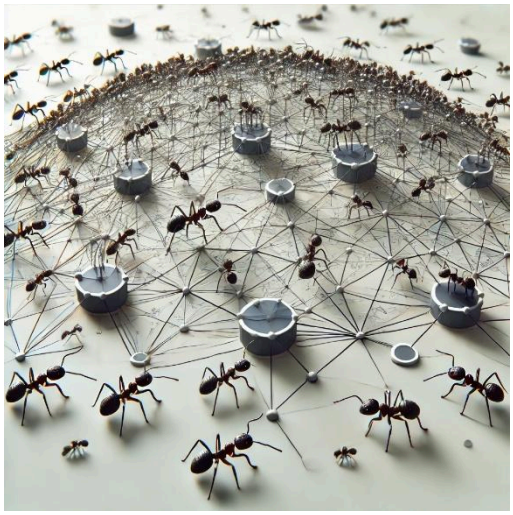
Citation Prediction using Ant Colony based Multi-Level Network Embedding



User Manual

Version 1.0

28/01/2025

Table of Contents

1. Introduction **3**

2. Getting Started **4**

3. Using the System **5**

3.1 Load Dataset 5
3.2 Configure Hyperparameters 6
3.3 Run Evaluation 8

4. Troubleshooting & Support **10**

4.1 Common Issues and Solutions 10
4.1.1 Application Becomes Unresponsive 10
4.1.2 Process Crashes or Errors During Execution 10
4.1.3 Difficulty Understanding Configuration Options 11

1. Introduction

This user guide provides detailed instructions for utilizing the **Citation Prediction using Ant Colony-based Multi-Level Network Embedding (ACE)** system, a cutting-edge tool designed to enhance the relevance and accuracy of academic citations through innovative graph embedding techniques.

The ACE system addresses the challenge of weak or irrelevant citations in academic papers, which can hinder the quality of research and the ability of readers to navigate vast amounts of academic data. By leveraging the principles of Ant Colony Optimization (ACO) and multi-level network embedding, the system creates hierarchical graph representations that accurately identify and evaluate citation relationships.

This manual will guide you through the application's functionalities and how to use them effectively.

2. Getting Started

To set up the environment, follow these steps:

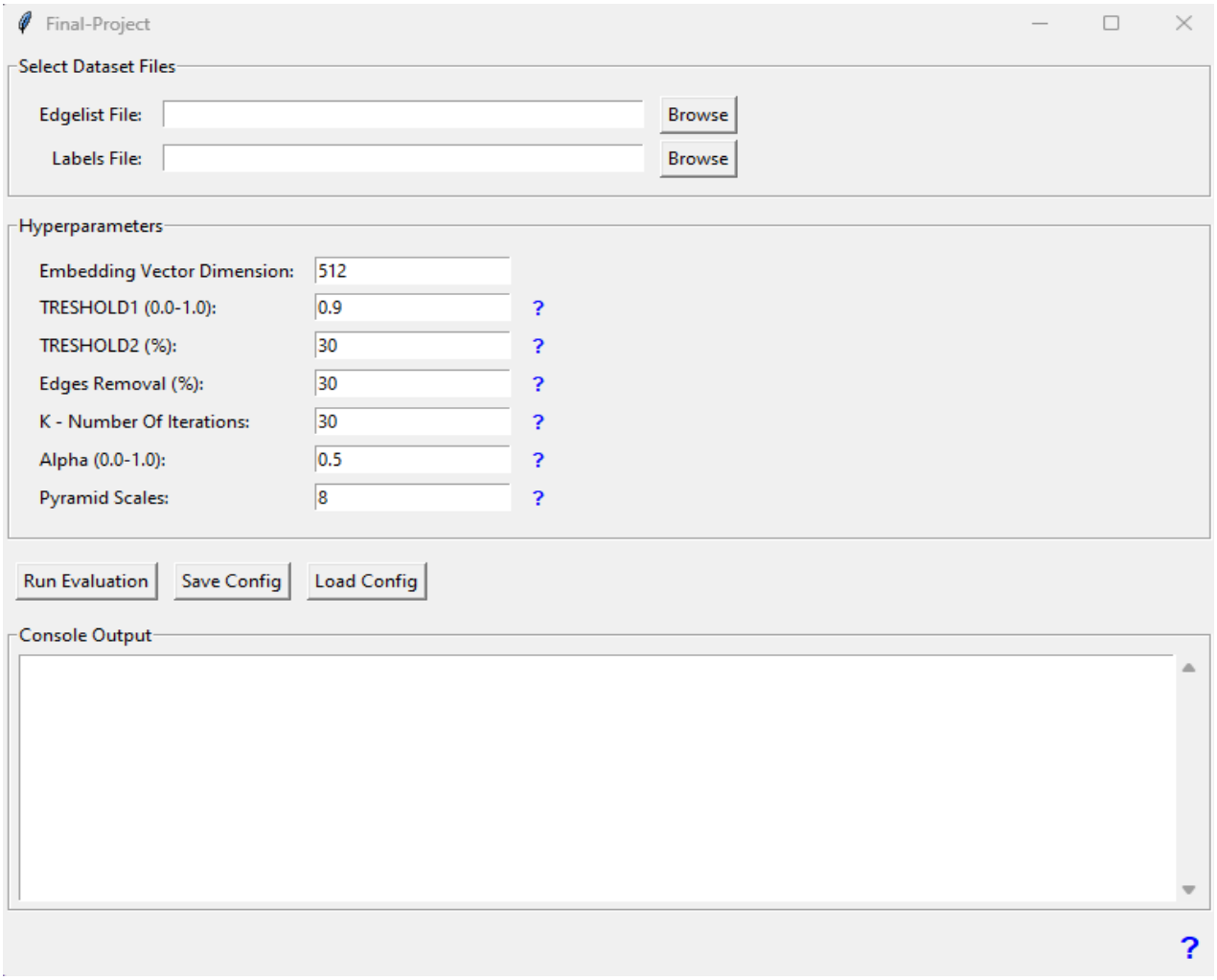
1. Open a terminal or command prompt.
2. Clone the repository from GitHub:

