ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Data Science Challenge

Margonis Phevos – Trantalidis Giannis

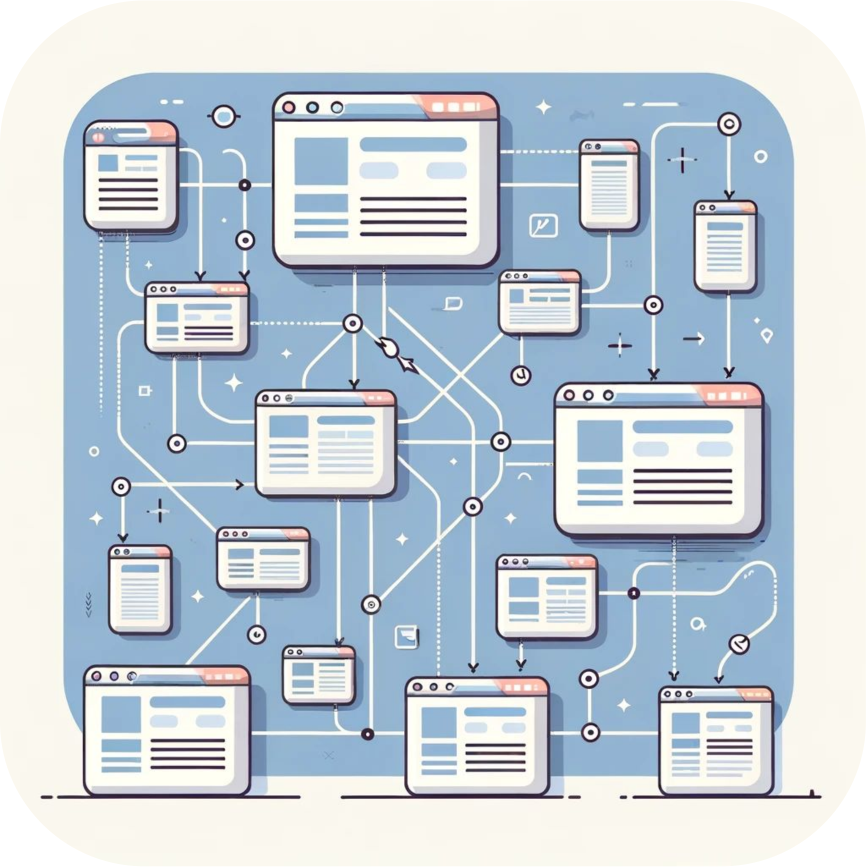
June 10, 2024

Contents

Introduction

Problem Definition

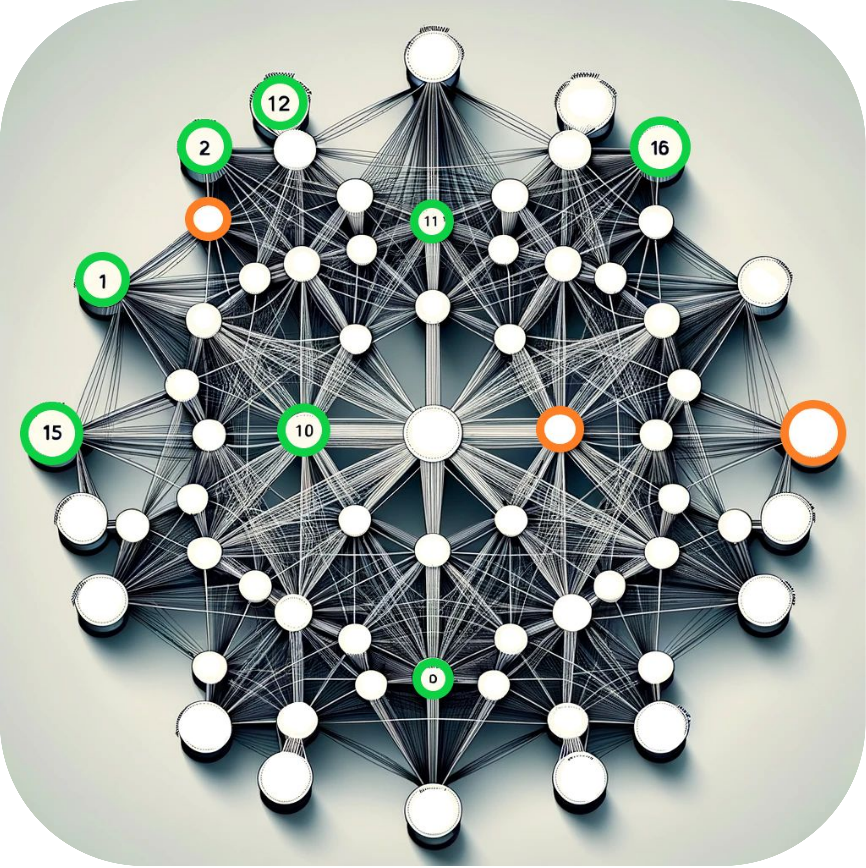
- Web pages
- Connected in a directed graph
- Have neighbors
- Have domain names
- Have subpage urls
- Have texts
- Single-label node (page) classification



