Natural Language Processing
Report

P.L.S.Sreedhar
201501029

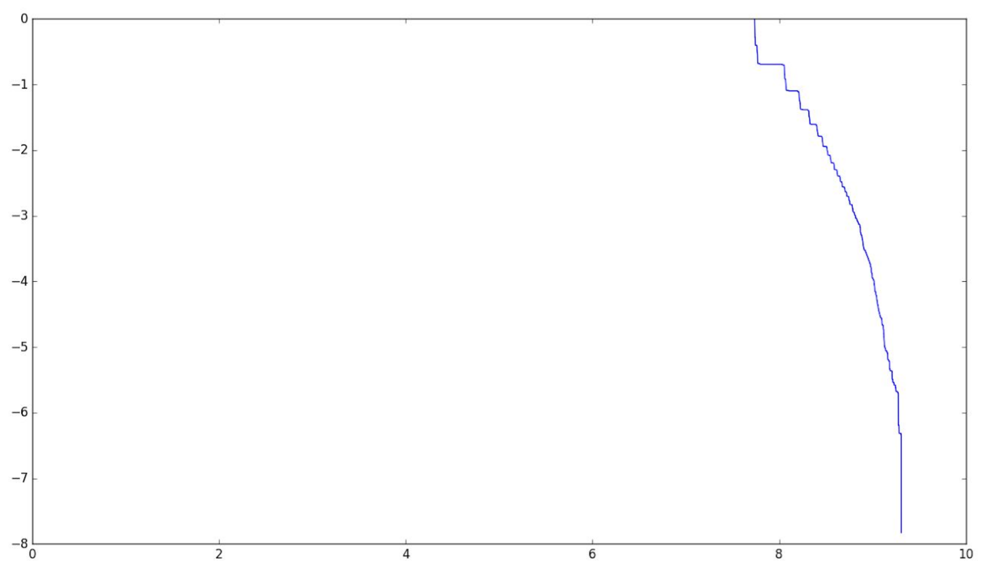
1. *Implement unigram, bigram and trigram language models.*
Implementation of the language models has been done in the code,
Steps require:

```
corpus=tokenise(file)
unigrams,unigrams_prob=get_unigrams(corpus)
bigrams,bigrams_prob = get_bigrams(corpus,unigrams)
trigrams,trigrams_prob = get_trigrams(corpus,bigrams)
```

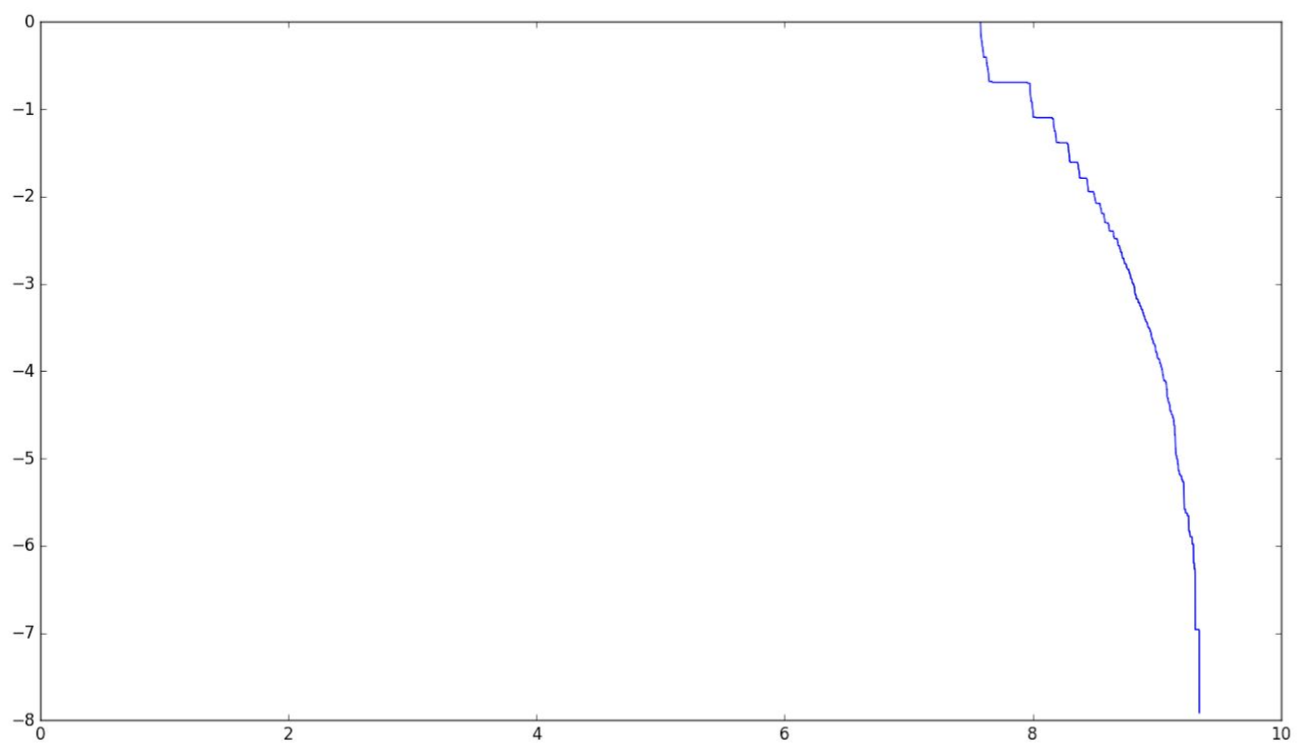
2.Plot log-log curve and zipf curve for the above:
Using

```
plot(sort_dict(unigrams_prob))
plot_log_log1(sort_dict(unigrams_prob))
```

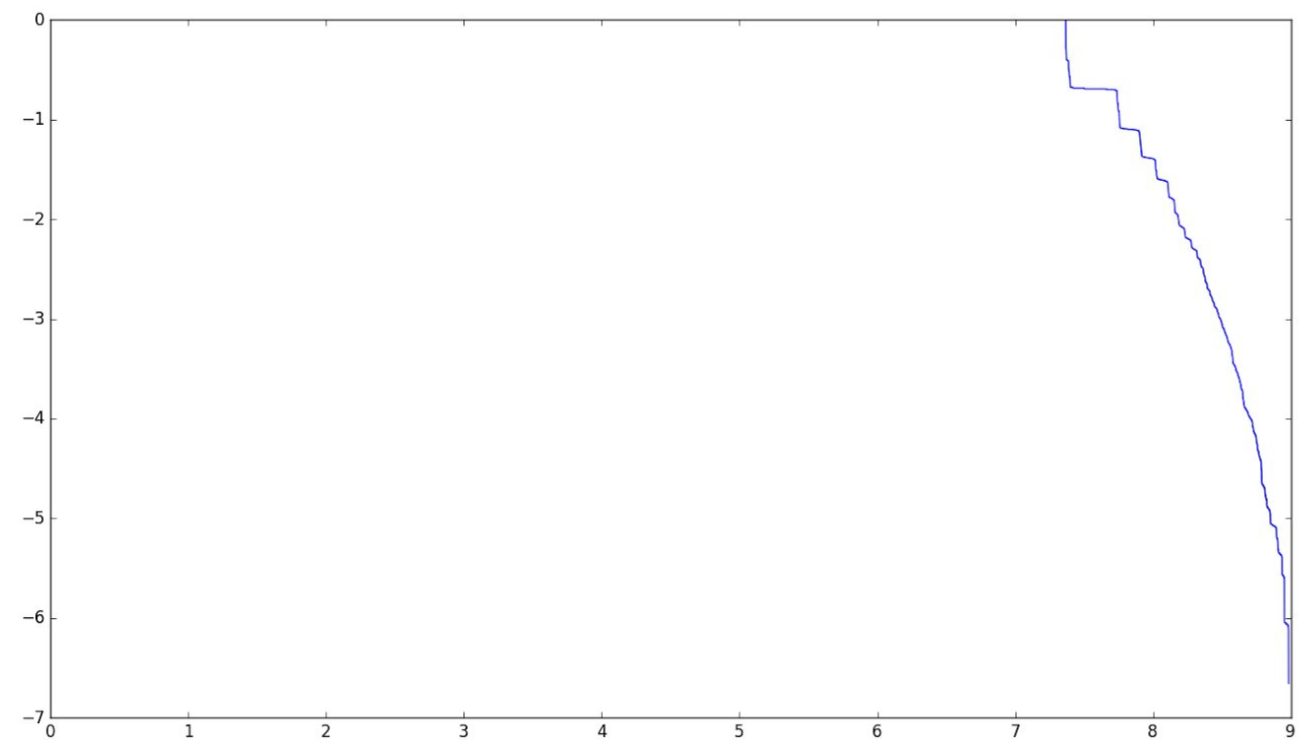
Different plots we got:
normal_bigram_log_anime



