**No Smoothing**

**Assumed Dependencies:**

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

**Python version:** Python 3.7.3

**How to run:**

**Command 1: This program runs Program 1 which implements No Smoothing**

|  |
| --- |
| python3 homework2\_AnanyaBanerjee.py no\_smoothing |

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

**Program 1:**

***File*:** no\_smoothing.py

***Running this file generates:***

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

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

*File 3: "counts\_unigram\_no\_smoothing.csv" contains the unigrams and their corresponding counts in given corpus*

*File 4: "counts\_bigram\_no\_smoothing.csv" contains the bigrams and their corresponding counts in given corpus*

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

My calculations of the given sentence **“*The standard Turbo engine is hard to work* ”** is as shown below**:**

**Bigrams:**

﻿['the', 'standard']

['standard', 'turbo']

['turbo', 'engine']

['engine', 'is']

['is', 'hard']

['hard', 'to']

['to', 'work']

**No Smoothing:**

*bigram : ['the', 'standard']* 🡺

|  |
| --- |
| Count(['the', 'standard'])=3  Count([‘the’])=3676  Prob1 = Count(['the', 'standard']) / Count([‘the’])  = 3/ (3676)  = 0.0008 (approx.) |

*bigram : ['standard', 'turbo']* 🡺

|  |
| --- |
| Count*(['standard', 'turbo']* )=2  Count([‘standard’])=10  Prob2 = Count*(['standard', 'turbo']* ) / Count([‘standard’])  = (2)/ (10)  = 0.2 (approx.) |

*bigram : ['turbo', 'engine'] 🡺*

|  |
| --- |
| Count([*'turbo', 'engine'*])=0  Count([‘turbo’])=2  Prob3 = Count([*'turbo', 'engine'*]) / Count([‘turbo’])  = (0)/ (2)  = 0 |

bigram : ['engine', 'is'] 🡺

|  |
| --- |
| Count(['engine', 'is'])=0  Count([‘engine’])=17  Prob4 = Count(['engine', 'is']) / Count([‘engine’])  = (0)/ (17)  = 0 |

bigram : ['is', 'hard'] 🡺

|  |
| --- |
| Count(['is', 'hard'])=0  Count([‘is’])=447  Prob5 = Count(['is', 'hard']) / Count([‘is’])  = (0)/ (447)  = 0 |

bigram : ['hard', 'to'] 🡺

|  |
| --- |
| Count(['hard', 'to'])=3  Count([‘hard’])= 4  Prob6 = Count(['hard', 'to']) / Count([‘hard’])  = (3)/ (4)  = 0.75 |

bigram : ['to', 'work'] 🡺

|  |
| --- |
| Count(['to', 'work'])=7  Count([‘to’])= 1551  Prob7 = Count(['to', 'work']) / Count([‘to’])  = (7)/ (1551)  = 0.0045 (approx.) |

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

Prob= 0.2\*0.0008 \*0\*0\*0\*0.75\*0.0045

=0

**Add One Smoothing:**

V=28268

﻿*bigram : ['the', 'standard']* 🡺

|  |
| --- |
| Count(['the', 'standard'])=3  Count([‘the’])=3676  Prob1 = Count(['the', 'standard']) + 1 / Count([‘the’]) + V  = 4/ (3676+28268)  = 4/ 31944  = 0.0001 (approx.) |

*bigram : ['standard', 'turbo']* 🡺

|  |
| --- |
| Count*(['standard', 'turbo']* )=2  Count([‘standard’])=10  Prob2 = Count*(['standard', 'turbo']* )+ 1 / Count([‘standard’]) + V  = (2+1)/ (10+28268)  = 3/ 28278  = 0.0001 (approx.) |

*bigram : ['turbo', 'engine'] 🡺*

|  |
| --- |
| Count([*'turbo', 'engine'*])=0  Count([‘turbo’])=2  Prob3 = Count([*'turbo', 'engine'*]) + 1 / Count([‘turbo’]) + V  = (0+1)/ (2+28268)  = 1/ 28270  = 0.00003 (approx.) |

bigram : ['engine', 'is'] 🡺

|  |
| --- |
| Count(['engine', 'is'])=0  Count([‘engine’])=17  Prob4 = Count(['engine', 'is']) + 1 / Count([‘engine’]) + V  = (0+1)/ (17+28268)  = 1/ 28285  = 0.00004 (approx.) |

bigram : ['is', 'hard'] 🡺

|  |
| --- |
| Count(['is', 'hard'])=0  Count([‘is’])=447  Prob5 = Count(['is', 'hard']) + 1 / Count([‘is’]) + V  = (0+1)/ (447+28268)  = 1/ 28715  = 0.00003(approx.) |

bigram : ['hard', 'to'] 🡺

|  |
| --- |
| Count(['hard', 'to'])=3  Count([‘hard’])= 4  Prob6 = Count(['hard', 'to']) + 1 / Count([‘hard’]) + V  = (3+1)/ (4+28268)  = 4/ 28272  = 0.0001 (approx.) |

bigram : ['to', 'work'] 🡺

|  |
| --- |
| Count(['to', 'work'])=7  Count([‘to’])= 1551  Prob7 = Count(['to', 'work']) + 1 / Count([‘to’]) + V  = (7+1)/ (1551+28268)  = 8/ 29819  = 0.0002(approx.) |

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

Prob=0.0001\*0.0001\*0.00003\*0.00004\*0.00003\*0.0001\*0.0002

= 7.2\* 10^(-30)