PES UNIVERSITY

PROJECT TITLE:

DETECTING ANOMALOUS SELF-CITATIONS USING RELATION NETWORK

PES2UG21CS170 : Farhaan Ebadulla
PES2UG21CS175 : Gaurav BV
PES2UG21CS182 : H Manoj
PES2UG21CS243 : Kruthik K

Project ID: 40

Project Guide: Dr. Nazmin Begum

ABSTRACT

Anomalous self-citations distort research impact by inflating an author's perceived importance. While some self-citations provide context or build upon prior findings, their disproportionate use threatens the integrity of research metrics.
Detecting these anomalous patterns is crucial for ensuring fair and accurate scholarly evaluation. This study develops a robust methodology to identify and address the issue of anomalous self-citation behavior.

SUGGESTIONS FROM PREVIOUS REVIEW

PES UNIVERSITY

- Find out about the type of embeddings in RN
s

PROJECT PIPELINE

Dataset Preprocessing : ✅
Get Self Citation Loops: ✅
Get TSR For the papers in the Self Citation Loop ✅
Use Bart for abstractive summarization of the abstracts to get a paper summary ✅
Use ParsCit to get the citation context ✅
Fine tune LLM
Feed these papers (Along with abstract or paper summary) to the RN with the CNN (to be replaced by LSTM or BiLSTM) for feature extraction

DATA PREPROCESSING

Data Collection:

The dataset is taken from a website called Aminer. The citation data is extracted from DBLP, ACM, MAG (Microsoft Academic Graph), and other sources. We are using version 14 with 5 million papers and 36 million citations.

Data Preparation:

Removing the data attributes like artworks that are irreverent to our study, and removing the fields that are not required to our study like
useless_attributes=
['lang','volume','v12_id','v12_authors','indexed_abstract','page_start','page_end','isbn','issn','doc_type','url','issue','doi','keywords','year']

DATA PREPROCESSING

Data Input:

After Data Preparation, the data input is in this schema:

- Author
- Source
- Destination
- Contents of source
- Contents of destinations

DATA PREPROCESSING

PES UNIVERSITY

Data Preprocessing:

- We convert the data points into a graph structure using the paper ID and references field from the cleaned dataset. This allows us to construct the citation network, where all edges are undirected. We then select random nodes and their K-hop neighborhoods.

- To find self-citations, we sample and construct a heterogeneous graph with two types of nodes—authors and papers—and two types of edges—cited and published

- We then convert this graph into a homogeneous graph where all nodes are of the same type. This is achieved by converting 'cited' edges into directed edges, similar to those found in natural networks, and converting 'published' edges into undirected or bidirectional edges.

