statistical_dependence

November 19, 2020

This sections tokenizes lowercased text with NLTK, computes unigram and bigram frequencies and computes the pmi for the whole dataset.

[1]:
```python
from nltk.tokenize import WordPunctTokenizer
from collections import Counter
import matplotlib.pyplot as plt
import math

def freq_corpus(corpus, threshold):
    with open(corpus, 'r') as f:
        data = WordPunctTokenizer().tokenize(f.read().lower())
        data = data[1:] # remove first unicode byte
    unigram_freq = Counter(data)
    bigrams = zip(data[:(len(data)-1)], data[1:])
    return (sum([v for k,v in unigram_freq.items() if v >= threshold]),␣
↪unigram_freq, Counter(bigrams))

def corpus_pmi(corpus_name, threshold=10):
    data_len, counter_uni, counter_bi = freq_corpus(f'../data/{corpus_name}.
↪txt', threshold)

    def pmi(w1, w2):
        if counter_uni[w1] < threshold or counter_uni[w2] < threshold:
            return None
        return math.log((counter_bi[(w1,w2)]*data_len)/
↪(counter_uni[w1]*counter_uni[w2]),2)

    pmis = [(x, pmi(x[0], x[1])) for x in counter_bi.keys()]
    pmis = [x for x in pmis if x[1] is not None]
    pmis.sort(key=lambda x: x[1], reverse=True)

    def md_table(pmis):
        print('w1|w2|pmi\n-|-|-')
        print('\n'.join([f'{x[0][0]}|{x[0][1]}|{x[1]:.2f}' for x in pmis])+'\n')

    print('Highest PMIs')
    md_table(pmis[:20])
```

```
    print('Lowest PMIs')
    md_table(pmis[(len(pmis)-20):])

corpus_pmi('King James Bible')
```

```
Highest PMIs
w1|w2|pmi
-|-|-
ill|favoured|14.64
judas|iscariot|14.43
curious|girdle|14.20
brook|kidron|14.19
poureth|contempt|14.12
measuring|reed|14.07
persecution|ariseth|13.99
divers|colours|13.97
mary|magdalene|13.89
overflowing|scourge|13.72
-|ward|13.70
wreathen|chains|13.56
fiery|furnace|13.54
sharp|sickle|13.54
committeth|adultery|13.52
earthen|vessel|13.51
perpetual|desolations|13.50
golden|spoon|13.46
bright|spot|13.44
tenth|deals|13.43

Lowest PMIs
w1|w2|pmi
-|-|-
of|will|-7.22
,|thee|-7.24
to|;|-7.27
into|,|-7.32
this|and|-7.33
,|me|-7.34
of|in|-7.35
of|,|-7.51
the|israel|-7.53
shall|of|-7.57
to|in|-7.59
will|and|-7.79
when|,|-7.81
with|,|-7.89
of|is|-8.08
```

```
the|said|-8.16
all|and|-8.35
of|he|-8.66
for|and|-9.02
of|to|-9.04
```

The output is the following. Bigrams, which occur only with each other have positive and the highest pointwise mutual information, because the two words occuring together is more probable than just the two words occuring independently. Because of this, pairs such as "ill" "favoured", "committeth" "adultery" or "http" "://" in Junglebook have such a high PMI. Mathematically, `p(w2|w1) > p(w1)` or `p(w1|w2) > p(w2)`. In contrast, words which occur in bigrams with lots of other words have negative PMI. It's negative, because the probability of these words occuring together is much lower than if it occured just by chance.

Since there are so many bigrams with especially so high PMIs, unigram models make a very strong and invalid assumption about the independence.

Highest PMIs

w1	w2	pmi
ill	favoured	14.64
judas	iscariot	14.43
curious	girdle	14.20
brook	kidron	14.19
poureth	contempt	14.12
measuring	reed	14.07
persecution	ariseth	13.99
divers	colours	13.97
mary	magdalene	13.89
overflowing	scourge	13.72
-	ward	13.70
wreathen	chains	13.56
fiery	furnace	13.54
sharp	sickle	13.54
committeth	adultery	13.52
earthen	vessel	13.51
perpetual	desolations	13.50
golden	spoon	13.46
bright	spot	13.44
tenth	deals	13.43

Lowest PMIs

w1	w2	pmi
of	will	-7.22
,	thee	-7.24
to	;	-7.27

w1	w2	pmi
into	,	-7.32
this	and	-7.33
,	me	-7.34
of	in	-7.35
of	,	-7.51
the	israel	-7.53
shall	of	-7.57
to	in	-7.59
will	and	-7.79
when	,	-7.81
with	,	-7.89
of	is	-8.08
the	said	-8.16
all	and	-8.35
of	he	-8.66
for	and	-9.02
of	to	-9.04