Exploratory Analysis

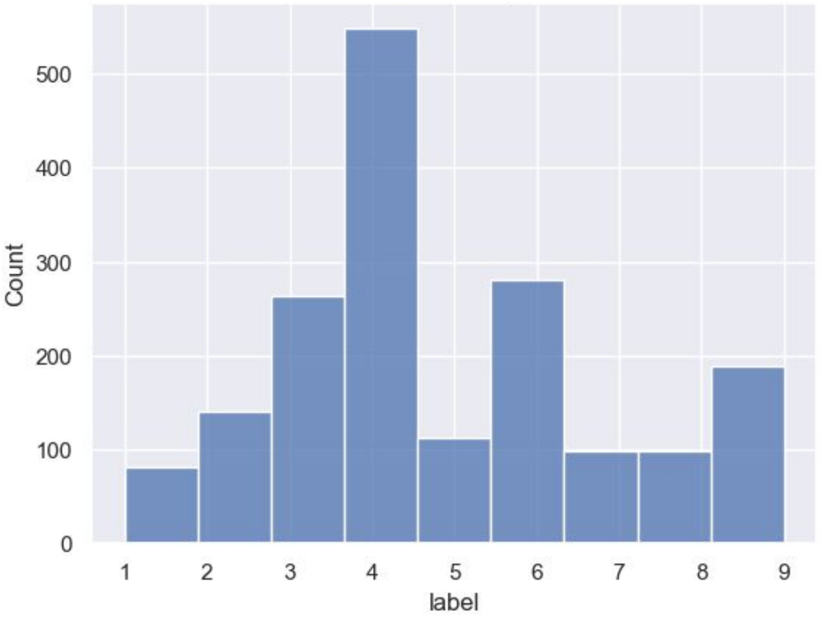
Exploratory Analysis

- Total nodes: > 65.000
- Labeled nodes: > 1.800
- Unlabeled test: < 700

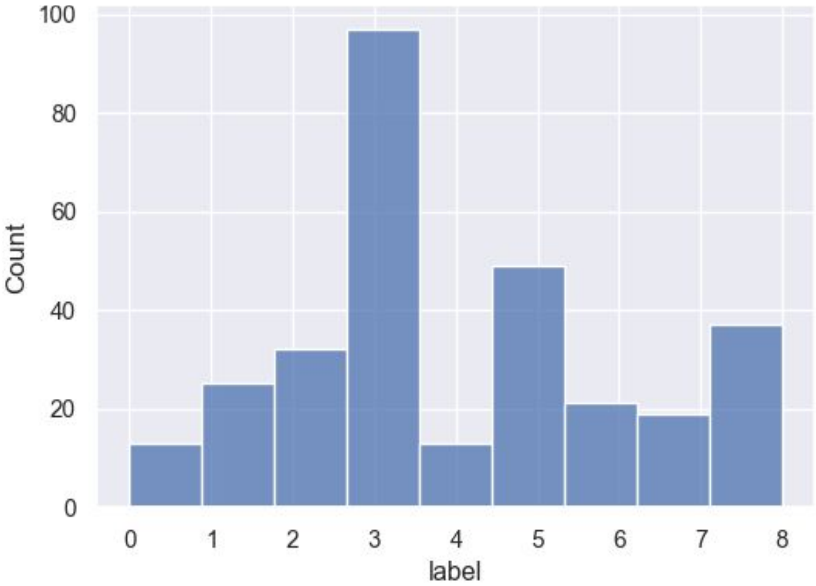


Label Distribution

Train Set:

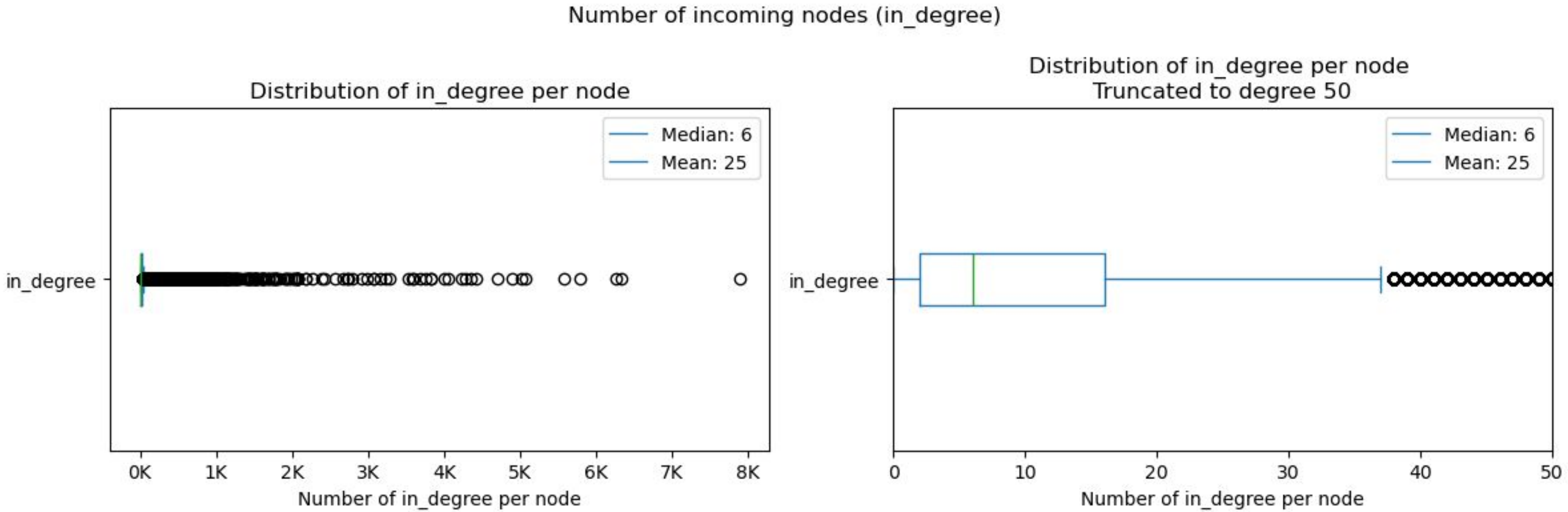


Test Set:



