*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

Abstract

Accurate assessment of research impact is of utmost importance because of its gravity in assessing scholarly contributions, which determines the fundings and promotions. While impartial assessments are of utmost need, anomalous self-citation--the practice of inflating one's own research with unmerited citation-may affect the dilution of the importance to be rendered to other research workers' work.

Using citation network analysis, machine learning techniques, and domain-specific knowledge, the study depicts a novel approach to identifying anomalous self-citation patterns.

To this end, it builds a comprehensive citation network through a large academic publication corpus, extracting features always at the author and paper level. This is to develop a predictive model to identify those authors who exercise statistically improbable self-citation levels and distinguish between regular oscillations and persistent self-citation patterns.

The study, thus, evaluates the suitability for various disciplines of approaches for the identification of unusual self-citations, contributing to discussions about the science of evaluation and research ethics and providing a robust approach that will, in the end, work towards cancelling out the skewing effect of excessive self-citation in the assessment of research impact.

Introduction

In academic research, it is crucial to reference relevant earlier studies. When you cite previous work, you are essentially acknowledging the debt of ideas you owe to others. Moreover, it helps provide context for your own discoveries.

The academic world is concerned about the self-citation phenomena, in which authors cite their own previous publications. Although self-citations are a valid way for researchers to place their most recent work within their body of work, using them excessively or disproportionately might skew how important and impactful one's scholarly contributions are seen to be.

The integrity of research assessment measures is manipulated by anomalous self-citation practices, in which authors consistently and incorrect cite their work. The quality and the impact of the scholarly output of the publication or research paper citation metrics like h-index, journal impact factor and citation counts are used. Important choices about funding distribution, tenure, hiring, and promotion are informed by these variables. However, they can be made unreliable by anomalous self-citation behaviour that artificially inflates the number of citations.

It is necessary to understand and combat such behaviors that artificially increase the number of citations as this helps to uphold the integrity and equity of research publishing. Although certain works have tried to explore the degree and reasons for excessive self-citation, there is still more need to develop efficient detection methods of such behaviours on a much broader scale. It is even more challenging to pinpoint self-citation for the purposes of accentuating one's citation index due to the collaborative cultures and creativity that come in with the disciplines.

This project aims to construct an in-depth structure, as well as the set of tools, for the detection of abnormal self-citation patterns in academic texts. By using such techniques as social network studies, graph theory algorithms, and knowledge of a specific area, we suggest a technique to tell the difference between a real self-citation and one that has been perpetrated to boost an author's citation numbers. The method that we propose here also includes the variety of self-citation strategies which allows for a more nuanced and contextualized analysis of self-citation patterns.

Problem Definition

Researchers rely on the assessment of articles and their citations in the process of academic publishing, which affects several crucial decisions such as funding or the reputation of an author. To determine the quality level of a publication or a citation of a research paper, publication metrics, such as the h-index, journal impact factor, and citation counts among others, are referred to. Generally, there is a problem with the reliability of such measurements. Therefore, the tendency of the authors to exaggerate the degree of relevance of their previous works to the current study often leads to the practice of self-citation. This and other similar opportunistic practices can contribute to an author's exaggerated media coverage which can distort the picture of his or her research output. This issue creates unfair advantages that can hurt researchers who play by the rules with their citations. It also damages the reputation of academic institutions and research organizations. The detection of anomalous self-citation patterns is a complex topic as it involves separating out anomalous self-citations from those which are legit and provide context. The problem is complicated for various of reasons:

- Due to differences in citation practices, collaborative patterns, and the nature of research in some subjects, self-citations might vary throughout various domains

- Researchers in the early stages might exhibit higher anomalous self-citations so that they can establish their research paths, while more experienced scholars will have diverse citation patterns

- Author's might mention the work of their colleagues, co-authorship research team composition can lead to anomalous self-citation

- Self-citations patterns such as citations rings will reveal themselves over time. Self-citation patterns will change over time, exhibiting brief spikes of self-citations which may not always be an indication of anomalous self-citation

By addressing these problems, we aim to contribute in maintaining the integrity of academic publications. In the end, identifying and reducing anomalous self-citation behaviours will promote a fairer evaluation.

Literature Review

3.1. **Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks."**

Introduction: This paper introduces NetGAN, a method for generating graphs using random walks while maintaining the structural properties.

Characteristics and Implementation: NetGAN uses an LSTM to handle sequence memory during random walk generation and employs latent space interpolation to smoothly change graph characteristics.

Features: NetGAN can generate graphs while maintaining structural properties without explicitly specifying them.

Evaluation: NetGAN's performance is evaluated by link prediction using sampled random walks from the trained generator.

3.2. **Avros, Renata, et al. "Detecting Pseudo-Manipulated Citations in Scientific Literature through Perturbations of the Citation Graph."**

Introduction: The paper proposes a novel approach to evaluate citation credibility by evaluating the stability of citation paths using network perturbations.

Characteristics and Implementation: The approach involves generating a new graph by using network structure information, initializing an array with zeros equal to the number of edges, creating temporary graphs at each iteration, and randomly removing nodes to evaluate network robustness and link stability.

Features: The method focuses on randomly removing nodes to assess network stability and credibility of individual links.

Evaluation: The paper highlights the limitations of the proposed method, the textual information can vary a lot, complexity of graph learning, and sensitivity to the sampling strategy. the method is sensitive to how you choose which data to use for the analysis.

3.3. **Yu, Shuo, et al. "Detecting outlier patterns with query-based artificially generated searching conditions."**

Introduction: This paper develops a method to identify outlier patterns(motifs) using user queries to find the structures.

Characteristics and Implementation: The method creates user queries with target motifs and search paths to guide searching conditions effectively. The method utilizes sub-graph and meta-path queries to develop a motif discovery algorithm and improve outlier pattern identification accuracy by computing motif similarity from node similarities.

Features: The objective of this method is to improve the understanding of motifs and outlier patterns.

