**Good Turing Discounting**

**Assumed Dependencies:**

**Libraries:** numpy, pandas, runpy, nltk, os, sys, itertools, operator

**Python version:** Python 3.7.3

**How to run:**

**Command : This program runs Program 3 which implements Good Turing Discounting**

|  |
| --- |
| python3 homework2\_AnanyaBanerjee.py good\_turing\_discounting |

*The probability for given sentence is printed on console output.*

**Program 3:**

***File*:** good\_turing\_discounting.py

***Running this file generates:***

*File 1:**"* bigrams\_good\_turing\_discounting\_smoothing.csv*” contains the bigrams and their corresponding probabilities*

*File 2:* unigrams\_good\_turing\_discounting\_smoothing.csv*" contains the unigrams and their corresponding probabilities*

*File 3: "* unigrams\_buckets\_good\_turing\_discounting\_smoothing.csv *" contains the unigrams and their corresponding counts or buckets in given corpus*

*File 4: "* bigrams\_buckets\_good\_turing\_discounting\_smoothing.csv *" contains the bigrams and their corresponding counts or buckets in given corpus*

﻿ You can open any of these files to have a look at the results.

**Good Turing Smoothing:**

N= 66517

﻿*bigram : ['the', 'standard']* 🡺

|  |
| --- |
| Count(['the', 'standard'])=3  N4: len(bucket with val 4) = 1363  N3: len(bucket with val 3) = 1813  Count\_star = [ (Count(['the', 'standard'])+1 ]\*[N4/N3]  = [ (3+1) \* (1363/1813) ]  = [ 4 / 0.75]  = 5.3  Prob1 = Count\_star / N  = 5.3/ 66517  = 0.00008 (approx.) |

*bigram : ['standard', 'turbo']* 🡺

|  |
| --- |
| Count*(['standard', 'turbo']* )=2  N2: len(bucket with val 2) = 6376  N3: len(bucket with val 3) = 1813  Count\_star = [(Count*(['standard', 'turbo']* )+ 1 ]\*[N3/N2]  = [ (2+1) \* (1813/6376) ]  = [ 3 / 0.28 ]  =. 10.71  Prob2 = Count\_star / N  = 10.71/ 66517  = 0.0001(approx.) |

*bigram : ['turbo', 'engine'] 🡺*

|  |
| --- |
| Count([*'turbo', 'engine'*])=0  Bigram not in corpus!!  N1: len(bucket with val 1) = 16520  Prob3 = N1/ N  = 16520/66517  = 0.24 (approx.) |

bigram : ['engine', 'is'] 🡺

|  |
| --- |
| Count(['engine', 'is'])=0  Bigram not in corpus!!  N1: len(bucket with val 1) = 16520  Prob4 = N1/ N  = 16520/66517  = 0.24 (approx.) |

bigram : ['is', 'hard'] 🡺

|  |
| --- |
| Count(['is', 'hard'])=0  Bigram not in corpus!!  N1: len(bucket with val 1) = 16520  Prob5 = N1/ N  = 16520/66517  = 0.24 (approx.) |

bigram : ['hard', 'to'] 🡺

|  |
| --- |
| Count(['hard', 'to'])=3  N4: len(bucket with val 4) = 1363  N3: len(bucket with val 3) = 1813  Count\_star = [(Count*(['hard', 'to']* )+ 1 ]\*[N4/N3]  = [ (3+1) \* (1363/1813) ]  = [ 4 / 0.75]  = 5.3  Prob6 = Count\_star / N  = 5.3/ 66517  = 0.00008 (approx.) |

bigram : ['to', 'work'] 🡺

|  |
| --- |
| Count(['to', 'work'])=7  N7: len(bucket with val 7) = 205  N8: len(bucket with val 8) = 184  Count\_star = [(Count*(['to, 'work']* )+ 1 ]\*[N8/N7]  = [ (7+1) \* (184/205) ]  = [ 8 / 0.89 ]  = 8.98  Prob7 = Count\_star/ N  = 8.98/ 66517  = 0.0001 (approx.) |

Prob= prob1\*prob2\*prob3\*prob4\*prob5\*prob6\*prob7

Prob= 0.00008 \*0.0001\*0.24 \*0.24 \*0.24 \*0.00008 \*0.0001

= 8.847\* 10 ^(-19)