   $ git clone https://github.com/yoni4600/Final-Project.git
3. Open the project in Python IDE with python version 3.9.0:

   $ cd src
   $ pip install -r requirements.txt
4. Build:

   $ python setup.py build_ext --inplace
5. Open the UI:

   $ python UI.py

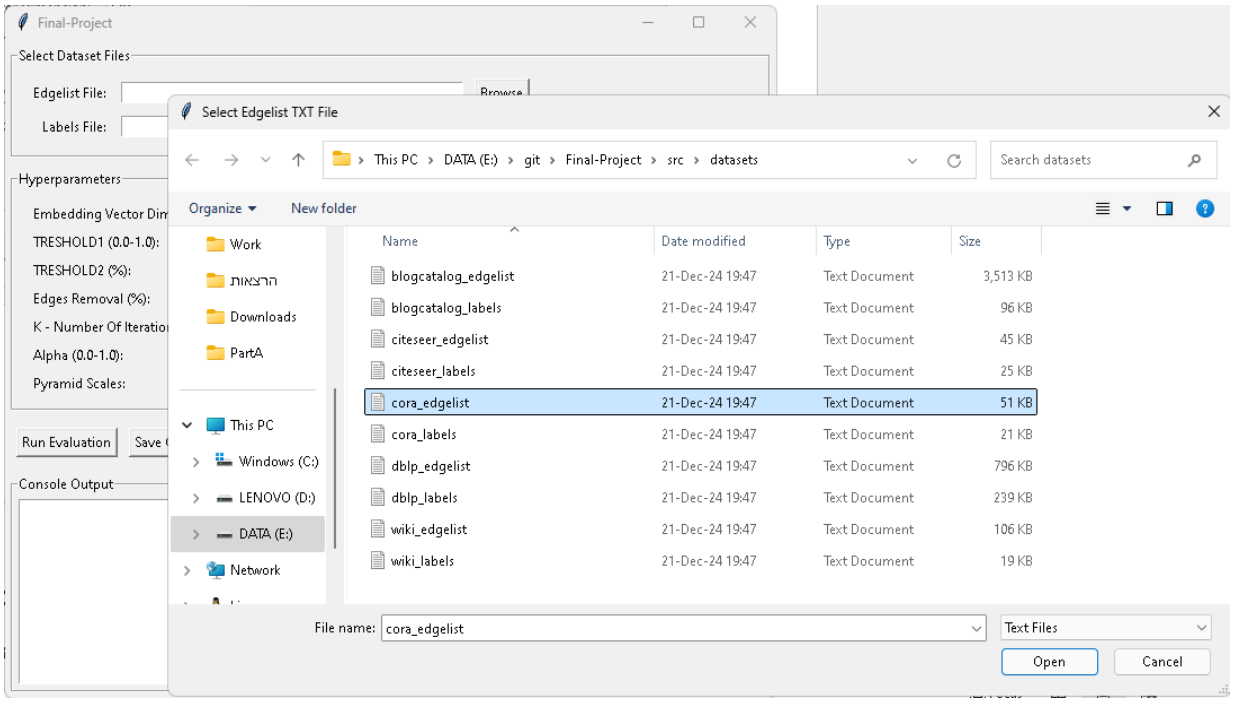


3. Using the System

3.1 Load Dataset

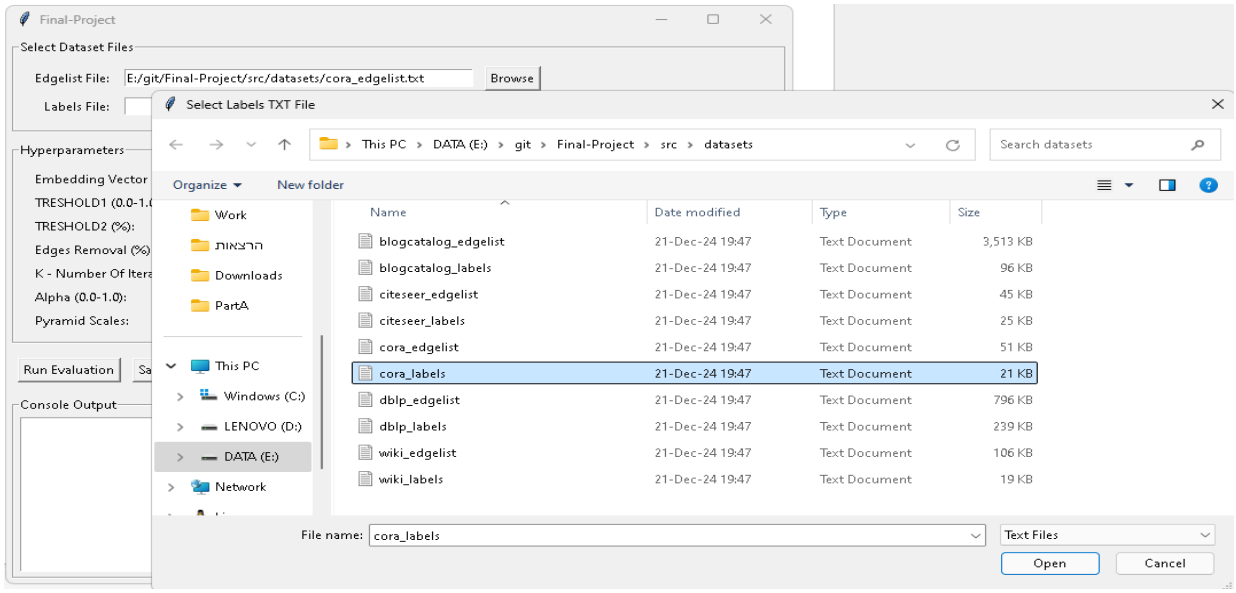
Load the graph's edge list:

Using the "Browse" button next to the Edgelist File label.

