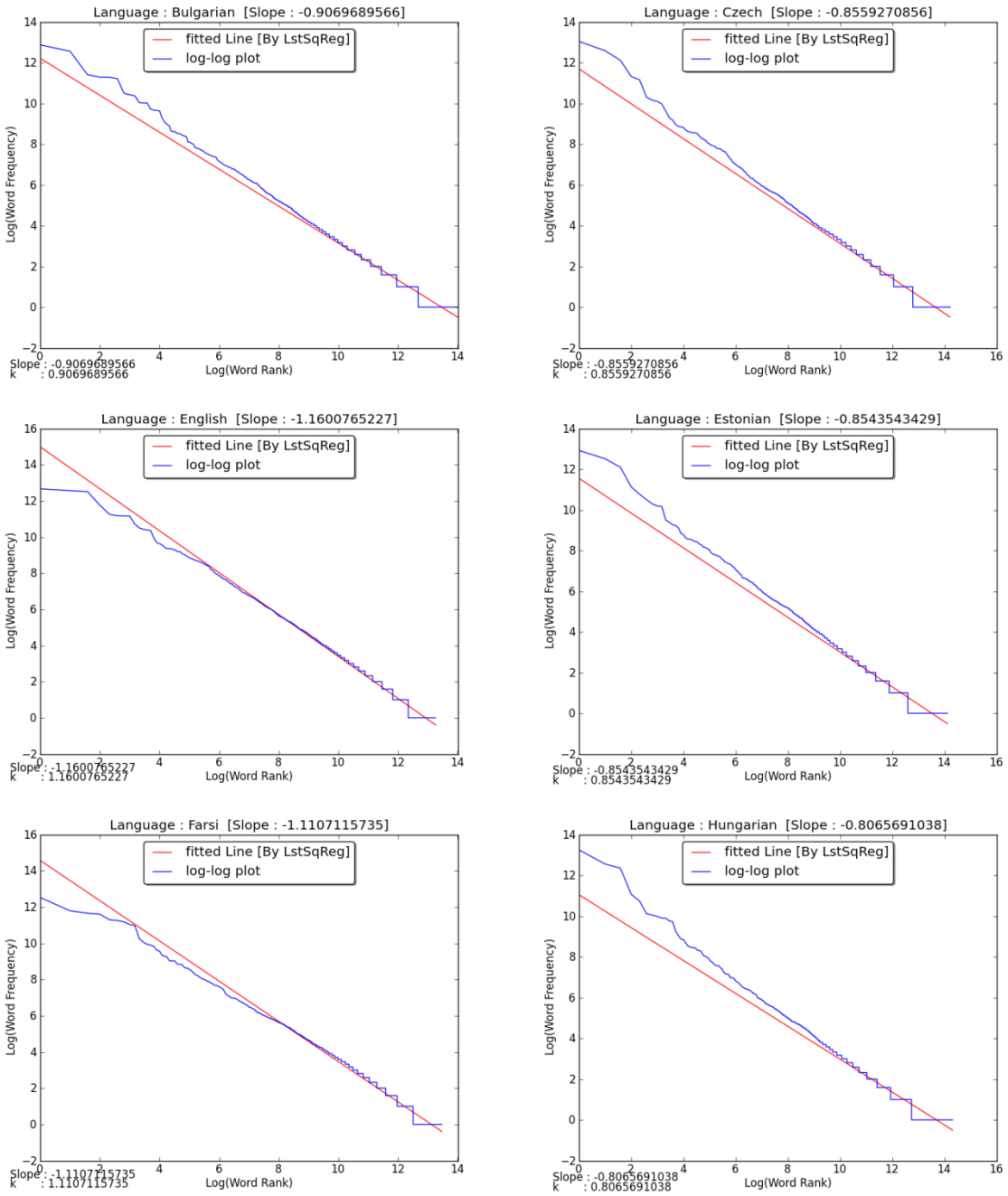

Language : Slovene
Log(Word Frequency)
Log(Word Rank)
Slope : -0.8889733682
k : 0.8889733682

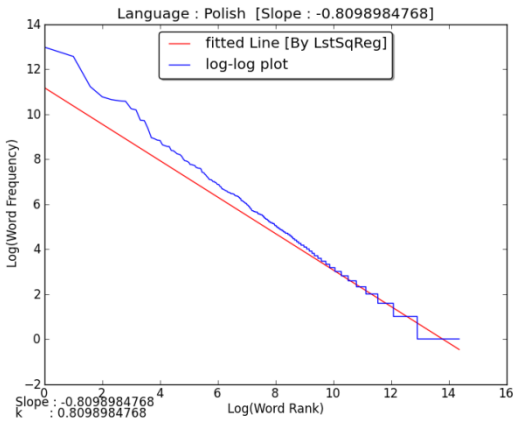

Language : Serbian
Log(Word Frequency)
Log(Word Rank)
Slope : -0.8928909865
k : 0.8928909865