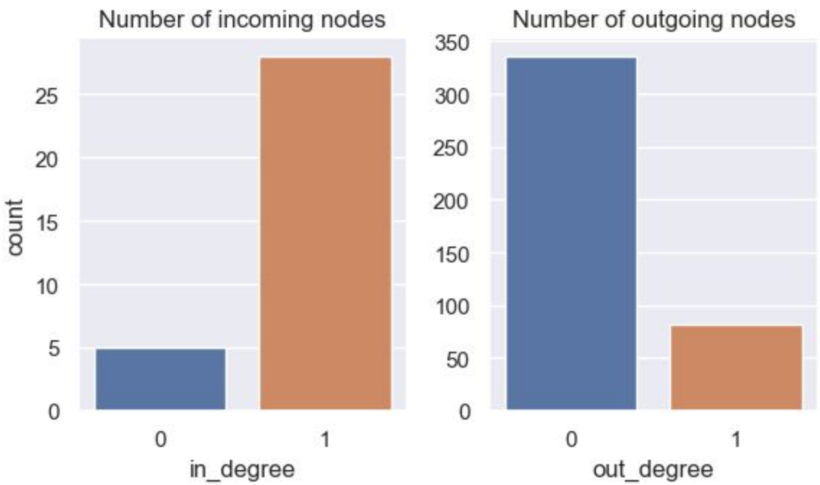
Labels follow the same distribuion in domain with and without text

Node Connectivity



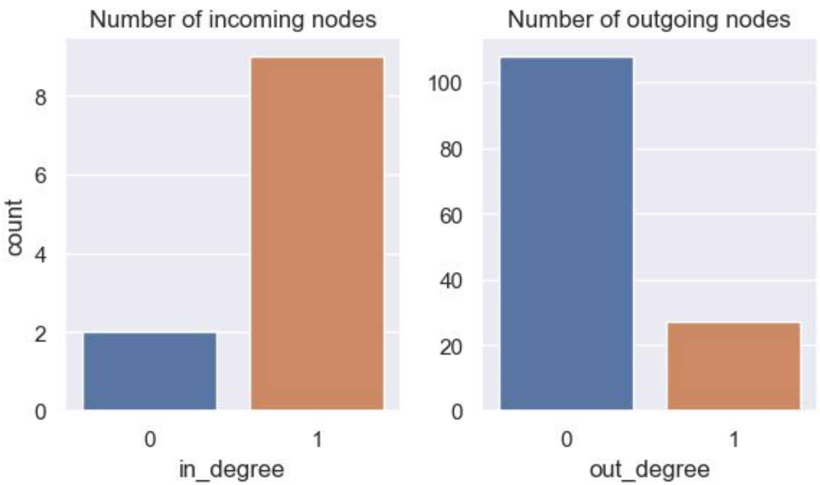
Dangling Nodes

Train Set:



In degree is more important due to 'message passing'

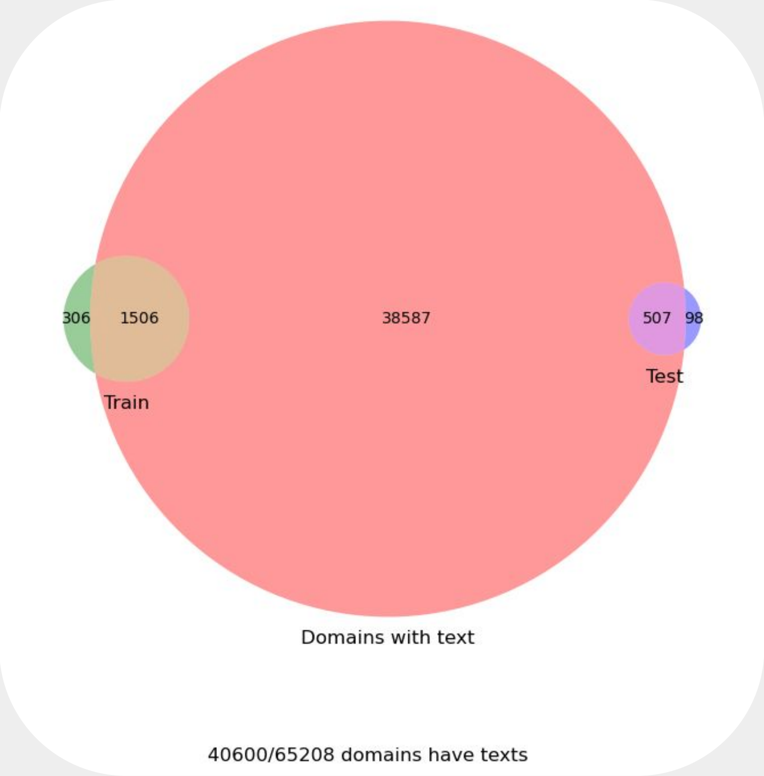
Test Set:



