Word Bigram Counts — X38851_en

A *word bigram* is a combination of two words appearing consecutively in a text. For instance the text *"tea for you and tea for me"* contains 5 different bigrams: *tea+for* (occurring twice), *for+you*, *you+and*, *and+tea*, *for+me*.
Write a program that reads a text given as input, counts the bigrams it contains, and produces a list with the total count of bigrams starting with each word, and the relative frequency of the second word. The list must contain only those words that happen more than once *as first word* in a bigram.
Note that the last word in the text is not the *first word* in any bigram, so it is not counted in the number of bigrams starting with that word.
For instance, in the sentence *"tea for you and tea for me and for him also tea"*, we obtain the following counts:

- Word *and* happens 2 times as a bigram first word. Once (50%) followed by *tea* and once (50%) followed by *for*.
- Word *for* happens 3 times as a bigram first word. Once (33%) followed by *you*, once (33%) followed by *me*, and once (33%) followed by *him*.
- Word *tea* happens 2 times as a bigram first word. Both times (i.e. 100%) is followed by *for*. Word *tea* also happens a third time (last word in the text) but since that occurrence is not first word of any bigram, it is not counted as such.
- Words *you*, *me*, *also*, and *him* happen only once as bigram first word, so they are not included in the final list.

So, the expected output would be:

```
and 2 : for 0.5 tea 0.5
for 3 : him 0.333 me 0.333 you 0.333
tea 2 : for 1.0
```

Input

The input is a text. It may consist of several lines.

Output

The output is a list where for each word appearing more than once, the number of occurrences is provided, followed by the words occurring right after, along with their relative frequencies.
Relative frequencies are rounded to 3 decimal places. Use `round(x,3)` to round a float value `x` to 3 decimals.
The word list is ordered alphabetically.
The list of second words seen after each word is also ordered alphabetically.
Follow the output format shown in the examples.

Sample input 1

```
tea for you and tea for me and for him also
```

Sample output 1

```
and 2 : for 0.5 tea 0.5
for 3 : him 0.333 me 0.333 you 0.333
tea 2 : for 1.0
```

Sample input 2

```
how much wood would a woodchuck chuck
if a woodchuck could chuck wood
he would chuck he would as much as he could
and chuck as much wood as a woodchuck
would if a woodchuck could chuck wood
```

Sample output 2

```
a 4 : woodchuck 1.0
as 4 : a 0.25 he 0.25 much 0.5
chuck 5 : as 0.2 he 0.2 if 0.2 wood 0.4
could 3 : and 0.333 chuck 0.667
he 3 : could 0.333 would 0.667
if 2 : a 1.0
much 3 : as 0.333 wood 0.667
wood 3 : as 0.333 he 0.333 would 0.333
woodchuck 4 : chuck 0.25 could 0.5 would 0.25
would 4 : a 0.25 as 0.25 chuck 0.25 if 0.25
```

Sample input 3

```
Write a program that reads a text given as input
counts the bigrams it contains
and produces a list with the total count of
bigrams starting with each word and the
relative frequency of the second word
```

Sample output 3

```
a 3 : list 0.333 program 0.333 text 0.333
and 2 : produces 0.5 the 0.5
bigrams 2 : it 0.5 starting 0.5
of 2 : bigrams 0.5 the 0.5
the 4 : bigrams 0.25 relative 0.25 second 0.25 total 0
with 2 : each 0.5 the 0.5
```

Problem information

Author :
Generation : 2024-07-11 11:31:34

© *Jutge.org*, 2006–2024.
https://jutge.org