Handout 6: Collocations

Bigrams and generators

1. Three unigram distributions

a. Consider:

```
>>> list(nltk.bigrams([’a’, ’b’, ’c’]))
[(’a’, ’b’), (’b’, ’c’)]
```

b. Unigrams in corpus: `a` 1, `b` 1, `c` 1.

c. As left member of bigram: `a` 1, `b` 1.

d. As right member of bigram: `b` 1, `c` 1.

e. The fix: add the wrap-around bigram `(’c’, ’a’)`.

2. Creating a generator: **yield**

a. Example:

```
def lengths (strings):
    for s in strings:
        yield len(s)
```

b. Using it:

```
>>> lengths([’my’, ’name’, ’is’, ’Ishmael’])
<generator object lengths at 0x100676c60>
>>> for n in lengths([’my’, ’name’, ’is’, ’Ishmael’]):
...     print(’n=’, n)
...
n= 2
n= 4
n= 2
n= 7
>>> list(lengths[’my’, ’name’, ’is’, ’Ishmael’])
[2, 4, 2, 7]
```

3. Exercise.

Redefine the function `bigrams`. It should take a list or text as input, and it should return a generator containing the bigrams of the text, including the wrap-around bigram.

```
>>> list(bigrams([’a’, ’b’, ’c’]))
[(’c’, ’a’), (’a’, ’b’), (’b’, ’c’)]
```

a. What are the unigram distributions now?

Conditional probability

4. Our little corpus: `tokens = list('abbdabdbbd')`

	a	b	d	
a	–	2/10	–	2/10
b	–	2/10	3/10	5/10
d	2/10	1/10	–	3/10
	2/10	5/10	3/10	

5. Consider bigram $(\mathtt{b}, \mathtt{d})$

a. $p(\mathtt{b}, \mathtt{d}) = 3/10$

b. In two steps: pick a position in the corpus at random. $p(\mathtt{b}) = 1/2$

c. **Conditional distribution** over bigram right, for bigrams beginning with `b`:

a	b	d
–	2/5	3/5

d. Second step: choose a random right member: $p(\mathtt{d}|\mathtt{b}) = 3/5$

e. Together: $p(\mathtt{b}, \mathtt{d}) = p(\mathtt{b})\, p(\mathtt{d}|\mathtt{b}) = (1/2)(3/5) = 3/10$

6. Another example

a. In *Moby Dick,* white whale occurs 31 times:

```
>>> bd = FreqDist(bigrams(text1))
>>> bd['white', 'whale']
31
```

b. Using circular bigrams, same number of unigrams and bigrams:

```
>>> fd = FreqDist(text1)
>>> fd.N()
260819
>>> bd.N()
260819
```

c. The word white occurs 191 times. $p(\mathit{white}) = 191/N$

d. Of those 191, *whale* comes next 31 times: $p(\mathit{whale}|\mathit{white}) = 31/191$

e. $p(\mathit{white}, \mathit{whale}) = p(\mathit{white})\, p(\mathit{whale}|\mathit{white}) = \frac{191}{N} \cdot \frac{31}{191} = 31/N$

7. Definition of conditional probability

a. Counts and probabilities: $p(x) = c(x)/N$

b. $c(\mathit{white}) = 191$, $c(\mathit{white}, \mathit{whale}) = 31$

c. Condition probability is 31/191:

$$p(y|x) = \frac{c(x,y)}{c(x)} = \frac{c(x,y)/N}{c(x)/N} = \frac{p(x,y)}{p(x)}$$

8. Summary

- **a.** The **chain rule**: $p(x)\,p(y|x) = p(x,y)$
- **b.** The **marginal probability**: $p(x)$
- **c.** The **conditional probability**: $p(y|x) \equiv p(x,y)/p(x)$
- **d.** The **joint probability**: $p(x,y)$

9. Exercises. What are these probabilities?

- **a.** When rolling a single die, the probability of an even number given that we roll a high number (≥ 4).
- **b.** When rolling a pair of dice, the probability of rolling "boxcars" (two sixes).
- **c.** The probability that we have rolled boxcars, if the number on the first die is a six.