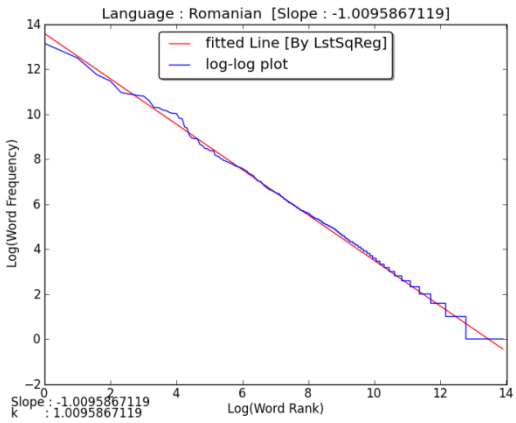
The log-log plots are almost linear which indicates that they obey power law. Also the slope is calculated for the best fitted line for each of the log-log plots. The negative of the slope gives the value of k in the zipf equation.

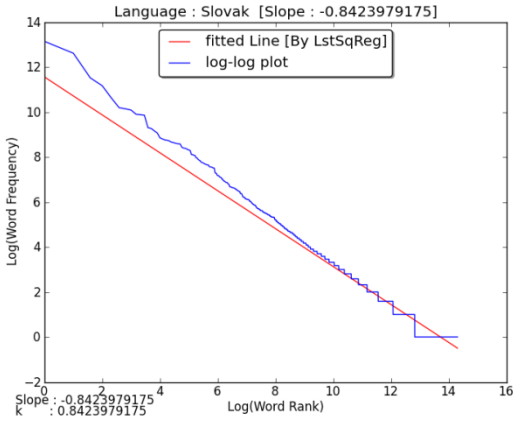
3. Answer to Question 3
    - On running the least squares linear regression on each language's log-log plot, the following plots are obtained:

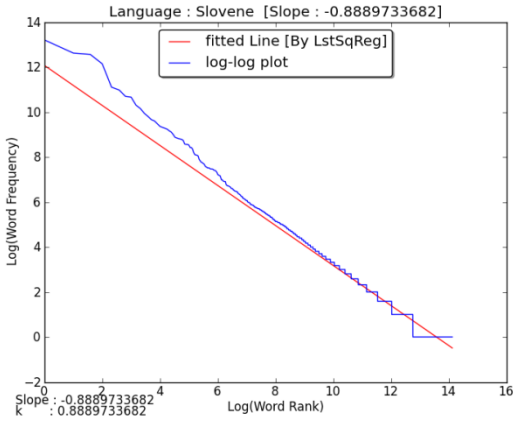
Fig 3.1 Log-Log Plots based on word frequency with most fitting line based on Least Squares Linear Regression

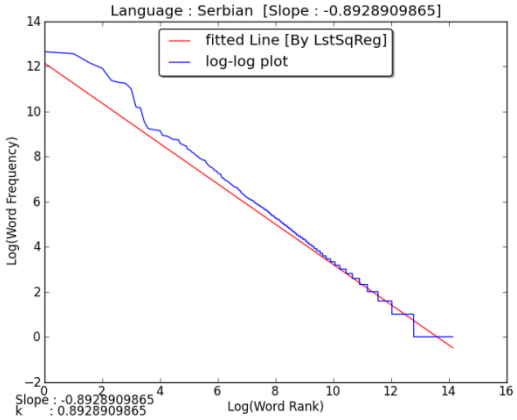












The negative of the slopes which indicates the **k** value in the zipf equation is found out in each case as listed in the figures.

- The mean and standard deviation of the slopes and k values are as below:

Attribute	**Slope**	**K (Negative of slope)**
Mean	-0.9216686406	0.9216686406
Standard deviation	0.1141354212	

- The values of **k** in all the cases are closer to **1** (mean : 0.9216686406) and they are almost similar across languages (standard deviation : 0.1141354212)