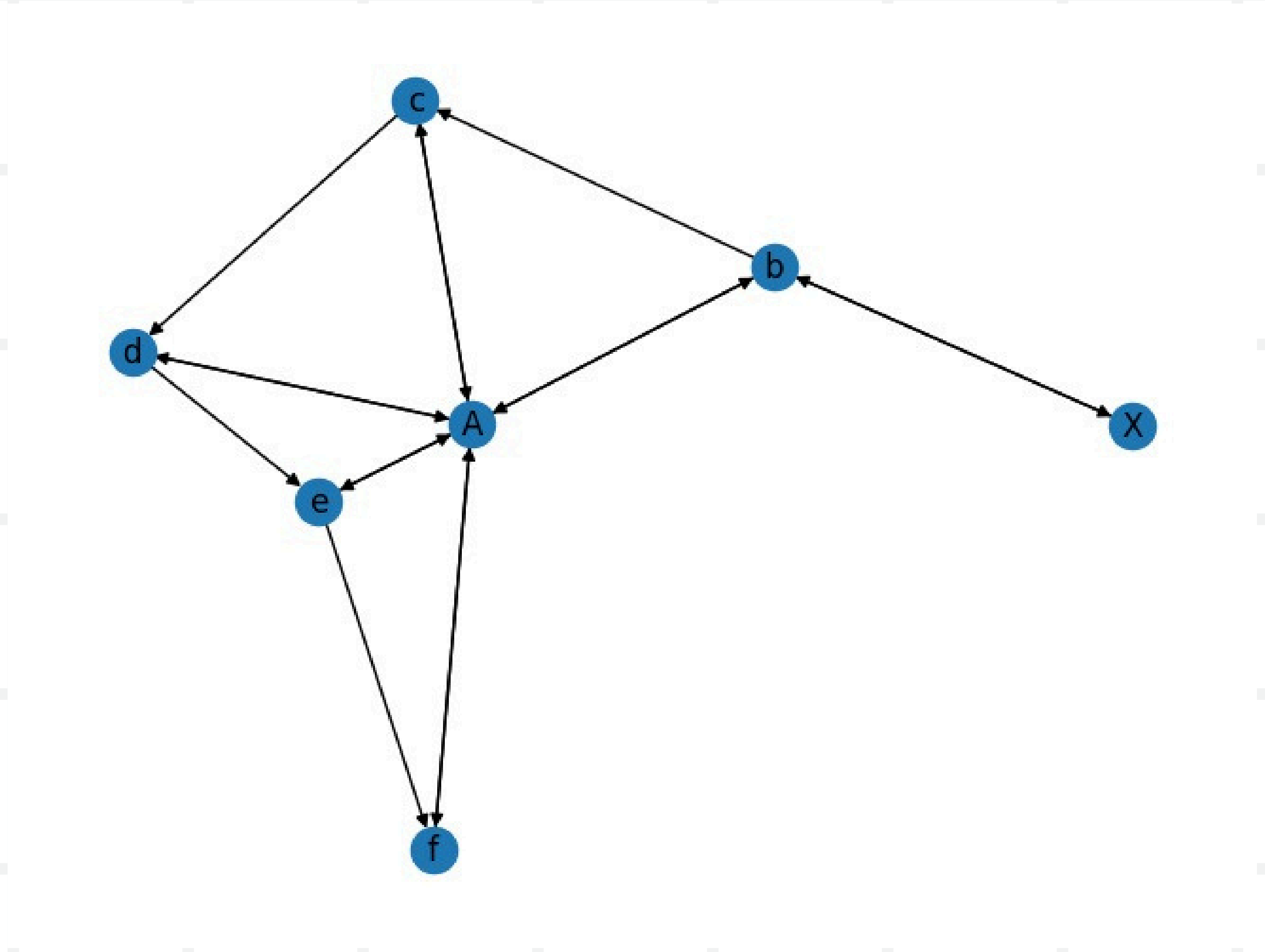
SELF CITATION DETECTION

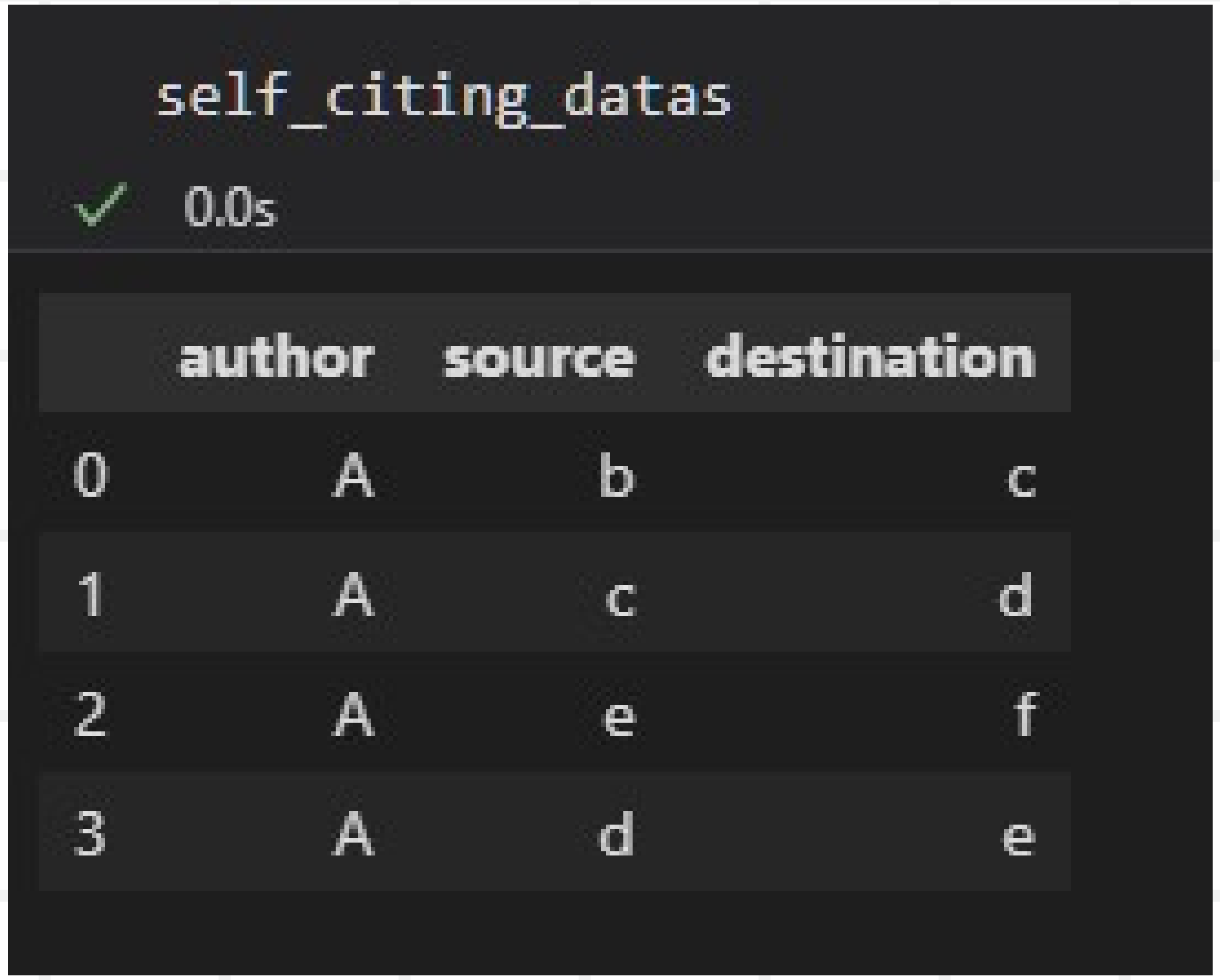
- The concept used to detect self-citations is the detection of triangles in a graph, which is a specific type of cycle, and in this case, it's used to identify potential self-citations through 3-cycles.

- If an author does not have a closed walk involving their papers (i.e., no self-citation through intermediaries), then Aij = 0.

- We compute A^3 for each author, where A is the adjacency matrix that includes the author and their papers. This identifies instances where the author is self-citing. We then focus on these authors, examine their one-hop neighborhoods, and apply our cycle-finding algorithm. This algorithm first considers the one-hop connections in A, then uses A^2 to re-examine their rows. This approach allows us to identify self-citing papers that other cycle-finding algorithms might miss.

SELF CITATION DETECTION





```
self_citing_datas
✓ 0.0s

   author  source  destination
0       A       b            c
1       A       c            d
2       A       e            f
3       A       d            e
```