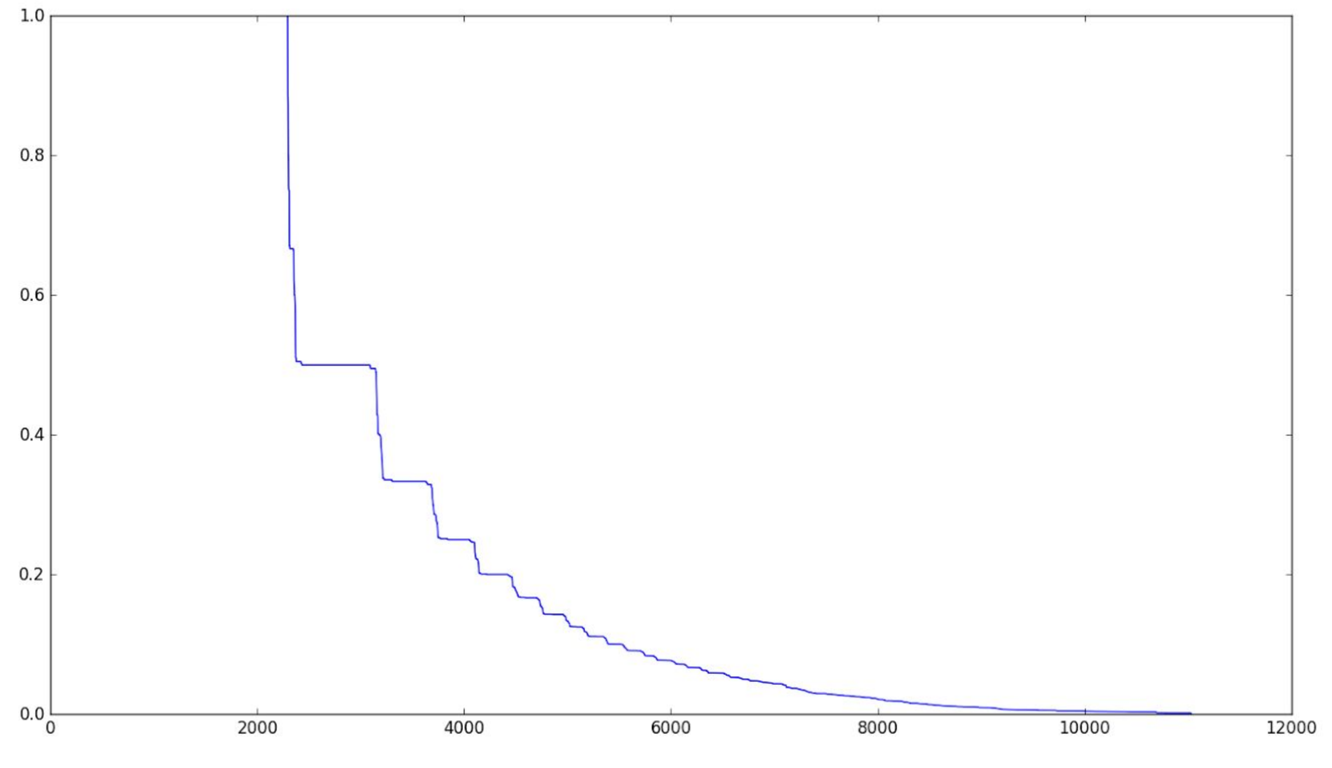
normal_bigram_log_movies



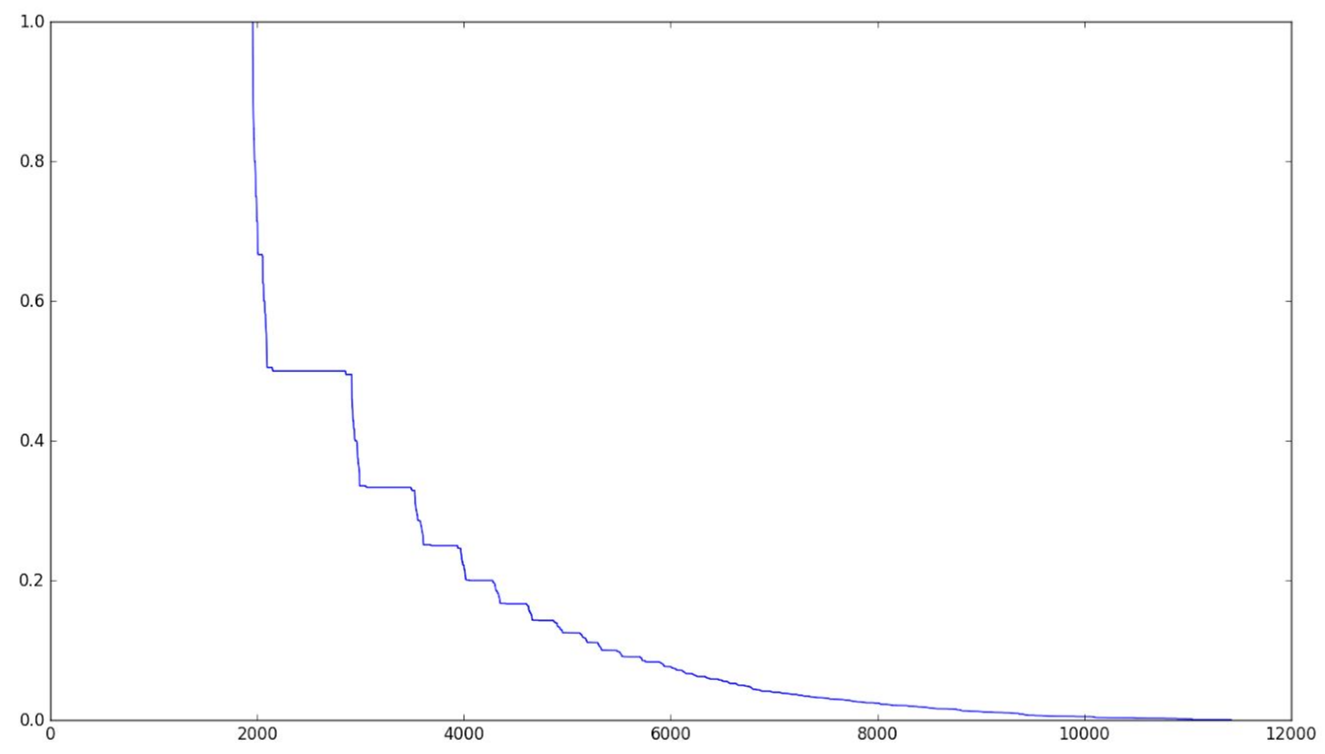
normal_bigram_log_news



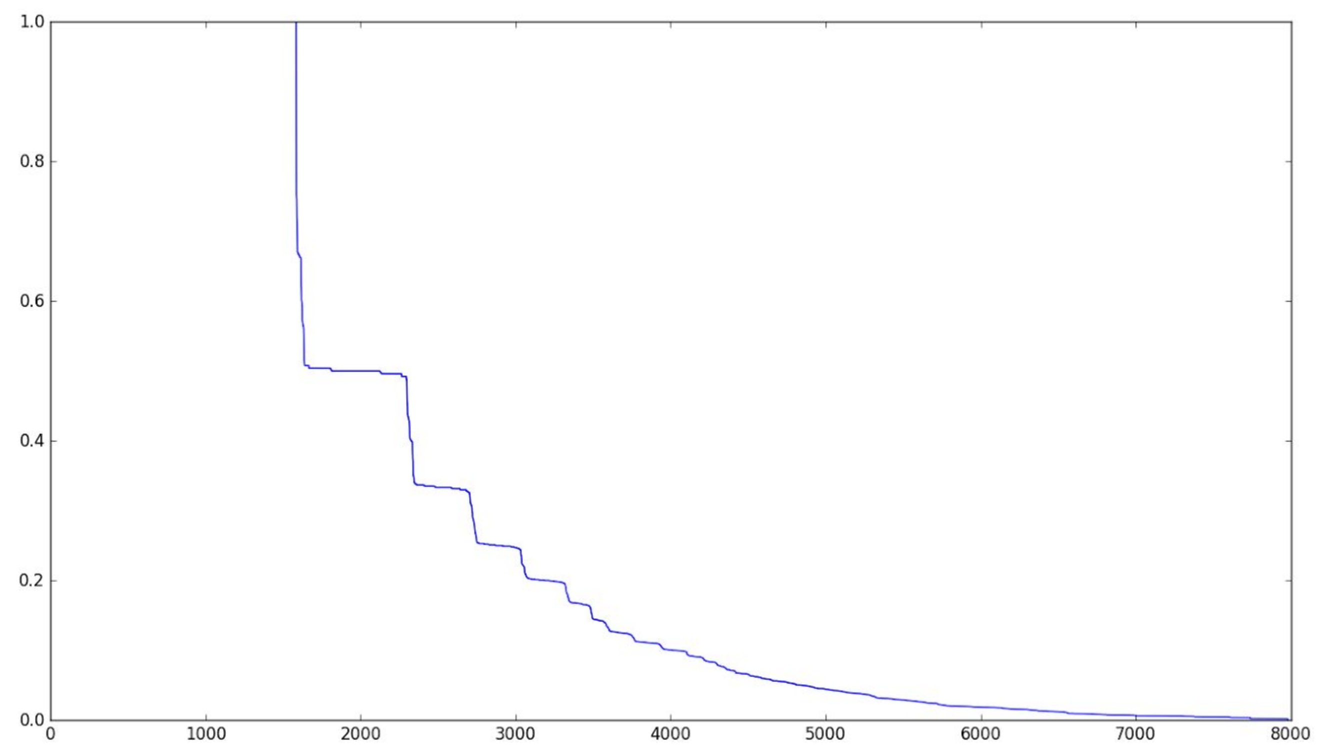
normal_bigram_zipf_anime



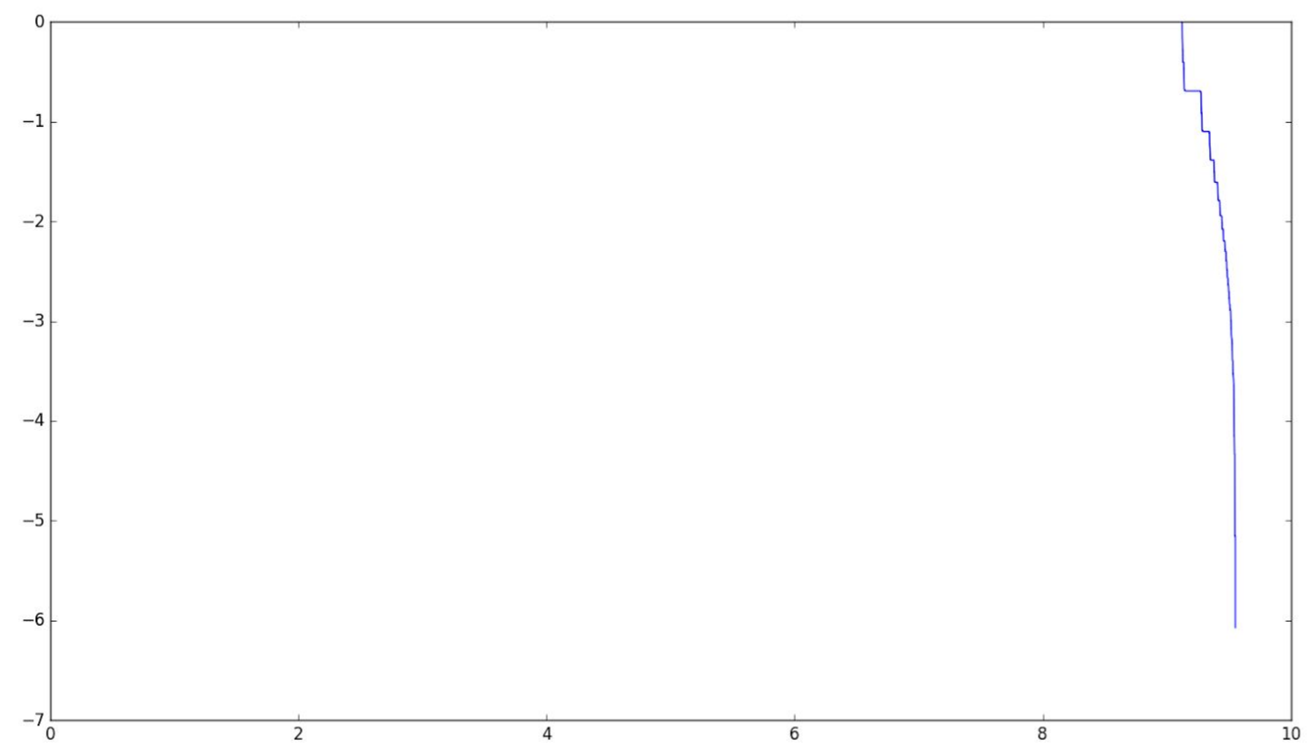
normal_bigram_zipf_movies



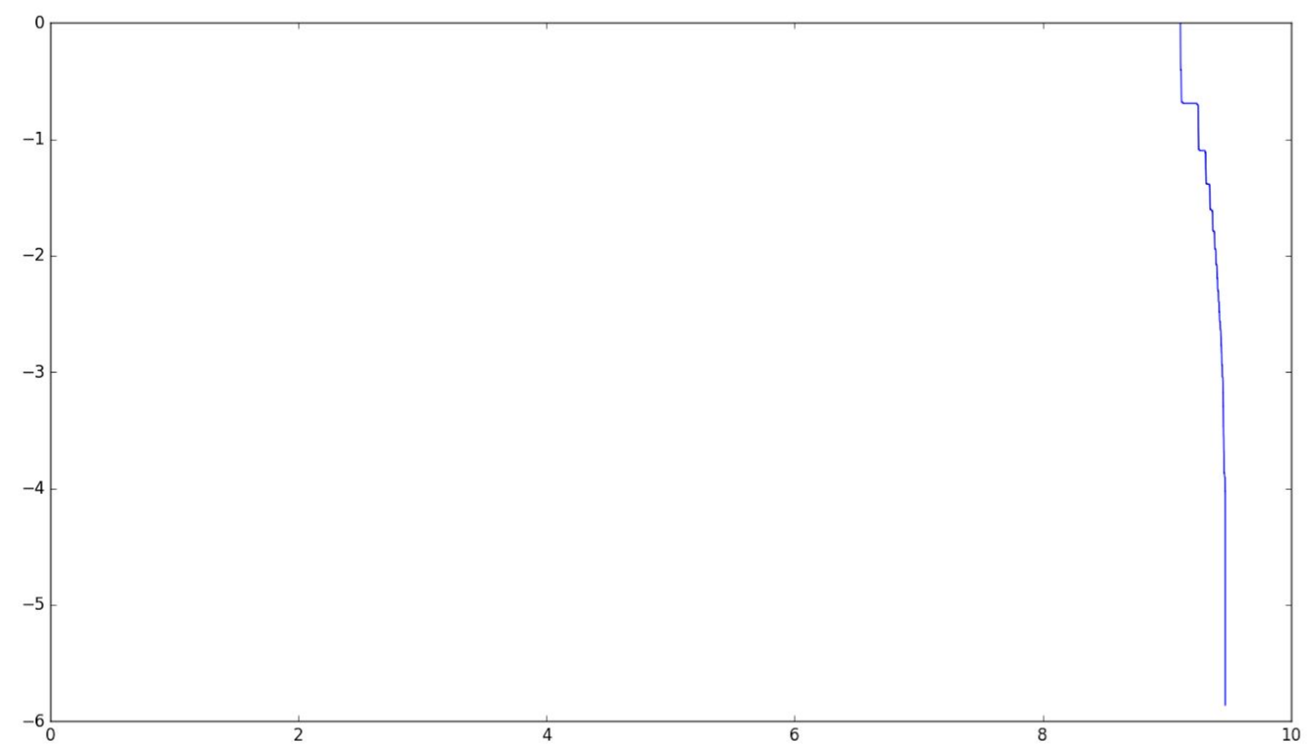
normal_bigram_zipf_news



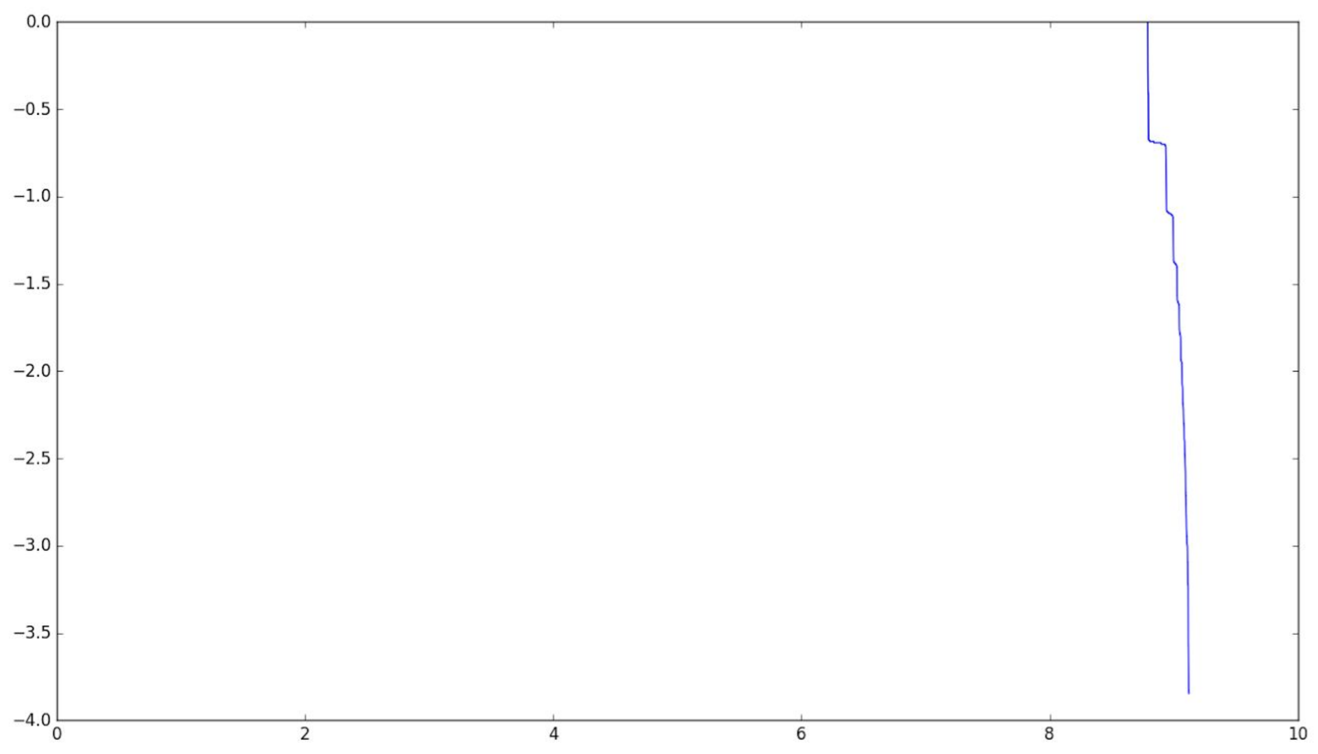
normal_trigram_log_movies



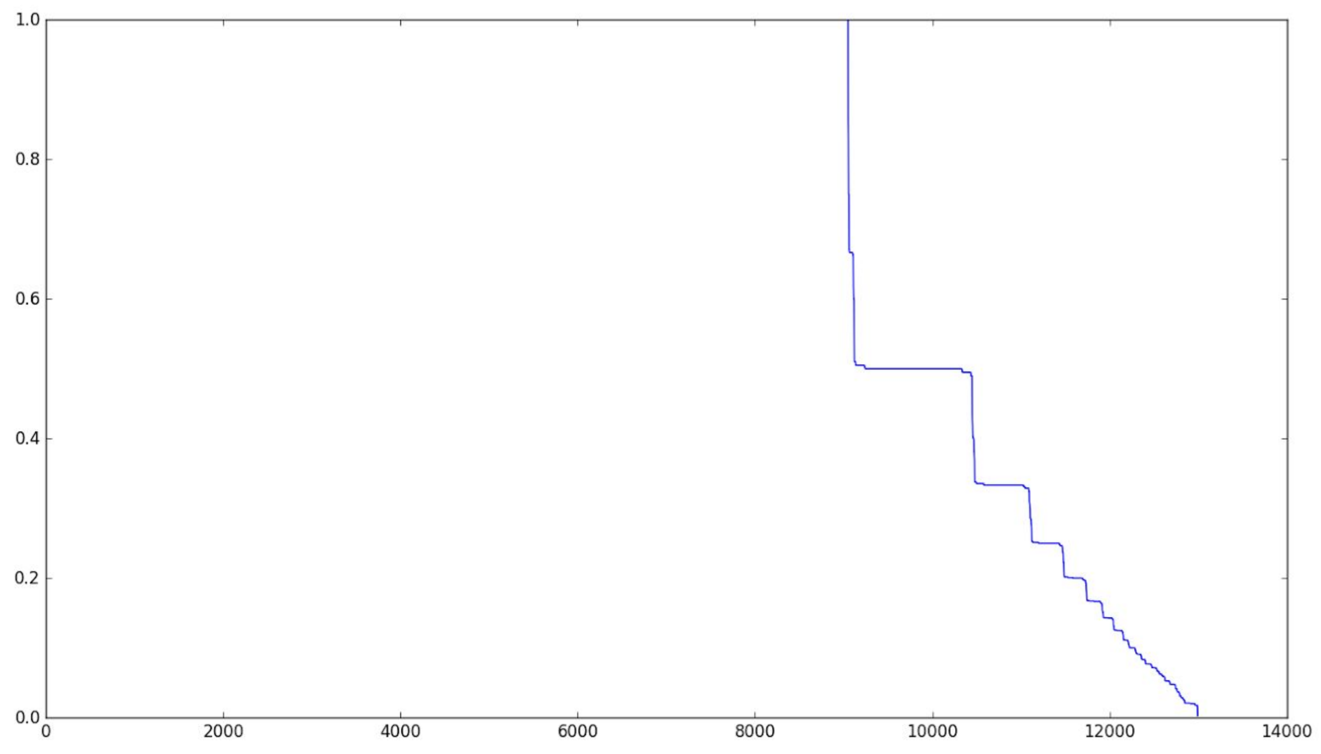
normal_trigrams_log_anime



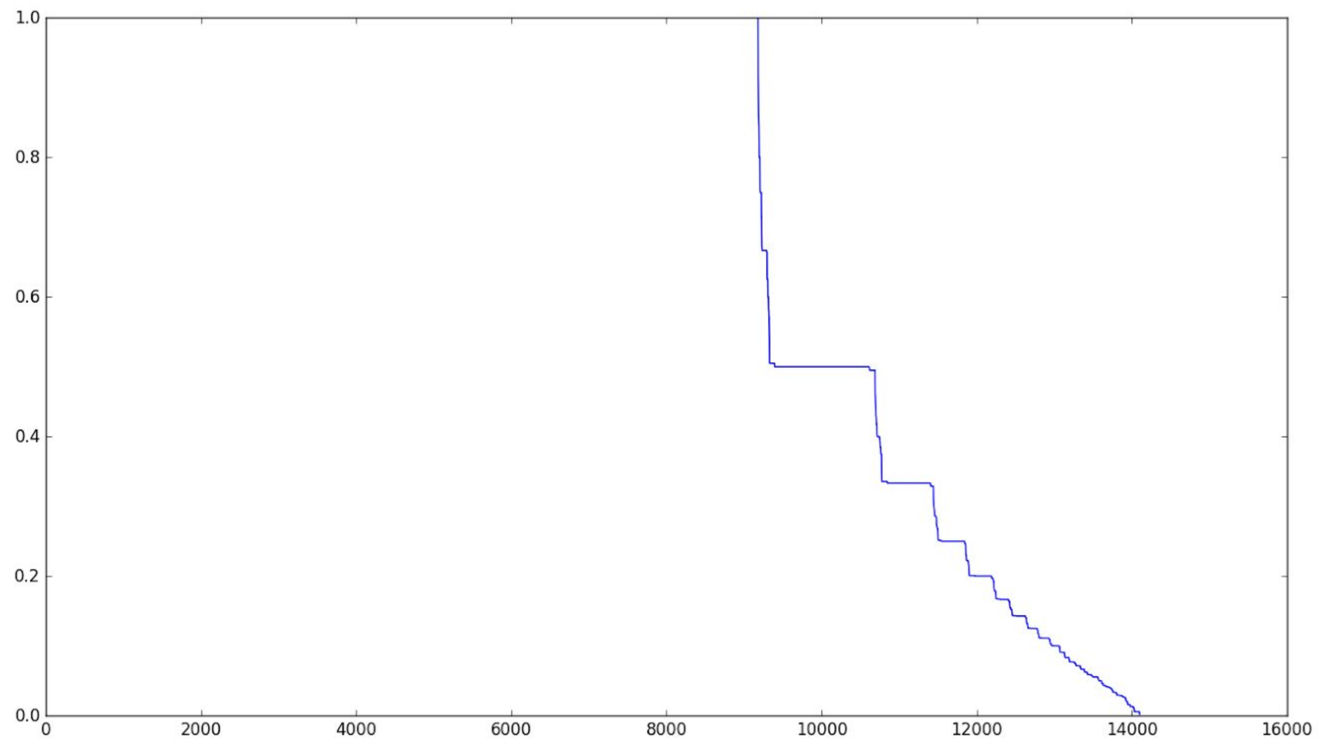
normal_trigrams_log_news



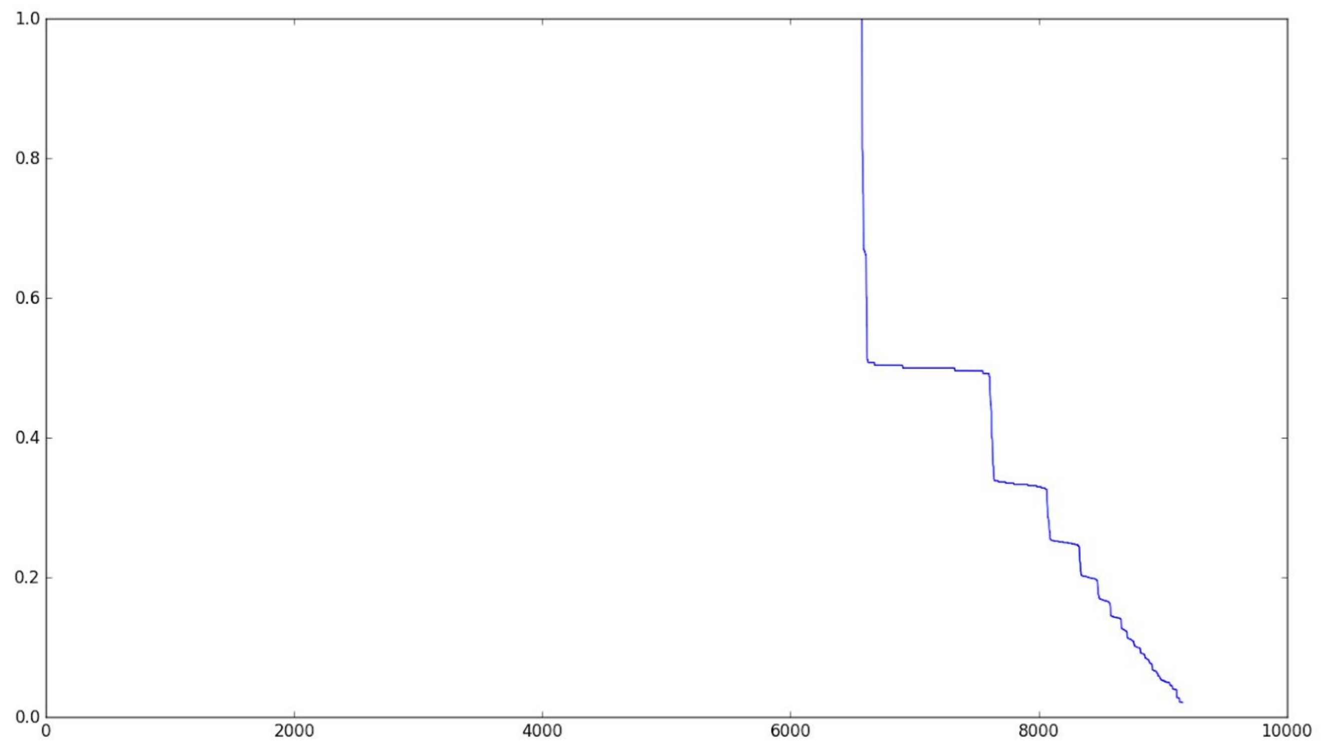
normal_trigrams_zipf_anime



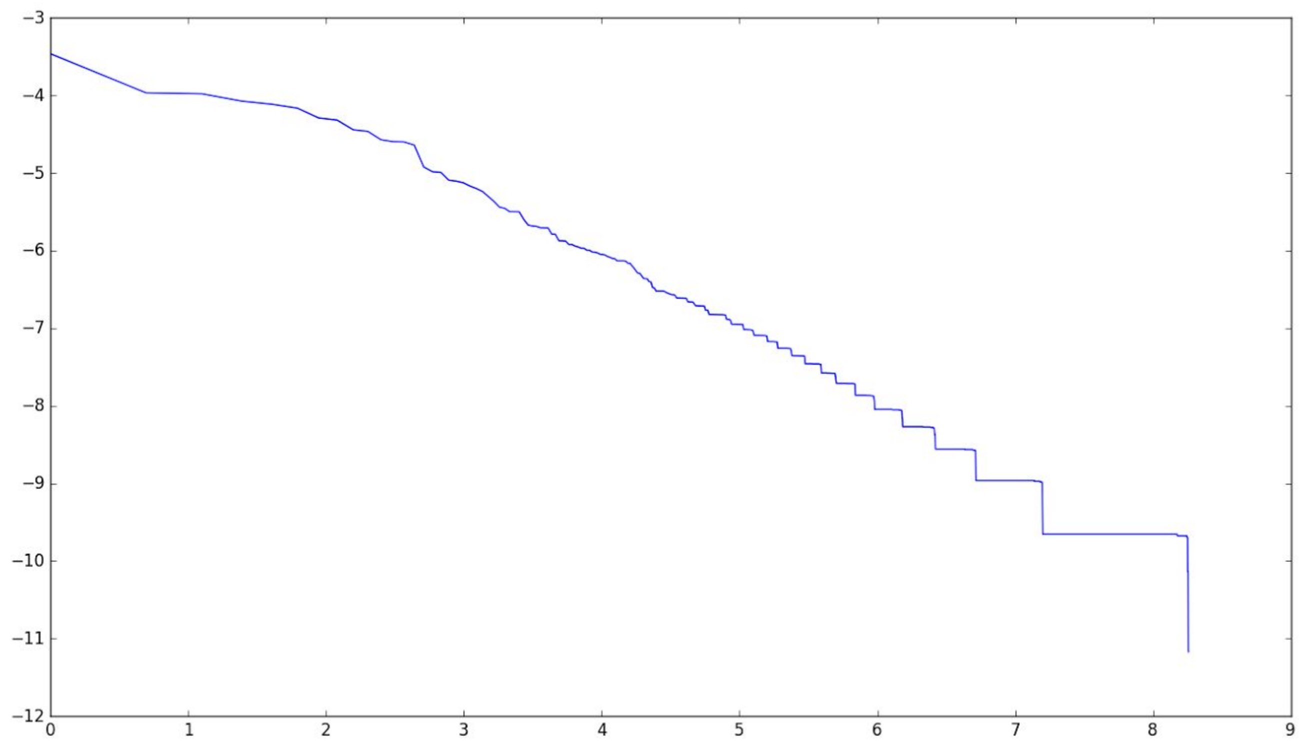
normal_trigrams_zipf_movies



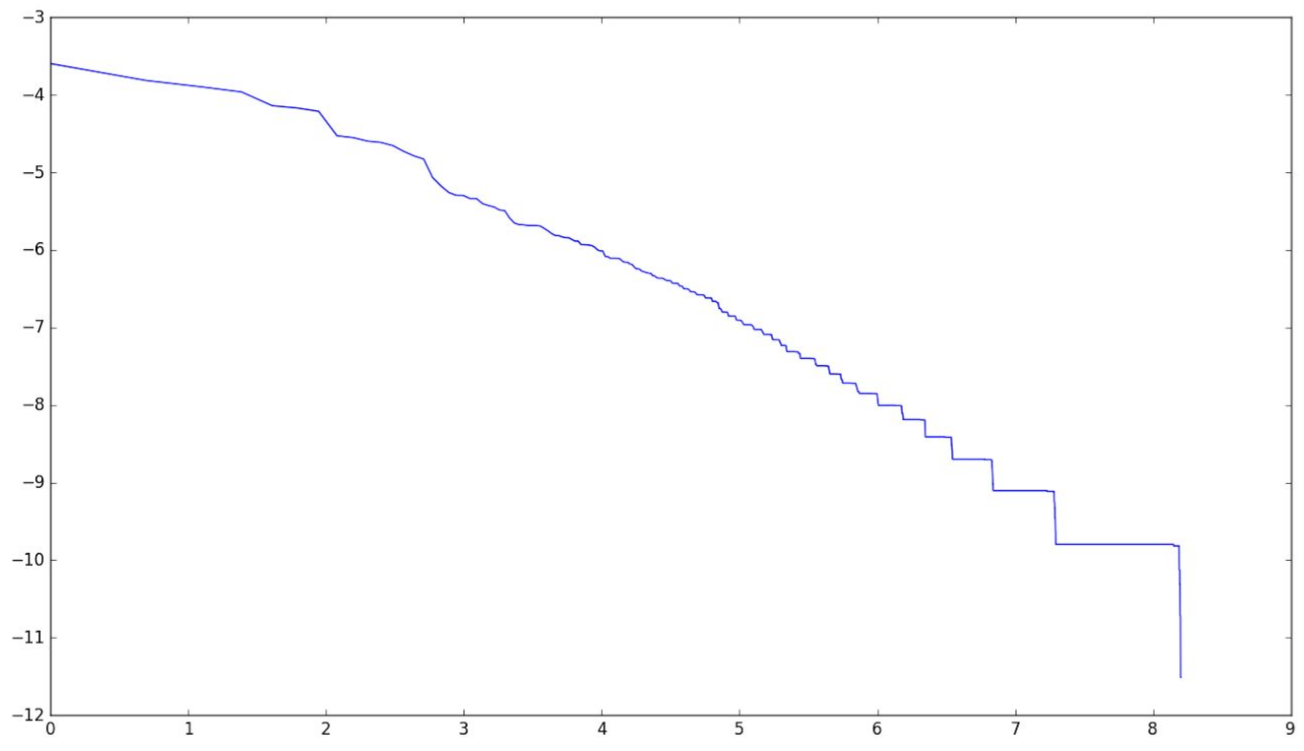
normal_trigrams_zipf_news



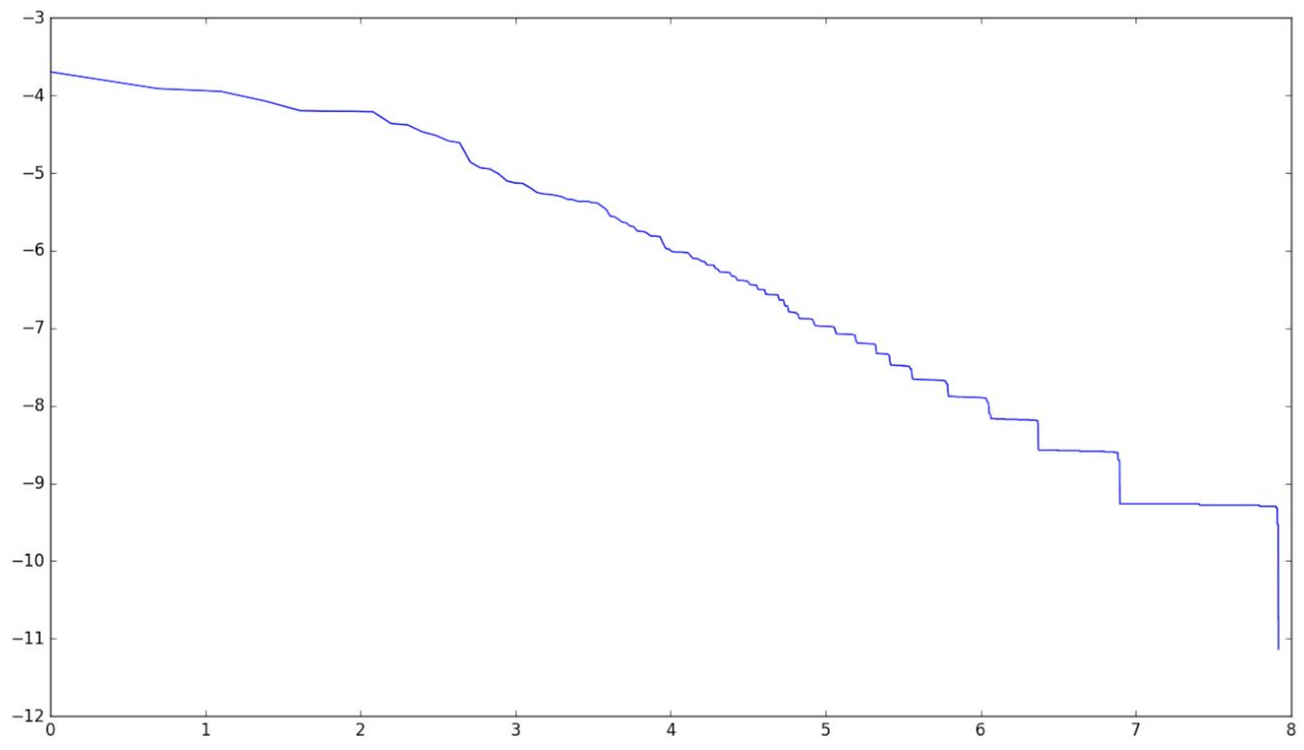
normal_unigram_log_anime



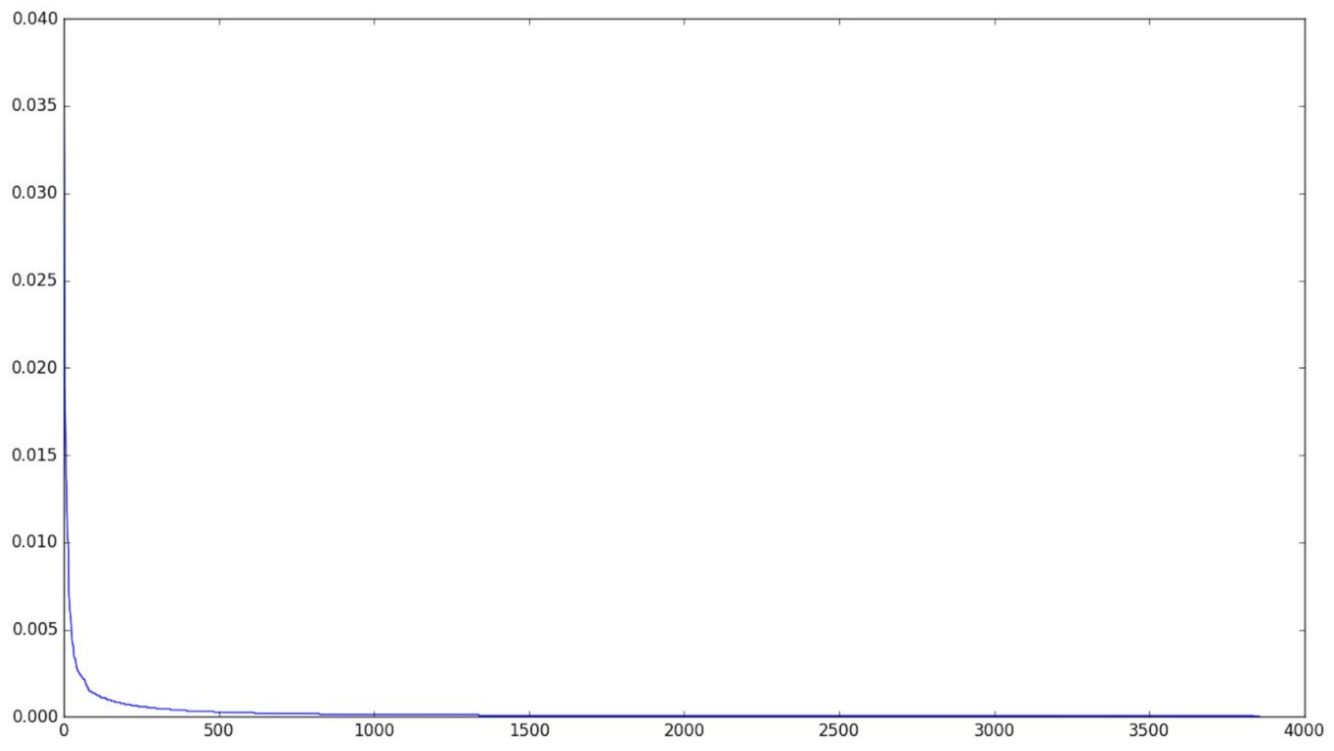
normal_unigram_log_movies



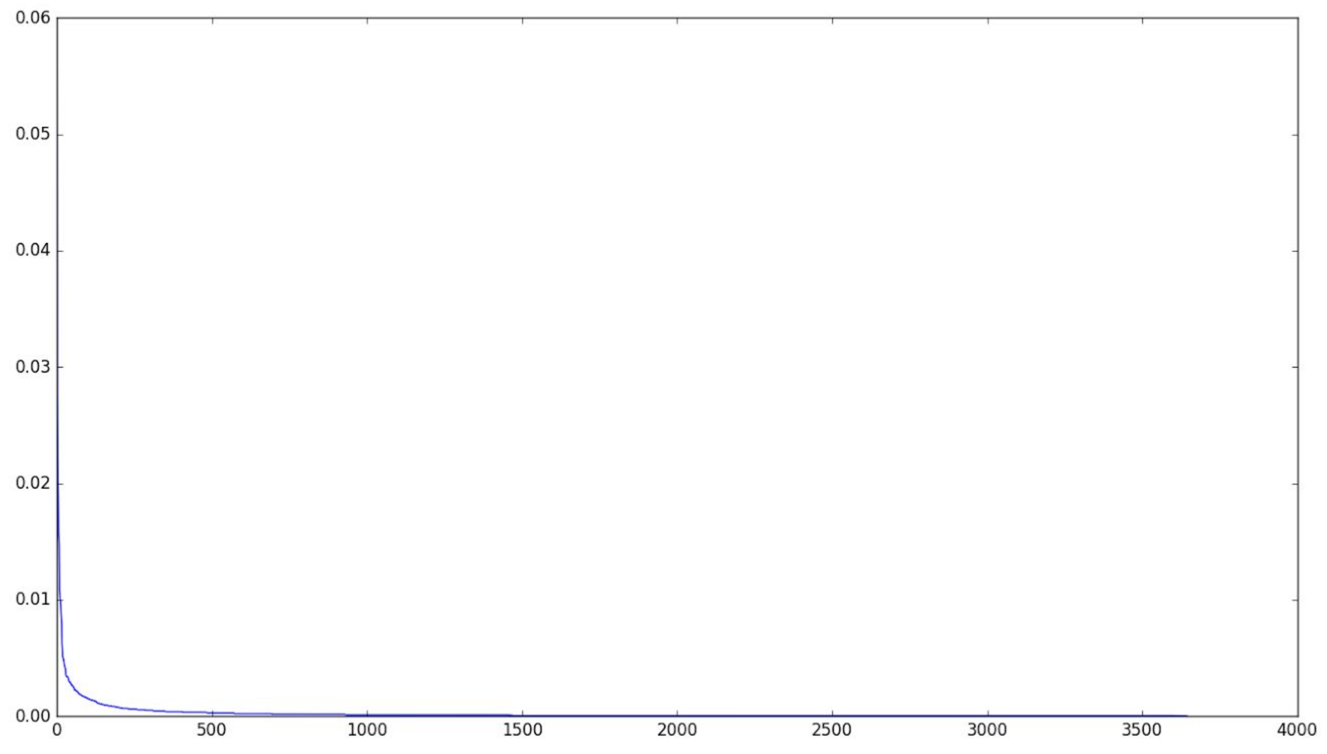
normal_unigram_log_news



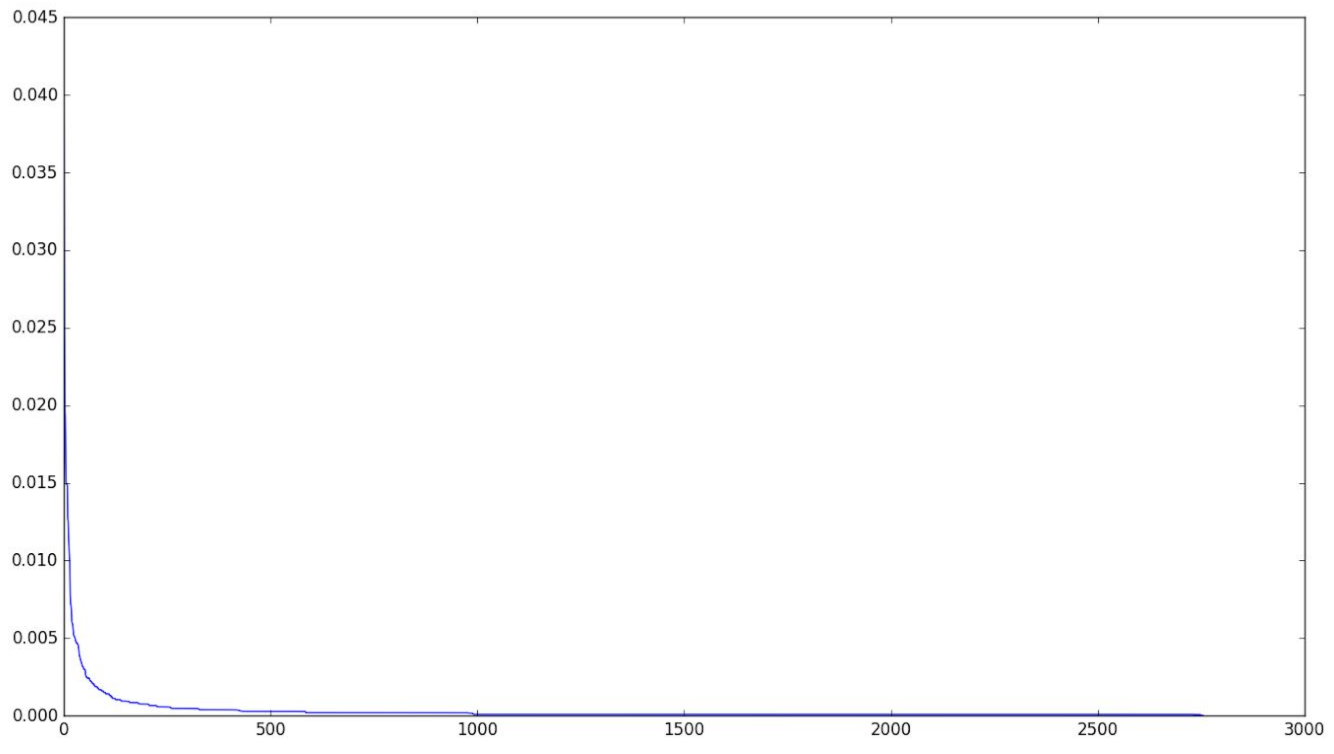
normal_unigram_zipf_anime



normal_unigram_zipf_movies

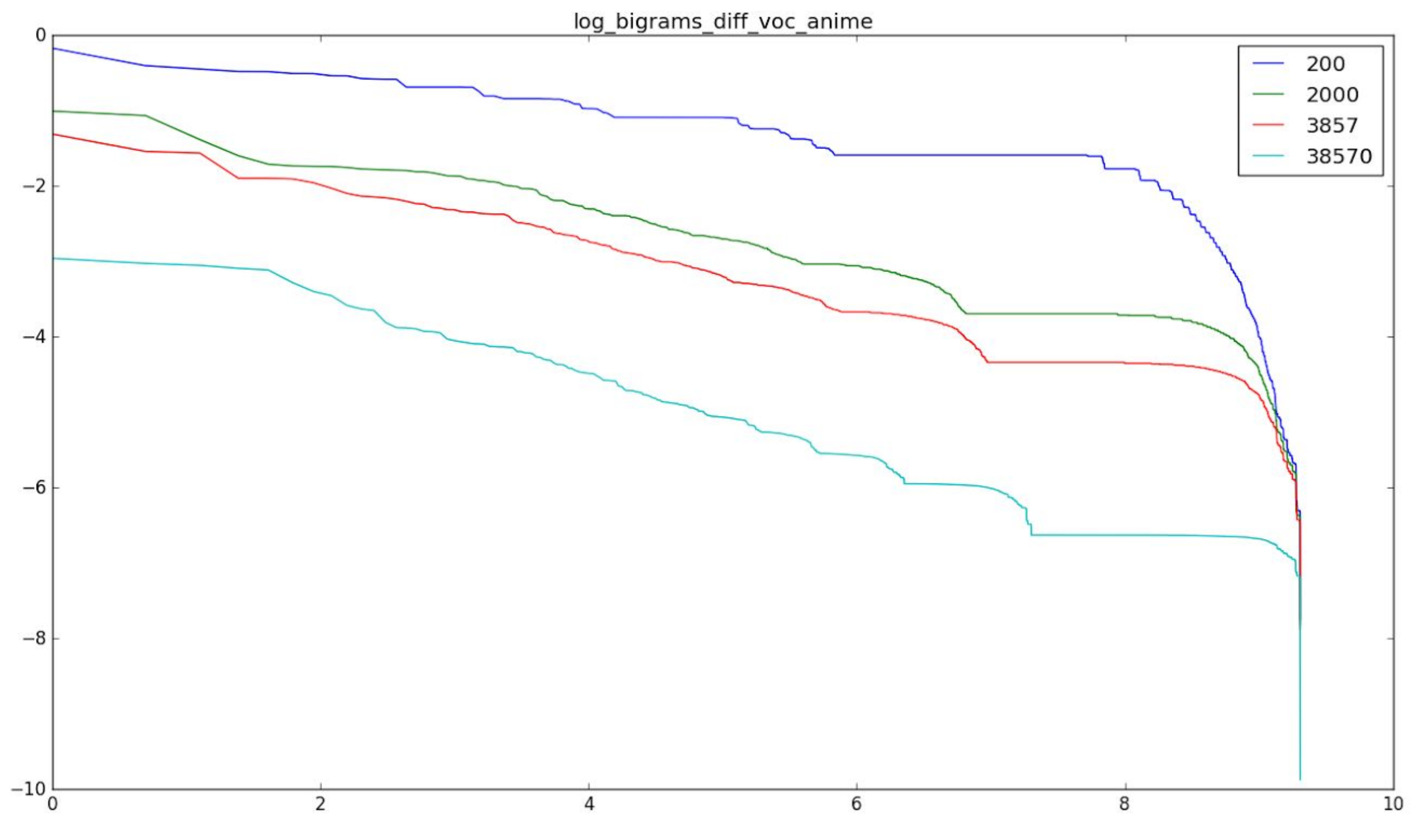