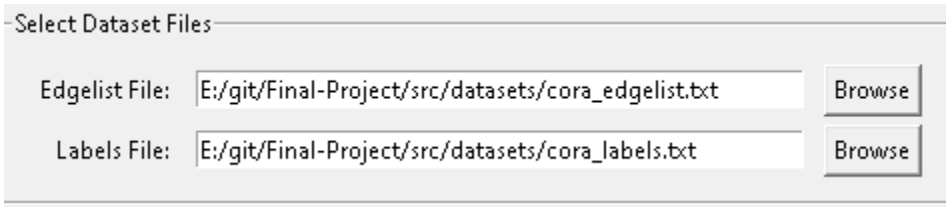


Load the graph's labels list:

Using the "Browse" button next to the Labels File label.



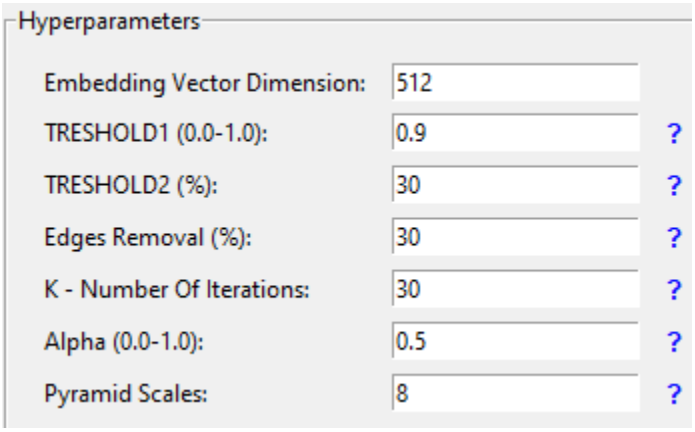
Final result:



3.2 Configure Hyperparameters

There are different ways to configure the Hyperparameters:

1. Select Hyperparameters for the current run manually:



Embedding Vector Dimension – The dimension of the embedding vector of a node in the graph.