Evaluation: Limitations pointed out by the paper include dependence on user queries, scalability, and evaluation metric consideration.

3.4. **Kashtan, Nadav, et al. "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs."**

Introduction: This paper creates a new algorithm for estimating subgraph concentrations and detecting motifs in several networks.

Characteristics and Implementation: The algorithm, which samples subgraphs for relative frequency, uses a probabilistic sampling technique for the estimates of subgraph concentrations

Features: The algorithm provides accurate estimation of subgraph concentrations and motif detection in large and complex networks.

Evaluation: There is a mention of limitations with respect to huge subgraphs in networks with highly populated nodes and potential sampling biases.

3.5. **Duke, Richard A., Hanno Lefmann, and Vojt eˇ ch Rödl. "A fast approximation algorithm for computing the frequencies of subgraphs in a given graph."**

Introduction: This paper presents an algorithm for frequency counting of subgraphs in graphs (motifs).

Characteristics and Implementation: This algorithm estimates the occurrence of labelled subgraphs(motifs) in more giant graphs using some variant of the regularity lemma.

Features: An algorithm provides efficient estimation of the number of induced subgraphs (motifs) with minimal error.

Evaluation: The paper talks about the dependency on the accuracy of approximation, which can affect the precision of the frequency estimate of the subgraph. This paper also talks about the limitation linked with the applicability of the algorithm on certain types of graphs (heterogenous graphs).

3.6. **Avros, Renata, et al. "Spotting Suspicious Academic Citations Using Self-Learning Graph Transformers."**

Introduction: This paper utilizes a masking mechanism in self-supervised learning for predicting masked nodes and explores heterogeneous graph inputs to enhance gMAE-based learning approaches.

Characteristics and Implementation: The paper implements self-supervised learning with masking on the Cite Seer dataset and develops an asymmetric encoder-decoder architecture for predicting masked nodes. It utilizes heterogeneous graph as inputs to enhance the predictive capabilities of the model.

Features: The approach aims to identify anomalous self-citations using self-learning graph transformers.

Evaluation: The limitation of this methods discussed are quality of the dataset, computational complexity, the size of the dataset and the quality of the masking process.

3.7. **Yang, Carl, and Jiawei Han. "Revisiting citation prediction with cluster-aware text-enhanced heterogeneous graph neural networks."**

Introduction: This paper addresses citation prediction, focusing on realistically predicting the average paper citations per year.

Characteristics and Implementation: The paper uses HGN, CA, and TE modules to model various object types and interactions accurately. It utilizes heterogeneous network for data representation, which can hold multiple object types and links. The cluster-aware module models interactions and nodes within research domains, while the text enhancing module performs term mining to improve data quality

Features: The approach aims to improve citation prediction between publications by using cluster-aware and text-enhanced components in heterogeneous graph neural networks.

Evaluation: The limitation of this method is the dataset quality and the effectiveness of the knowledge graphs and the decoders.

3.8. **Zhao, Ying, et al. "Enhancing inter-sentence attention for Semantic Textual Similarity."**
Introduction: This paper introduces Enhanced Inter-sentence Attention (EIA) for Semantic Textual Similarity task and develops a new multi-head architecture for self-attention for comparing sentence pairs.

Characteristics and Implementation: The approach proposes general and specific enhancements in the attention mechanism, leveraging WordNet semantic relations. It integrates original and enhanced attention representations through gated fusion and evaluates performance enhancement on BERT and RoBERTa models, particularly on smaller datasets.

Features: The EIA method aims to improve the performance of STS tasks by enhancing inter-sentence attention.

Evaluation: The paper mentions limitations such as limited applicability, effectiveness variability, and computational overhead.

3.9. **Deepak Agarwal and Bee-Chung Chen. "fLDA: matrix factorization through latent Dirichlet allocation."**

Introduction: The goal of this paper is to recommend scientific papers using a hybrid technique that combines concepts from collaborative filtering based on a relevance-based language model with content analysis based on probabilistic topic modelling.

Characteristics and Implementation: The main aim is to increase the effectiveness of recommendations by utilising latent topics of interest in researcher publications and utilising the social structure of researchers as trustworthy sources of knowledge. The method conducts experimental research using the DBLP dataset.

Features: The paper aims to provide effective scientific paper recommendations by combining content analysis and collaborative filtering techniques.

Evaluation: For improved suggestions, the research proposes expanding the method for creating a heterogeneous network to incorporate item qualities as well as structure, but it acknowledges overfitting to noise as a drawback.

3.10. **B. Hu, C. Shi, W. X. Zhao, and P. S. Yu. "Leveraging meta-path-based context for top-N recommendation with a neural co-attention model."**

Introduction: This paper develops the Relative Self-Attention Transformer for encoding molecular features in graph data and enhancing Transformer-based models by incorporating atom relations, positional encoding, and neighbourhood features.

Characteristics and Implementation: The paper implements Graph Transformers for direct operation on molecular graph structures, utilizes relative positional encoding to encode information about relative distances between atoms, and incorporates neighbourhood, bond, and distance embeddings to capture physicochemical features comprehensively.

Features: The Relative Self-Attention Transformer aims to improve the representation and modelling of molecular features in graph data.

Evaluation: The paper notes the dependency on Graph Convolution layers, positional information handling as limitation

3.11. **Maziarka, Łukasz, et al. "Relative molecule self-attention transformer."**

Introduction: This paper develops GOAt, a local-level Graph Neural Network (GNN) explainer, to address limitations of existing GNN explainers, such as insufficient discriminability and inconsistency on sameclass data samples.

Characteristics and Implementation: GOAt expands GNN outputs into scalar products and attributes each product to input features, demonstrating superior explanation ability using a great deal of testing on both hypothetical and real databases.

