Assignment 3

Assignment 3

- **Documents All Pairs Similarity**

- 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
    - varying number of workers
    - max 2 pages

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:

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

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