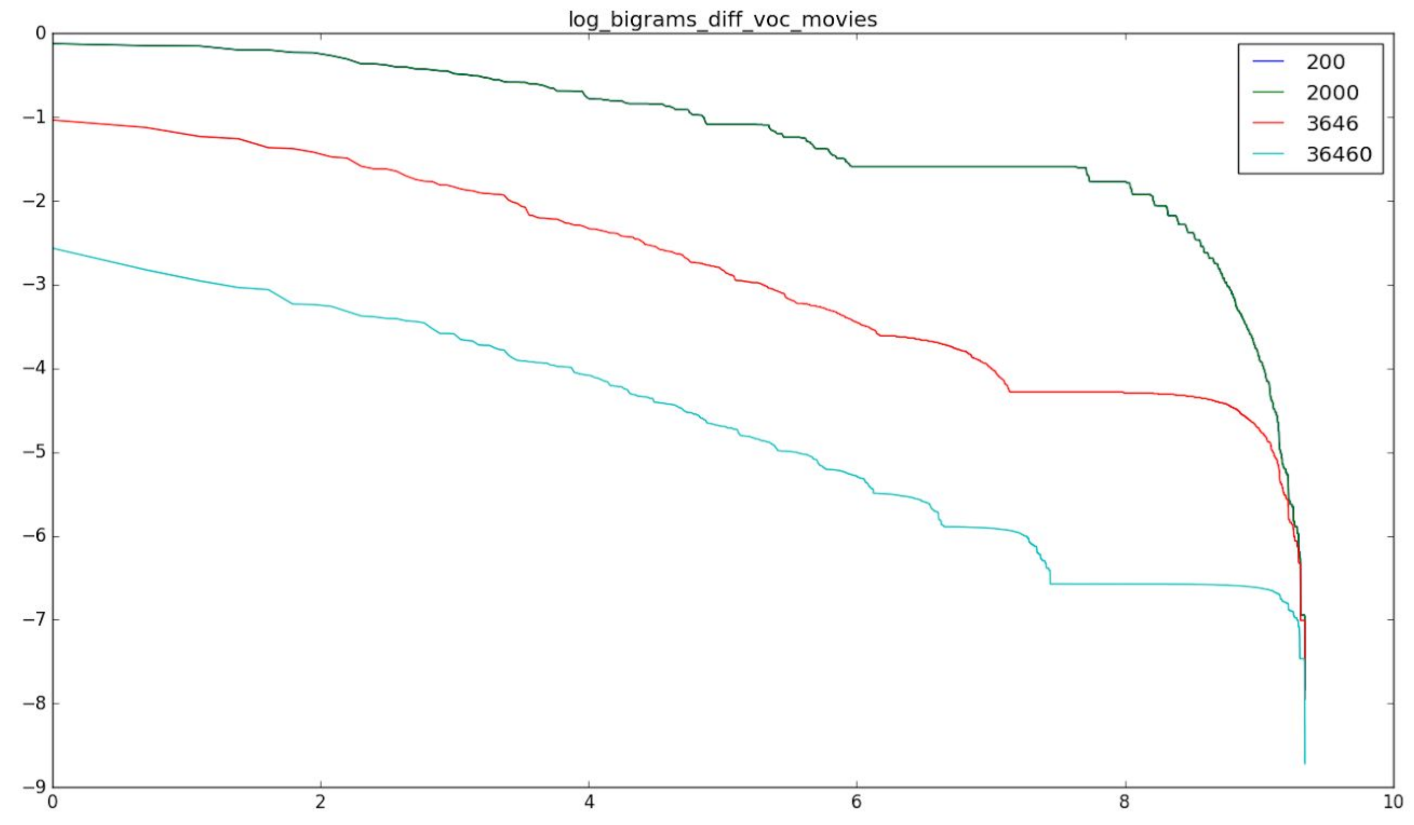
normal_unigram_zipf_news

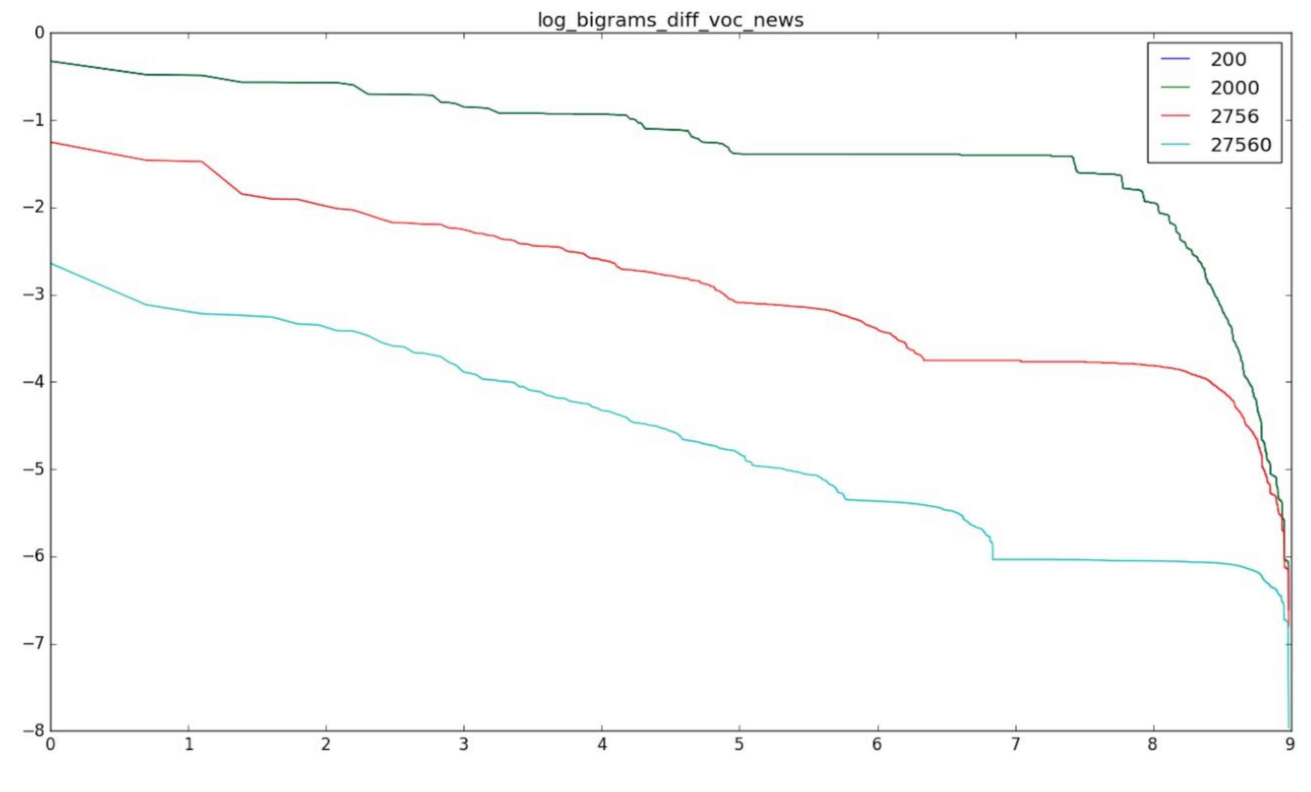


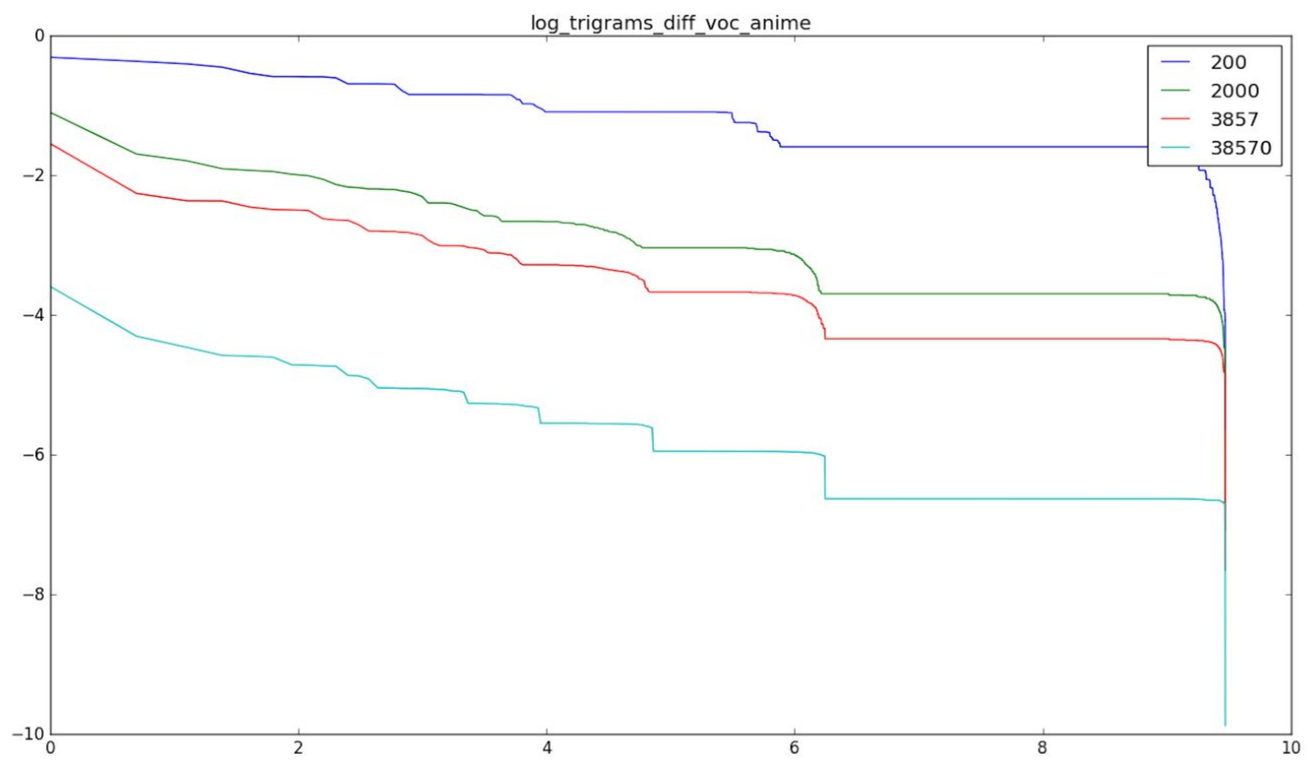
3.*Implement laplace smoothing. Compare the effect of smoothing on different values for V (200, 2000, current size of vocabulary, 10*size of vocabulary). Plot these to compare.*

```
laplace_unigrams_prob = get_laplace_unigrams(unigrams,200)
laplace_unigrams_prob2 = get_laplace_unigrams(unigrams,2000)
laplace_unigrams_prob3 = get_laplace_unigrams(unigrams,len(unigrams))

laplace_bigrams_prob1 = get_laplace_bigrams(unigrams,bigrams,200)
laplace_bigrams_prob2 = get_laplace_bigrams(unigrams,bigrams,2000)
laplace_bigrams_prob3 = get_laplace_bigrams(unigrams,bigrams,len(unigrams))
laplace_bigrams_prob4 = get_laplace_bigrams(unigrams,bigrams,10*len(unigrams))

laplace_trigrams_prob1 = get_laplace_trigrams(unigrams,bigrams,trigrams,200)
laplace_trigrams_prob2 = get_laplace_trigrams(unigrams,bigrams,trigrams,2000)
laplace_trigrams_prob3 = get_laplace_trigrams(unigrams,bigrams,trigrams,len(unigrams))
laplace_trigrams_prob4 = get_laplace_trigrams(unigrams,bigrams,trigrams,10*len(unigrams))
```