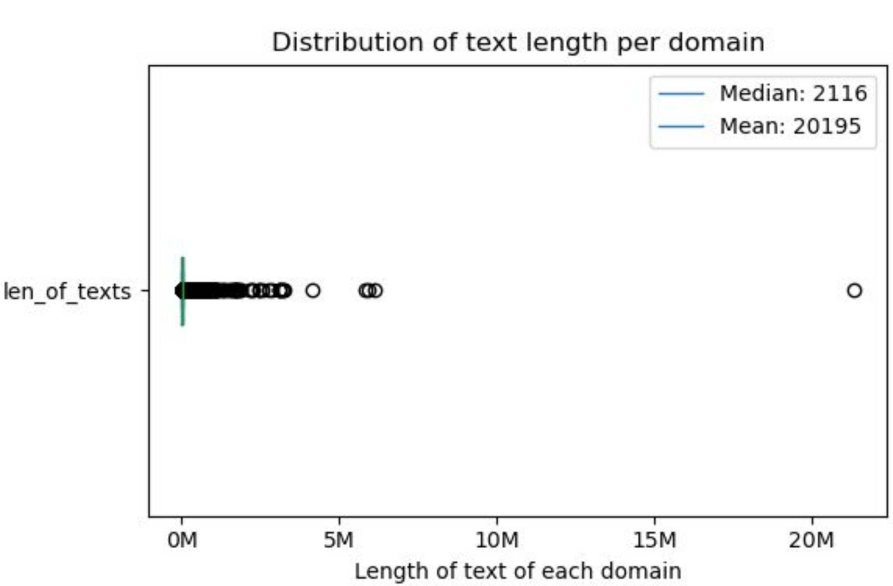
In degree is more important due to 'message passing'

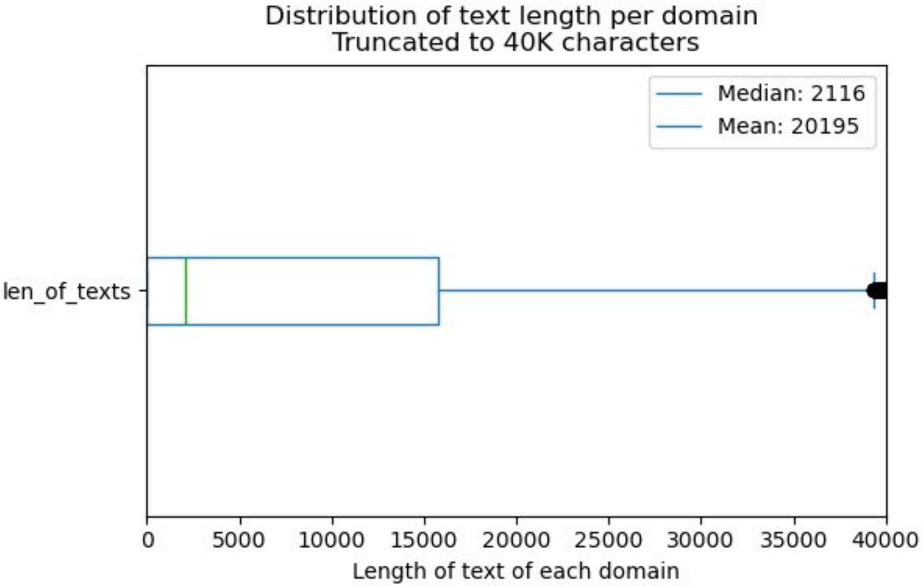
Page Content

How many pages have text available?



