Most contributed articles
(extended version)

Big Data Computing Final Project

Mehrdad Hassanzadeh
1961575

September 2023

Introduction

Task: Graph Analysis + Information Retrieval System

We are looking for: given a query, retrieve the most relevant articles

Data: .json file containing article information such as title and list of references. Data can be found here.

Approaches:

- Degree Centrality
- PageRank
- TF-IDF + Cosine similarity
- Cosine similarity + PageRank -> ContentLink score

Components used:

- PySpark 3.4.1
- Python 3.10
- Google Colab Pro
- Google Drive

Data

Json

File type

≈ 12GB

File size

4,894,504

Articles

45,564,149

Citations

Data Pre-processing

01 Json

12GB file size

02 Pickle

Dictionary storing the relevant data
(id, title, references)
≈ 660 MB file size

03 Csv

- Filtering **top 500,000** articles

- **Article.csv** (id, title)
≈ 40 MB file size

- **Citations.csv** (citations)
≈ 120 MB file size
5,635,143

Graph Structure

Nodes

Articles

Edges

Citation relationship

a to b (a -> b)
if article a has cited article b

Unweighted

Directed

Approaches

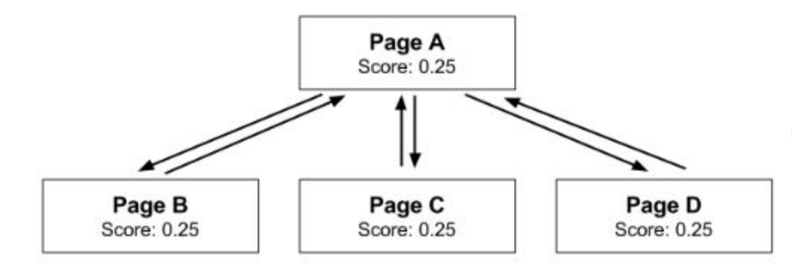
Degree centrality

- *Undirected graph:* defines the importance of a node v as the **number of the neighbors of node v**
- *Directed graph*: defines the importance of a node v based on the **number of the incoming and outgoing edges**
- In our case, the **higher the number of the incoming edges**, the **higher is the importance of the node.**

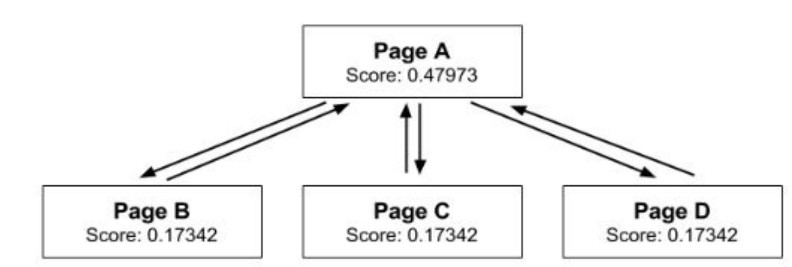
PageRank

- PageRank works by counting the **number** and **quality** of links to a page to determine a **rough estimate of how important the website** is. “Google”
- The PageRank algorithm outputs a **probability distribution** used to represent the **likelihood that a person randomly** clicking on links will arrive **at any particular page**.
- Initial probability (rank) = 1/N, N = # nodes
- The rank of a node would be updated at the end of each iteration based on the **rank of the preceding nodes** and the **probability of being randomly chosen**.
- **Stopping criteria:** when we reach a point of convergence

Example



100 iterations



- Setting the decay factor d = 0.85

- The **new rank of page A** would be updated to:
  d*(0.25/1) + d*(0.25/1) + d*(0.25/1) + (1 - d)/4 = 0.675

https://courses.cs.washington.edu/courses/cse373/17au/project3/project3-3.html

Results

Degree centrality results

InDegree

Distinctive Image Features from Scale-Invariant Keypoints

#Citations received: **10,123**

OutDegree

Deep Reinforcement Learning

#Citations given: **307**

In our work, the **higher a node's InDegree**, the **more important** that specific node is.