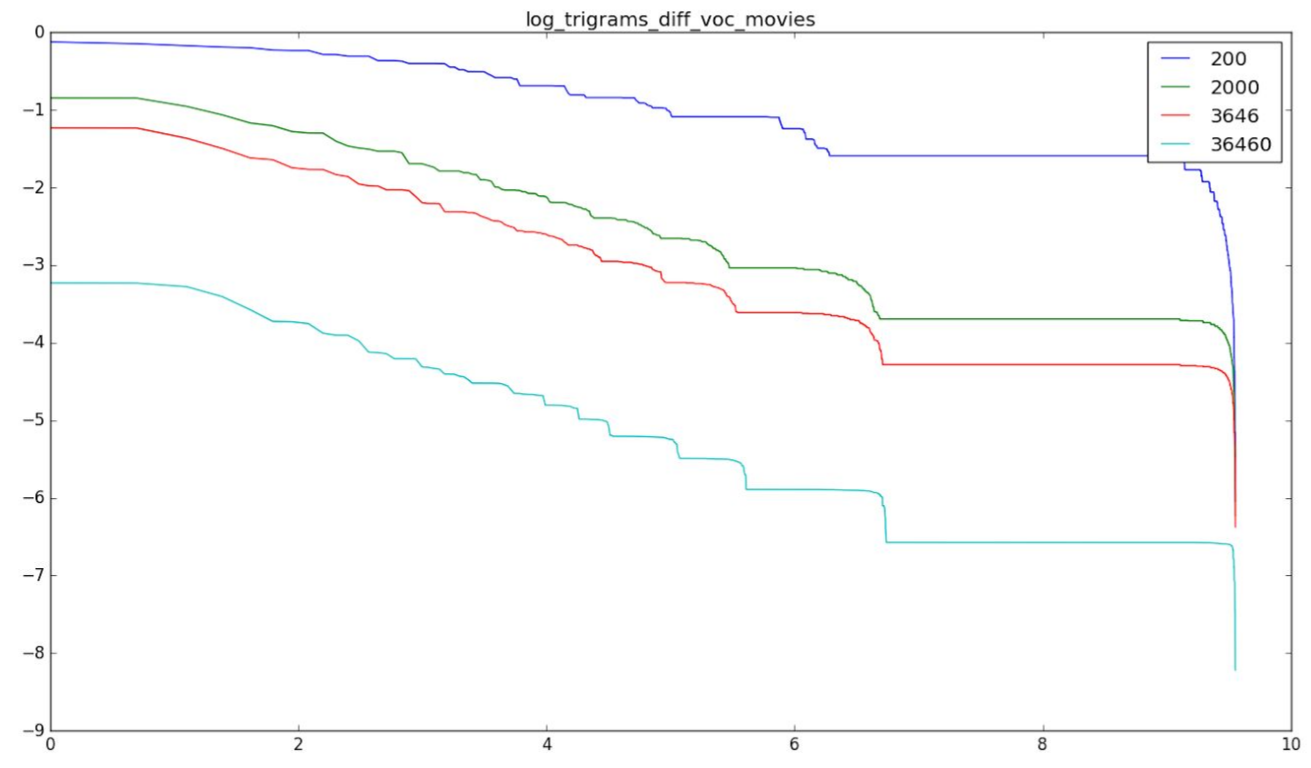

log_bigrams_diff_voc_anime
200
2000
3857
38570
0
−2
−4
−6
−8
−10
0
2
4
6
8
10


log_bigrams_diff_voc_movies
200
2000
3646
36460
0
−1
−2
−3
−4
−5
−6
−7
−8
−9
0
2
4
6
8
10

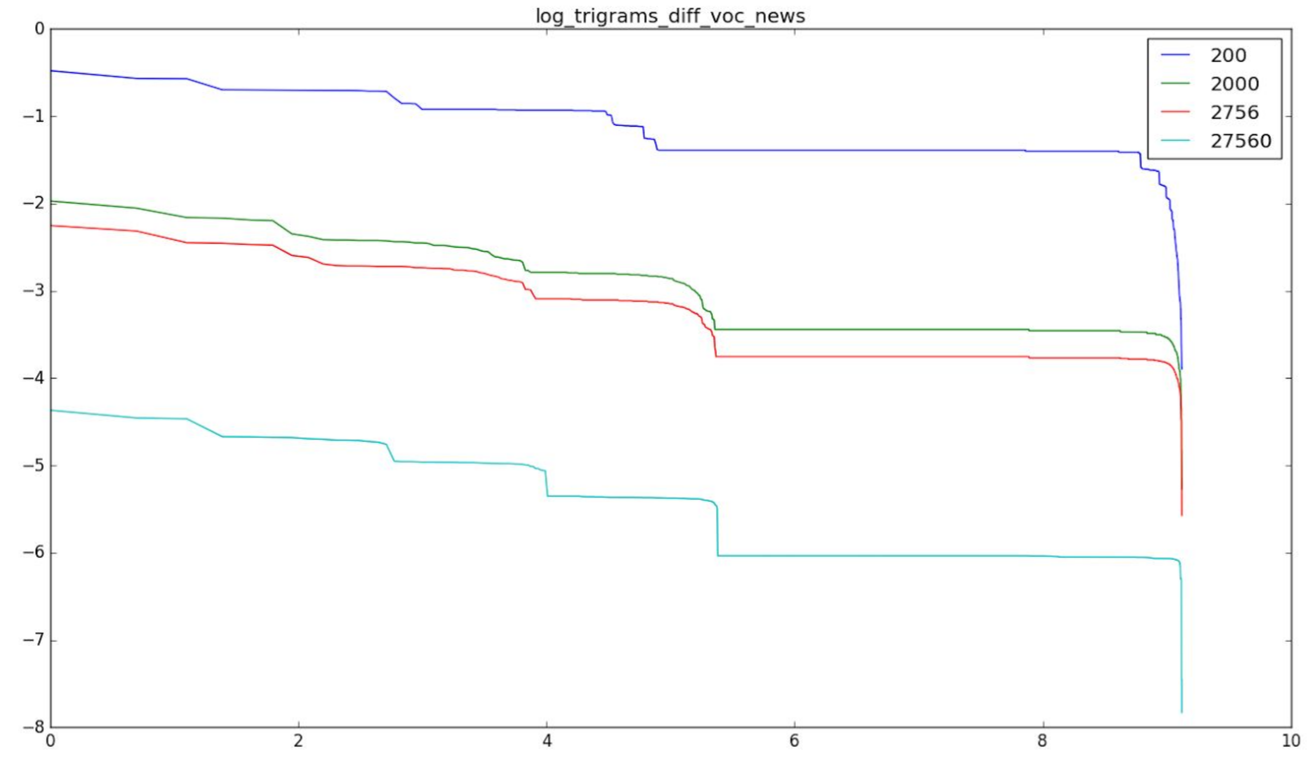

log_bigrams_diff_voc_news
200
2000
2756
27560
0
−1
−2
−3
−4
−5
−6
−7
−8
0
1
2
3
4
5
6
7
8
9



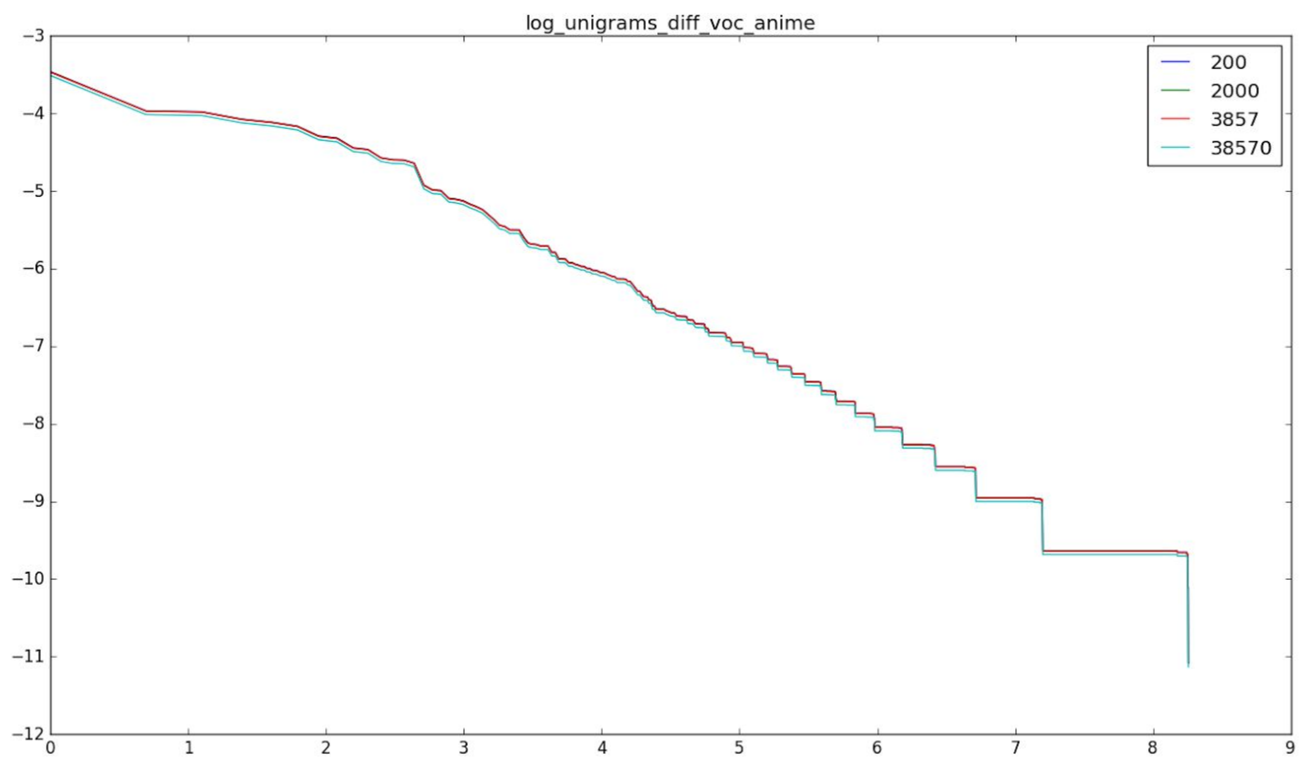
log_trigrams_diff_voc_anime
200
2000
3857
38570
0
−2
−4
−6
−8
−10
0
2
4
6
8
10

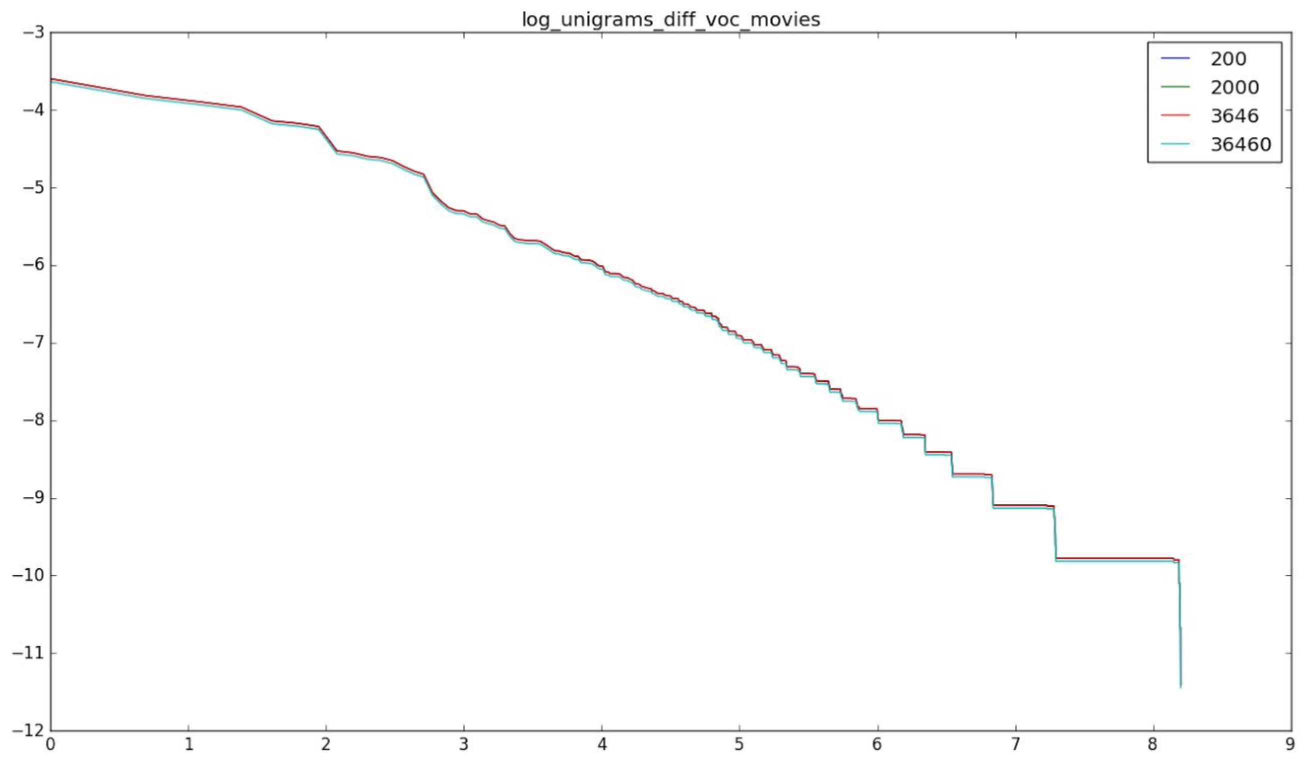


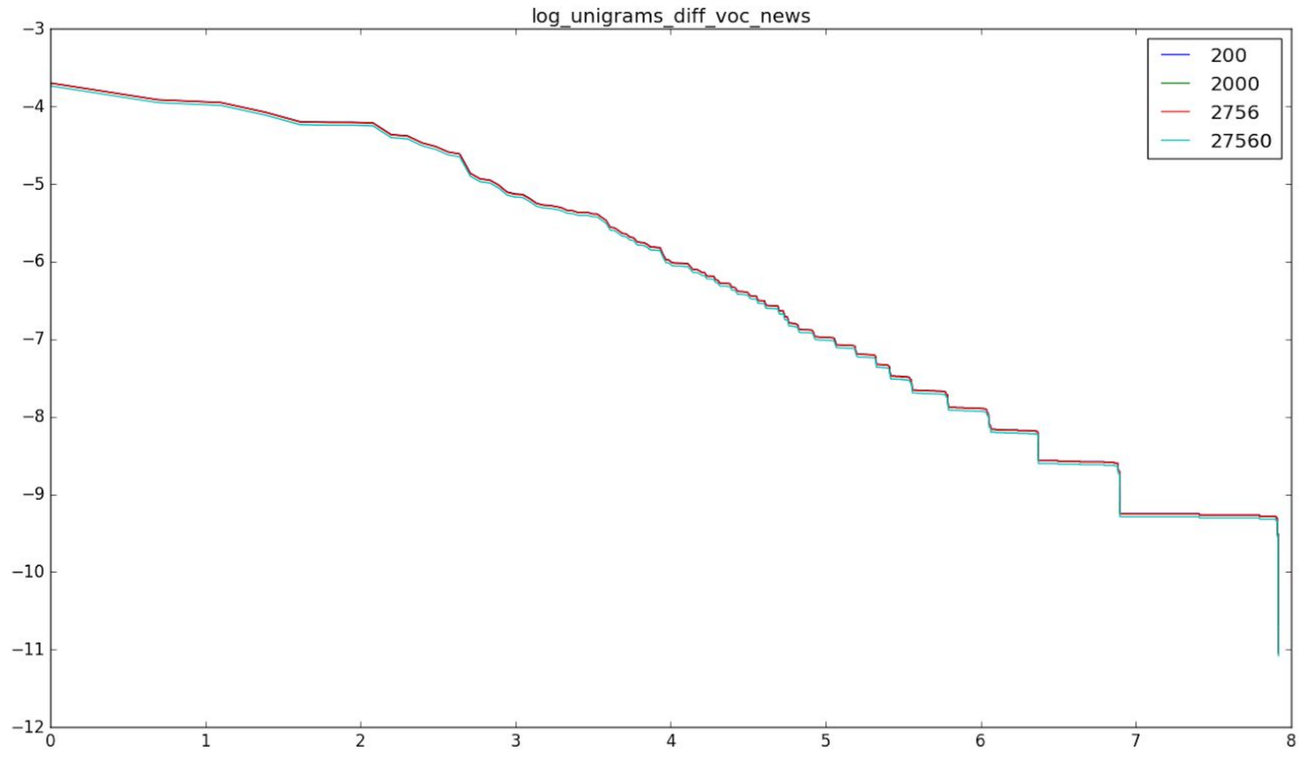
log_trigrams_diff_voc_movies
200
2000
3646
36460
0
−1
−2
−3
−4
−5
−6
−7
−8
−9
0
2
4
6
8
10

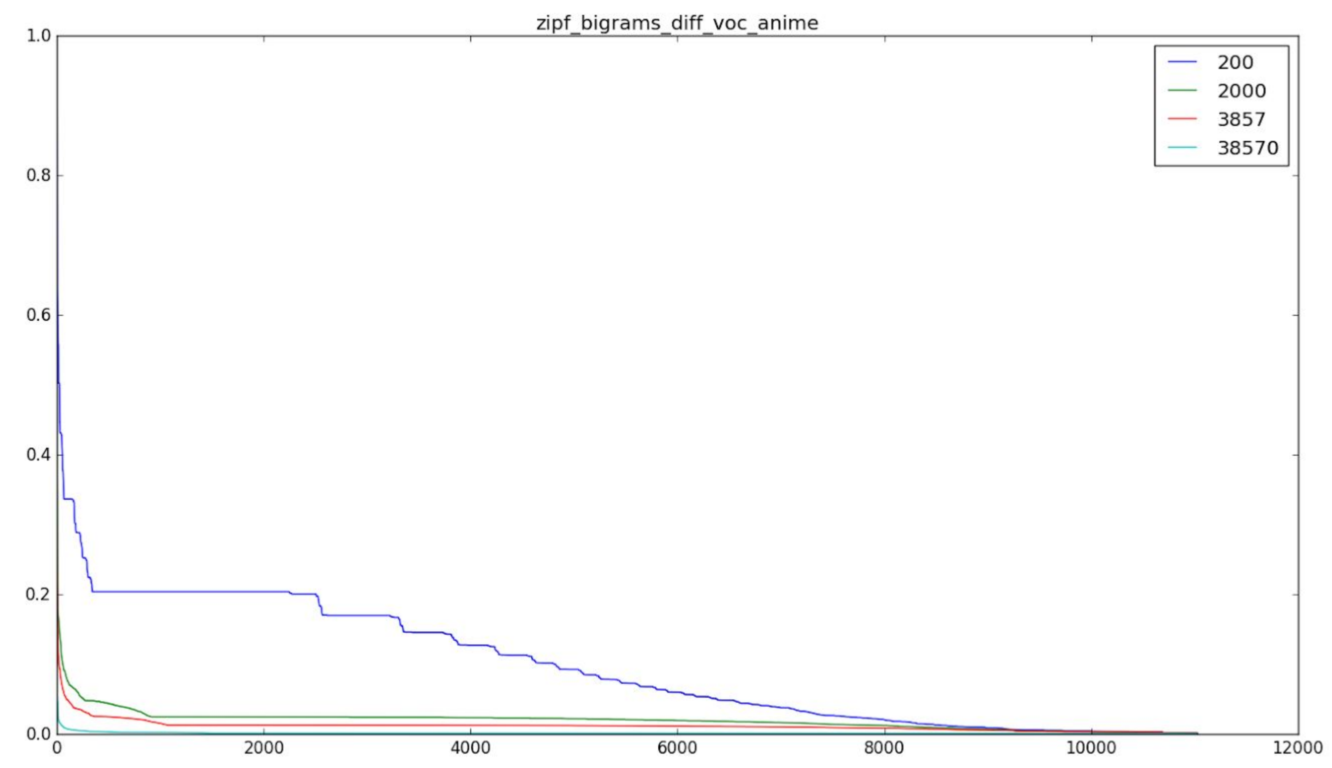


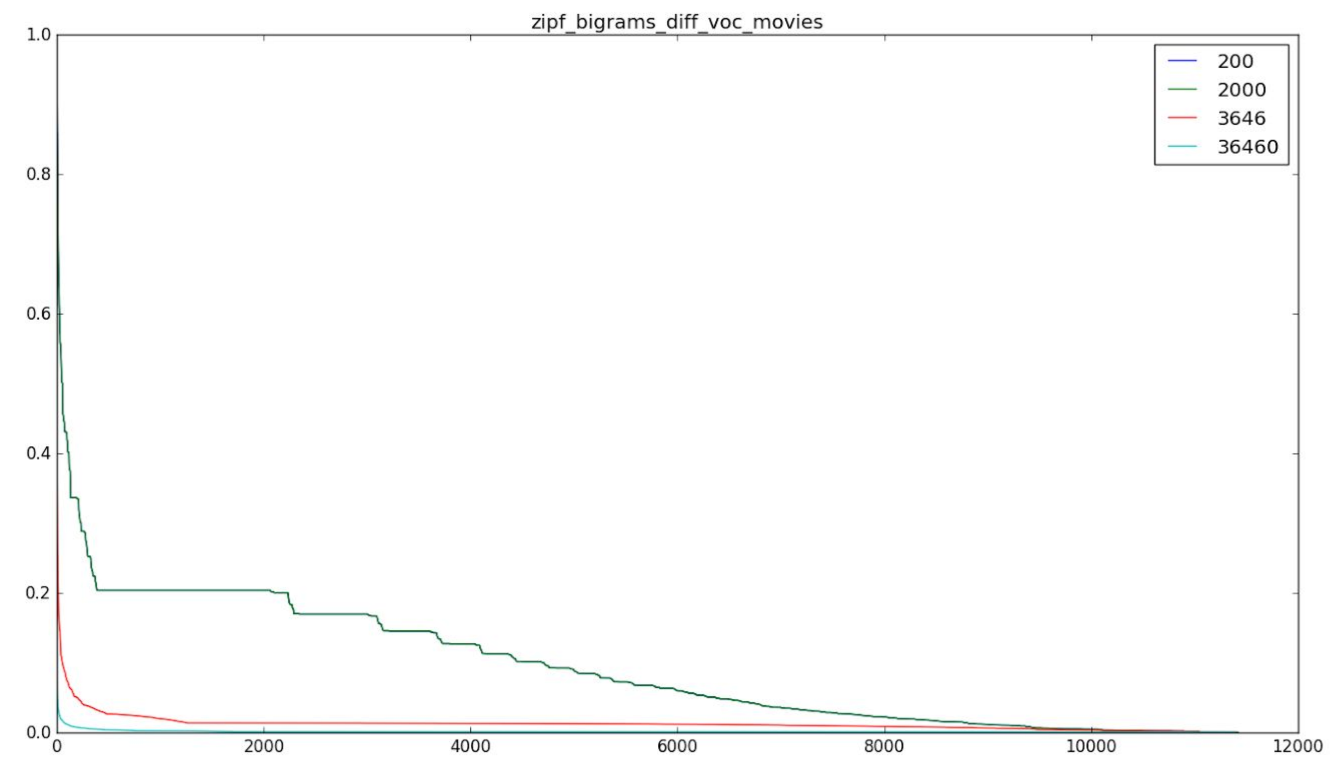
log_trigrams_diff_voc_news
200
2000
2756
27560
0
−1
−2
−3
−4
−5
−6
−7
−8
0
2
4
6
8
10