Features: GOAt aims to provide local explanations for Graph Neural Network (GNN) outputs by attributing each output component to the input features

Evaluation: The paper states that GOAt overcomes limitations like insufficient discriminability and inconsistency compared to existing GNN explainers through extensive evaluations on synthetic and real datasets. However, it mentions similarity with decomposition methods and overfitting to noise as potential limitations.

3.12. **Lu, Shengyao, et al. "GOAt: Explaining Graph Neural Networks via Graph Output Attribution."**

Introduction: This paper utilizes Graph Convolutional Networks (GCNs) for analysing hyperspectral images (HSI).

Characteristics and Implementation: The approach involves spectral-spatial graph construction, dual self-expression attention-based fusion, and clustering. It extracts information using a sliding window and k-nearest neighbours to construct a graph for the GCN, applies the GCN to extract neighbourhood information, fuses the information to construct a final affinity graph, and clusters this graph.

Features: The method utilizes GCNs' learning ability to analyse topological relationships between nodes in HSI data.

Evaluation: The paper mentions comparison limitations, a single node focus, and model limitations.

3.13. **Li, Xianju, et al. "Multiview Subspace Clustering of Hyperspectral Images based on Graph Convolutional Networks."**

Introduction: This paper aims to make large language models (LLMs) faster by using Graph Neural Networks (GNNs).

Characteristics and Implementation: The paper proposes Linguistic Graph Knowledge Distillation (LinguGKD), which trains a lightweight GNN student model using an LLM teacher model. It involves teacher feature learning via the LLM, extracting semantically-enriched node features, and student feature learning via the GNN model.

Features: The approach combines LLMs and GNNs to develop faster language models.

Evaluation: The paper mentions similarity with decomposition methods and overfitting to noise as potential limitations. methods and overfitting to noise as potential limitations.

3.14. **Huang, Mu-hsuan, and Pei-shan Chi. "A comparative analysis of the application of h-index, g-index, and a-index in institutional-level research evaluation."**

Introduction: This paper evaluates scientific indices and the performance of researchers, aiming to provide a more comprehensive analysis.

Characteristics and Implementation: The paper highlights the limitations of existing indices in evaluating researchers' performance and introduces new indices (th-index and tA-index) that offer a more comprehensive and detailed analysis of the quality of publications in the htail.

Features: The new indices aim to improve the evaluation of researchers and their scientific impact by addressing limitations of existing indices.

Evaluation: The paper notes the lack of consideration for the quality of publications in the h-tail beyond the h-core, only limited information provided about the distribution, difficulty in comparing researchers' work as limitations.

3.15. **Ferrara, Emilio, and Alfonso E. Romero. "Scientific impact evaluation and the effect of self-citations: Mitigating the bias by discounting the h-index."**

Introduction: This paper presents a new way, called the h-index, to solve problems with current evaluation methods like the h-index. It makes evaluations fairer by including self-citations

Characteristics and Implementation: The paper provides real-world examples to demonstrate how self-citations can affect rankings and highlights the importance of considering self-citations in impact evaluation.

Features: The dh-index aims to mitigate the bias introduced by self-citations in research impact evaluation.

Evaluation: The paper demonstrates the effectiveness of the dh-index in mitigating bias through real-world examples from the field of computer science, but notes limitations such as dependency on specific datasets and ranking services, and the potential for not considering all factors influencing impact assessments.

3.16. **Majeed, Fiaz, et al. "Self-citation analysis on Google Scholar dataset for H-index corrections."** **Introduction:** This paper compares two datasets and analyzes them to determine the impact of self-citation, aiming to provide a better comprehensive evaluation and highlight the significance of a revised h-index.

Characteristics and Implementation: To improve the h-index calculation for Google Scholar, the paper quantifies and removes self-citations from the h-index. It examines two datasets to determine how self-citations affect the h-index: one with scientists who have won awards, and the other with scientists who have not.

Features: The paper introduces a revised h-index calculation by accounting for self-citations.

Evaluation: The paper talks about its limitations such as lack of data sources and the lack of detailed explanation on excluding a citation.

Product Requirement Specifications:

Product Perspective

Product Features

Key features of this anomalous citation detection system include:

- **Citation Network Analysis**: Builds and models a citation network using data from Aminer, mapping relationships between papers and identifying self-citation loops using graph random walks.
- **Self-Citation Detection**: Identifies self-citation loops using graph-based techniques, focusing on three-step random walks to detect recurrent citation patterns.
- **Summarization Pipeline**: This process takes the paper identified in self-citation loops and summarizes information, using BERT for page-level summarization and Gemini for final aggregated summaries.
- **Citation Context Extraction**: Captures context surrounding citations, specifically four lines before and after citation instances, to provide clear examples for analysis.
- **Classification Model**: Utilizes fine-tuned LLMs (GPT-4-o, Gemini, and Llama 3.2) with established rules to classify citations as essential or non-essential based on defined criteria.

User Classes and Characteristics

The primary user classes include:

• Academic Researchers: The system will be used by individuals who have the responsibility of conducting and publishing scholarly research, to ensure the integrity of the citation practices to avoid unintentionally violating anomalous self-citation rules.

• Journal Editors and Publishers: The system will be used by those who have the duty of ensuring that academic publications maintain quality and professionalism, to check manuscripts before their publications.

• University and Research Institution Administrators: The system will be utilized by those committed to overseeing the conduct of research in universities and ensuring academic integrity by monitoring over proper practices by faculty and researchers

• Research Funding Agencies: The system will be used by those allocating grants and funding to research programs to check credibility and establish unbiased and non-misleading information that will guide them in issuing funding decisions.

Operating Environment

The system will be implanted on high-performance computing infrastructure, supplemented with research databases and citation management systems. Essential components:

- Computing Clusters: For training and deploying large language models and graph-based analysis tools.
- Distributed Storage Systems: For handling citation and paper data at scale.