PageRank results

PageRank

Learnability and the Vapnik-Chervonenkis dimension

Rank: **0.00269**

Iterations: 10

Comparison

InDegree

Distinctive Image Features from Scale-Invariant Keypoints

Citations **10,123**

Rank: **0.0013**

PageRank

Learnability and the Vapnik-Chervonenkis dimension

Citations: **296**

Rank: **0.00269**

Search Engine

TF-IDF

- It evaluates the **importance** of a word (or term) within a document relative to a collection of documents (corpus).

$$\text{Tf\_Idf}(t, d, D) = \text{Tf}(t, d) * \text{Idf}(t, D)$$

- **Tf (t,d):** # occurences of a term t in a document d

$$\text{Tf}(t, d) = \log(1+\text{freq}(t, d))$$

- **Idf (t, D):** quantifies how common a term is among a set of documents D

$$\text{Idf}(t, D)=\log(|D| / \text{count}(d \in D : t \in d))$$

Text preprocessing

- We are going to focus on the terms in the title of the articles. In order to make our data suitable for our work we have to go through a text preprocessing:

    - **Tokenization**
    - **Lowercasing**
    - **Stopwords removal**
    - **Punctuations removal**
    - **Stemming**

Cosine similarity

- Given a query and a set of documents with their terms' TF-IDF, we define the cosine similarity between each query and a document as follows:

cosine similarity (q, d) = dot_product(q,d) / (||q||*||d||)

- The **higher the cosine similarity**, the **more relevant** the document is for the given query

Note: we have assumed that we have the TF-IDF of the terms in the query and the document in a vector.

Results

Cosine similarity results

Our query **“Big Data Computing”**

Retrieved articles with their cosine similarity :

- “Big Data – A State-of-the-Art” : **0.86**
- “Big data” : **0.86**
- “Big Data over Networks” : **0.81**
- “Content-Centric and Software-Defined Networking with Big Data” : **0.81**
- “Efficient computation of the well-founded semantics over big data” : **0.79**

ContentLink score

ContentLink score

We have defined a score that will combine the cosine similarity and the pageRank metrics.

Steps:

- **Filtering:** Filtering the articles that have <u>at least one</u> of the terms in the given query in their title.
- **Scaling:** Scaling the similarities and the pageRanks for these articles to be in [0, 1]

ContentLink score = alpha * cosine_similarity(q, d) + beta * pageRank(d)

alpha, beta in [0,1] are the rate of the contributions

Results

ContentLink score

- Our query **“Big Data Computing”**
- **alpha = 0.5, beta = 0.5**

Retrieved articles with their score :

- “Computer Processing of Line-Drawing Images” : **0.65**
- “Big data” : **0.50**
- “Big Data – A State-of-the-Art” : **0.50**
- “Big Data over Networks” : **0.47**
- “Content-Centric and Software-Defined Networking with Big Data” : **0.47**

ContentLink score

- Our query **“Big Data Computing”**
- **alpha = 0.6, beta = 0.4**

Retrieved articles with their score :

- “Big data” : **0.60**
- “Big Data – A State-of-the-Art” : **0.60**
- “Computer Processing of Line-Drawing Images” : **0.58**
- “Big Data over Networks” : **0.56**
- “Content-Centric and Software-Defined Networking with Big Data” : **0.56**

ContentLink score

- Our query **“Big Data Computing”**
- **alpha = 0.4, beta = 0.6**

Retrieved articles with their score :

- “Computer Processing of Line-Drawing Images” : **0.72**
- “Specification and implementation of resilient, atomic data types” : **0.49**
- “A Survey of Data Structures for Computer Graphics Systems” : **0.42**
- “Clustering categorical data: an approach based on dynamical systems” : **0.42**
- “Big data” : **0.40**

Thank you for your attention