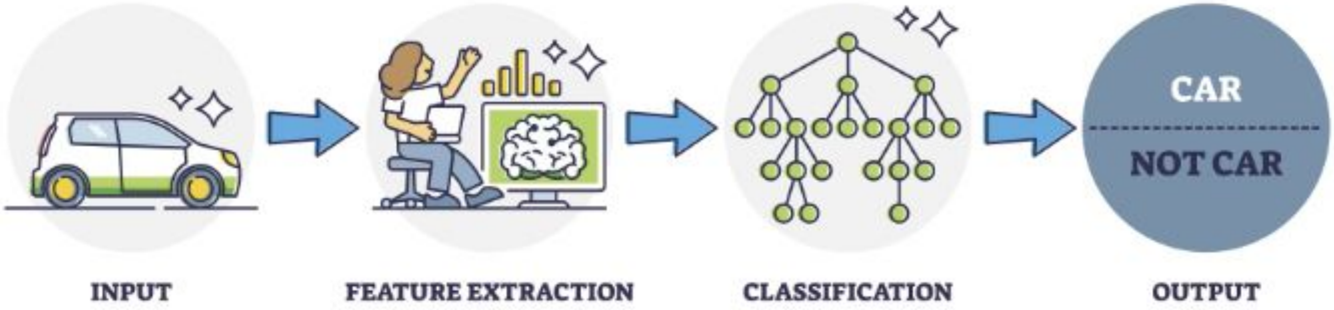
Unlearning Mechanisms in Graph Models and Document Classification

By Adam Ge and Aadya Goel (Mentor Mayuri Sridhar)
MIT PRIMES October Conference: October 12th, 2024

Introduction

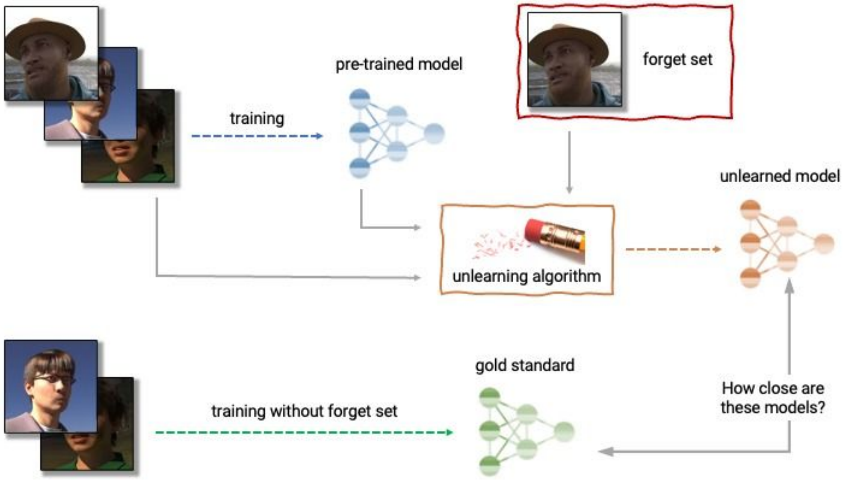
What is Machine Learning

- *Machine Learning* is the process of training a model, typically by tuning parameters, to make predictions



What is Machine Unlearning

- *Gold Standard*: Remove data to be unlearned and retrain the model
  - Can be too costly
- *Machine Unlearning* is the process of removing training data without retraining the entire model