General Constraints, Assumptions, and Dependencies

- **Constraints**:
  - Data privacy must be upheld, especially around sensitive research information.
  - Availability of high-performance computational resources, such as GPUs.
  - Near real-time processing capabilities to accommodate new citation data.
- **Assumptions**:
  - Access to reliable, complete citation databases and publication sources.
  - Compatibility with scholarly databases and citation management tools.
- **Dependencies**:
  - Graph Database for citation network storage and queries.
  - Machine Learning and NLP **Libraries** for model training, text analysis, and citation context extraction.
  - Generative AI Models for citation pattern simulation and anomaly detection.

3.5 Risks

Potential risks include:

- Data Quality and Availability: Incomplete or inconsistent data affecting model accuracy.
- Model Bias: Possible bias due to lack of representation across diverse research fields or demographics.
- Scalability and Performance: Increasing complexity with large data volumes, requiring efficient computational resource allocation.
- Interpretability: Ensuring that detected anomalies and citation patterns are interpretable and explainable to users.

4. Functional Requirements

1. Data Collection and Preprocessing: Ingest citation data from the Aminer database, retaining essential fields (title, abstract, references).
2. Network Construction and Loop Detection: Constructs a citation network and identifies self-citation loops using graph random walks.
3. Summarization Pipeline:
    - Page-Level Summarization: Summarizes individual pages using BERT.
    - Aggregate Summarization: Summarizes entire papers using Gemini.
4. Citation Context Extraction: Extracts relevant citation context by isolating text around citation markers.
5. Classification Module: Classifies citations as essential or non-essential using LLMs fine-tuned with a rule-based approach.
6. Reporting and Visualization: Provides clear, interpretable reports of detected anomalies for investigation.

5. External Interface Requirements

5.1 User Interfaces

- Citation Analysis Dashboard: Displays metrics, citation patterns, and flagged anomalous citations.
- Detailed Anomaly View: Visualizes specific anomalous citation loops, involved authors, and citation context.

5.2 Hardware Requirements

- High-Performance Clusters: For model training and inference tasks.
- Distributed Storage: For citation data and network information storage.
- User Workstations: Accessing dashboard, visualizations, and insights.

5.3 Software Requirements

- Machine Learning Frameworks: (e.g., TensorFlow, PyTorch) for model training and fine-tuning.
- Graph Database: (e.g., Neo4j) for citation network storage and querying.
- NLP Libraries: For processing and extracting citation context.
- Generative AI Models: For simulation and anomaly detection in citation networks.

5.4 Communication Interfaces

- REST APIs: For integration with scholarly databases.
- Secure Data Transfer Protocols: Ensuring secure data exchange between components.

6. Non-Functional Requirements

6.1 Performance Requirements

- Training and Inference Efficiency: Model training and inference on citation data must be performant.
- Real-Time Processing: Near real-time analysis of new citations.
- Scalability: System should handle increasing citation data volume effectively.

6.2 Safety Requirements

- High Availability: Minimizing downtime for continuous anomaly detection.
- Fault Tolerance: Redundant system design for uninterrupted operation.

6.3 Security Requirements

- Data Privacy and Confidentiality: Robust encryption and access controls for citation data.
- User Access Control: Role-based access to secure sensitive information.

7. Other Requirements

- Interpretability: Clear, interpretable output explaining detected anomalies and reasoning.
- User Experience: Intuitive interfaces for efficient data exploration and analysis.
- Integration and Interoperability: Seamless integration with scholarly databases through defined APIs.
- Compliance and Auditing: Maintain ethical standards for citation analysis and reporting.

System Design:

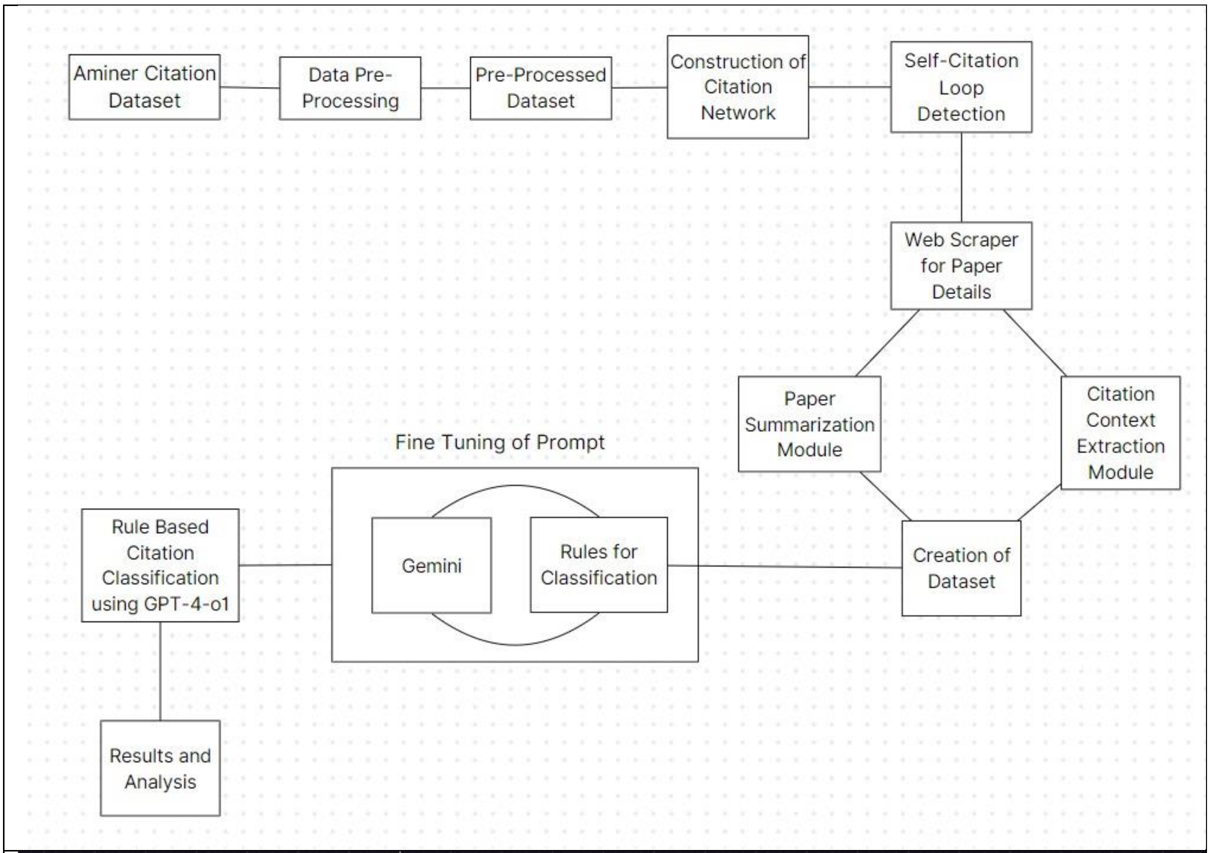
Design Goals

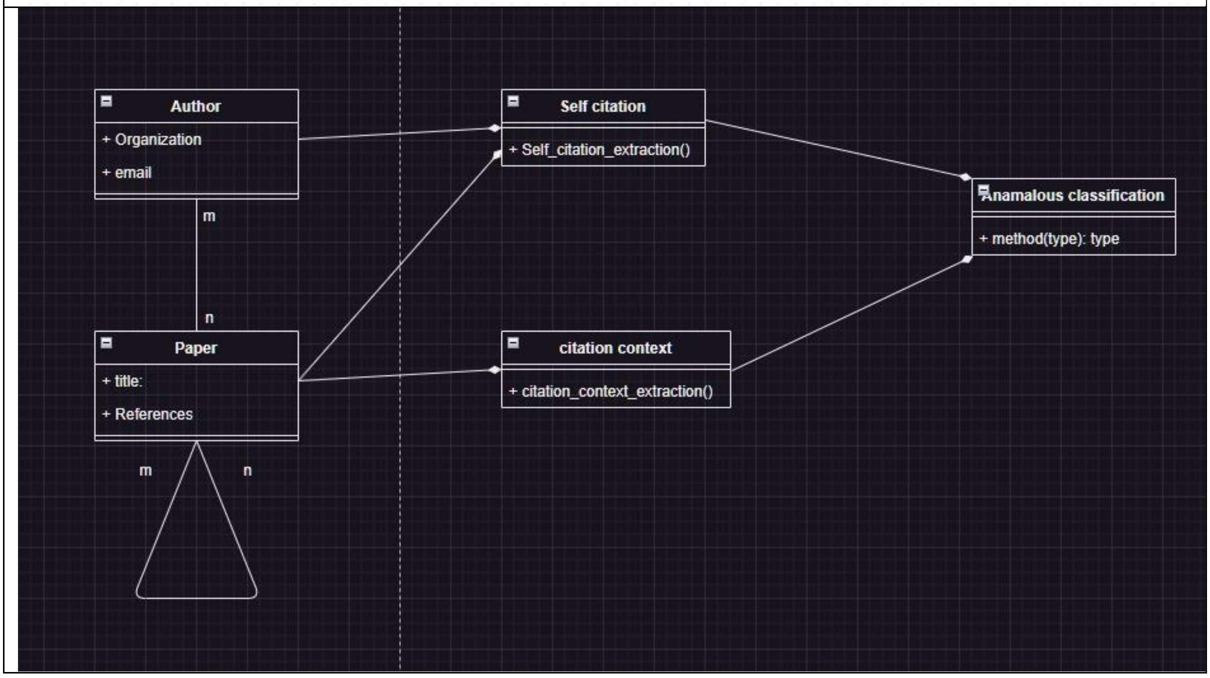
Performance:

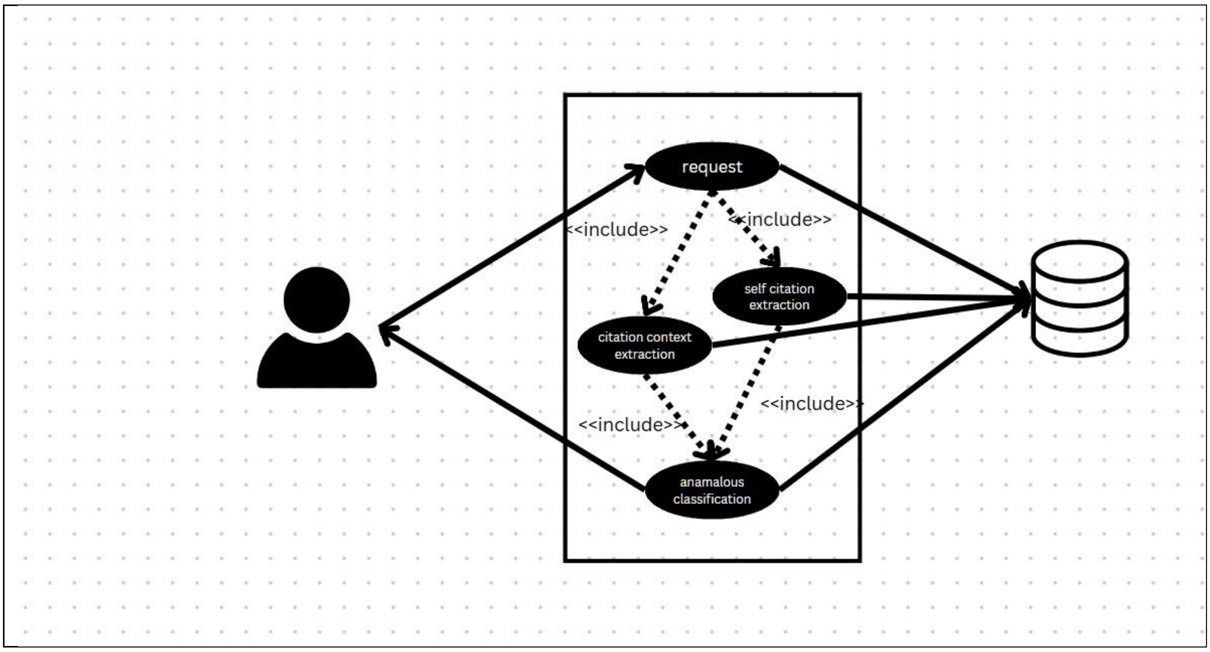
• Handling large-scale citation networks with millions of papers and relations requires optimizations in data processing, graph construction, and model training pipelines. Parallel computing, distributed processing, and efficient data structures and algorithms may be needed to ensure high performance.