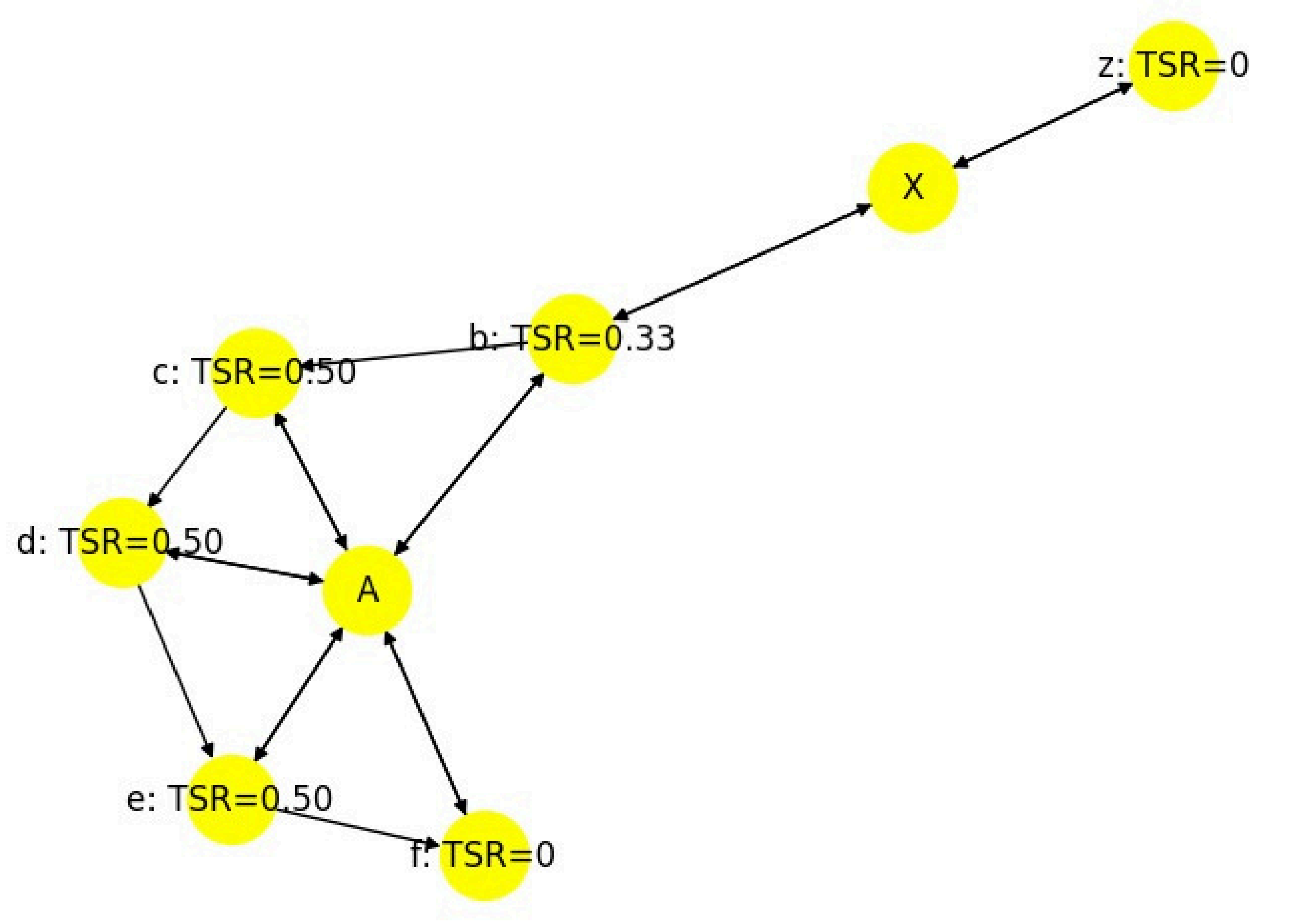
TOTAL SELF CITATION RATE

For each paper in a Self Citation Loop, the TSR value is calculated

$$TSR = \frac{TotalNumberOfSelfCitations}{TotalNumberOfCitations}$$

The Mean TSR for all the papers is calculated, and only the papers with TSR greater than the mean are retained to further reduce the size of the dataset.

TOTAL SELF CITATION RATE



ABSTRACTIVE SUMMARY

To summarize the paper, we perform a multi-level summarization (two levels) using the BART model, specifically Facebook's bart-large-cnn.