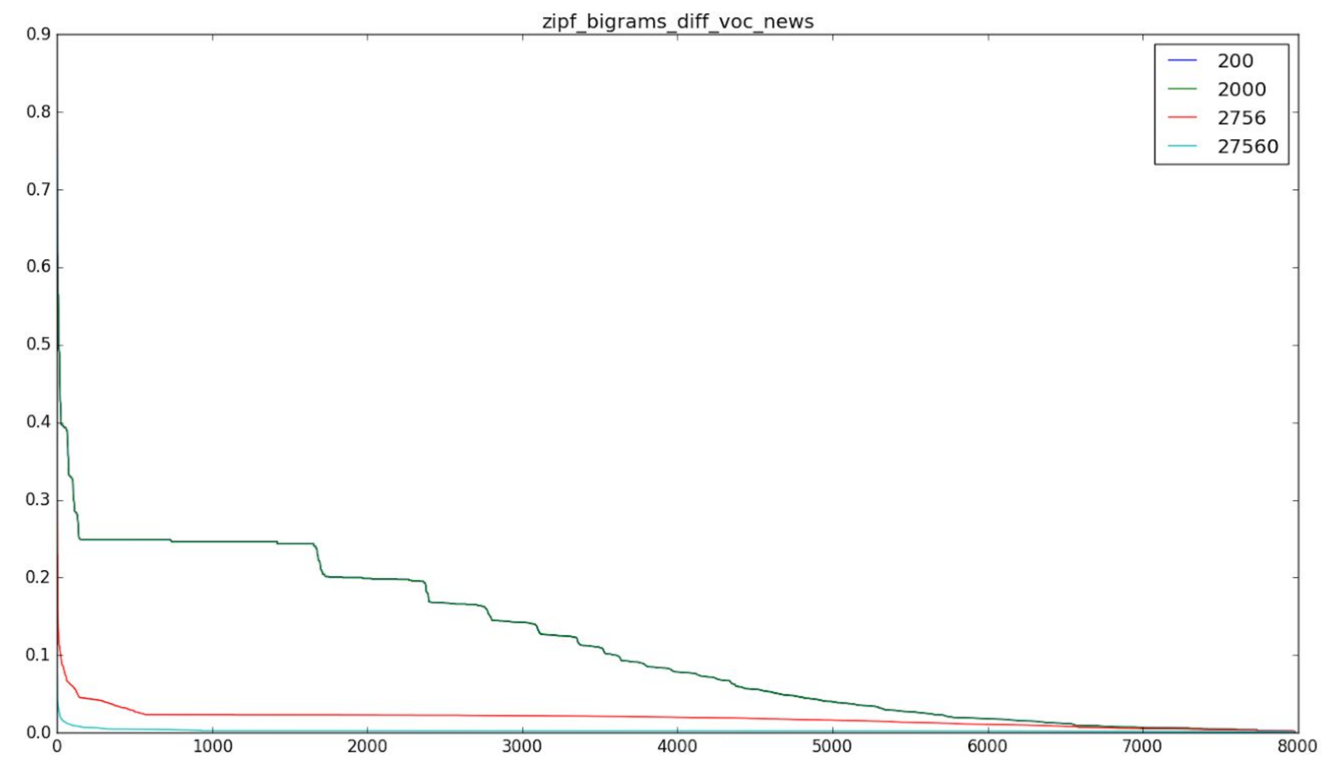

log_unigrams_diff_voc_anime
200
2000
3857
38570
−3
−4
−5
−6
−7
−8
−9
−10
−11
−12
0
1
2
3
4
5
6
7
8
9

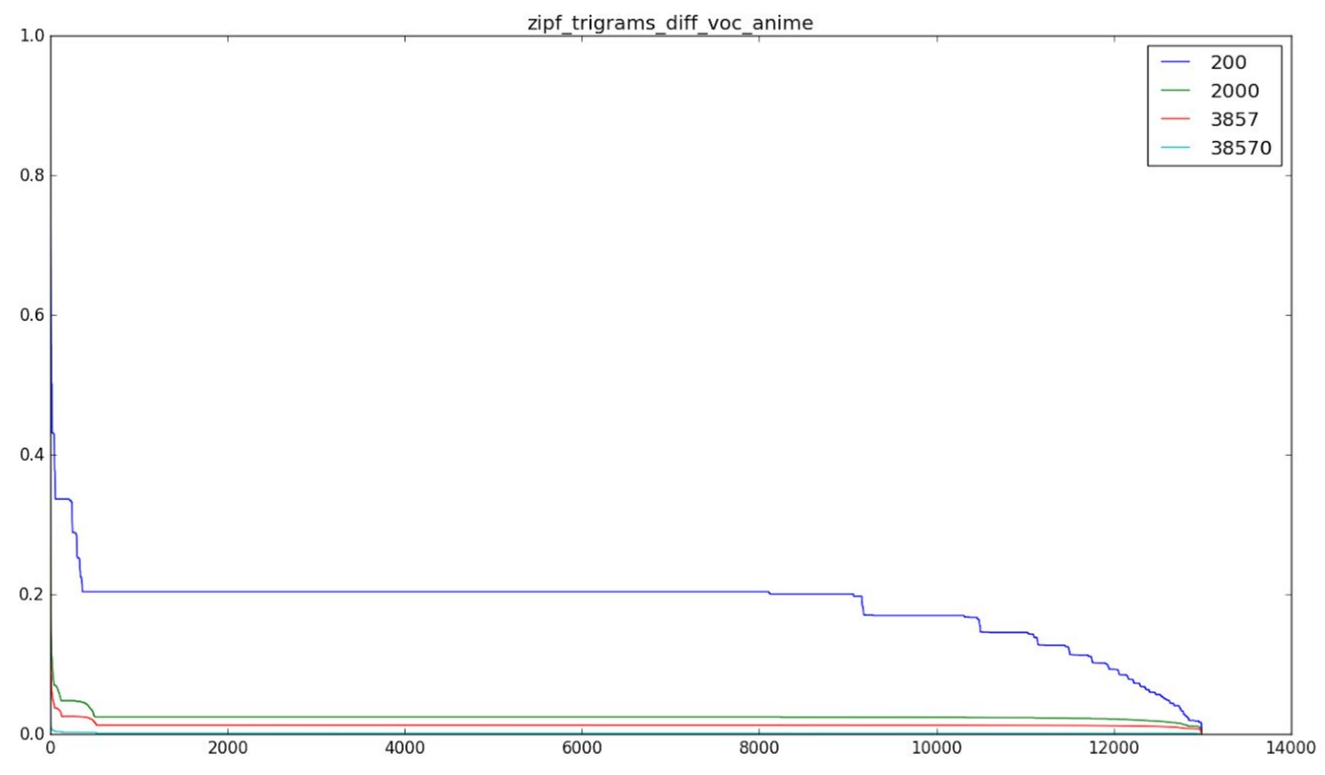

log_unigrams_diff_voc_movies
200
2000
3646
36460
−3
−4
−5
−6
−7
−8
−9
−10
−11
−12
0
1
2
3
4
5
6
7
8
9

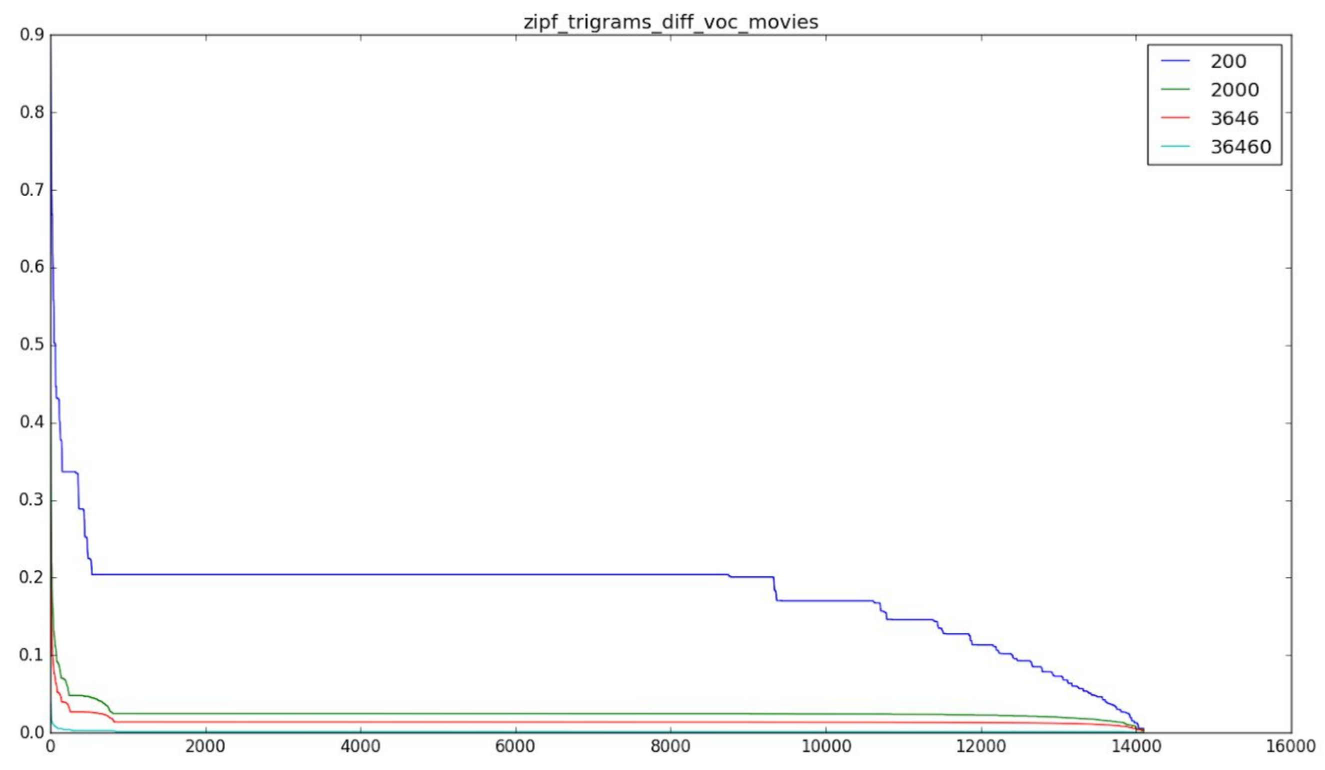

log_unigrams_diff_voc_news
200
2000
2756
27560
−3
−4
−5
−6
−7
−8
−9
−10
−11
−12
0
1
2
3
4
5
6
7
8

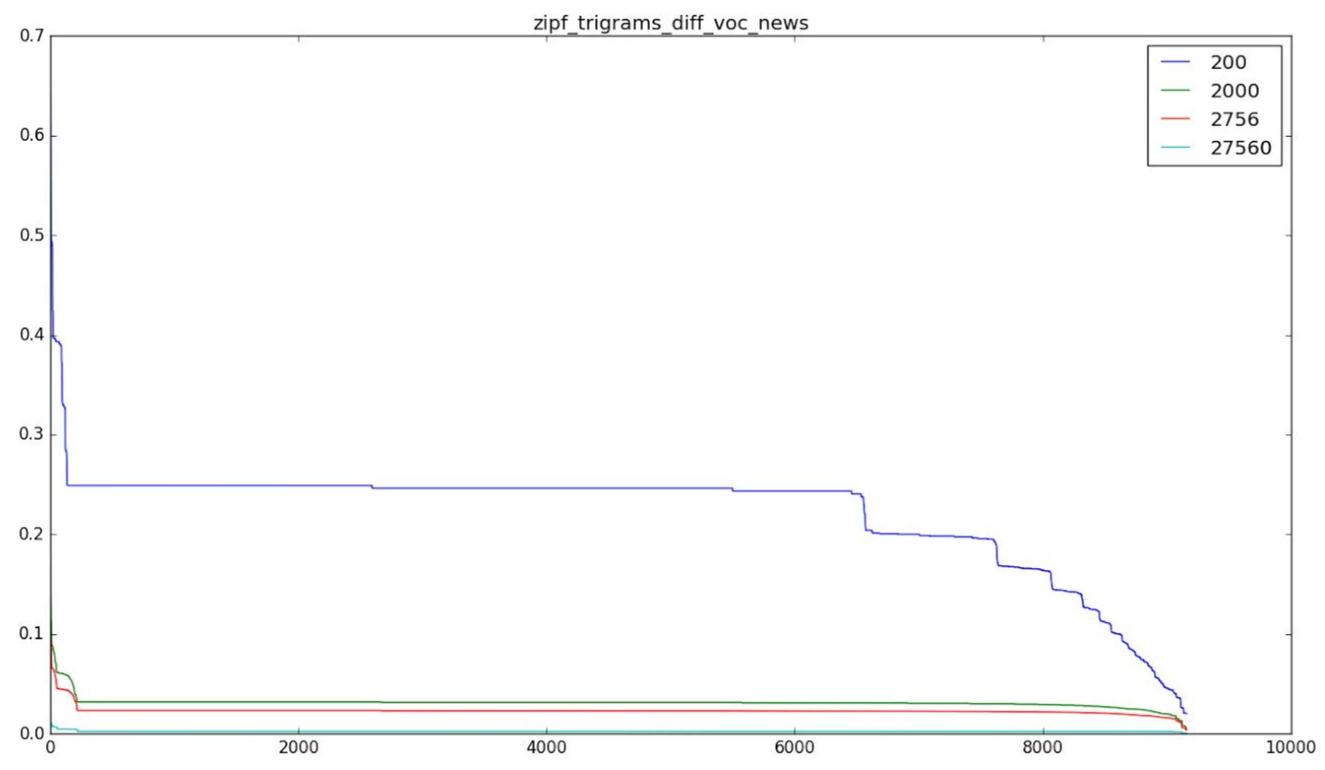

zipf_bigrams_diff_voc_anime
200
2000
3857
38570
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
12000

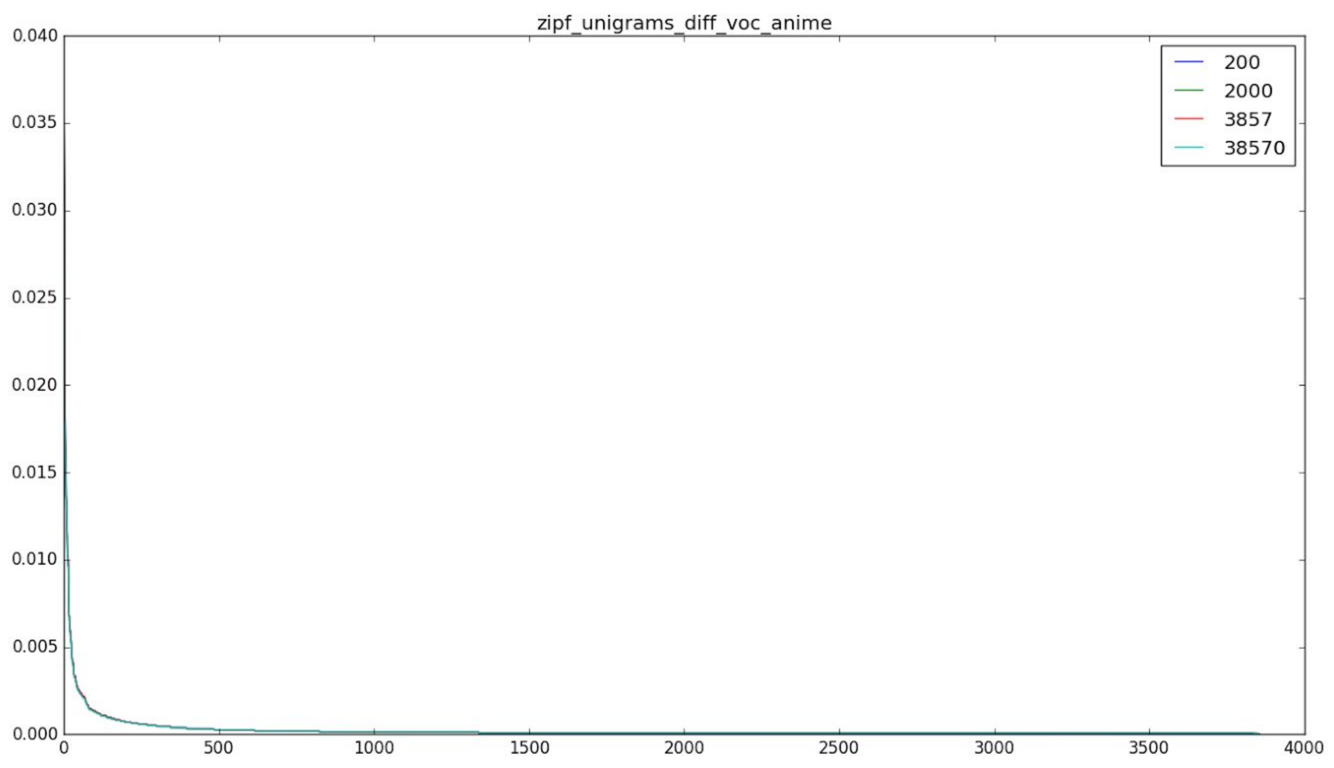

zipf_bigrams_diff_voc_movies
200
2000
3646
36460
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
12000

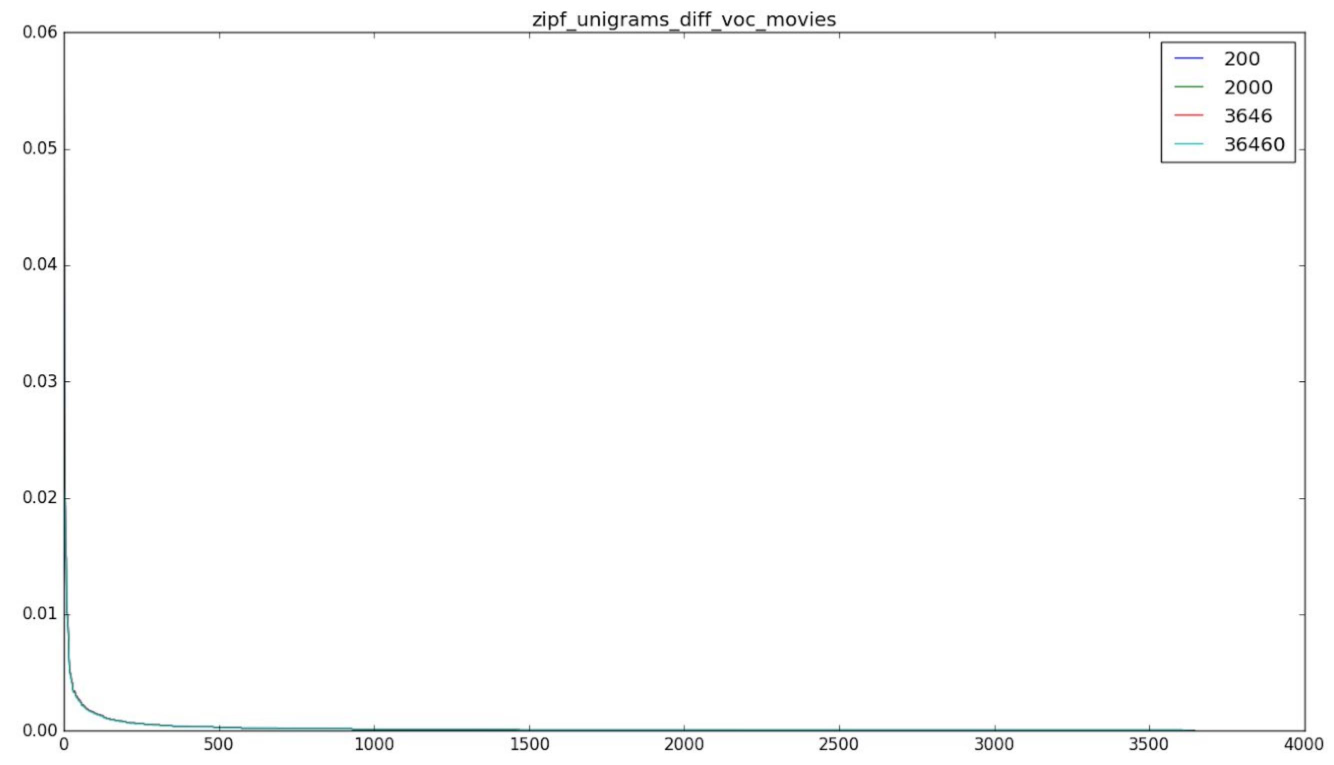

zipf_bigrams_diff_voc_news
200
2000
2756
27560
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
1000
2000
3000
4000
5000
6000
7000
8000

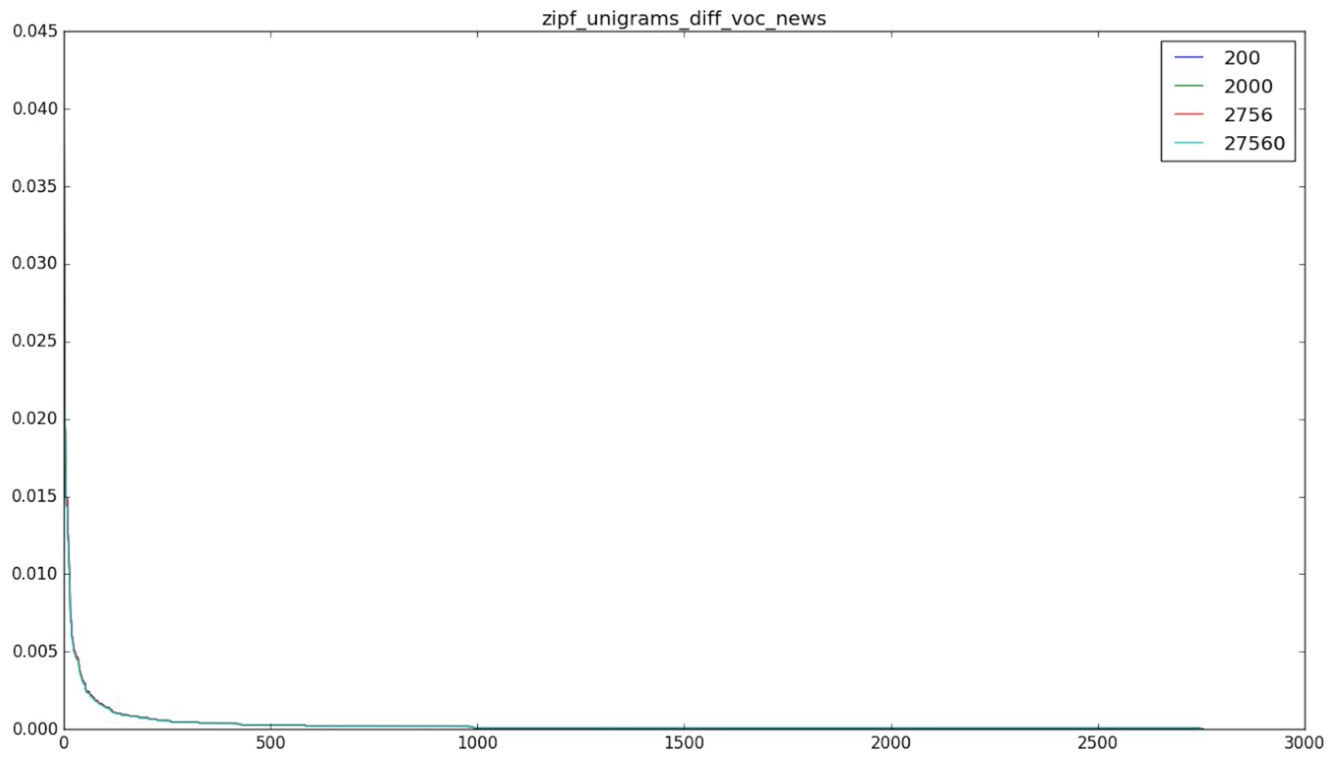

zipf_trigrams_diff_voc_anime
200
2000
3857
38570
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
12000
14000


zipf_trigrams_diff_voc_movies
200
2000
3646
36460
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2000
4000
6000
8000
10000
12000
14000
16000


zipf_trigrams_diff_voc_news
200
2000
2756
27560
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2000
4000
6000
8000
10000