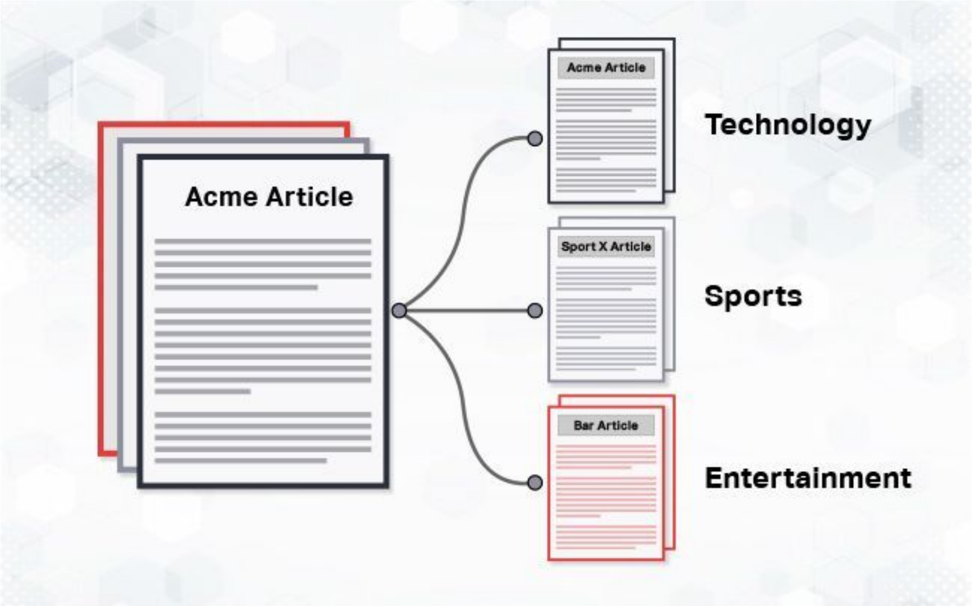
Why Machine Unlearning

- Training may use data collected from individuals, could be private/sensitive
    - Recent legislation mandated the erasure of personal data when requested
- Model maintenance may be necessary to remove false data

Document Classification

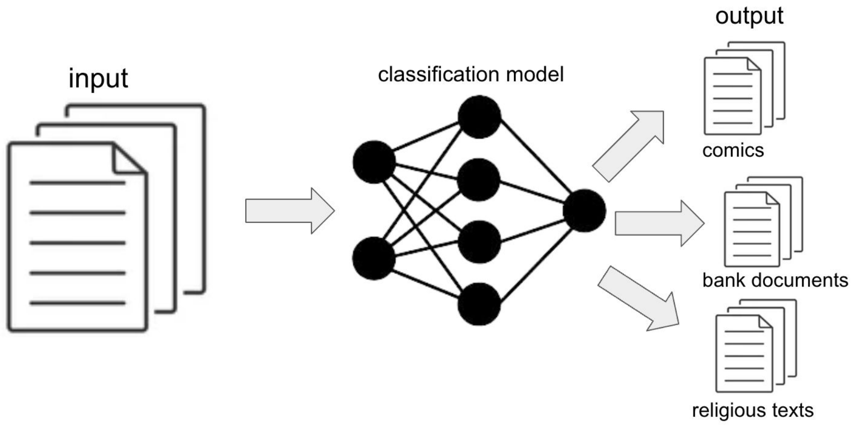
What is Document Classification

- Document classification is the process of assigning documents to different categories or classes



Document Classification Models

- Done through image classification or **text classification**
- Large task of NLP (Natural Language Processing), commonly done with *word embedding*
    - Process of representing words as a vector of real numbers
    - Vectors capture information about the words so those with similar meanings are near each other in the vector space



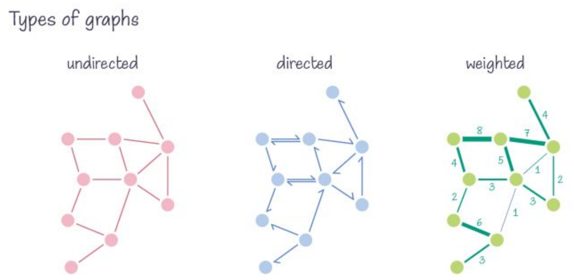
Unlearning Documents

- Removing an entire category/document label
- Current work randomly distributes documents within the remaining classes
- Sort the documents into the next top class possible
- Graph Models

Graph Models and Unlearning

Graph Theory

- Graphs are data structures where nodes/vertices represent entities and edges represent relationships between vertices
- Two nodes are *neighboring* if they are connected with an edge
  - The *neighborhood of a node is the set of all its neighbors*
- Machine unlearning of graphs is called *graph unlearning*
  - Includes *edge unlearning* and *node unlearning*