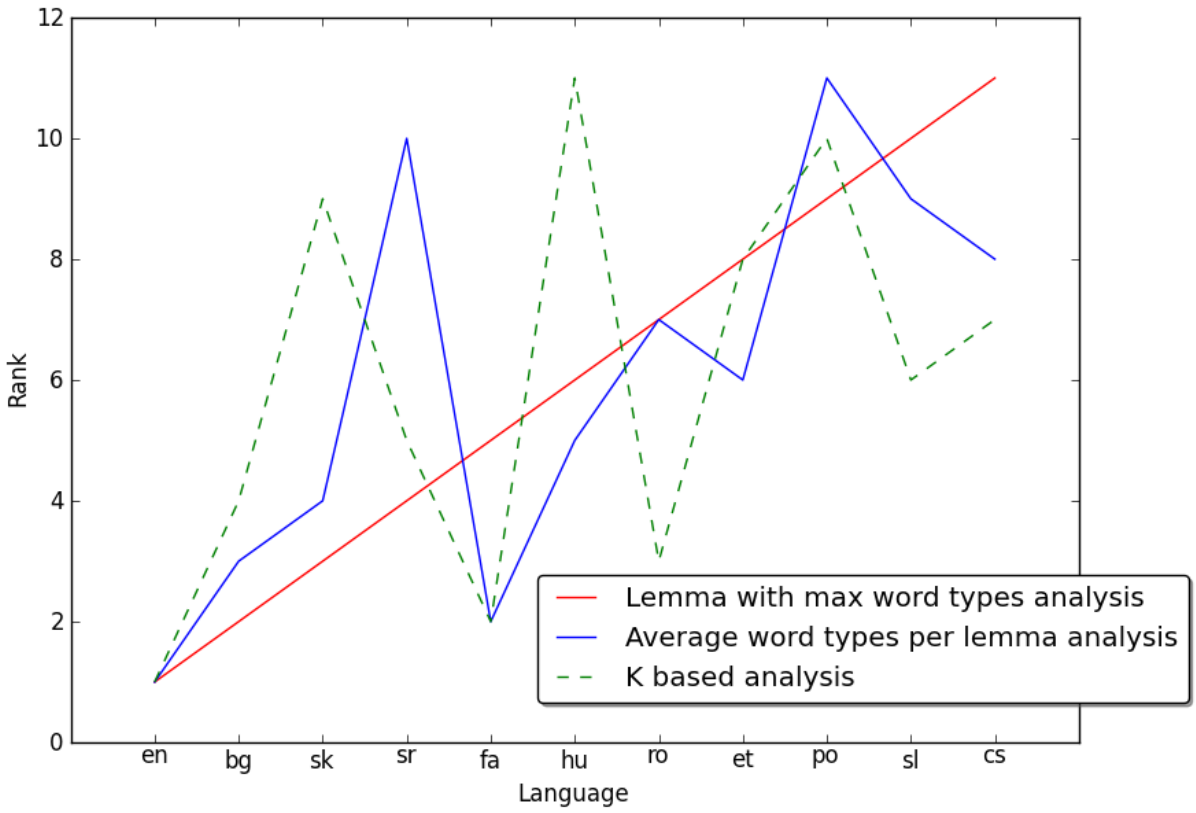
- Comparing the cross language differences in k with the cross language differences in complexity observed above,

The various k values for the languages are:

Language	**Language code**	**k**	**1-k**
English	en	1.1600765227	0.1600765227
Farsi	fa	1.1107115735	0.1107115735
Romanian	ro	1.0095867119	0.0095867119
Bulgarian	bg	0.9069689566	0.0930310434
Serbian	sr	0.8928909865	0.1071090135
Slovene	sl	0.8889733682	0.1110266318
Czech	cs	0.8559270856	0.1440729144
Estonian	et	0.8543543429	0.1456456571
Slovak	sk	0.8423979175	0.1576020825
Polish	po	0.8098984768	0.1901015232
Hungarian	hu	0.8065691038	0.1934308962

Higher k value indicates that fewer words are more frequent and more words are less frequent than in other languages with lesser k value. This gives an idea on the morphological complexity existing in these languages. So, comparing this with previous analysis on morphological complexity,

Fig 3.2 Comparison of Language's morphological complexity based on Least Squares Linear Regression using word frequency with previous methods



- Now lets use lemma frequency to estimate k,

Now the mean and Standard deviation are as follows:

