Information Retrieval

Assignment 5:
Finding Frequent Word Co-Occurrences

Patrick Schäfer (patrick.schaefer@hu-berlin.de)

Marc Bux (buxmarcn@informatik.hu-berlin.de)

Collocation

- two terms co-occur if they appear together in a sentence
- a collocation is a sequence of tokens that correspond to some conventional way of saying things [MS99]
  - examples: excruciating pain, crystal clear, whisper softly, cosmetic surgery
- one way to find collocations is to search for co-occurrences that appear more often than would be expected by chance

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

Assignment 5: Finding Frequent Co-Occurrences

- only parse the titles and plot descriptions from the plot.list
- tokenization (at " .,:!?") and lower-case-conversion as in assignment 2
- since we don't detect sentence borders, we only consider as co-occurrence subsequent occurrences of the two tokens
- disregard tokens that are stop words based on the "Default English stopwords list" from 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 occurrences
  - 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
- sort co-occurrences by descending 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

- sentences (“about”, “against”, “and”, “be”, “me”, “the”, “this”, “was”, and “with” are stop words):
  - *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:
  - $F(\text{crystal}) = 2, F(\text{clear}) = 3, F(\text{water}) = 1, F(\text{rose}) = 3, F(\text{sky}) = 2, \ldots$
  - $F(\text{crystal}, \text{clear}) = 2, F(\text{water}, \text{rose}) = 1, F(\text{rose}, \text{sky}) = 0, \ldots$
- scores:
  - $s(\text{crystal}, \text{clear}) = \frac{2 \cdot F(\text{crystal,clear})}{F(\text{crystal}) + F(\text{clear})} = \frac{2 \cdot 2}{2+3} = \frac{4}{5}$
  - $s(\text{water}, \text{rose}) = \frac{2 \cdot F(\text{water,rose})}{F(\text{water}) + F(\text{rose})} = \frac{2 \cdot 1}{1+3} = \frac{1}{2}$
  - $s(\text{rose}, \text{sky}) = \frac{2 \cdot F(\text{rose,sky})}{F(\text{rose}) + F(\text{sky})} = \frac{2 \cdot 0}{3+2} = 0$
  - …

Top-10 Output (plot.list changes, your output may differ!)

los angeles 0.8932607215793057

hong kong 0.7493632195618951

las vegas 0.7398075240594926

u s 0.70640263377721

united states 0.6942972495584153

hip hop 0.629205402290623

san francisco 0.6093043290975663

martial arts 0.4953350296861747

beverly hills 0.48834080717488787

award winning 0.3101041554815849

Deliverables

- by Thursday, 9.2., 23:59 (midnight)

- submission: archive (zip, tar.gz)
  - contains Java source files, any used libraries, and your executable (and ready-to-be-executed) Jar
  - file name (of submitted archive): your group name

- upload to https://hu.berlin/ue_ir_5
  - if this doesn't work, send via mail to buxmarcn@informatik.hu-berlin.de

Your Program

- no Java code frame given this time
  - you may reuse code from assignment 2 (parser?, positional index?)
- output of your Jar: top 1000 co-occurrences sorted (from high to low) by score, which is also printed
  - format / syntax: <token> <token> <score of bigram as double> (see slide 5)
- your Jar must be
  - named CoOccurrences.jar
  - executable from the command line by running
    java -jar CoOccurrences.jar plot.list
  - tried and tested on gruenau2 before submission

Presentation of Solutions

- presentations will be given on 13./14. 2.

- no dudle this time
- via Agnes, we will announce who will (yet have to) present
- remember, having presented at least once is a prerequisite for the exam admission

- one team will present their word filter (stop words, infrequent words)
- one team will present their bigram counter and odds scorer

Competition

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

- buffering / pre-computation of results is not allowed
  - we will delete files created by your Jar prior to the next run

Checklist

before submitting your results, make sure that you

1. named your jar CoOccurrences.jar
2. named your submitted archive according to your group name
3. included your source code in the submitted archive
4. tested your Jar on gruenau2 by running
   java -jar CoOccurrences.jar plot.list
   (you might have to increase Java heap space, e.g. -Xmx6g)
5. made sure the output is similar in content and identical in syntax to the output on slide 5

Next (Final) Steps

- 30./31. Jan (next week):
  - evaluation of assignment 4
  - presentation of solutions for assignment 4
  - Q/A session for assignment 5
- 6./7. Feb
  - Q/A session for assignment 5
  - attendance is optional
- 9. Feb:
  - submit assignment 5
- 13./14. Feb
  - evaluation of assignment 5
  - presentation of solutions for assignment 5
  - award & farewell ceremony