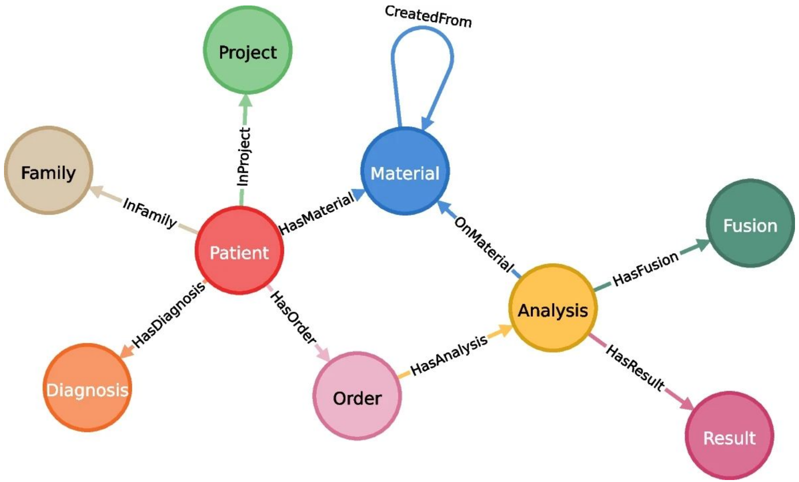
Why Graphs for Document Unlearning?

- Capture the structural information of a text
- Mitigate the effects of “curse-of-dimensionality”
- Helps represent the similarities between documents
  - Assess the importance of a word for a whole set of documents
- GNNs: Help capture complex patterns, improving classification
  - Finds the connection in content of the documents

Unlearning Mechanisms in Graph Models

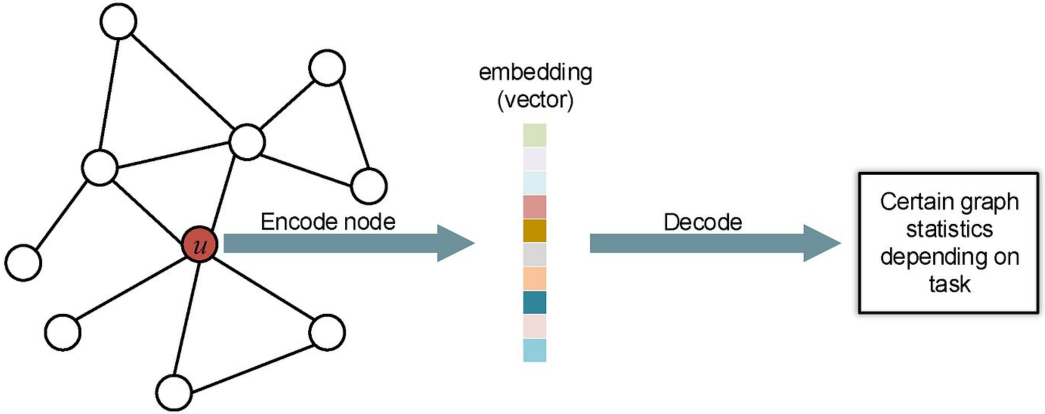
Social Network Graph

Heterogeneous Healthcare Graph



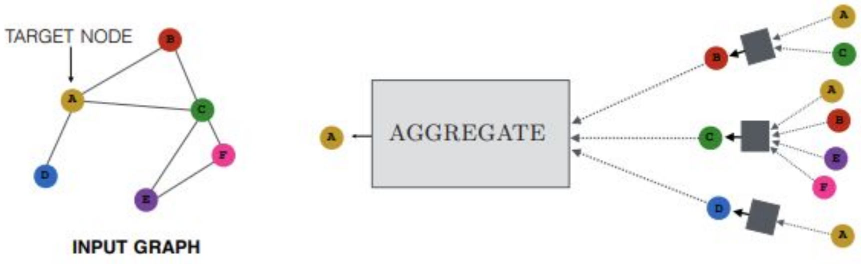
Graph Neural Networks

- *Graph Neural Networks (GNNs)* are neural networks that operate on graph-structured data
- Composed of multiple layers
- Each node has a *feature vector*, which represents its attributes



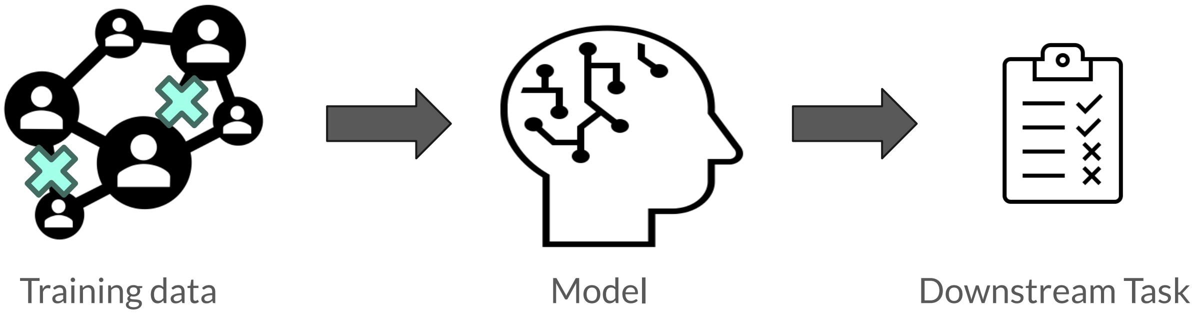
Graph Convolutional Networks

- Each node in a GCN sends its current feature information to its neighbors
- Aggregates them (e.g. by averaging) and applies a non-linear transformation to update the node's feature vector
- Done recursively for several layers