<u>First Level of Summarization:</u> We start by summarizing the paper page by page to ensure that no important information is lost, using the default parameters: num_beams=4 and early_stopping=True. Beam search, a heuristic search algorithm, explores a graph by expanding the most promising nodes within a limited set. In text generation, it tracks multiple candidate sequences (beams) at each step, selecting the top num_beams sequences based on their probabilities. With early_stopping=True, the generation process stops as soon as all beams have generated the end-of-sequence token, speeding up the process and preventing the model from generating unnecessarily long sequences.

ABSTRACTIVE SUMMARY

Second Level of Summarization: We then take the summaries from each page, concatenate them, and feed them back into the model with different parameters: num_beams=8 and early_stopping=False. This adjustment helps reduce information loss, as all the extracted information from the first level is important and needs to be preserved.

CITATION CONTEXT

- To assess the relevance of self-citations, we plan to extract the citation context using the citation parsing tool available in the ParsCit repository.

- The process involves extracting the text surrounding a citation reference, known as the citation context. This context provides insight into how and why the authors referenced their own work, which is essential for determining whether the self-citation is relevant and meaningful or merely an attempt to inflate citation metrics.

CITATION CONTEXT

PES UNIVERSITY

- To streamline our analysis, we will focus on extracting the citation contexts only from those papers that contain self-citations. These extracted contexts will be 3 lines before the citation and 5 lines after the citation. This structured data will allow us to systematically evaluate the relevance of self-citations across multiple papers.

- However, while extracting the citation context using ParsCit, we have encountered an issue where formulas and mathematical expressions within the citation context are being converted into special characters. This issue requires further processing of the extracted context to ensure that the original content, including formulas, is accurately represented

FINE-TUNING LLM

- We will fine-tune large language models (LLMs) by providing them with the summary of a source paper and the citation context within a destination paper. The LLM will compare the similarity between these two pieces of text to help flag whether the citation context is anomalous.

- We will use both LLAMA 3 and Google's Gemini to individually assess these text pairs, flagging any anomalies. The average of their similarity scores will be taken, and if the score exceeds a predetermined threshold, the data point will be flagged as anomalous. This process will be repeated for 50 out of 200 data points in the dataset.

FINE-TUNING LLM

Dataset Selection:

- Use the SICK (Sentences Involving Compositional Knowledge) dataset for fine-tuning the large language models (LLMs).

Fine-Tuning Process:

- Format Data: Prepare the dataset with sentence pairs and their semantic similarity scores.

Train Model:

- Fine-tune the LLM on the SICK dataset to learn the relationship between sentence pairs and their similarity.
- Adjust Weights: During training, adjust the model's weights to predict similarity scores that match the ground truth in the dataset.

FINE-TUNING LLM

Post-Fine-Tuning Application:

- Evaluate Similarity: Use the fine-tuned model to assess the similarity between paper summaries and citation contexts.
- Generate Scores: For each pair, the model generates a similarity score.

Anomaly Detection:

- Calculate Anomaly Score: Compute 1−similarity score for each pair.
- Threshold Comparison: Compare the adjusted score to a predefined threshold.
- Flag Anomalies: If the adjusted score exceeds the threshold, flag the citation context as anomalous, indicating a potential mismatch or inconsistency.

RELATION NETWORK

Relation Networks are a type of neural network architecture designed for few-shot learning tasks. Introduced by Sung et al. in 2018 for few-shot classification. RNs learn to compare pairs of examples and determine their similarity