Length of texts

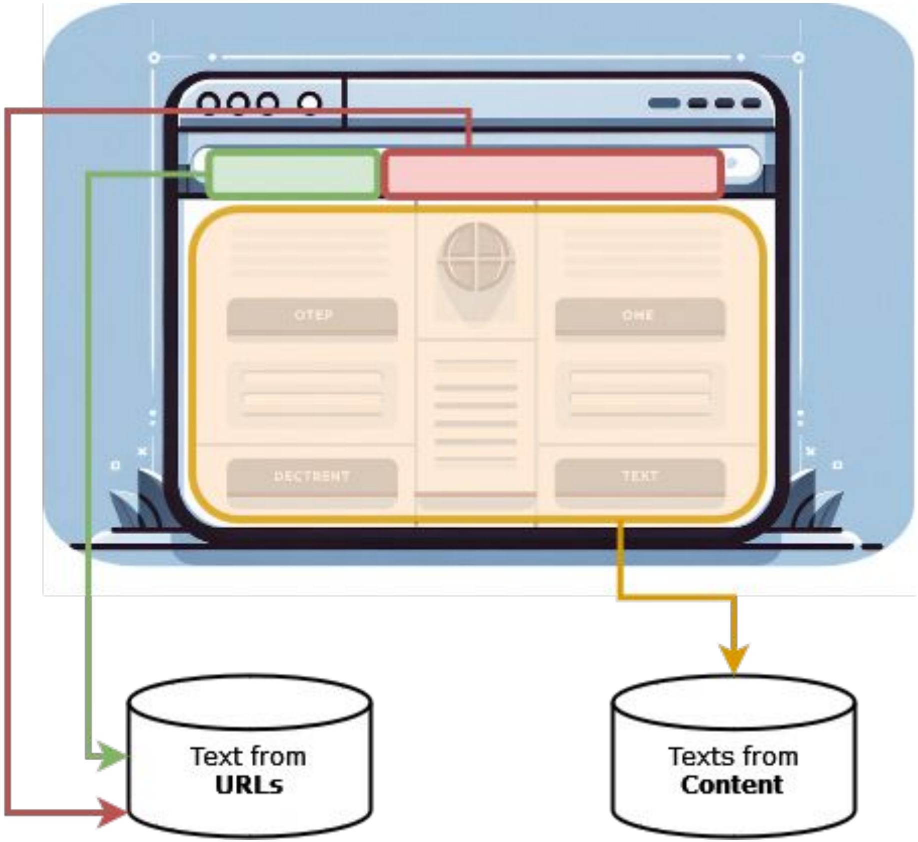




Data Cleaning

Clean Texts

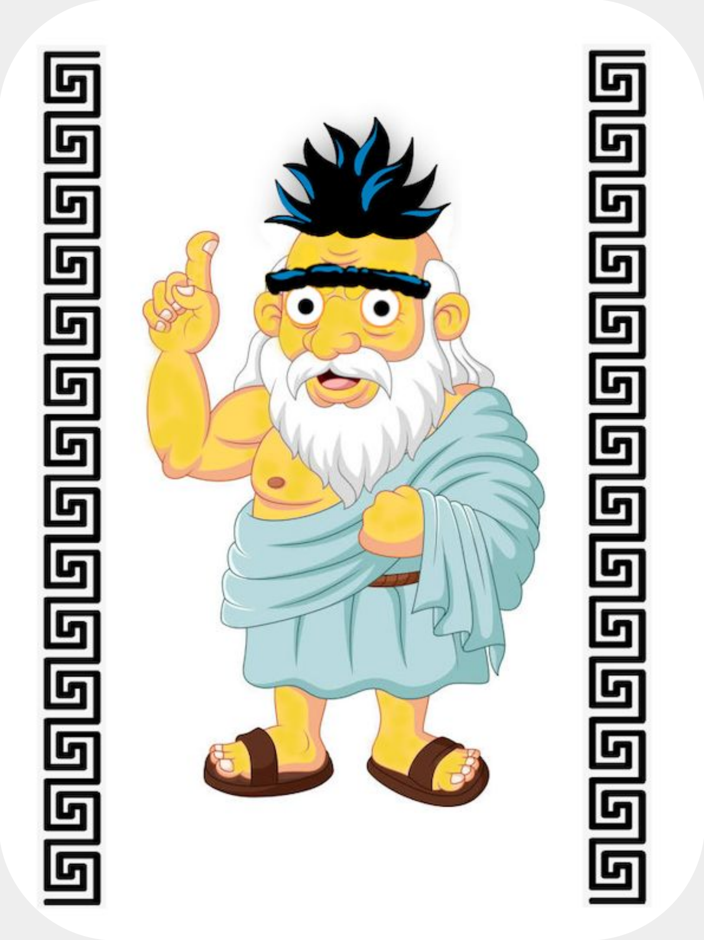
1. Clean page contents
2. Extract domain names
3. Extract urls from pages
4. Post-process URLs
    a. Unigram tokenization
    b. Transliteration
5. Truncate texts to 512 words