Treshold1 – The threshold used to classify “strong” or “weak” relation of an edge for each iteration in the Research Plan.

Threshold2 – The threshold used to decide whether an edge is “strong” or “weak” after all K iterations.

Edges Removal – The percentage of edges removed in each iteration in the Research Plan.

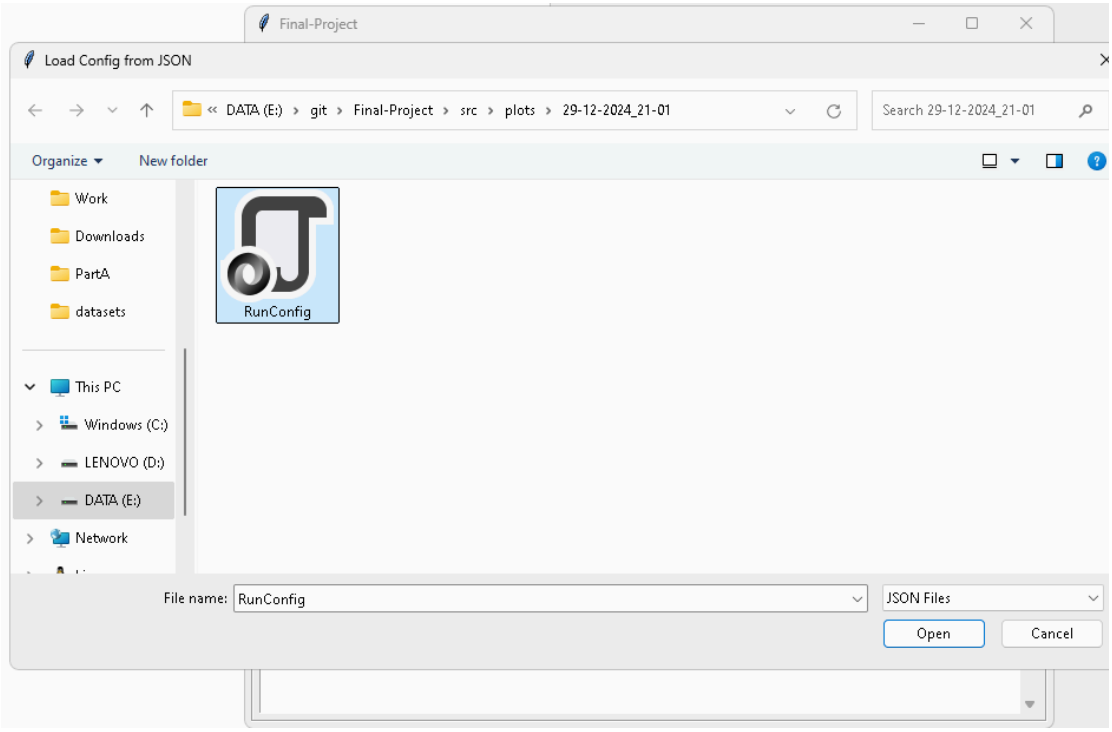
K – Number of iterations in the Research Plan.