Two main components: embedding module and relation module

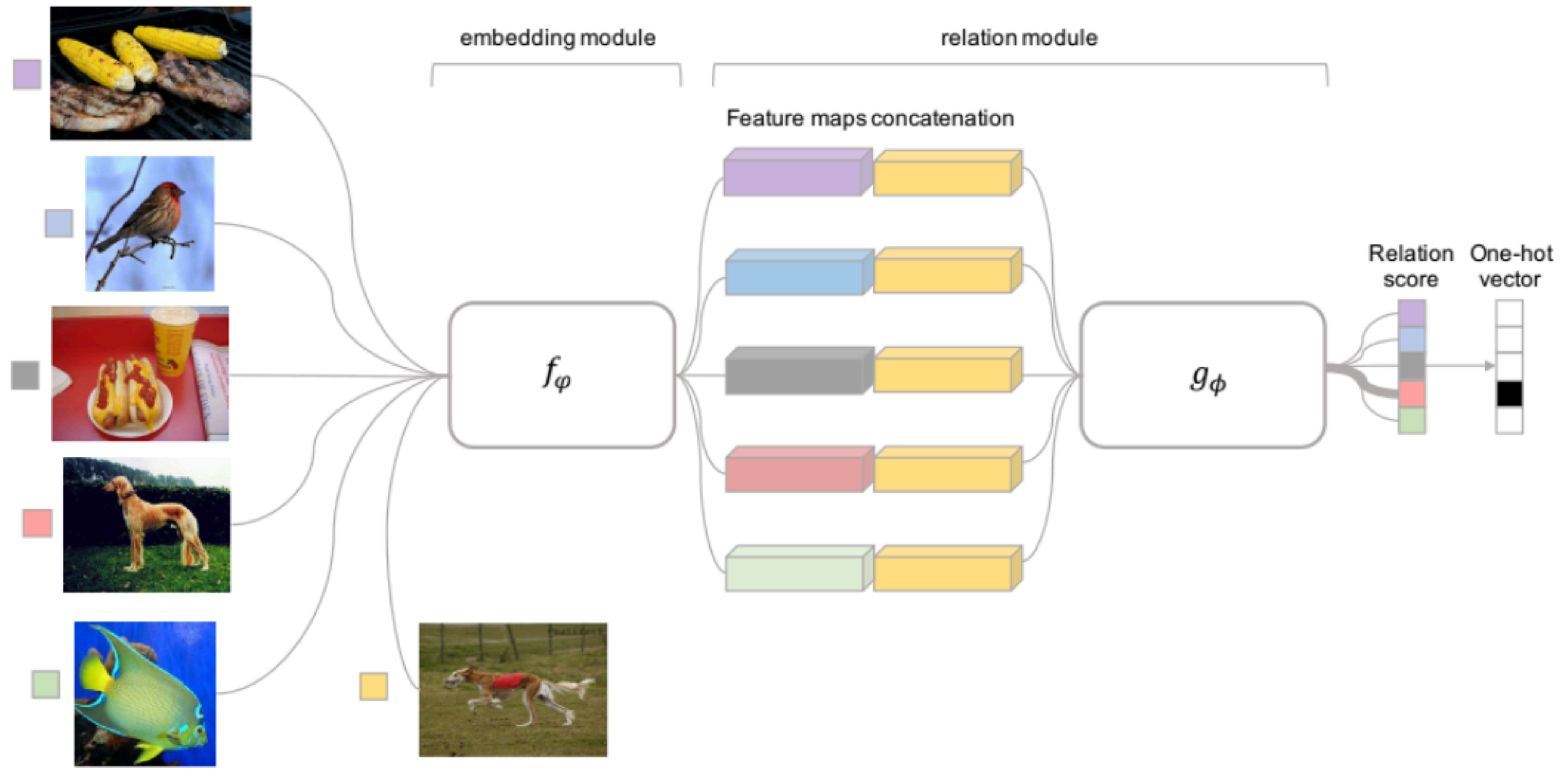
- Embedding module: Processes input features
- Relation module: Compares pairs of embedded examples

Learns to generate similarity scores between examples

RELATION NETWORK

Support set: Small set of labeled examples (e.g., 5 examples per class)
Query set: New examples to be classified

RN compares each query to the support set and predicts class based on similarity scores