4.*Implement Witten-Bell backoff.*

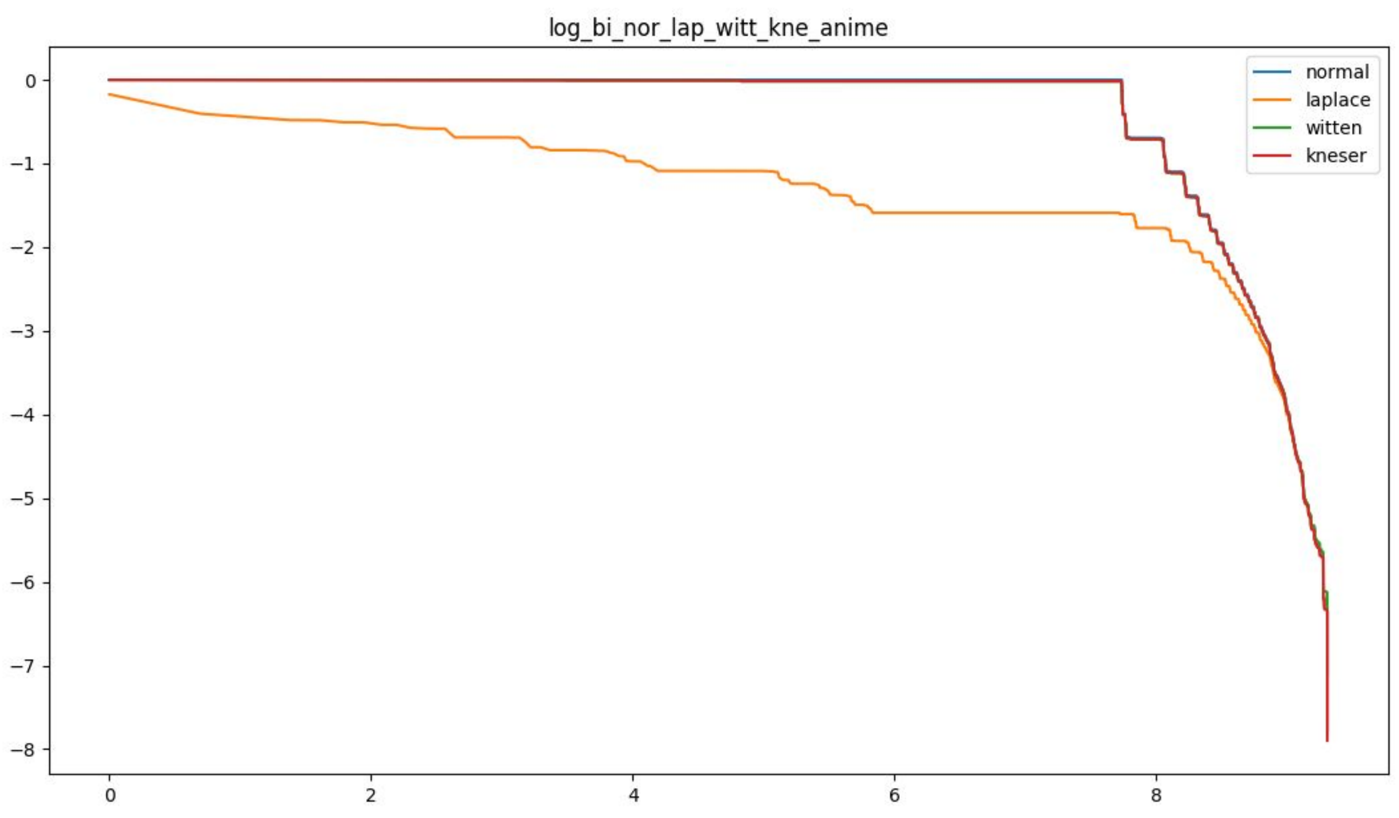
This functions could be called as shown below:

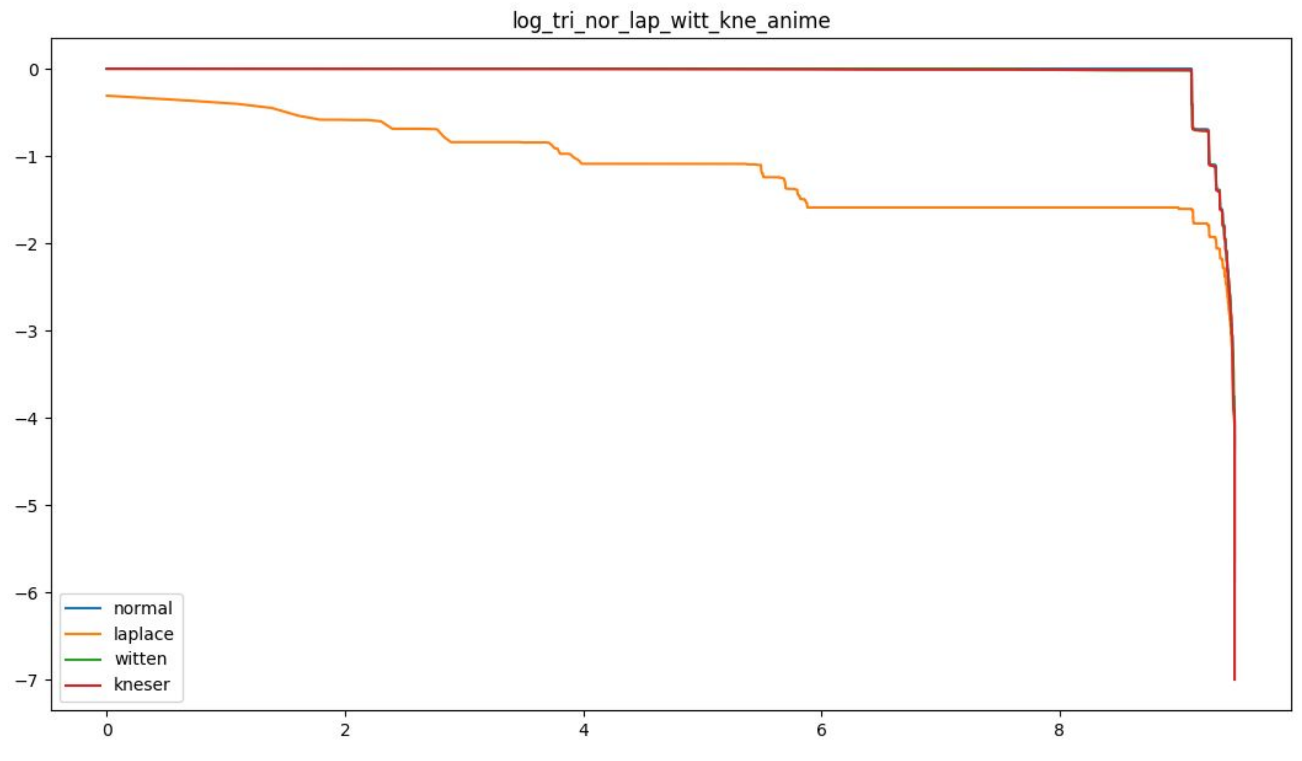
```
wittenbell_bigrams_prob = get_wittenbell_bigrams(unigrams,bigrams,unigrams_prob,wittenbell_unigrams_prob)
wittenbell_trigrams_prob = get_wittenbell_trigrams(unigrams,trigrams,trigrams_prob,wittenbell_bigrams_prob)
```

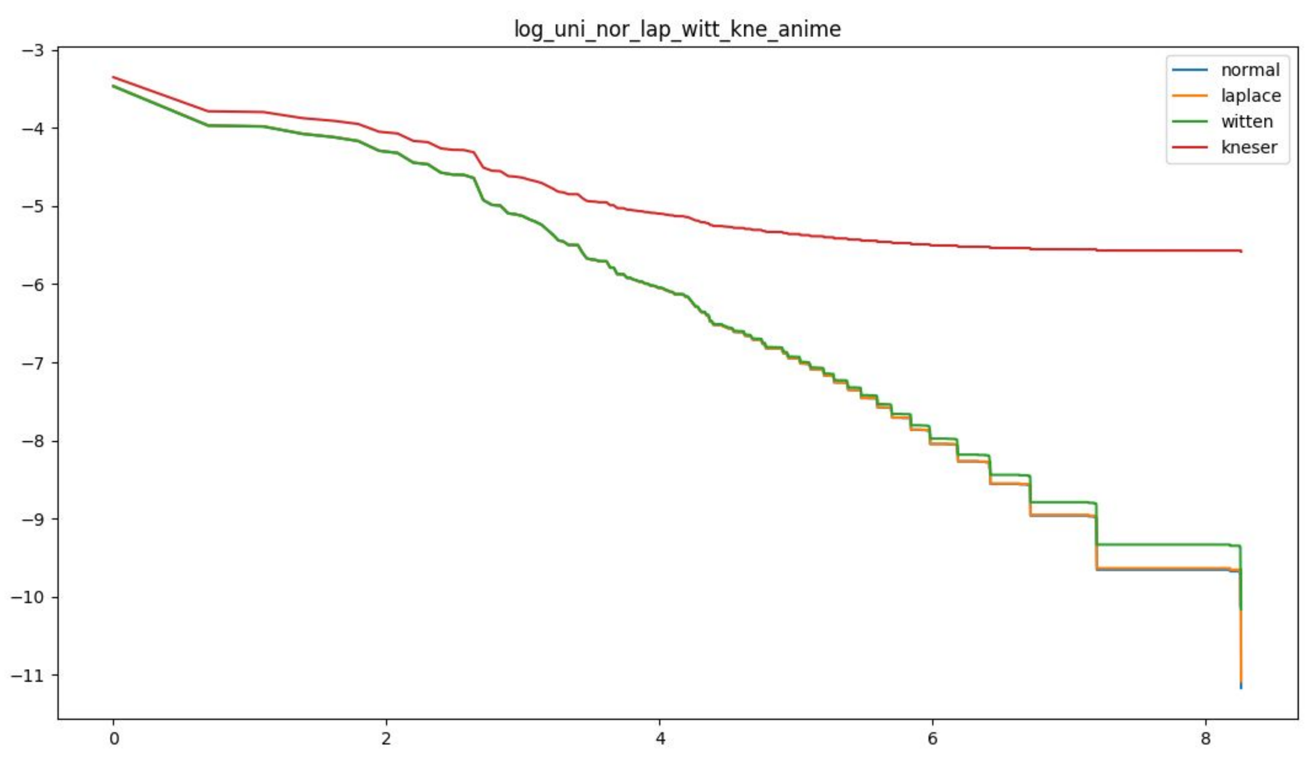
4.*Implement Kneser-Ney smoothing.*
Kneser-Ney smoothing is implemented which could be called using below functions

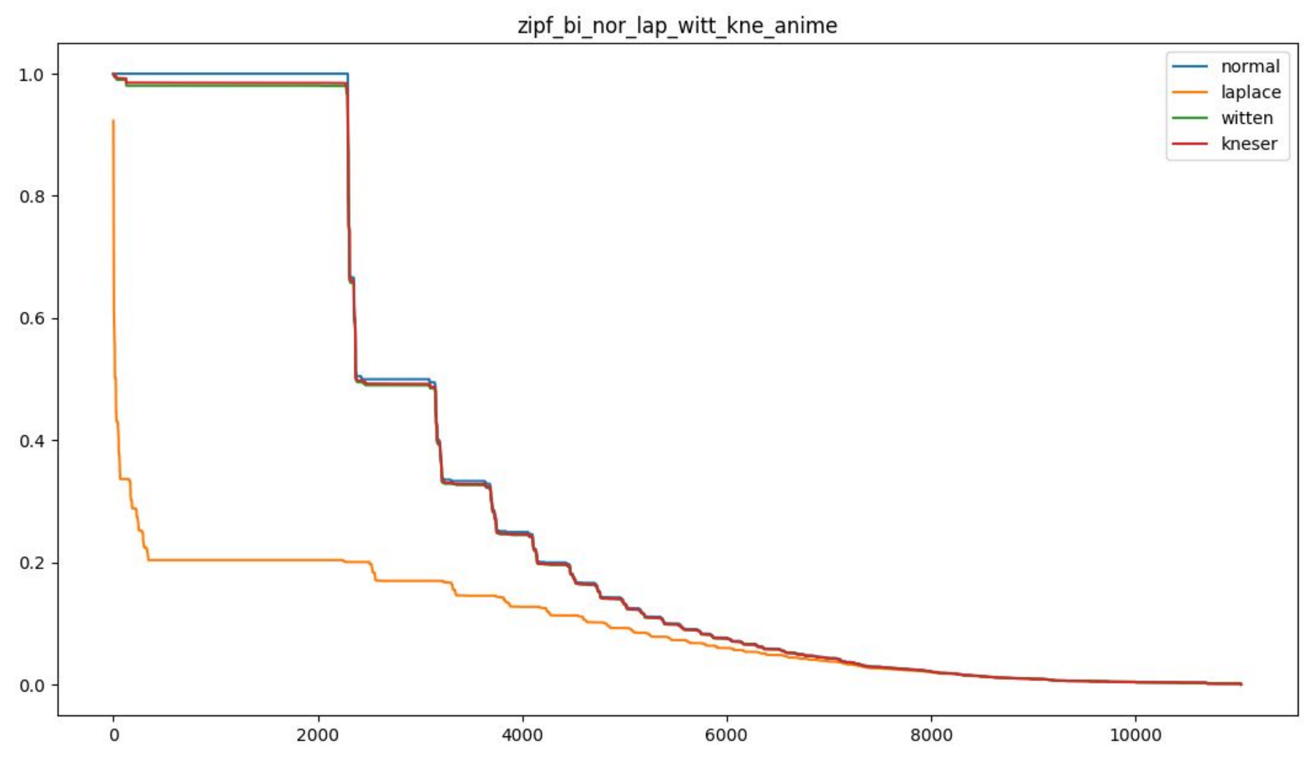
```
kn_unigrams_prob = get_kn_unigrams(unigrams,200)
kn_bigrams_prob =  get_kn_bigrams(unigrams,bigrams)
kn_trigrams_prob =  get_kn_trigrams(unigrams,bigrams,trigrams)
```

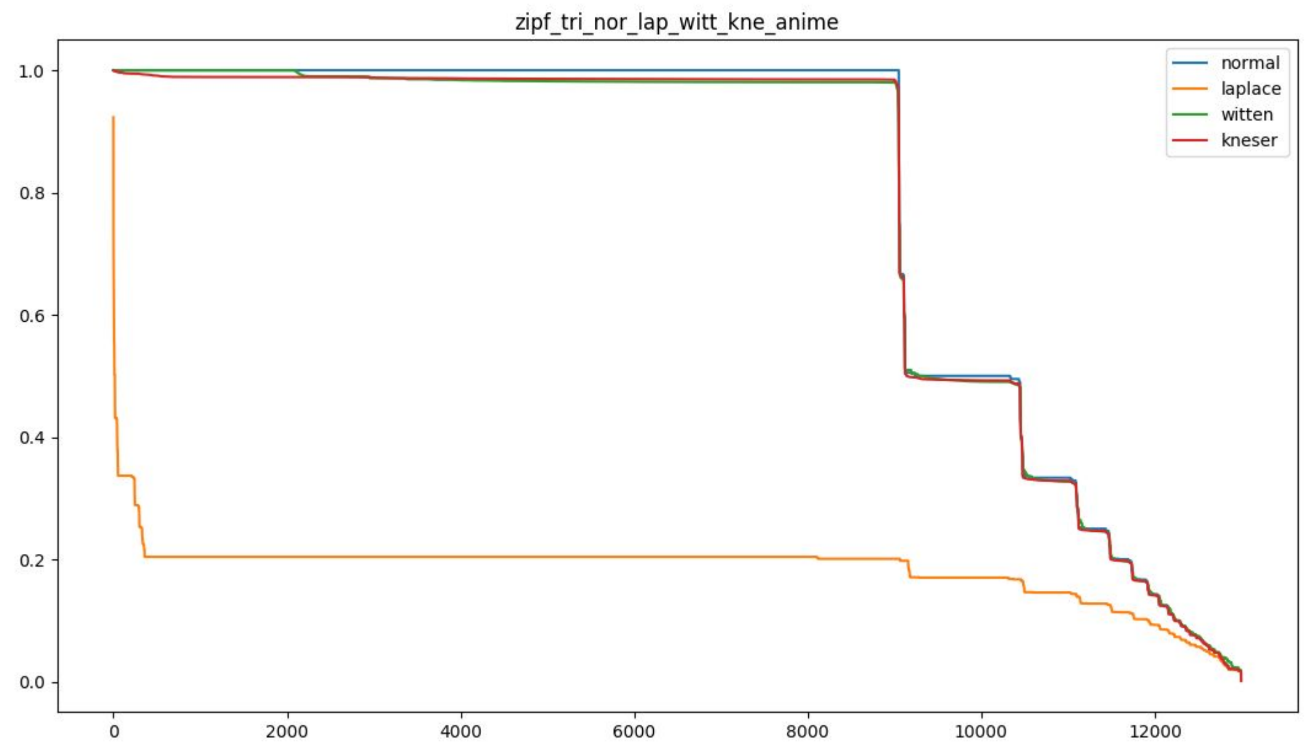
5. *Compare the three smoothing techniques:*

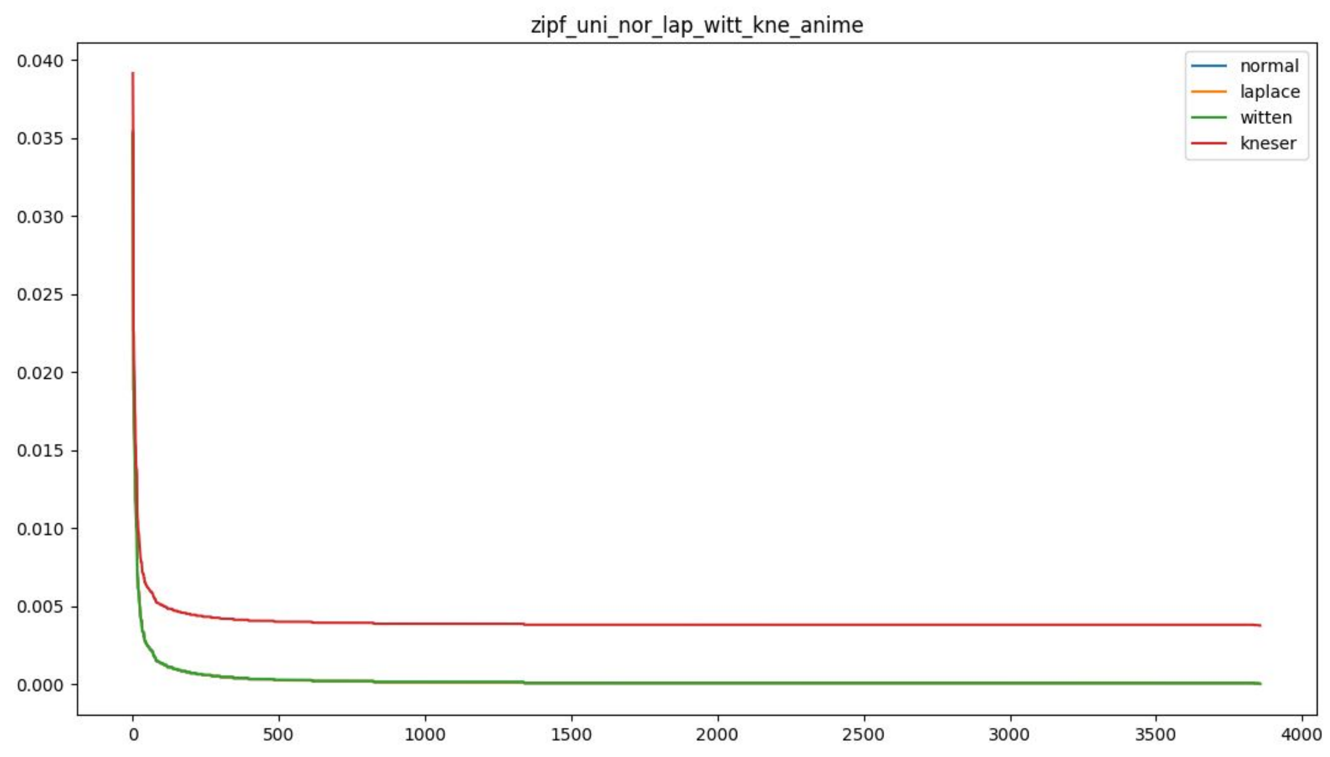




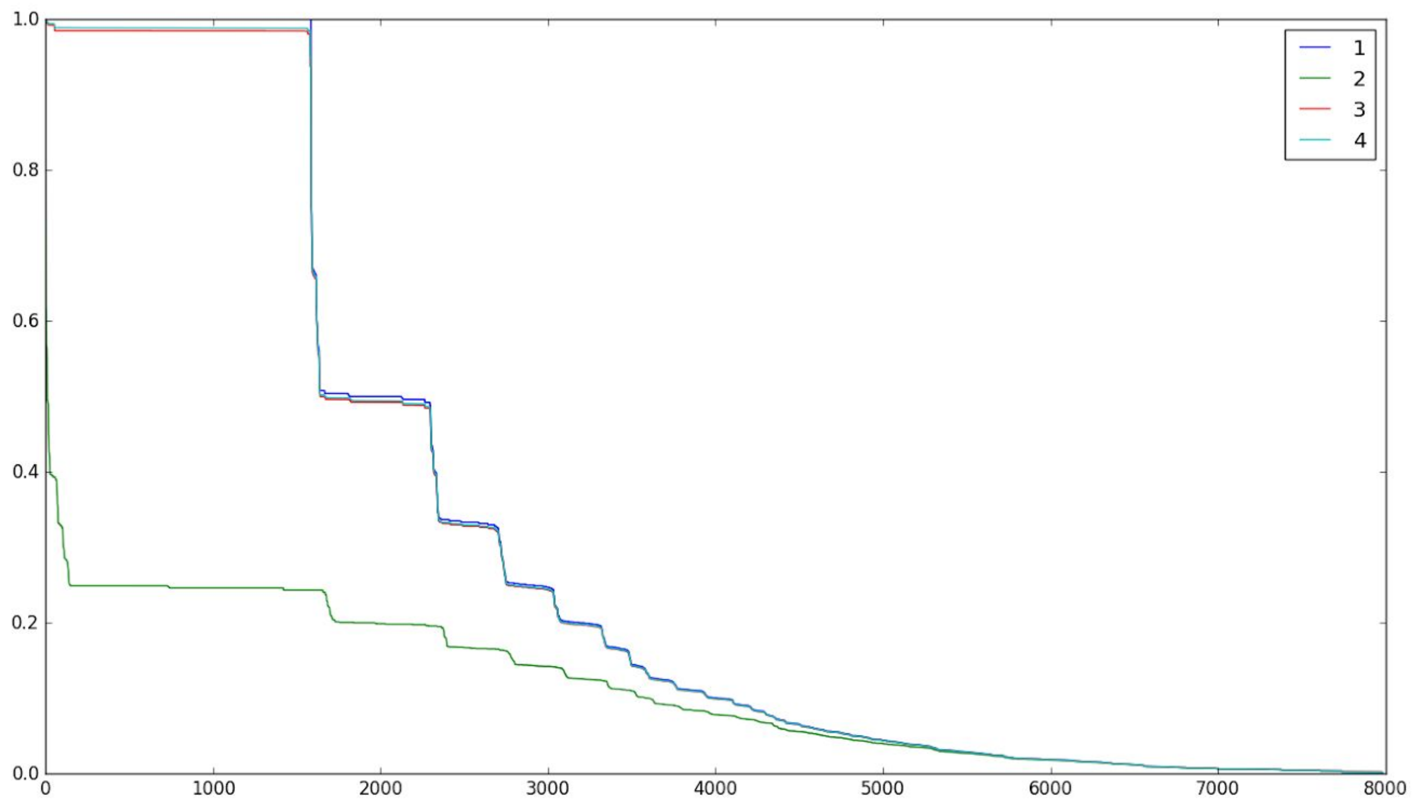

log_uni_nor_lap_witt_kne_anime
normal
laplace
witten
kneser
−3
−4
−5
−6
−7
−8
−9
−10
−11
0
2
4
6
8


zipf_bi_nor_lap_witt_kne_anime
normal
laplace
witten
kneser
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000