Reliability:

• The system should provide consistent and reliable results, as its outputs may have significant implications for researchers and academic institutions.


Aminer Citation Dataset
Data Pre-Processing
Pre-Processed Dataset
Construction of Citation Network
Self-Citation Loop Detection
Web Scraper for Paper Details
Paper Summarization Module
Citation Context Extraction Module
Creation of Dataset
Fine Tuning of Prompt
Gemini
Rules for Classification
Rule Based Citation Classification using GPT-4-o1
Results and Analysis


Author
+ Organization
+ email
m
n
Paper
+ title:
+ References
m
n
Self citation
+ Self_citation_extraction()
citation context
+ citation_context_extraction()
Anamalous classification
+ method(type): type



Proposed Methodology

Data Collection and Preprocessing

- Data Source: The dataset originates from Aminer, a large-scale academic citation database containing metadata for scholarly papers, including titles, abstracts, authors, and reference lists.
- Data Filtering and Cleaning:
    - Key Fields Extraction: We keep only the essential fields: paper title, abstract, and references.
    - Noise Reduction: Remove any non-relevant or noisy data, ensuring clean metadata. This step is crucial for accurate citation network modeling.
    - Formatting for Consistency: Standardize field formats to ensure uniformity in references, author names, and citation structure, which simplifies subsequent network analysis.

Citation Network Construction

- Building the Citation Graph:
    - Nodes and Edges: Nodes represent individual papers, and directed edges represent citation relationships. Each directed edge from Paper A to Paper B indicates that Paper A cites Paper B.

- Self-Citation Definition: Self-citations are identified by author overlap between a citing paper and a cited paper. Self-citation loops are then defined as sequences where an author or group of authors repeatedly cite their own prior work.

- Self-Citation Loop Detection Using Graph Random Walks:
  - Random Walks of Length 3: To identify potential self-citation patterns, perform random walks of length 3 on the graph. These three-step paths allow the detection of citation "loops" where an author may cite themselves multiple times across different papers.
  - Self-Citation Loop Criteria: Loops where all papers have overlapping authorship are flagged, indicating potential self-citation patterns. These loops form the initial pool of citation instances for further analysis.

Web Scraping for Full-Text Access

- Scraping Target Papers in Loops:
  - For each paper identified in a self-citation loop, a scraper retrieves the paper's full text from online sources.
  - This full-text access allows deeper content analysis beyond abstracts and enables comprehensive understanding through summarization.

Summarization Pipeline for Contextual Understanding

- Two-Level Summarization Approach:

Page-Level Summarization with BERT:

Summarize each page of a paper separately to extract key points from the full text.

This segmentation ensures that no critical detail is overlooked, and reduces the load for the second-level summarizer by providing concise summaries per page.

- Aggregated Summarization with Google's Gemini:

After obtaining page-level summaries, apply Gemini to summarize across these summaries.

This step produces a cohesive summary for the entire paper, capturing core arguments, contributions, and contexts in a concise format, suitable for comparing against citation contexts.

Citation Context Extraction

- Extracting Citation Context:
  - For each citation instance, extract four lines of text preceding and following the citation bracket ([]), which is often used to mark references in scientific papers.
  - This "context window" provides enough surrounding text to capture the author's intent and the citation's purpose (whether it supports the current work or simply acknowledges prior work).

  - These citation contexts are key data points for training the classification model.

Definition and Fine-Tuning of Essential and Non-Essential Citation Rules

- Rule Development for Citation Classification:
  - Initially, manually analyze citation instances to develop a set of rules that distinguish essential citations (integral to the paper's argument or results) from non-essential ones (non-crucial self-references or citations that do not add value to the work).
  - These rules cover aspects such as:
    - Importance of Citation: Whether the citation substantiates a key finding or background information.
    - Novelty or Redundancy: If the citation repeats prior work without new insights.
    - Author Self-Referencing: Instances where the author appears to self-cite extensively without clear justification.
- Fine-Tuning the Rules Using Few-Shot Learning with Gemini:
  - Few-shot learning is employed to "teach" Google's Gemini using examples from the dataset.
  - The fine-tuning process involves feeding a curated set of citation contexts labeled manually as essential or non-essential. The model learns to generalize these examples, effectively applying the rules across new, unlabeled contexts.

Classification Pipeline for Anomaly Detection

- Input Preparation:
  - Each input to the model includes the citation context, the summary of the citing paper, and the summary of the cited paper.
- Models Used:
  - Gemini: Used to fine tune the initial set of rules based on a few anomalous self-citations found manually
  - GPT-4-o1: Used to classify the self-citations as essential or non-essential based on the rules which were fine-tuned by Gemini

Implementation:

Data sampling Steps:

In this data sampling step, we identify the k-hop neighborhood of a given node in a citation network and extract the subgraph containing nodes and their relationships about a particular domain.

Function hop_neighbour(node, k)

1. Find all nodes within `k` hops from the given `node`:

    a. Use `single_source_shortest_path_length` to calculate the shortest path from `node` to all other nodes in the graph `G`, up to a maximum path length of `k`.

    b. Store these nodes and their distances from `node` in `k_hop_neighborhood`.