Feature Extraction

1.Greek-BERT *for* Text Representations

1. Train-Validation split texts
2. Finetune for 3 epochs
3. Combine texts and urls
4. Extract CLS



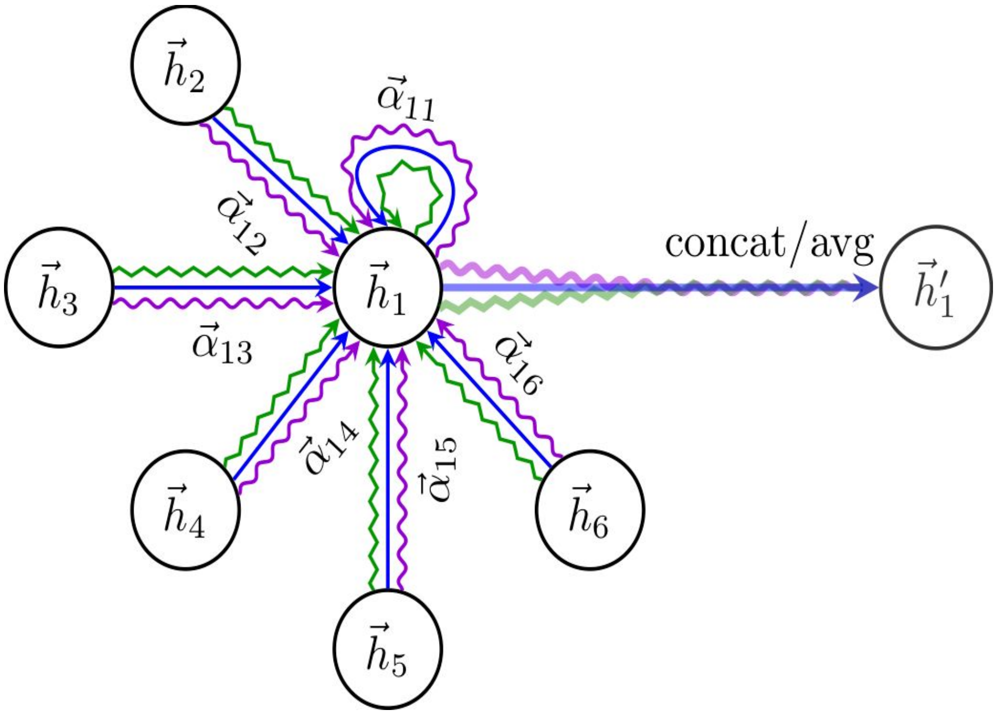
2. Deep Walk

- **Series of short random walks = Sentences**
- **50 random walks from each node**
- **Walk length 70**
- **When visiting a node we extend the sentence by selecting randomly 4 of its neighbors and continue the walk from 1 of the 4 neighbors**
- **Pass Sentences to Word2Vec to obtain 130-dimensional node embeddings**

Models

- 4 GAT models
- GAT: neural network that operates on graph
- GAT Input Features: Each node in the graph has a feature vector
- Nodes aggregate the features of their neighbors
- For each layer, output feature of each node is a weighted sum of the linear transformation of its neighbors' features, using the normalized attention coefficients as weights
- Attention: Messages from some neighbors may be more important than messages from others
- Give an attention coefficient to each neighbor, indicating the importance of that neighbor's features (using a learnable weight vector)
- Attention score for each pair of connected nodes normalized across each node's neighborhood using Softmax

$$\mathbf{h}_i^{(t+1)} = \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(t)} \mathbf{W}^{(t)} \mathbf{h}_j^{(t)}\Big)$$



Multi-head attention (3 heads)

GAT 1 Configuration

- **Input Features**:
  - BERT text CLS embeddings
- **Architecture**:
  - **2 Layers**:
    - **Layer 1**:
      - 13-dimensional output per head.
      - 4 heads concatenated to form a 52-dimensional output vector for each node.
    - **Layer 2**:
      - Produces 9-dimensional output class probabilities.
      - Uses a single head.

GAT 2 Configuration w/ Skip Connections

- **Input Features**:
  - BERT text CLS embeddings
- **Architecture**:
  - **2 Layers**:
    - **Both Layers**:
      - 25-dimensional outputs per head.
      - 2 heads concatenated, resulting in a 50-dimensional output vector from each layer.
  - **Output**:
    - Outputs from the two layers are concatenated to form a 100-dimensional vector.
    - This 100-dimensional vector is passed to an MLP to compute probabilities for 9 classes.

GAT 3 Configuration w/ Skip Connections

- **Input Features**:
  - Deep Walk embeddings
- **Architecture**:
  - **2 Layers**:
    - **Both Layers**:
      - 27-dimensional outputs per head.
      - 4 heads concatenated, resulting in a 108-dimensional output vector from each layer.
  - **Output:**
    - Outputs from the two layers are concatenated to form a 216-dimensional vector.
    - This vector is passed to an MLP to compute probabilities for 9 classes.

GAT 4 Configuration w/ Skip Connections

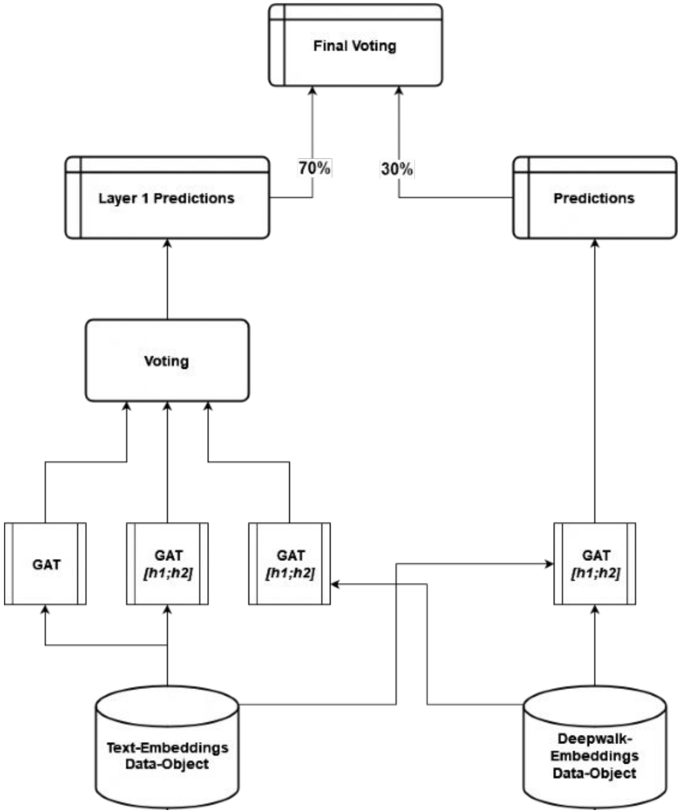
- **Input Features (concatanated)**:
  - BERT text CLS embeddings
  - Deep Walk embeddings
- **Architecture**:
  - **2 Layers**:
    - **Both Layers**:
      - 16-dimensional outputs per head.
      - 2 heads concatenated, resulting in a 32-dimensional output vector from each layer.
  - **Output**:
    - Outputs from the two layers are concatenated to form a 64-dimensional vector.
    - This vector is passed to an MLP to compute probabilities for 9 classes.