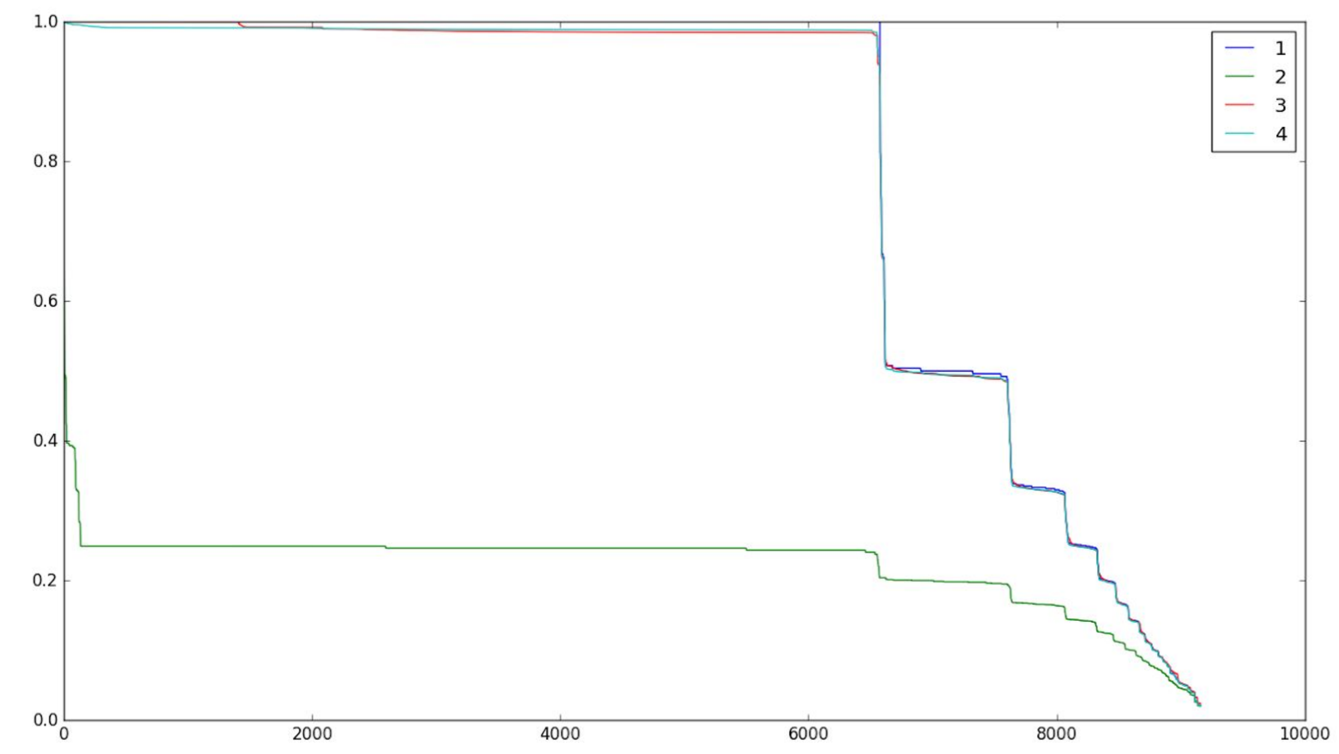

zipf_tri_nor_lap_witt_kne_anime
normal
laplace
witten
kneser
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
12000



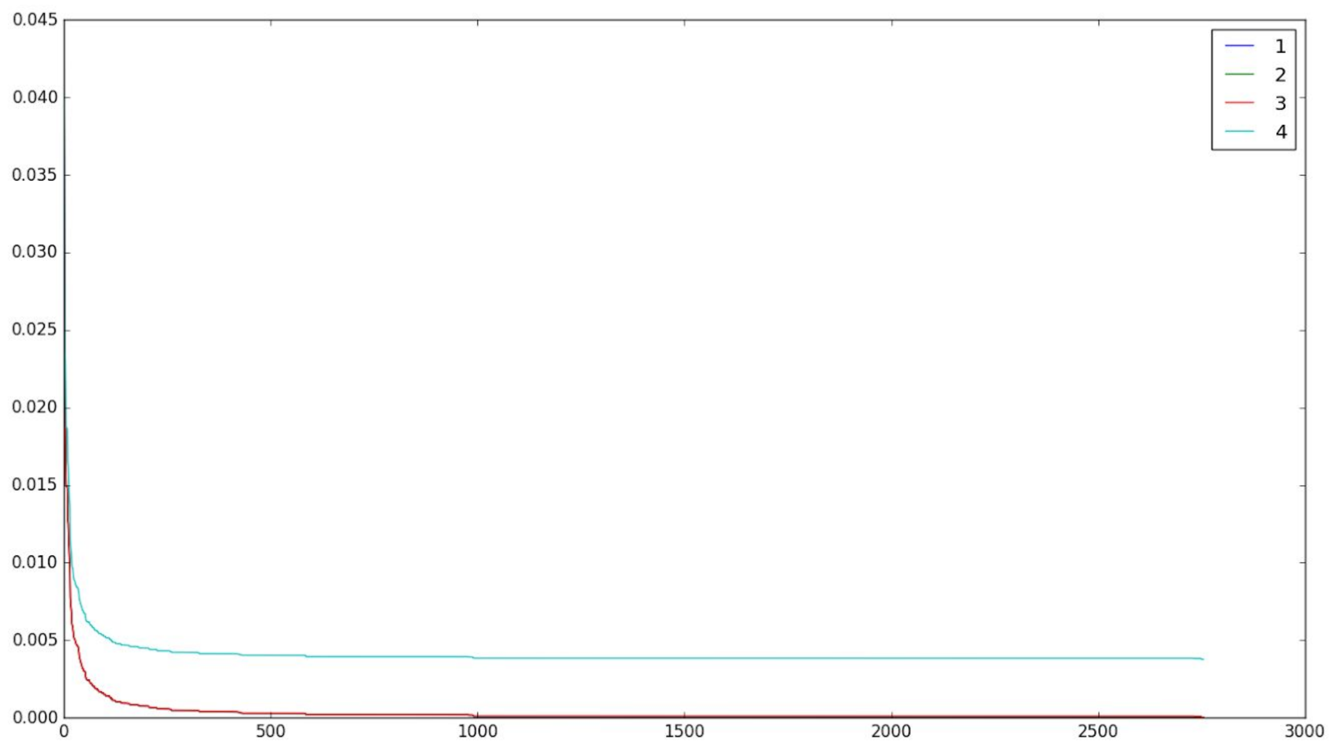
Zipf Bigram News



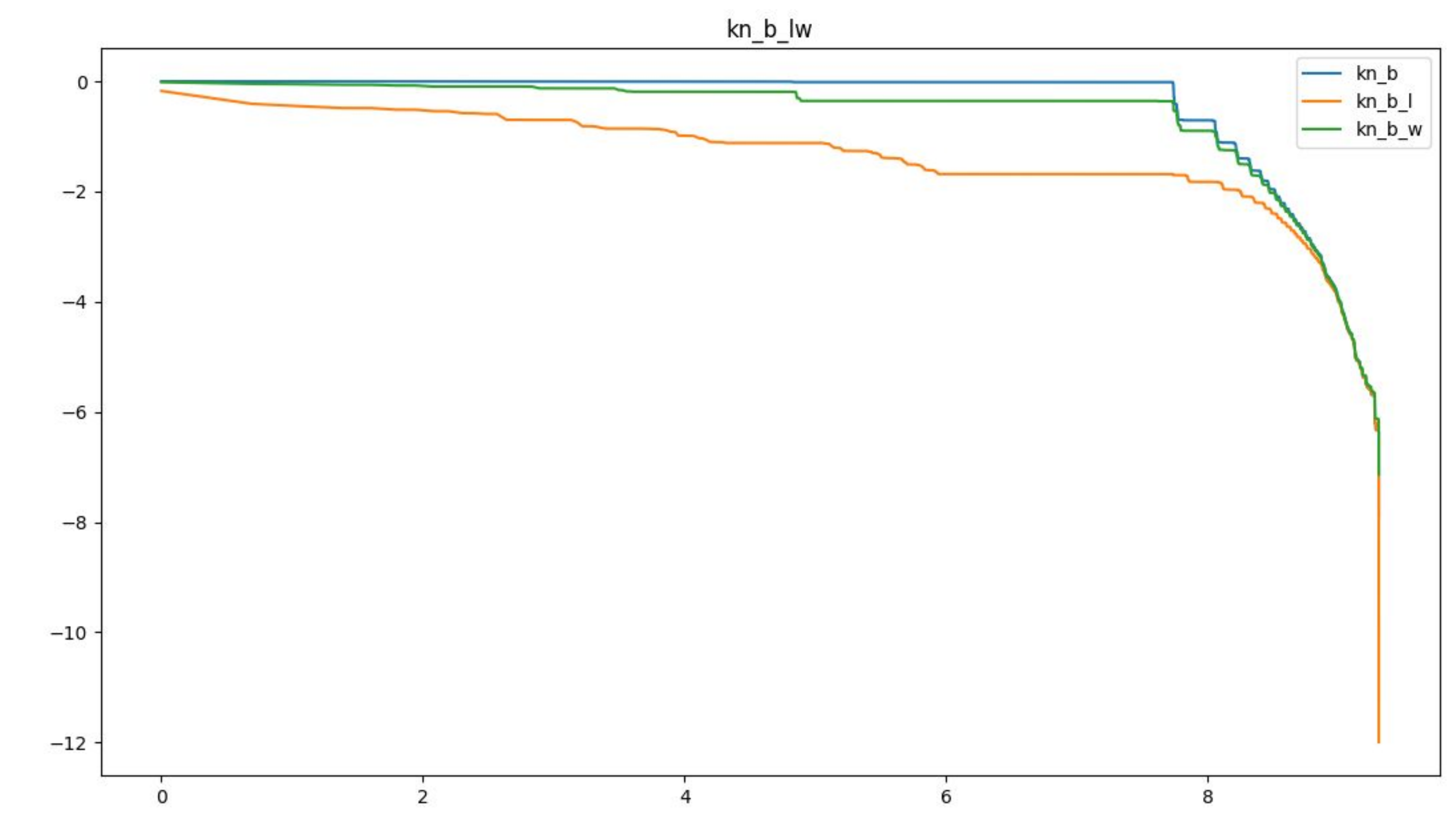
Zipf trigram News

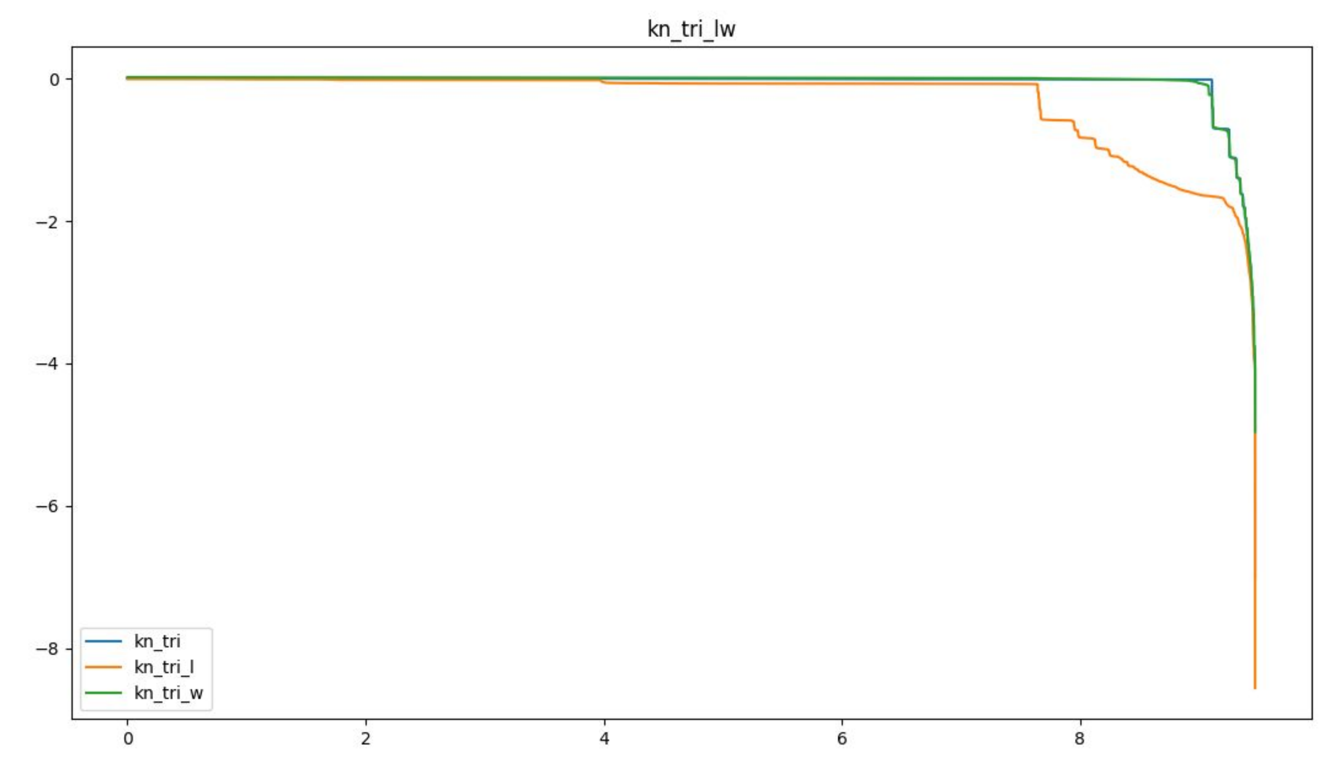


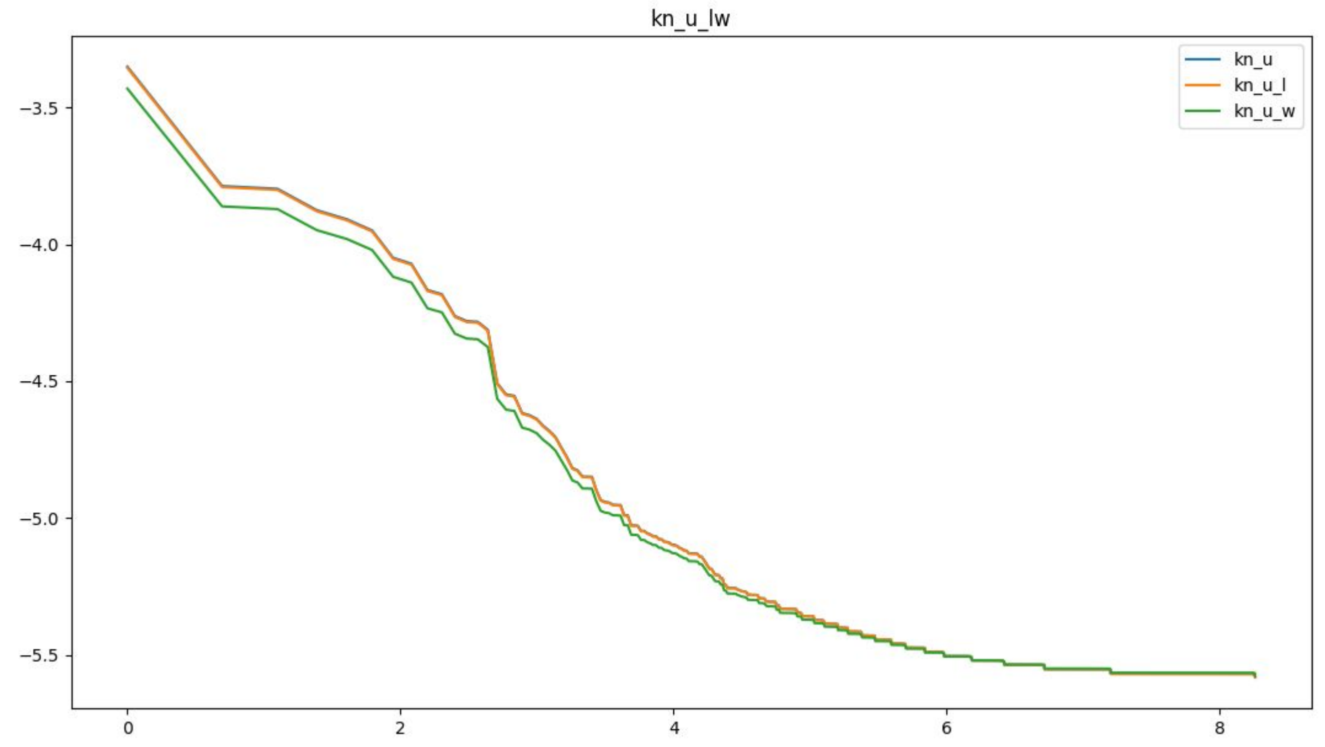
Zipf Unigram News

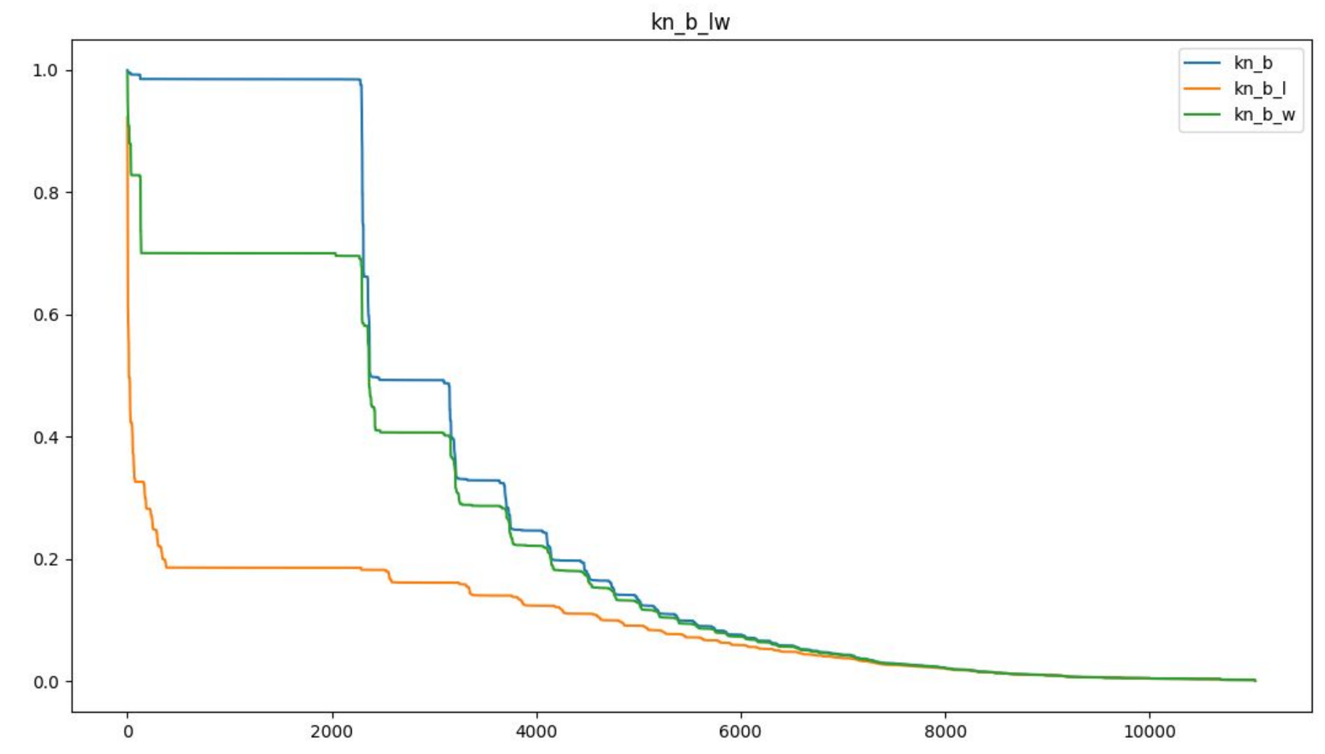


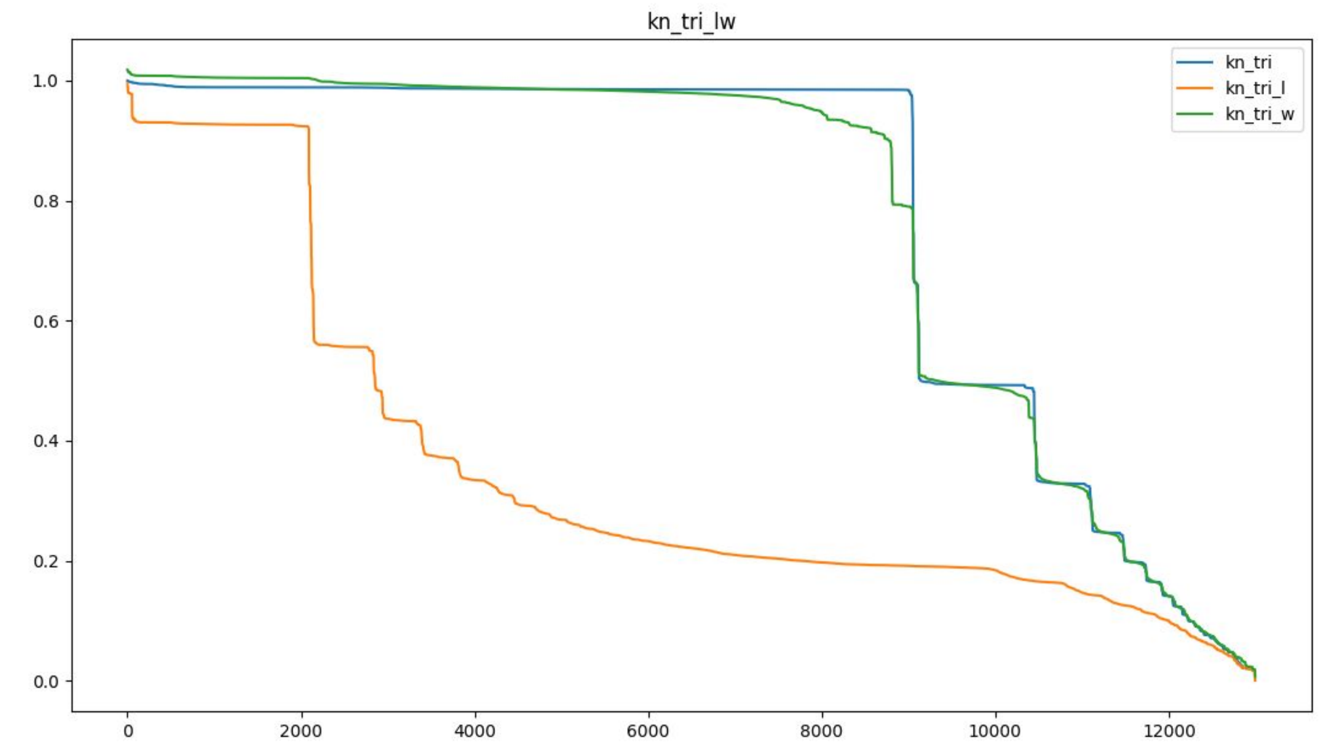
7. In Kneser-Ney, what happens if we use the estimates from laplace and wittenbell in the absolute discounting step ?