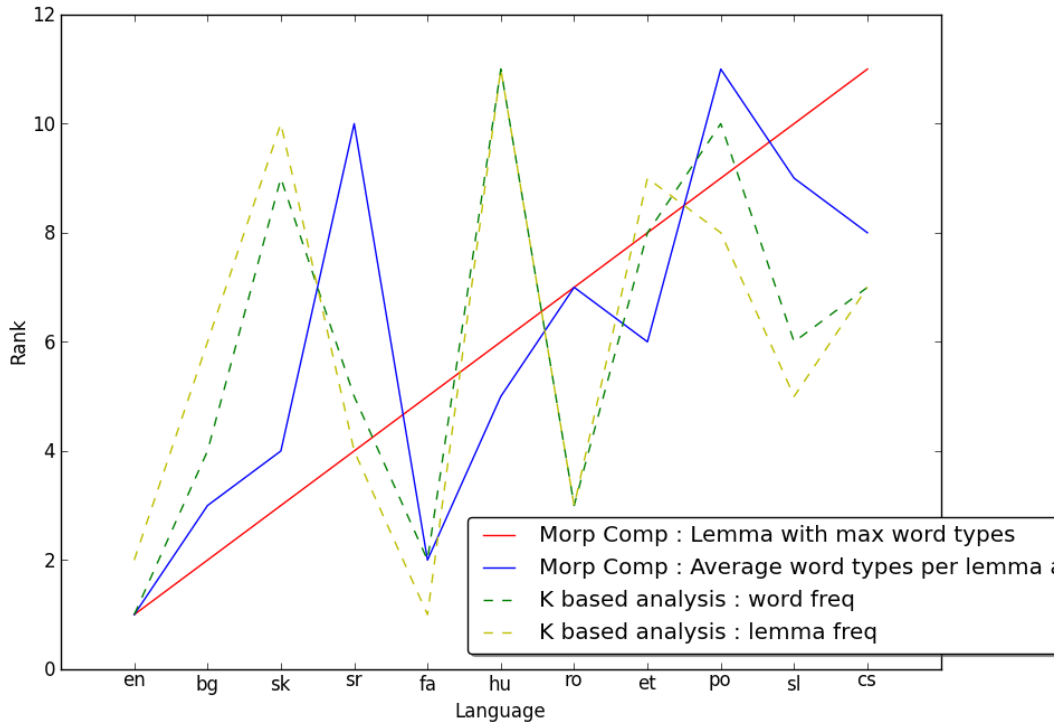
Attribute	**Slope**	**K (Negative of slope)**
Mean	-1.1726987696	1.1726987696
Standard deviation	0.0685249043	

The various k values for the language are:

Language	**Language code**	**k**	**1-k**
Farsi	fa	1.3021333775	0.3021333775
English	en	1.2665200119	0.2665200119
Romanian	ro	1.2594472162	0.2594472162
Serbian	sr	1.1775284227	0.1775284227
Slovene	sl	1.1679606026	0.1679606026
Bulgarian	bg	1.1599982432	0.1599982432
Czech	cs	1.1248045419	0.1248045419
Polish	po	1.1227551645	0.1227551645
Estonian	et	1.1134450001	0.1134450001
Slovak	sk	1.1130917186	0.1130917186
Hungarian	hu	1.0920021660	0.0920021660

The new k values are also closer to 1 and they appear to have more similar frequency distribution. Similar comparison with morphological complexity yields,

Fig 3.3 Comparison of Language's morphological complexity based on Least Squares Linear Regression using lemma frequency with previous methods



Code

- **LanguageAnalyzer.py**

  This is the main program. It has the LanguageMeta and LanguageAnalyzer classes implemented. The LanguageMeta is used to store all relevant analysed meta for each language and LanguageAnalyzer provides with functionalities to draw plots and analyse data stored in LanguageMeta.

- **OrwellDataParser.py**

  This is used by the LanguageAnalyzer to parse the input data files and to get specific information from the input files abstracting the method by which it is obtained.

- **loggerUtils.py**

  This is a logging utility used to capture warnings and special cases which is used for debugging.

Plots

- **orwell-<language-code>.txt.out**
  It has the list of all lemmas sorted based on the count of word types associated with it.

- **Plots_Zipfian_Analysis_rank_1.pdf**
  It consists of plots for all languages with word rank at x-axis and word frequency at y-axis starting with rank = 1.

- **Plots_Zipfian_Analysis_rank_300.pdf**
  It consists of plots for all languages with word rank at x-axis and word frequency at y-axis starting with rank = 300.

- **Log_Plots_Zipfian_Analysis_rank_1.pdf**
  It consists of plots for all languages with log(word rank) at x-axis and log(word frequency) at y-axis starting with rank = 1.

- **Least_Square_Regression_Word.pdf**
  It consists of least square regression analysis plots with log(word rank) at x-axis and log(word frequency) at y-axis with most fitting line along with slope and k value calculations.

- **Least_Square_Regression_Lemma.pdf**
  It consists of least square regression analysis plots with log(lemma rank) at x-axis and log(lemma frequency) at y-axis with most fitting line along with slope and k value calculations.