This files will be used as a database of words

|  |  |
| --- | --- |
| [count\_1w.txt](http://norvig.com/ngrams/count_1w.txt) | The 1/3 million most frequent words, all lowercase, with counts. |
| [count\_2w.txt](http://norvig.com/ngrams/count_2w.txt) | The 1/4 million most frequent two-word (lowercase) bigrams, with counts. |

**Q1:** You have entered text:

shep

Find minimum edit distance (by Levenshtein) for all candidates (from count\_1w.txt, you can take 3 of them) that starts with she

**Q2:** (Implement code) Find language model probability of all candidates in distance 3 from the previous example.

For minimum edit distance, you can use any ready implementations. E.g.

In Java (<https://www.programcreek.com/2013/12/edit-distance-in-java/> )

**Q3:** (Implement code) Find language model probability for any entered sentence. Use count\_1w.txt and count\_2w.txt

*"I regret and condemn the decision of the United States to continue construction of a wall that, for years, has divided us instead of uniting us."*

*“Sea do go citizen going Thunder”*

The result shouldn't be zero.

Use laplace smoothing

