
# Information Retrieval Exercises

## Assignment 5:

## **Finding frequent word co-occurrences**

Mario Sänger (saengema@informatik.hu-berlin.de)

# Collocations

- A collocation is a sequence of tokens that correspond to some conventional way of saying things [MS99]
  - Examples: strong tea, crystal clear, whisper softly
- One way to find collocations is to search for co-occurrences that appear more often than would be expected by chance
  - Two terms co-occur if they appear together in a context (e.g. a sentence or a window of *n* words)

- Assignment 5: Find all over-represented co-occurrences among a reduced set of words in the IMDB corpus

# Finding Frequent Co-occurrences

- Parse the title and plot descriptions from the plot.list
- Use pre-processing from assignment 2
  - Tokenization at spaces, line breaks, dots, commas, colons, question marks and exclamation marks ([ .,:!?])
  - Lower-case all tokens

- Since we don't detect sentence borders, we only consider subsequent occurrences of the two tokens as co-occurrence!

# Finding Frequent Co-occurrences

- Disregard all tokens that are stop words based on the "Default English stopwords list" from ranks.nl
  - See http://www.ranks.nl/stopwords
  - Don't remove stop words from the corpus, only disregard co-occurrences containing them
- Disregard infrequent tokens with less than 1000 total occurrences in the corpus

- Again, both tokens have to subsequent to one another (in the corpus) and neither may be a stop word or appear less than 1000 times

# Finding Frequent Co-occurrences

- Sort co-occurrences (descending) by the following score:

$$s(t,t') = \frac{2 \cdot F(t,t')}{F(t)+F(t')}$$

  - *F(t)* is the frequency of token *t* in the corpus
  - *F(t,t')* is the frequency of bigram *t,t'* in the corpus
  - A bigram is a sequence of two adjacent tokens

- Report the top 1000 co-occurrences along with their score

# Example

- Stop words:
  - about, against, and, be, me, the, this, was, and, with
- Sentences
  - the crystal clear water rose against the coast, merging with the sky
  - let me be crystal clear about this, Rose
  - the red sun rose and the sky turned clear
- Token and bigram frequencies
  - F(crystal)=2,  F(clear)=3, F(water)=1, F(rose)=3, F(sky) =2, ...
  - F(crystal,clear)=2, F(water,rose)=1, F(rose,sky)=0, ...

# Example

- Token and bigram frequencies
  - F(crystal)=2,  F(clear)=3, F(water)=1, F(rose)=3, F(sky) =2,...
  - F(crystal,clear)=2, F(water,rose)=1, F(rose,sky)=0,...

- Co-occurrence scores:

$$s(\text{crystal}, \text{clear}) = \frac{2 \cdot F(\text{crystal,clear})}{F(\text{crystal}) + F(\text{clear})} = \frac{2 \cdot 2}{2+3} = \frac{4}{5}$$

$$s(\text{water}, \text{rose}) = \frac{2 \cdot F(\text{water,rose})}{F(\text{water}) + F(\text{rose})} = \frac{2 \cdot 1}{1+3} = \frac{1}{2}$$

$$s(\text{rose}, \text{sky}) = \frac{2 \cdot F(\text{rose,sky})}{F(\text{rose}) + F(\text{sky})} = \frac{2 \cdot 0}{3+2} = 0$$

# Computation details

- Regard title and plot as well as plots from different authors as different texts

MV: "The Simpsons" (2018) {Springfield Splendor}

PL: The best

PL: episode.

BY: foo@example.com

PL: A rather dull episode.

**Potential bigrams:**

the simpsons

the best

best episode

a rather

rather dull

dull episode

- Note: Some of these bigrams will later be discarded due to containing a stop word or infrequent word!

# Submission

- No Java class skeleton given this time
  - You may reuse your code from the other assignments!
- Submit executable JAR CoOccurrencesFinder.jar
  - Syntax: java –jar CoOccurrencesFinder.jar *<plot-file>* *<output-file>*
- Write top 1000 co-occurrences sorted (desc) by score to *<output-file>*
  - Syntax: <token>\t<token>\t<score>\n

| | | |
|---|---|---|
| los | angeles | 0.8932607215793057 |
| hong | kong | 0.7493632195618951 |
| las | vegas | 0.7398075240594926 |
| u | s | 0.70640263377721 |
| united | states | 0.6942972495584153 |

# Submission

- **Group 1: Wednesday, 11.07., 23:59 (midnight)**
- **Group 2: Friday, 13.07., 23:59 (midnight)**

- Submit a ZIP archive named *ass5_<group-name>.zip*
  - Java source files of your solution
  - Compiled and executable CoOccurrencesFinder.jar

- Upload archive to the HU-BOX:
  - https://box.hu-berlin.de/u/d/0a3e0548ea7e4bd5b8d2/

# Presentation of the solutions

- The presentation of the solutions will be given on 16.07. resp. 18.07.
- You are be able to pick when and what you'd like to present (first-come-first-served):
  - Group 1 (Mo): https://dudle.inf.tu-dresden.de/ire_ass5_mo/
  - Group 2 (We): https://dudle.inf.tu-dresden.de/ire_ass5_we/
- Keep in mind that every group has to present at least once to pass the exercise!

# Competition

- Parse corpus and compute co-occurrences as fast as possible
- Use memory abundantly (you have up to 50 GB)

# Checklist

- Before submitting your results, make sure that you ...
  - ... named your jar CoOccurrencesFinder.jar
  - ... named your submitted archive according to your group name
  - ... included your source code in the submitted archive
  - ... tested your executable JAR on gruenau hosts by running
    java -jar CoOccurrencesFinder.jar plot.list output.txt
    (you might have to increase Java heap space, e.g. -Xmx6g)
  - ...made sure the output is syntactically correct

# Roadmap for the last weeks

- **09./11.07.2018**
  - Evaluation and presentation of assignment 4 solutions
  - Q/A for assignment 5
- **11./13.07.2018**
  - Submission deadline for assignment 5
- **16./18.07.2018**
  - Evaluation and presentation of assignment 5 solutions
  - Feedback, award & farewell ceremony