RELATION NETWORK

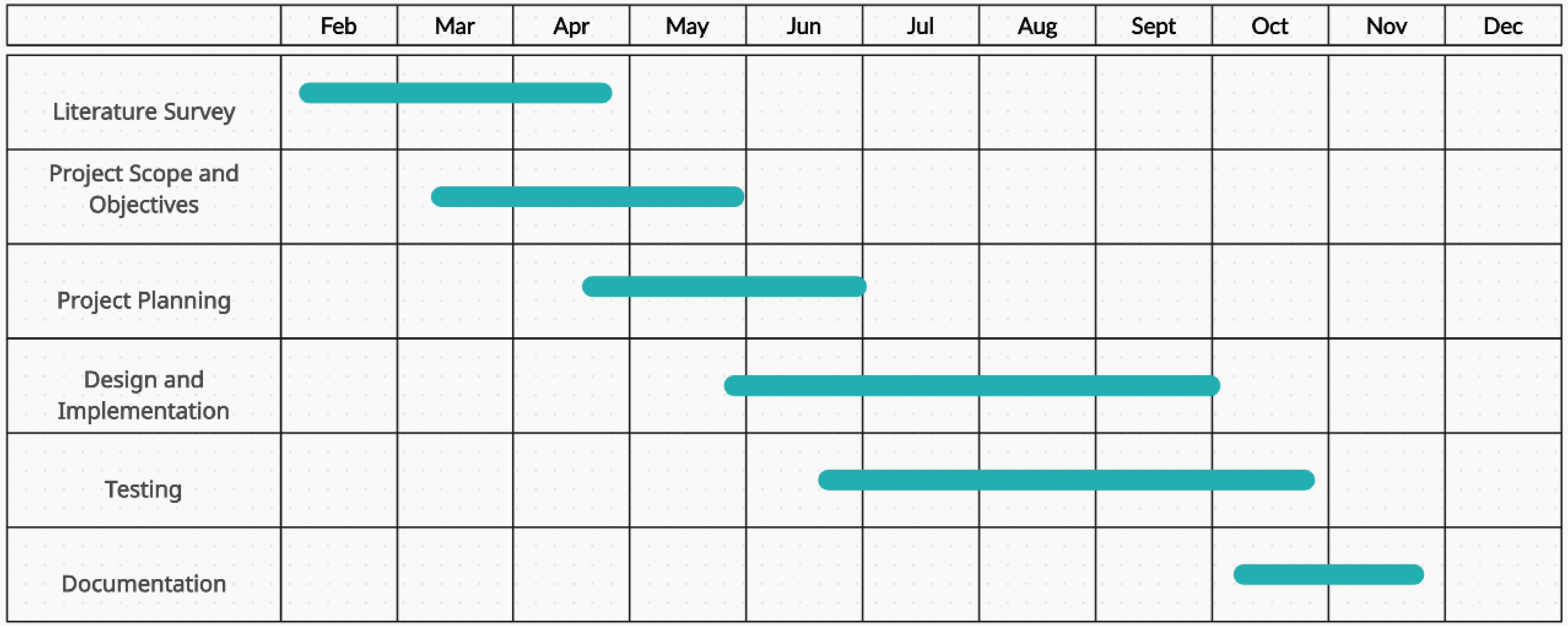
Task: Flag 150 unlabeled examples using 50 labeled examples
Input: Paper summaries and citation contexts
Output: Binary classification (anomalous or not anomalous)

RN Architecture for Our Task

- Embedding module: Process text features
- Relation module: Compare citation pairs
- Output: Similarity score indicating likelihood of anomalous self-citation

RELATION NETWORK

- Use 50 labeled examples to create training pairs
- Train RN to distinguish between anomalous and non-anomalous pairs
- Compare each unlabeled example with labeled examples
- Use confident predictions to augment labeled dataset
- Retrain model and repeat classification
- Apply threshold to flag remaining examples
- Advantages: Works well with limited labeled data, learns to compare examples


Feb
Mar
Apr
May
Jun
Jul
Aug
Sept
Oct
Nov
Dec
Literature Survey
Project Scope and Objectives
Project Planning
Design and Implementation
Testing
Documentation

THANK YOU