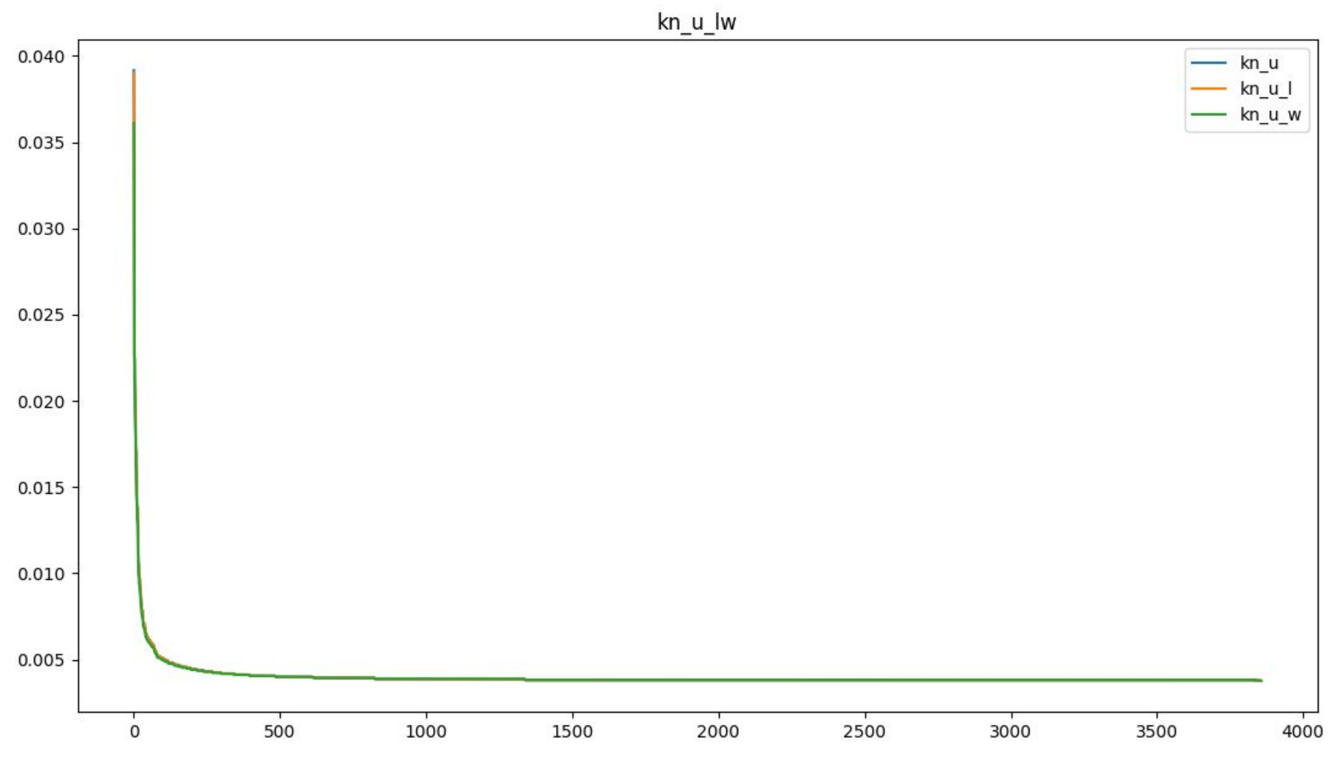

kn_b_lw
kn_b
kn_b_l
kn_b_w
0
−2
−4
−6
−8
−10
−12
0
2
4
6
8


kn_tri_lw
kn_tri
kn_tri_l
kn_tri_w
0
−2
−4
−6
−8
0
2
4
6
8


kn_u_lw
kn_u
kn_u_l
kn_u_w
−3.5
−4.0
−4.5
−5.0
−5.5
0
2
4
6
8







8. U*sing KN-estimates from the three sources, generate text with unigram, bigram and trigram probabilities.*

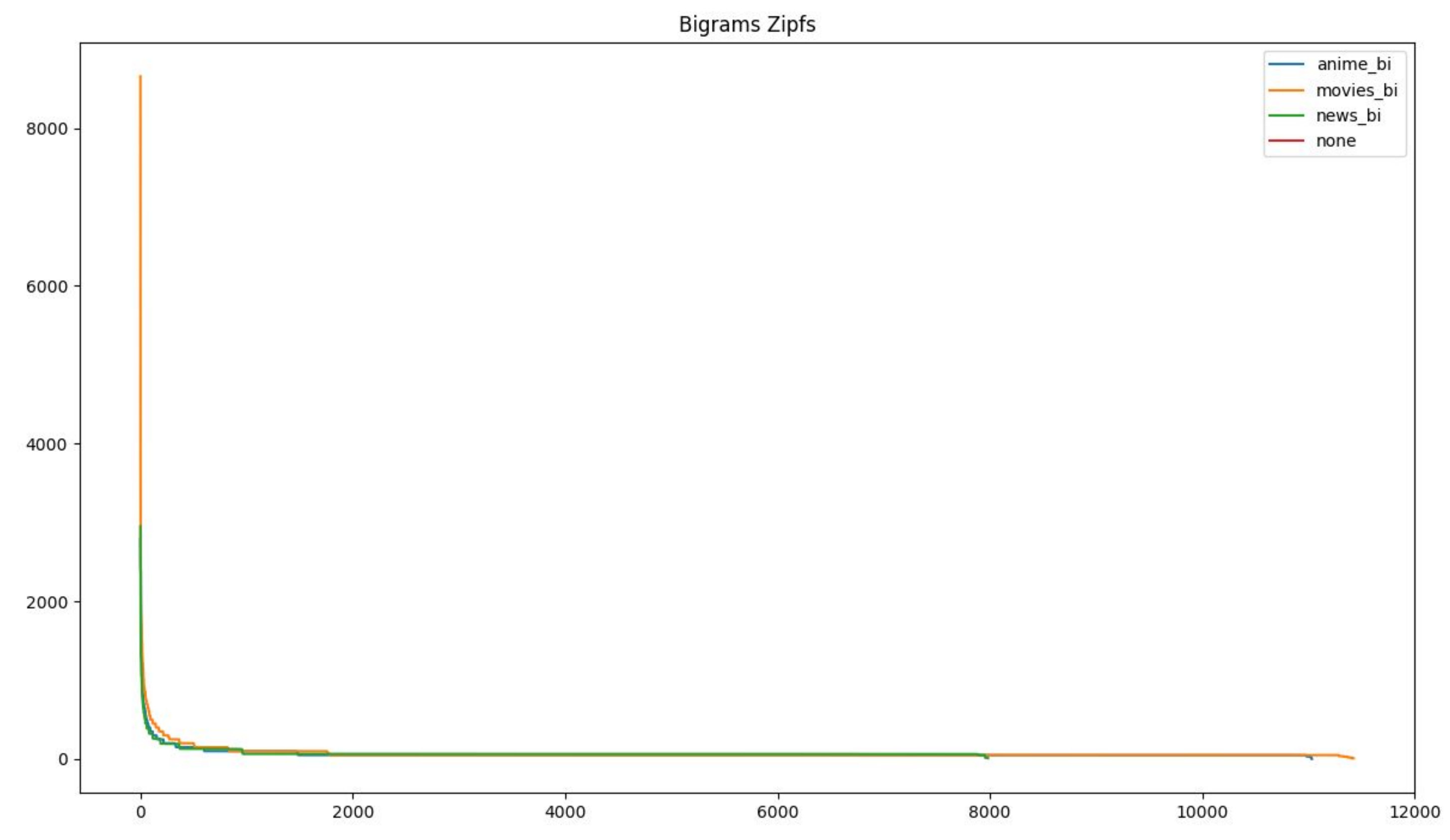
Generated Texts:

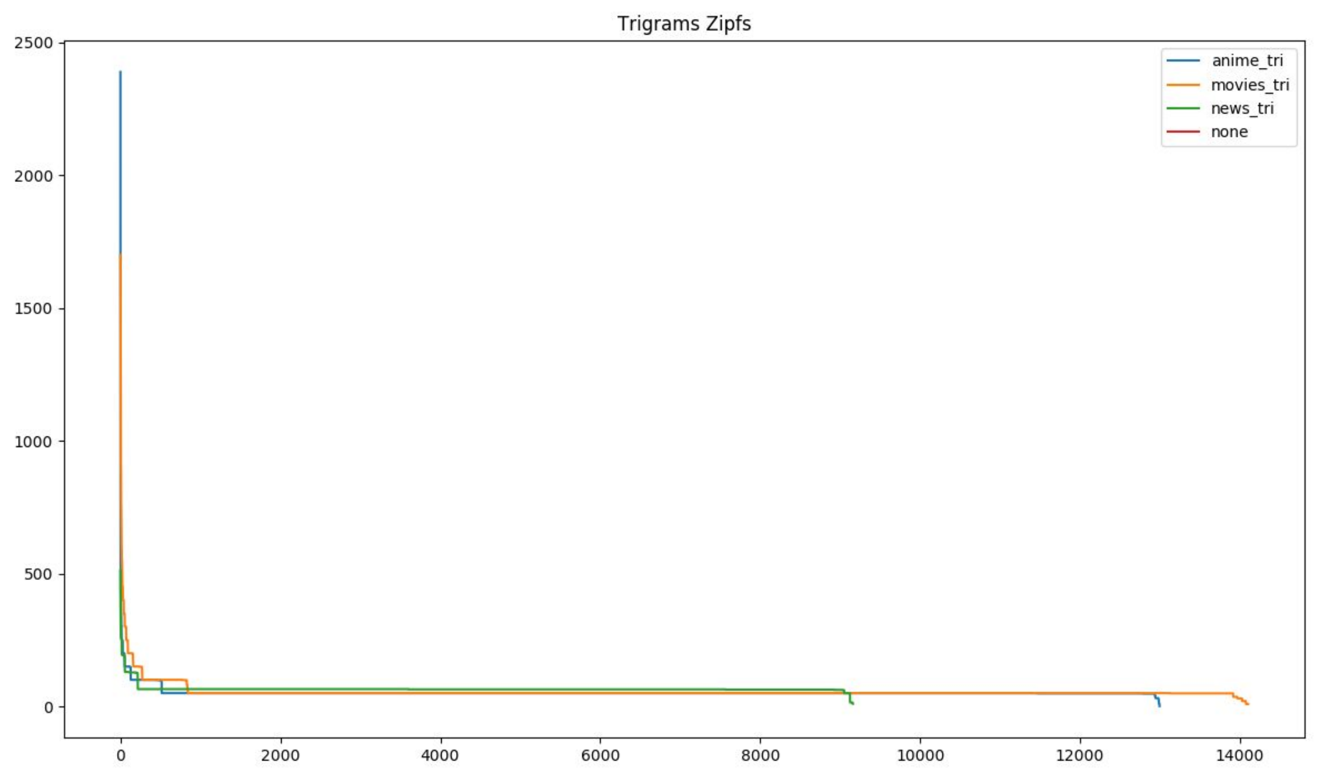
For trigram:
['team', 'is', 'back', 'it', 'was', 'a', 'little', 'too', 'much', 'and']
['man', 'am', 'i', 'the', 'only', 'one', 'of', 'my', 'favorite', 'anime']

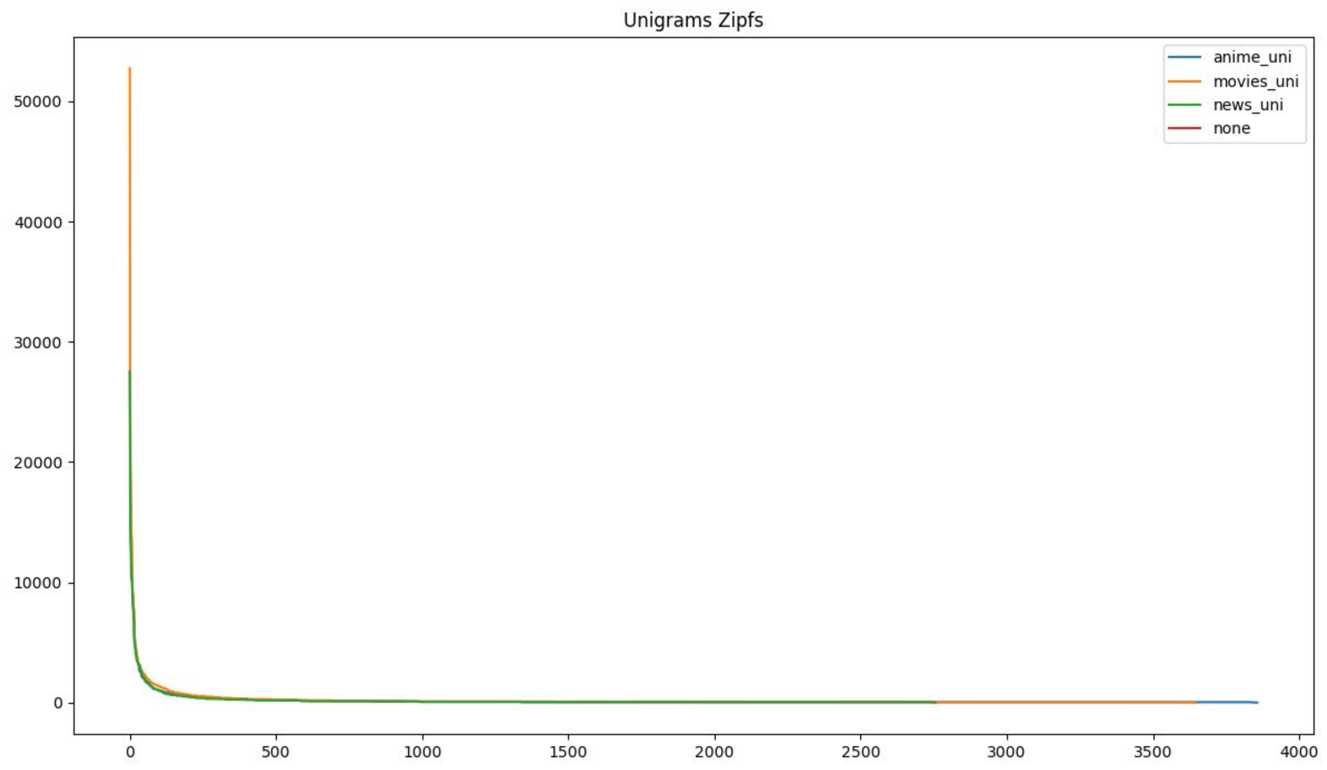
For Bigrams:
['planet', 'but', 'i', 'm', 'not']
['im', 'thoroughly', 'enjoying', 'seeing', 'misaki', 's', 'real', 'personality', 'as', 'well']

Naive Bayes

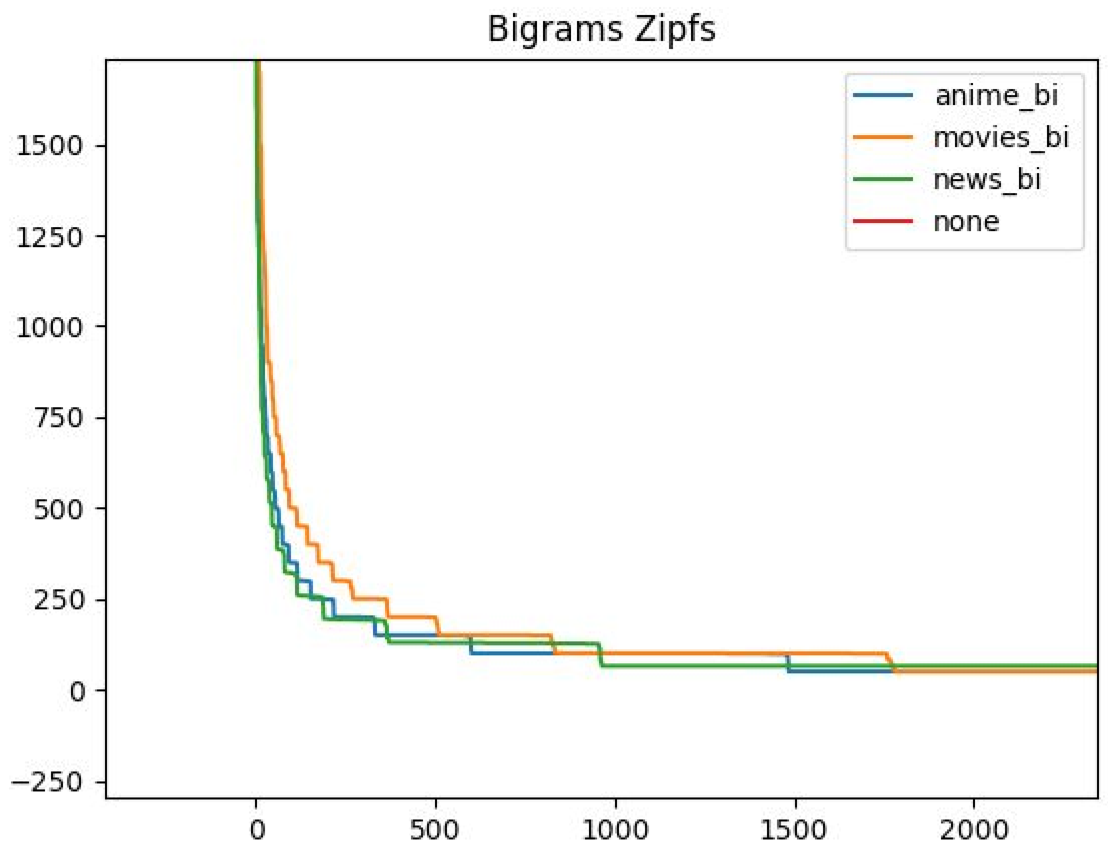
Plot the zipf's curves of all the three sources on one graph. Where do they match ? Where don't they match ?

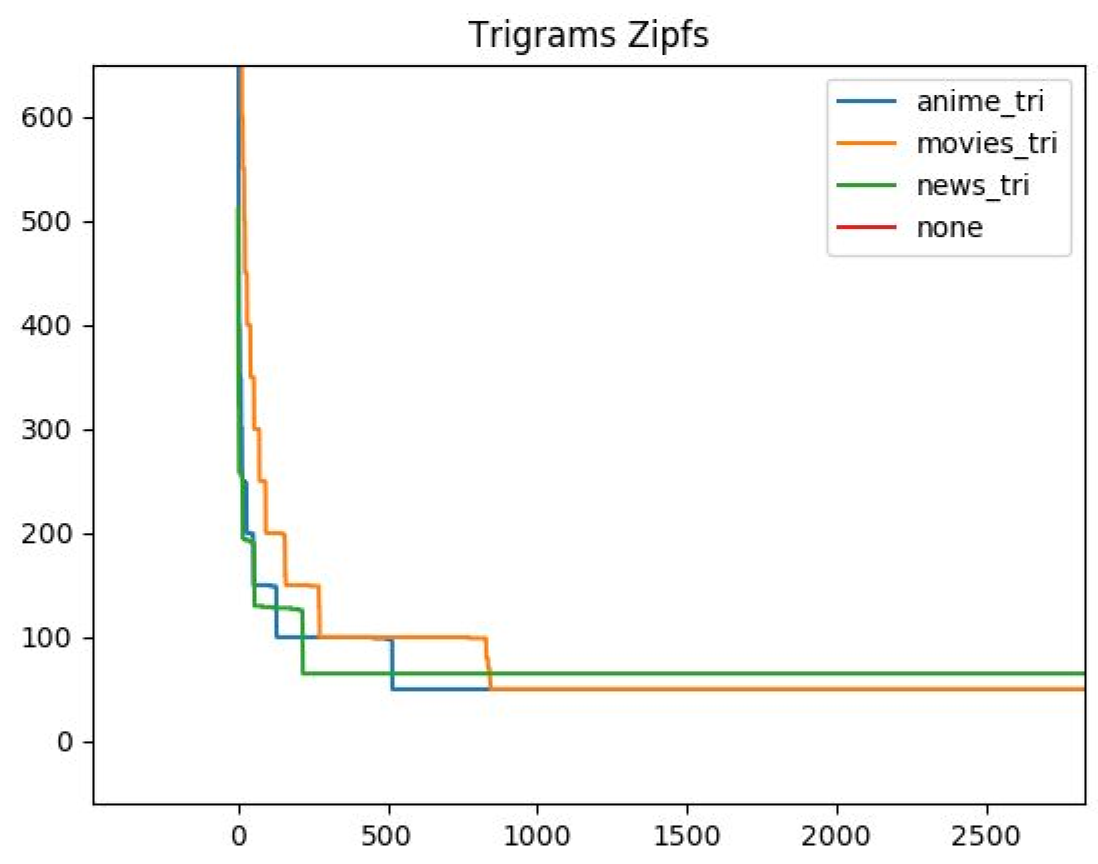


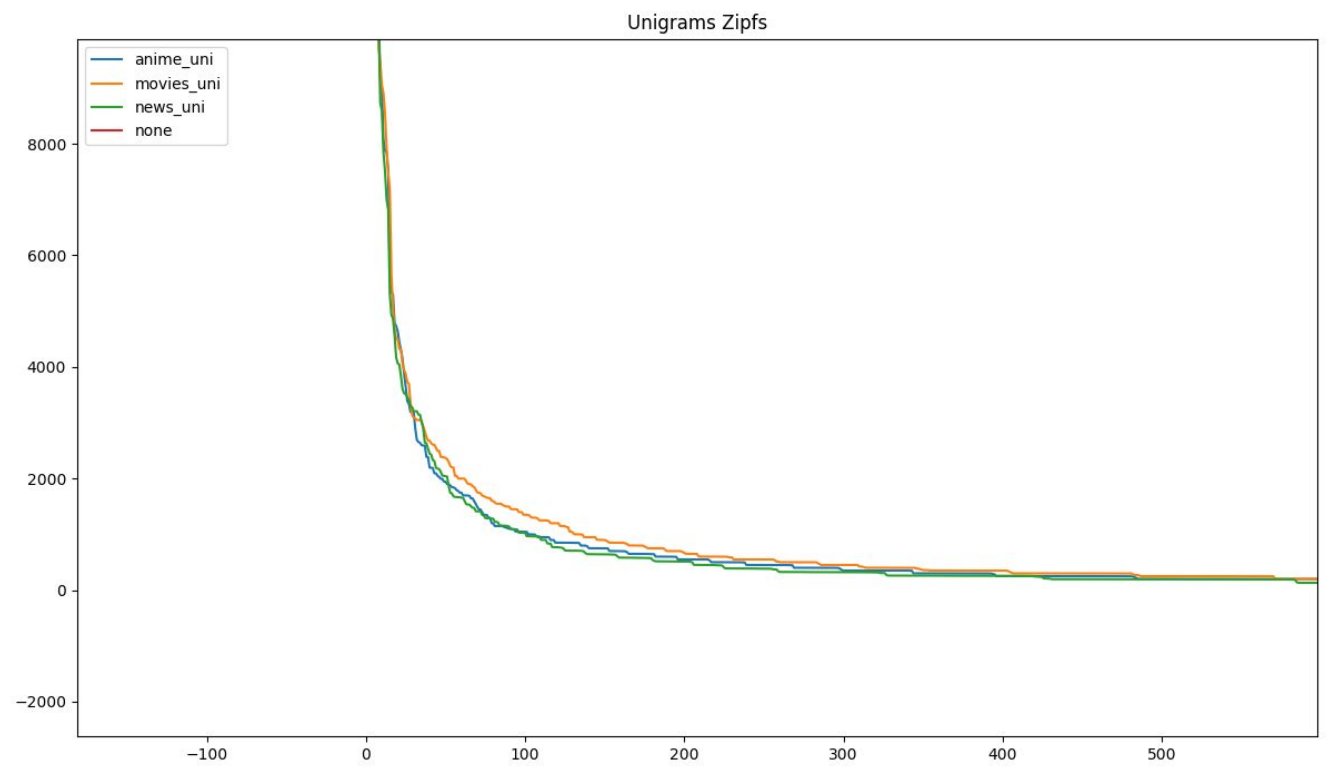




To Clearly view the meeting point I have Zoomed the pictures leading to pictures as below




Trigrams Zipfs
anime_tri
movies_tri
news_tri
none
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500


Unigrams Zipfs
anime_uni
movies_uni
news_uni
none
8000
6000
4000
2000
0
−2000
−100
0
100
200
300
400
500