- Skip Connections: combine layer outputs, capturing both local and global information - richer features
- Hyperparameter tuning with optuna
- Learn parameters with Backpropagation - Adam optimizer
- CrossEntropy Loss

Final Model Architecture

- Voting first 3 models with equal weights

- Final Voting - Submission: 70% first Voting & 30% model 4



Results

Model	Validation Loss	Public Loss	Private Loss
CLS 1 GAT	0.7931	0.8343	0.8439
CLS 2 GAT_concat	0.7703	-	-
WALK GAT_concat	0.8067	-	-
CLS+WALK GAT_concat	0.75	0.7848	0.7631
First layer voting	0.69	0.6693	0.7255
Final Voting-submission	-	0.6691	**0.7059**

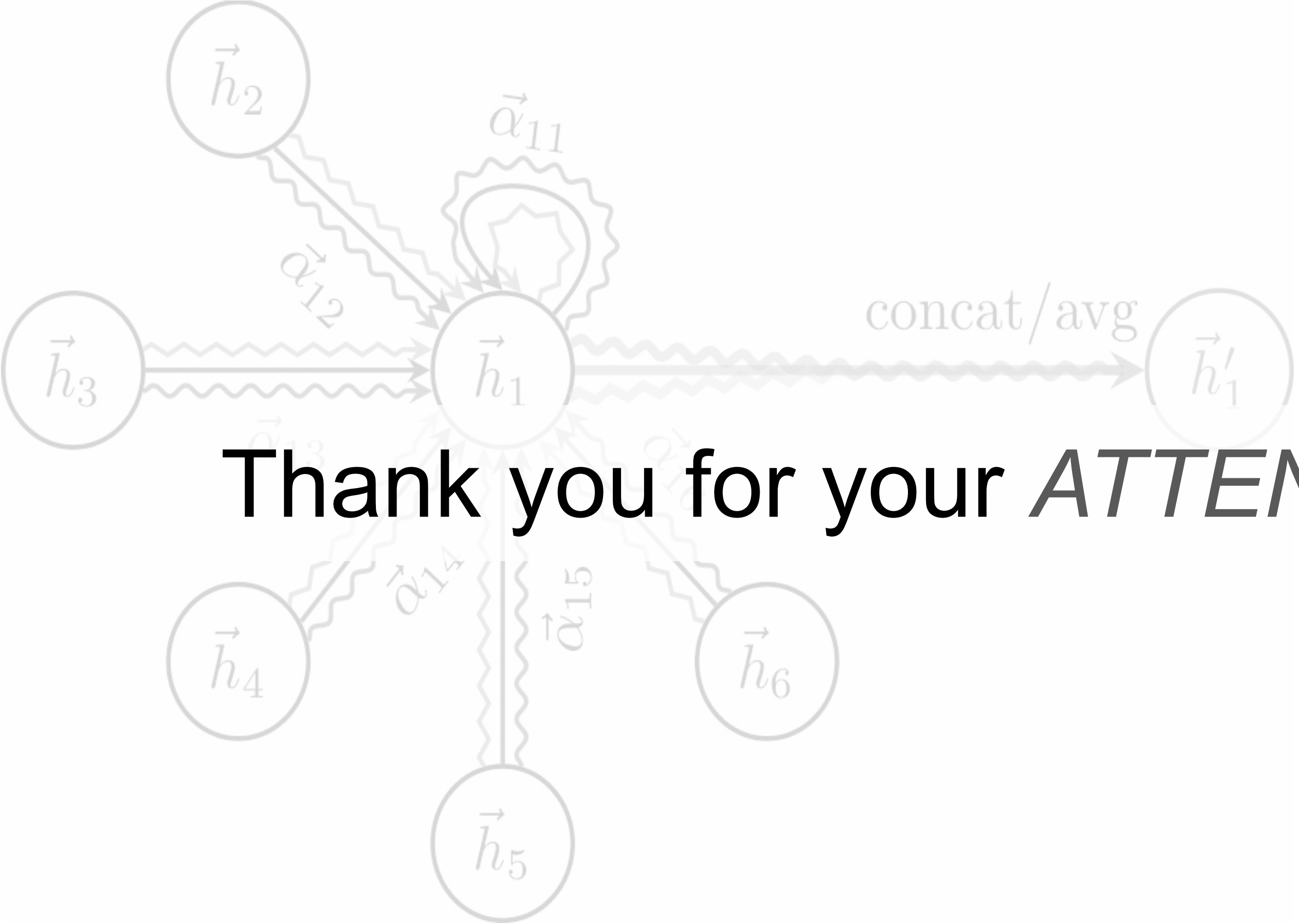
Failed Experiments

Experiments

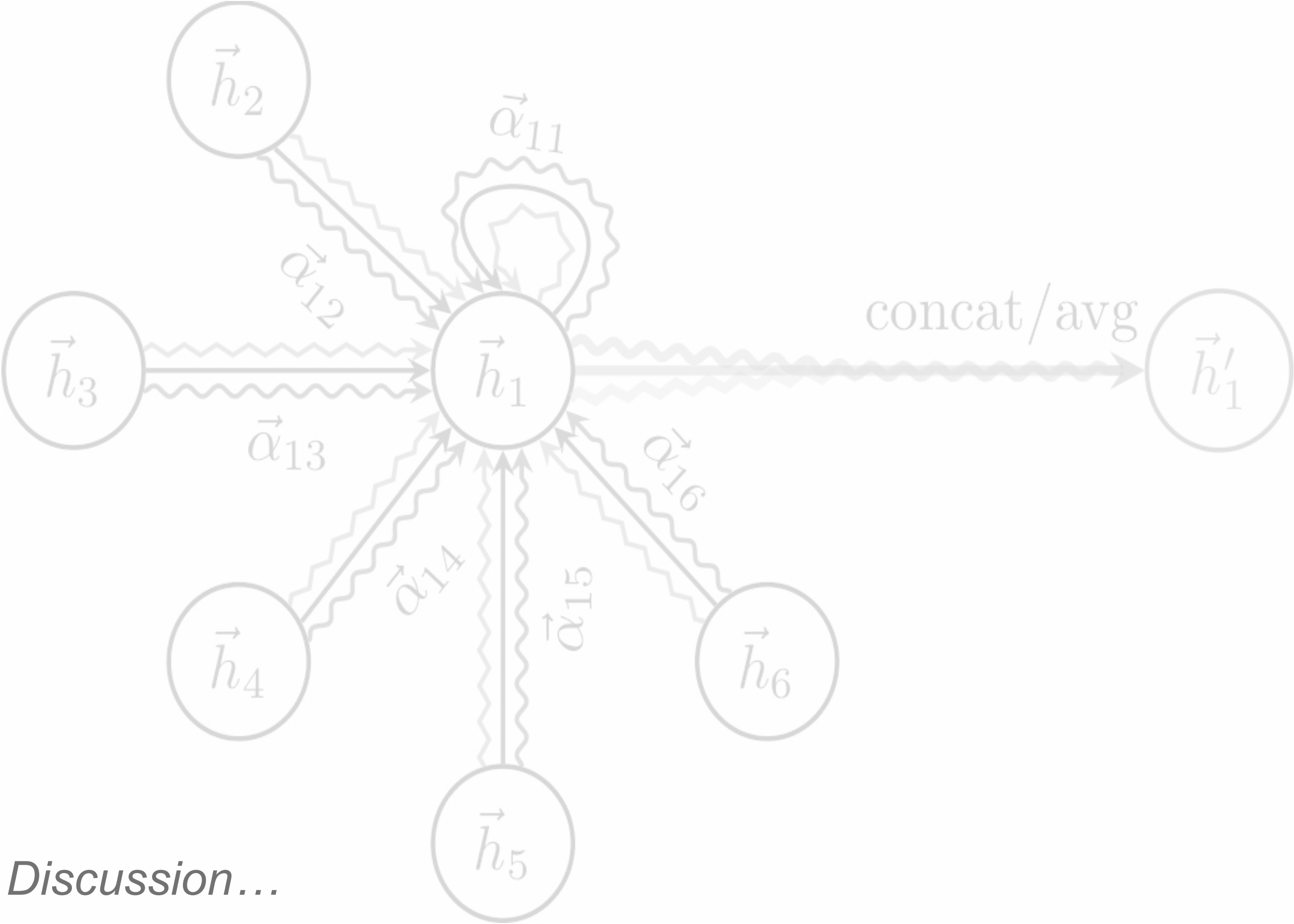
1. GCNs
2. Neighbor-Loader with GraphSage
3. Vectorizers: Count/TF-IDF
4. Multilingual BERT
5. Node2Vec
6. GAE / VGAE
7. Link-Prediction

References

1. Chalamandaris, A., Protopapas, A., Tsiakoulis, P., & Raptis, S. (n.d.). *All Greek to me! An automatic Greeklish to Greek transliteration system*. Institute for Language and Speech Processing. Epidavrou & Artemidos 6, 15125 Maroussi, Greece.