10. Conditional frequency distributions

a. Our little example:

```
>>> from nltk import ConditionalFreqDist
>>> cfd = ConditionalFreqDist(bigrams('abbdabdbbd'))
>>> cfd.tabulate()
     a   b   d
a    0   2   0
b    0   2   3
d    2   1   0
```

b. Row labels are **conditions**.

```
>>> cfd.conditions()
['a', 'b', 'd']
```

c. Rows are FreqDists.

```
>>> cfd['b']
FreqDist({'d': 3, 'b': 2})
```

d. Rows represent **conditional probabilities**. $p(\mathtt{d}|\mathtt{b}) = 3/5$

```
>>> cfd['b'].freq('d')
0.6
```

e. Contrast with joint probability $p(\mathtt{b},\mathtt{d})$

```
>>> bd = FreqDist(bigrams('abbdabdbbd'))
>>> bd.freq(('b', 'd'))
0.3
```

11. Exercise. How do we get the most-likely item following b?

(In)dependence

12. Conditional dependence

- **a.** Is `d` more likely following `b`? $\quad p(\texttt{d}|\texttt{b}) > p(\texttt{d})$?

```
>>> fd.freq('d')
0.3
>>> cfd['b'].freq('d')
0.6
```

- **b.** When rolling a single die, is an even number more likely, knowing that you rolled high?
- **c.** When rolling two dice, is a six more likely on the second die, knowing that you rolled six with the first die?

13. **Expected** count

- **a.** If x and y are **independent**, then $p(y|x) = p(y)$
- **b.** Joint probability

$$p = p(x)\, p(y|x)$$

- **c.** Expected joint probability, assuming independence:

$$\hat{p} = p(x)\, p(y)$$

- **d.** Expected count, assuming independence:

$$\hat{c}(x, y) = N\hat{p}(x, y)$$

- **e.** What is the expected count of $(\texttt{b}, \texttt{d})$, assuming independence?

```
>>> fd.N() * fd.freq('b') * fd.freq('d')
1.5
```

- **f.** Actual count is 3: twice the expected count

14. The **dependence ratio**: ratio of actual count to expected count

- **a.** Same as ratio of $p(y|x)$ to $p(y)$

$$r = \frac{c(x, y)}{\hat{c}(x, y)} = \frac{Np(x, y)}{N\hat{p}(x, y)} = \frac{p(x)\, p(y|x)}{p(x)\, p(y)} = \frac{p(y|x)}{p(y)}$$

- **b.** Example: $p(\texttt{d}|\texttt{b})/p(\texttt{d}) = \frac{3}{5}/\frac{3}{10} = 2$

15. Exercises.

- **a.** What is the dependence ratio for $(\texttt{d}, \texttt{a})$? More or less than $(\texttt{b}, \texttt{d})$?
- **b.** What is the dependence ratio between rolling an even number and rolling a high number?

16. A **collocation** is a word pair with a high degree of dependence.

a. Is white whale a collocation? Expected count:

```
>>> fd = FreqDist(text1)
>>> fd.N() * fd.freq('white') * fd.freq('whale')
0.6634716029123645
```

b. What is the actual count?

```
>>> cfd = ConditionalFreqDist(bigrams(text1))
>>> cfd['white']['whale']
31
```

c. It occurs $\boxed{?}$ times as often as we expect

```
>>> 31 / 0.6634716029123645
46.723928897518576
```

d. Computing from $p(y|x)$ and $p(y)$:

```
>>> cfd['white'].freq('whale')
0.16230366492146597
>>> fd.freq('whale')
0.003473673313677301
>>> cfd['white'].freq('whale') / fd.freq('whale')
46.723928897518576
```

17. Mutual information

a. (Pointwise) mutual information is the log of the degree of dependence

```
>>> from math import log10
>>> def pmi (x, y, fd, cfd):
...     return log10(cfd[x].freq(y) / fd.freq(y))
...
>>> pmi('white', 'whale', fd, cfd)
1.6695393543691355
>>> 10 ** pmi('white', 'whale', fd, cfd)
46.72392889751858
```

b. Pmi = 1 means $\boxed{?}$ times as likely

c. Pmi = 2 means $\boxed{?}$ times as likely

d. Pmi = $\boxed{?}$ means equally likely

e. Pmi = 0.5 means 3.2 ($= \sqrt{10}$) times as likely

f. Pmi = 1.5 means $\boxed{?}$ times as likely

g. What does a negative pmi mean?

18. Differing collocation strengths:

```
>>> pmi('the', 'whale', fd, cfd)
0.8888516296076806
>>> pmi('sperm', 'whale', fd, cfd)
2.1825403779780537
```

19. Symmetry and assymetry

a. What is the dependence ratio between rolling a high number and rolling an even number?

b. Is the dependence ratio symmetric in general?

$$\frac{p(y|x)}{p(y)} = \frac{p(x,y)}{p(x)\,p(y)} = \frac{p(x|y)}{p(x)}$$

c. But be careful:

$$\frac{\Pr[R=y|L=x]}{\Pr[R=y]} = \frac{\Pr[L=x|R=y]}{\Pr[L=x]}$$

$$\frac{\Pr[R=y|L=x]}{\Pr[R=y]} \neq \frac{\Pr[L=y|R=x]}{\Pr[L=y]}$$

d. Example:

```
>>> pmi('whale', 'the', fd, cfd)
-1.3771447408873985
```