2. Extract the list of nodes from `k_hop_neighborhood`.

3. Return the list of nodes within `k` hops of `node`.

End Function

Function k_hop_subgraph(G, node, k)

1. Call `hop_neighbour(node, k)` to get a list of nodes within `k` hops from `node`.

2. Create a subgraph from `G` that includes only the nodes found in the `k`-hop neighborhood.

3. Return the subgraph containing nodes within `k` hops of `node` and the edges between them.

End Function

Detecting Self Citation Loops:

This code identifies self-citation loops for each author in a citation network. A self-citation loop occurs when an author or group of authors cities their own work in a cycle, creating a closed path in the citation graph.

1. Calculate adjacency matrix `A` for graph `G`.

2. Compute `A_squared` by multiplying `A` with itself, representing 2-hop paths between nodes.

3. Compute `A_cube` by multiplying `A_squared` with `A`, representing 3-hop paths between nodes.

4. Initialize empty list `self_citing_datas` to store self-citation cycles for each author.

5. For each author `i` in the list of authors:

    a. Find all cycles involving node `i` using the `find_cycles_for_node` function.

    b. Initialize empty list `flag` to store self-citation cycles for author `i`.

    c. For each cycle `j` in the list of cycles:

        i. Locate the index of author `i` within cycle `j`.

        ii. Rotate cycle `j` so that it begins with author `i`, using the `rotate_list` function.

iii. Append the rotated cycle to `flag`.

d. Convert `flag` into a DataFrame `df1` with columns ['author', 'source', 'destination'].

e. Append `df1` to `self_citing_datas`.

6. Concatenate all DataFrames in `self_citing_datas` into a single DataFrame

7. Reset the index of the final concatenated DataFrame and store it back in `self_citing_datas`.

8. Return or use `self_citing_datas` for further analysis.

Summarization Module:
This module summarizes the content of a research paper using a two-level approach. First, each page is summarized using a BART model. Then a higher-level summary of the document is generated by summarizing the combined text.

Initialize and load models:

1. Load BART summarization model and tokenizer from pretrained models.

2. Configure Google Gemini API for advanced summarization.

Function summaries(text):

1. Define prompt for Gemini model to summarize the input `text` in 500 words.

2. Call Google Gemini model with prompt.

3. Wait briefly (e.g., 13 seconds) to ensure compliance with API rate limits.

4. Return the summary text from the response.

Function level1(path):

1. Open PDF file at the specified `path`.

2. Initialize empty string `summary` to store combined page summaries.

3. For each page in the PDF:

a. Extract text from the page and remove line breaks.

b. Tokenize text for BART model, truncating to model's maximum length.

c. Generate a summary for the page using BART, with beam search for higher-quality summaries.

d. Decode the summary tokens and add it to the `summary` string.

4. Return the final combined `summary` text, which includes summaries from all pages

End Functions

Classification Code:

This module identifies and labels citations in a text as either “Essential” or “Non-Essential” to the passage’s main contributions or findings.

Initialize API configurations:

1. Configure Google Gemini and OpenAI API keys.

Function lol_prompts(text):

1. If `text` is of type float (e.g., NaN), return an empty string since it is not valid input.

2. Define the prompt for Google Gemini:

   a. Explain the classification criteria:

      i. "Essential": The citation directly influences the passage's main findings or provides significant insights.

      ii. "Non-Essential": The citation only acknowledges related work without impacting the main analysis

3. Append the input `text` (i.e., passage containing citation) to the prompt.

4. Call the Google Gemini model to classify the citation based on the prompt.

5. Wait for 10 seconds to manage API rate limits.

6. Return the classification label ("Essential" or "Non-Essential") from the model’s response.

End Function

Function anomalous_check(text):

1. Define prompt for OpenAI GPT-4:

   a. Specify the task of identifying all instances of a reference in the passage.

   b. Classify each instance as either "Essential" or "Non-Essential" based on its contribution to the passage’s findings.

2. Append the input `text` to the prompt.

3. Call OpenAI’s GPT-4 model to generate a response based on the prompt.

4. Extract and return the model’s classification, providing a single-word response: "Essential" or "Non-Essential".

End Function

Results and Discussion

Through the application of the developed citation network analysis and machine learning techniques, we were able to uncover several insights into the prevalence and characteristics of anomalous self-citation patterns in academic publications. The results of our analysis are summarized below:
a)

1. **Citation Context**

- **Reference [8]** is mentioned in a passage discussing various bounds on the Price of Anarchy for the KP model and its variants. The reference is cited alongside other works that also explore these bounds.

2. Summary of Paper

- **Paper Title**: *Nash Equilibria in Discrete Routing Games with Convex Latency Functions*
- This paper introduces discrete routing games as a model that combines aspects of the KP and Wardrop models, focusing on non-cooperative routing over parallel links. It examines the Price of Anarchy (PoA), representing the efficiency loss when users act selfishly rather than collaboratively.
- **Key Contributions**:
    - **Fully Mixed Nash Equilibria**: Analyses properties of fully mixed equilibria under conditions like identical users and arbitrary links.
    - **Price of Anarchy Bounds**: Establishes upper bounds for the PoA in discrete routing games.
    - **Pure Nash Equilibria**: Explores single-link strategies and presents the ComputeNash algorithm, which finds these equilibria for arbitrary link configurations.
    - **Complexity of Equilibria**: Investigates the NP-completeness of computing best and worst pure Nash equilibria.
    - **Algorithms**: Proposes the ComputeNash algorithm, which runs efficiently for large configurations using a greedy approach.
    - **Open Problems**: Highlights potential future work, such as finding a Polynomial Time Approximation Scheme (PTAS) for optimizing pure Nash equilibria.
- The study significantly contributes to understanding the interaction between mixed and pure strategies in routing, the PoA under varying conditions, and computational challenges in identifying optimal routes.

3. Results

- **Non-Essential Citation**: The citation to Reference [8] was classified as non-essential.
- **Reasoning**: The reference appears in a list of related works, providing general background on the Price of Anarchy bounds without directly contributing to the passage's primary analysis or critical insights.