2. Koutsikakis, J., Chalkidis, I., Malakasiotis, P., & Androutsopoulos, I. (2020). GREEK-BERT: The Greeks visiting Sesame Street. In *11th Hellenic Conference on Artificial Intelligence (SETN 2020)* (pp. 110–117). Association for Computing Machinery. https://doi.org/10.1145/3411408.3411440
3. Nikolentzos, G. (2024). INF342: Domain name classification challenge. *Kaggle*. https://kaggle.com/competitions/inf342-datachallenge-2024
4. Toumazatos, A., Pavlopoulos, J., Androutsopoulos, I., & Vassos, S. (n.d.). *Still all Greeklish to me: Greeklish to Greek transliteration*. Department of Informatics, Athens University of Economics and Business, Greece; Archimedes/Athena RC, Greece; Helvia.ai.
5. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph attention networks*. arXiv. https://arxiv.org/abs/1710.10903
6. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (n.d.). *Graph neural networks: A review of methods and applications*.



Thank you for your *ATTENTION!*


$\vec{h}_2$
$\vec{\alpha}_{11}$
$\vec{\alpha}_{12}$
$\vec{h}_3$
$\vec{h}_1$
concat/avg
$\vec{h}'_1$
$\vec{\alpha}_{13}$
$\vec{\alpha}_{16}$
$\vec{\alpha}_{14}$
$\vec{\alpha}_{15}$
$\vec{h}_4$
$\vec{h}_6$
$\vec{h}_5$

Discussion…