Assignment 3

Assignment 3

- **Documents All Pairs Similarity**

  ↳ find all doc. that are similar at least a threshold (es. 0.95)

  (usefull for ex. in duplicate search or spam removal)

- To Be Delivered:
  - Sequential implementation (Python-Numpy/Java/Scala)
  - Parallel implementation
    - MapReduce / Apache Spark (Python/Java/Scala)
  - Report discussing performance figures of the proposed parallel implementation
    - varying datasets (and samples), similarity thresholds (es. 0.5 or 0.95)
    - varying number of workers (→ CPU, thread, machine)
    - max 2 pages

(LOOK AT THE PREVIOUS REPORT COMMENTS)

Documents All Pairs Similarity

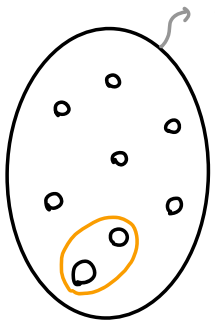
- A document is a vector *d* of *N* elements
  - *N* is the number of distinct words in the corpus (the lexicon)
  - d[i]: stores the frequency of the term *i* in document *d* (*tf(i)*).
    Then d is normalized (divided by its L_2 norm)
  - additionally you can use *tf-idf*:

$$tf\text{-}idf(i) = tf(i) \cdot \ln \frac{N_{docs}}{df(i)}$$

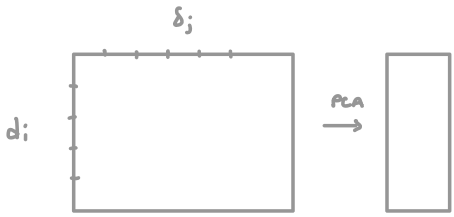
- There are many ways to measure similarity
  - Cosine:

$$s(a,b) = \sum_{i=1\ldots N} a[i] \cdot b[i]$$

DOCUMENT ALL PAIRS SIMILARITY SEARCH



SIMLARITY $\geq \tau$



What is a document?

Can be any object $\rightsquigarrow$ I need to find a repr.

For textual doc I can : d_i

$\hookrightarrow$ tf-idf $\rightsquigarrow$ usually $tf(t) \cdot \log\left(\frac{N_f}{|d \text{ containing } t|}\right)$

For movies dataset I have : m_i

Similarity can be computed using COSINE SIMILARITY that is basically a dot product $(d_i \cdot d_j)$

$\rightsquigarrow$ I want the EXACT similarity (but I can compare it with approximations)

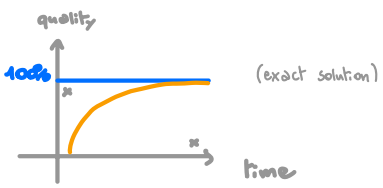
- CLUSTER docs and inside cluster do search (run alg. on clusters) (can be done on clusters in parallel)
- LSH
- PCA or JOHNSON - LINDENSTRAUD

$\downarrow$

if $\geq \tau$ doc it's append to result

$\frac{tf\text{-}idf}{\|d_i\|_2}$ $\rightsquigarrow$ to compute the score of a doc

I can plot the threadoff comparison (investigate the tradeoff error - speed)



using for ex. JACCARD sim. BETWEEN RESULTS (exact and approx.)

COMPUTING THE EXACT SOLUTION :

For each d_i :
  For each d_j :
    compute sim (d_i, d_j)

$\Rightarrow$ $O(n^2)$

THERE ARE SOME TECHNIQUES THAT SPEED UP THE COMPUTATION BUT NOT REDUCE COMPLEXITY

WE CAN ALSO SORT BY DOC LENGTH AND SKIP ALL THE DOC SHORTER THAN A TRESHOLD BECAUSE THE DOT PROD SURELY DON'T REACH THE VALUE (SKIP A LOT OF DOC IN ONE PASS WITH SORTING, RATHER THEN SKIPPING THEM ONE AT A TIME)

Sequential Algorithm

- Given the minimum required similarity *threshold*

  - SIM_DOCS = 0
  - For-each document d_1 in the corpus D :
    - For-each document d_2 in the corpus D :
      - if $d_1 != d_2$ and $s(d_1, d_2) >= threshold$:
        - SIM_DOCS += 1

- Note: usually you are interested to the similar document pairs, rather than to the number of similar document
- *Try your optimizations!!*

- Datasets:
  - https://github.com/beir-cellar/beir
  - https://grouplens.org/datasets/movielens/

Deadline and Evaluation

- Delivery before *May 26*:
  - *+1* point if positively evaluated, *+0.5* if sufficient, re-submit if insufficient
- Or, delivery at written exam
  - +1 point if positively evaluated, +0 if sufficient, exam not passed if insufficient
- Positive Evaluation means:
  - good report, good code, good analysis