4. Explanation

- **Role of Citation**: Reference [8] serves to acknowledge similar studies on PoA bounds in the KP model context without elaborating on specific methods or findings from the cited work. Its role is more supportive than integral, as it does not directly impact the passage's arguments or conclusions.

b)

1. Citation Context

- **Reference [9]** appears in a passage discussing various conjectures and studies related to the Fully Mixed Nash Equilibrium Conjecture within discrete routing games. It is mentioned alongside other references ([4], [8], [17], [19]) as part of a broader discussion on the topic.

2. Summary of Paper

- **Paper Title**: *Nash Equilibria in Discrete Routing Games with Convex Latency Functions*
- This paper presents a discrete routing model combining aspects of the KP and Wardrop models, focusing on non-cooperative routing among parallel links. Key contributions include analysis of the Price of Anarchy (PoA), representing efficiency loss in scenarios where users act independently rather than collaboratively.
- **Key Contributions**:
    - **Fully Mixed Nash Equilibria**: Analyses equilibria where users employ mixed strategies under various conditions, including identical users and arbitrary links.
    - **Price of Anarchy Bounds**: Establishes bounds for PoA in discrete routing games, providing insights into the model's performance under competitive conditions.
    - **Pure Nash Equilibria**: Examines equilibria where users choose single links, introducing an efficient algorithm (ComputeNash) for computing these equilibria.
    - **Best and Worst Pure Nash Equilibria**: Investigates computational challenges in finding optimal equilibria, proving NP-completeness for identifying the best and worst pure Nash equilibria.
    - **Open Problems**: Suggests areas for future research, such as developing a Polynomial Time Approximation Scheme (PTAS) for optimizing pure Nash equilibria.
- This paper enhances understanding of routing efficiency in decentralized systems by examining latency's role in PoA and computational complexities in finding optimal routes.

3. Results

- **Non-Essential Citation**: The citation to Reference [9] is classified as non-essential.
- **Reasoning**: The reference is part of a list of related studies, serving to acknowledge broader research on conjectures related to Fully Mixed Nash Equilibria without contributing directly to the passage's primary analysis.

4. Explanation

- **Role of Citation**: Reference [9] serves an acknowledgment role, recognizing similar studies without directly impacting the main findings. It provides context for the Fully Mixed Nash Equilibrium Conjecture discussion but is not central to the core arguments or conclusions of the passage.

c)

Citation Context
Reference [26] appears in a passage discussing the behaviour of algorithms (LocalHeur and DeltaHeur) in a Locally Minimum Cost Forwarding (LMCF) game. The passage specifically examines algorithmic performance in terms of stretch, cost deviation, and node choice, with Reference [26] used to explain the behaviour of the Shortest Path Tree (SPT) at higher powers.

Summary of Paper
Paper Title: On a Locally Minimum Cost Forwarding Game
This paper examines the Locally Minimum Cost Forwarding Game (LMCF) in wireless networks, where nodes selfishly select paths to minimize individual costs. The focus is on determining optimal Nash equilibria under various conditions. Key contributions include:

- **Existence of Nash Equilibria**: Demonstrates that Nash equilibria exist in LMCF, with minimum spanning trees as a notable example.
- **Price of Anarchy (PoA) and Stability**: Analyses efficiency losses from selfish behaviour in LMCF, showing a linear PoA under certain cost objectives.
- **NP-Hardness**: Establishes the NP-hard nature of finding optimal Nash equilibria and introduces the Delta Heuristic to address this.
- **Simulation Results**: Demonstrates that DeltaHeur generally outperforms the Minimum Spanning Tree (MST), making it useful for controlling network topology in wireless settings.

The paper emphasizes understanding node behaviour in LMCF as critical for designing efficient wireless networks and positions DeltaHeur as a practical tool for achieving this.

Results
Essential Citation: The citation to Reference [26] is classified as essential.

Reasoning: Reference [26] is integral to explaining a core theoretical concept that supports the analysis of LocalHeur's behaviour in the passage. It provides theoretical justification for SPT's performance in the LMCF game, tying directly to the authors' main analysis on algorithmic behaviour under powered conditions.

Explanation
Role of Citation: Reference [26] explains the observed behaviour of LocalHeur compared to DeltaHeur within the LMCF game framework. Its inclusion is essential for understanding LocalHeur's effectiveness, which is central to the primary findings.

Conclusion:

In this study, we proposed a method for classing citations into Essential and Non-Essential ones, utilizing advanced language models to evaluate the role of the citation in relation to the principal findings or contributions of the text. By interpreting the intent behind each reference, our method uses the semantics of the passage to identify citations necessary to validate the main argument or method of the paper, rather than those containing no key propositions. As a result, our work can distinguish Essential citations from those that appear as credits to the other paper(s). We demonstrated how our developed model can recognize the importance of citations in academic content. Its application could inform citation practices, and at least identify patterns that could indicate repeated and unreferenced self-citations as a candidate for dubious scholarly integrity. Overall, this classification framework offers a structured approach for analysing citation relevance and can contribute to improved understanding of citation behaviours within academic networks.

Future work

While this method effectively distinguishes essential citations from non-essential ones, several improvements can further refine the approach. One limitation is that the model currently does not account for redundant citations across publication timelines. Ideally, a paper should cite the most recent research relevant to its field. For example, if a paper published in 2015 covers similar content as a paper from 2005, which in turn builds upon a 1999 study, then only the 2015 paper should be cited. Future iterations could incorporate this temporal filtering to avoid redundant or outdated citations. Additionally, the current model does not handle citations that follow certain formats or regex patterns, such as those using "[Suri et al.]" citations. This format is common in specific academic styles, and handling these cases with customized regex-based recognition could improve classification accuracy. Future work could include adapting the method to process such citations effectively, enabling more nuanced and comprehensive citation analyses across diverse academic formats.