Collocation Extraction

SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI
EST. 1888
FACULTY
OF ECONOMICS
AND BUSINESS
ADMINISTRATION

Collocation extraction in NLP

- Extracting collocations from a text corpus might help us in **data exploration** and analysis.
- Collocations are simply words which **commonly occur together**[1].
- Collocations might be bigrams, trigrams etc.
- **Statistical measures** for finding collocations – frequency, *t* test, Pointwise Mutual Information (PMI), Chi-square and other.

[1]Athough this is probably the most frequently used definition of a "collocation", there are other valid definitions with a slightly different meaning. For a detailed discussion on the subject – check section 5.5, chapter 5, Manning, C., & Schutze, H. (1999). **Foundations of statistical natural language processing**. MIT press.

Pointwise Mutual Information (PMI)

- **PMI** is one useful approach for finding collocations in a text corpus.
- PMI - a **measure of association.** It compares the probability of two words occurring together to the probability if the two words are independent.
- **Implementation** in Python - NLTK :: Sample usage for collocations

Pointwise Mutual Information (1)

- Mathematical formulation:

$$PMI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, \text{where:}$$

w_1 – a given word part of the text corpus (word 1).
w_2 – a given word part of the text corpus (word 2).
$P(w_1)$ – individual probability of occurence of word 1.
$P(w_2)$ – individual probability of occurence of word 2.
$P(w_1, w_2)$ – joint probability of occurence of word 1 and word 2.

Pointwise Mutual Information (2)

- Calculation of PMI when two words are independent:

$$PMI(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} = \log \frac{P(w_1)P(w_2)}{P(w_1)P(w_2)} = \log 1 = 0$$

*In case of independence: $P(w_1, w_2) = P(w_1)P(w_2)$

In this case we observe perfect independence between the two words and PMI=0.

Pointwise Mutual Information (3)

- Calculation of PMI when two words **always occur together** in a given text corpus:

$$PMI(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} = \log \frac{P(w_1)}{P(w_1)P(w_2)} = \log \frac{1}{P(w_2)}$$

In this case we observe perfect dependence between the two words and reach the maximum value for $PMI(w_1, w_2)$ (in a given corpus).

Pointwise Mutual Information (4)

- When $\boldsymbol{PMI = 0} \Rightarrow$ perfect independence $\Rightarrow$ the two words are independent and do not form an interesting and meaningful collocation.
- When $\boldsymbol{PMI > 0} \Rightarrow$ the two words appear more frequently than we would expect under an independence assumption. **Might** indicate interesting collocations. We aim at finding collocations with **higher PMI**!
- PMI might also have **negative values**. This means that the two words appear less frequently than we would expect by chance. In practice, negative values of PMI are often **set to zero** due to unreliability and interpretability issues (**positive pointwise mutual information-PPMI**).

An example (1)

- Assume that our sample contains the following three movie reviews (already tokenized):

[“the”, “movie”, “is”, “ok”]
[“the”, “beginning”, “was”, “boring”, “very”, “boring”]
[“the”, “movie”, “was”, “very”, “good”]

- Number of tokens in the sample (with repetition) = 15

An example (2)

- Calculate $PMI(\text{very}, \text{boring})$:

$$p(very) = \frac{\#\text{ of times the word occurs}}{\#\ all\ tokens} = \frac{2}{15}$$

$$p(\text{boring}) = \frac{\#\text{ of times the word occurs}}{\#\ all\ tokens} = \frac{2}{15}$$

$$p(\text{very}, \text{boring}) = \frac{\#\text{ of times the two words occur together}}{\#\ all\ tokens} = \frac{1}{15}$$

$$PMI = \log_2 \frac{p(\text{very}, \text{boring})}{p(very)\ \times\ p(\text{boring})} = \log_2 \frac{\frac{1}{15}}{\frac{2}{15} \times \frac{2}{15}} = \log_2 \frac{15}{4} = \log_2 3{,}75 = 1.9068905956$$

$\Longrightarrow$ Please, check out the solution in Python – "Calculate PMI in Python.py"

PMI – more practical information

- PMI is not a very accurate measure when applied for **low-frequency events** since it depends on the frequency of individual words.
- "**Bigrams composed of low-frequency words will receive a higher score than bigrams composed of high-frequency words**." - Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing.
- It is often useful to filter out collocations that occur **too infrequently** (for example, ignore collocations that occur less than 3 times). The choice of a cutoff value depends on sample characteristics.

Thanks!

Any questions?

You can find me at:

g.hristova@feb.uni-sofia.bg