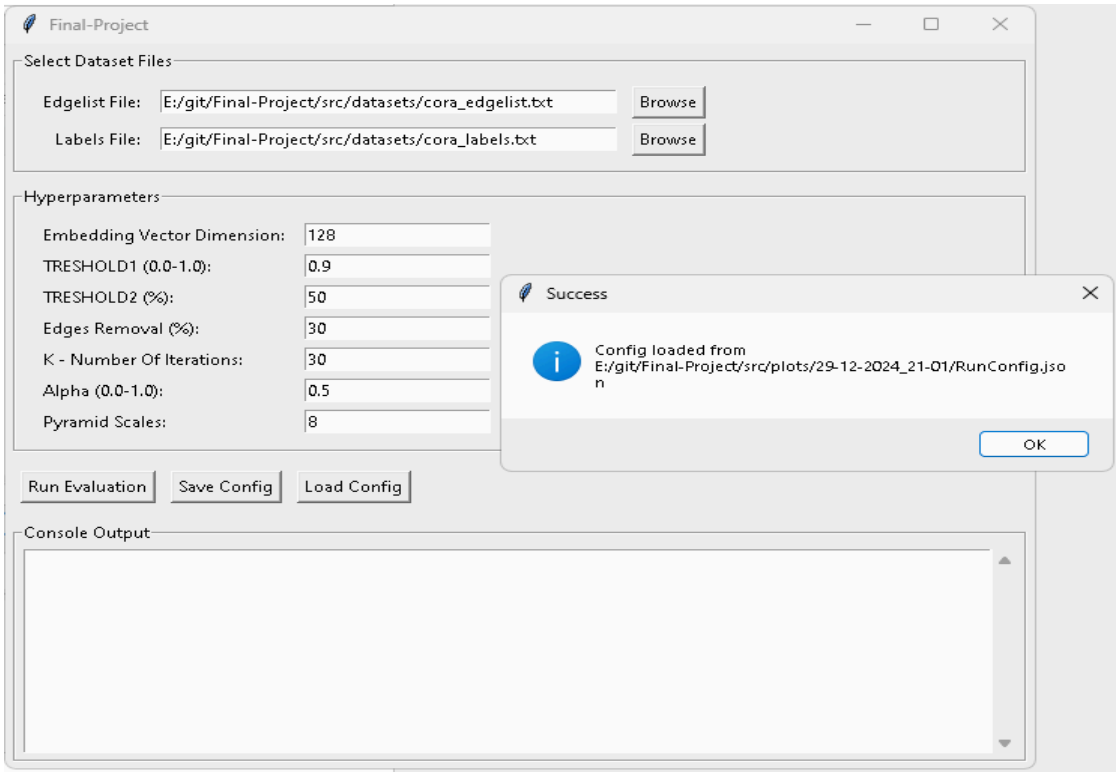
Alpha – Scaling factor used in the Ant colony transition probabilities formula.

Pyramid Scale – Number of graph layers that will be created by the Ant colony algorithm.

2. Load Hyperparameters configuration:

Using the “Load Config” button you can load an existing desired configuration from a .json file.





3. Save Configuration:

Using the “Save Config” button you can save current configuration as a .json file.

3.3 Run Evaluation

After everything is ready, using the “Run Evaluation” button we can run the Evaluation Plan on the selected Dataset with the current Hyperparameters configuration.

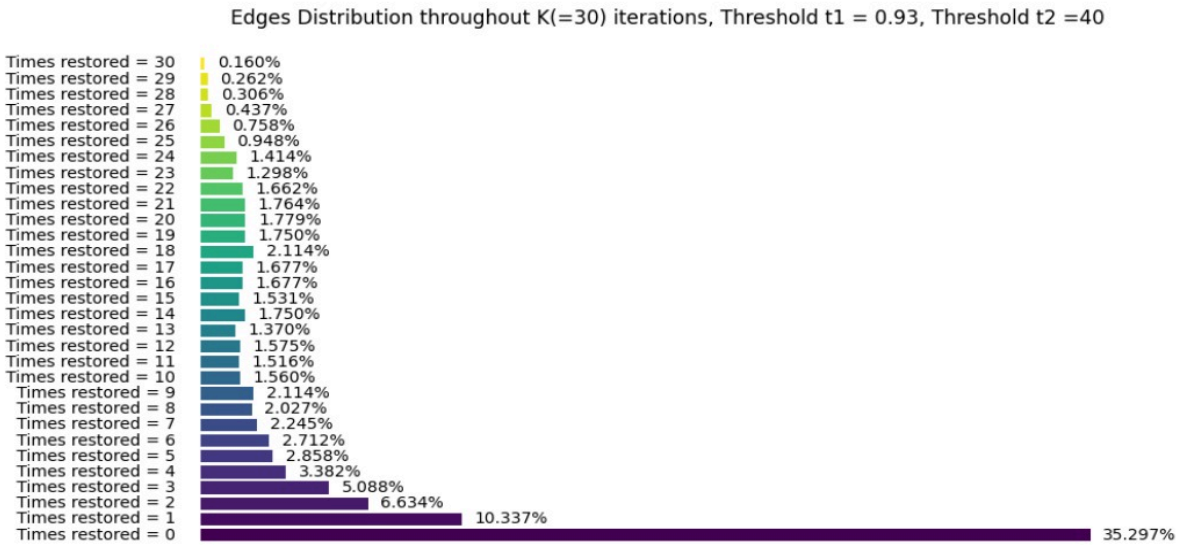
Every step in the process is documented in the Console Output within the UI:

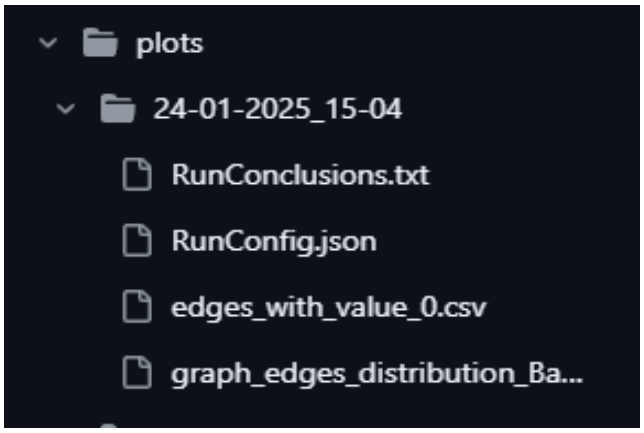
*Notice that the runtime of this program may be very long even with strong computational resources.

The process will run until it will display the statistics and conclusions:

In addition to the Console results you can find in the project directory after the process finished a new directory with a timestamp. Inside this directory you can find the configuration that was used in the specific run, a bar chart of the edges distribution and more other statistics.

For example:


Edges Distribution throughout K(=30) iterations, Threshold t1 = 0.93, Threshold t2 =40
Times restored = 30 0.160%
Times restored = 29 0.262%
Times restored = 28 0.306%
Times restored = 27 0.437%
Times restored = 26 0.758%
Times restored = 25 0.948%
Times restored = 24 1.414%
Times restored = 23 1.298%
Times restored = 22 1.662%
Times restored = 21 1.764%
Times restored = 20 1.779%
Times restored = 19 1.750%
Times restored = 18 2.114%
Times restored = 17 1.677%
Times restored = 16 1.677%
Times restored = 15 1.531%
Times restored = 14 1.750%
Times restored = 13 1.370%
Times restored = 12 1.575%
Times restored = 11 1.516%
Times restored = 10 1.560%
Times restored = 9 2.114%
Times restored = 8 2.027%
Times restored = 7 2.245%
Times restored = 6 2.712%
Times restored = 5 2.858%
Times restored = 4 3.382%
Times restored = 3 5.088%
Times restored = 2 6.634%
Times restored = 1 10.337%
Times restored = 0 35.297%


plots
24-01-2025_15-04
RunConclusions.txt
RunConfig.json
edges_with_value_0.csv
graph_edges_distribution_Ba...

4. Troubleshooting & Support

This section is designed to assist users in diagnosing and resolving issues that may arise while using the application. It covers common error messages, potential challenges related to hardware limitations, and tips to ensure smooth operation.

4.1 Common Issues and Solutions

Users may encounter various error messages while interacting with the application. Below, we identify common error messages, their likely causes, and possible solutions:

4.1.1 Application Becomes Unresponsive

- **Cause**: The process consumes high computational resources, and systems running on CPU-only hardware may struggle to keep up with the demands, causing the UI to freeze or turn black.
- **solution**:
  1. Stop the current process immediately.
  2. Quit the application and restart it.
  3. If restarting does not resolve the issue, try one of the following:
     - Switch to a system with more robust resources, such as one equipped with a GPU.
     - Adjust the hyperparameters in the application to reduce resource usage, such as lowering the number of iterations or decreasing the embedding vector size.

4.1.2 Process Crashes or Errors During Execution

- **Cause**: Insufficient system resources (e.g., limited CPU or memory) or unsuitable hyperparameter configurations.
- **Solution**:

  1. Verify that your system meets the recommended hardware requirements.
  2. Reduce the computational load by:
     - Lowering the number of iterations.
     - Reducing the embedding vector dimension.
     - Modifying other hyperparameters in the configuration menu.
  3. Ensure no other resource-heavy applications are running in the background.

4.1.3 Difficulty Understanding Configuration Options

- **Cause:** Users may be unfamiliar with the purpose of specific hyperparameters or features.
- **Solution**:

  1. Hover over the **‘?’** icons in the UI for detailed explanations of each setting.

  2. Refer to the user guide's section on **Hyperparameter Configuration** for additional details.