Graph Unlearning

- Sensitive data stored in graphs (attributes of nodes or edges)
- We focus on edge unlearning
    - e.g. friendships in social networks



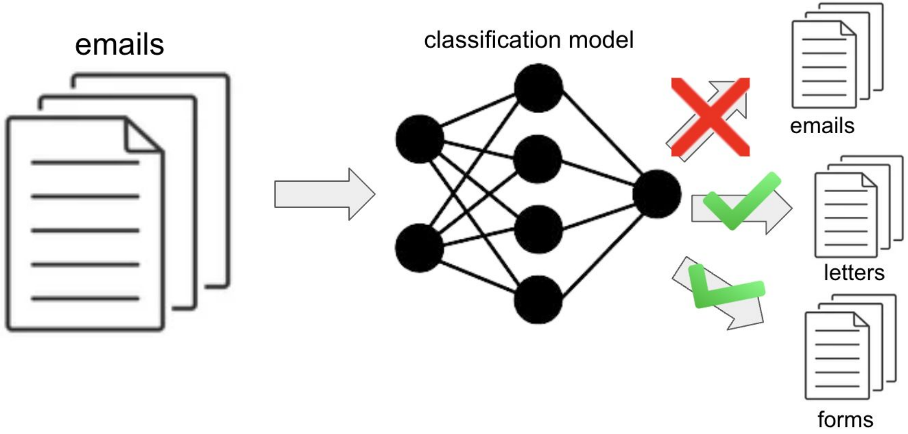
The Problem

The Adversary in Document Classification

- Gains access to document content but not the classification label
  - Finding the label can exploit potential vulnerabilities
- Cannot figure out the label for multiple reasons:
  - Redacted Information: sensitive sections are obscured, making interpretation hard without a machine
  - Technical Jargon: language is complex, making it hard for the adversary to understand
  - Volume of Data: large number of documents make manual reading impractical

Unlearning Classification Label Problem

- Consider a group of documents with sensitive data (e.g. bank documents)
- Resort this documents into other labels
  - Sensitivity is within the label (contents are unknown)
- Current methods randomly distribute
  - Can lead to decreased model utility
  - Falsely sort other documents as well



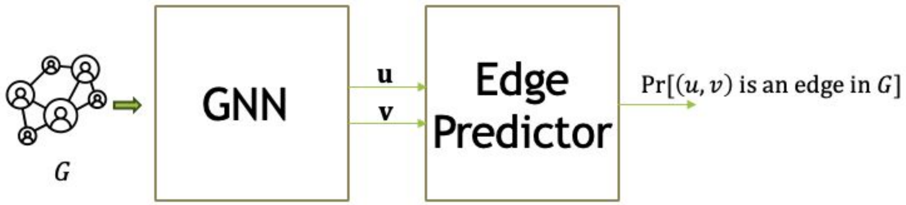
Problem with GNNDelete

- The paper *GNN Delete: A General Strategy for Unlearning in Graph Neural Networks* presents a method for graph unlearning from a trained GNN Model
    - *Deleted Edge Consistency (DEC) loss:* $\min[\mathbb{P}(e(u,v)) - \mathbb{P}(e(\text{nonexistant edge}))]$
- Changes the predicted probability of an edge being between *u* and *v* (endpoints of unlearned edge) to be about the same as between a random pair of nodes
    - Erases too much information, thus affecting model utility
    - In graphs with many communities, *u* and *v* may have many common neighbors, should have a higher predicted probability of an edge

Implementation and Solutions

Verifying Problem with GNNDelete

- Trained GCN Model with Cora Dataset but with 3 edges deleted, outputs node embeddings
- Embeddings are fed into an Edge Predictor model (multi-layer perceptron model) which trains the model
- Predicted probability of the deleted edges was much higher than that of a nonexistent edge.



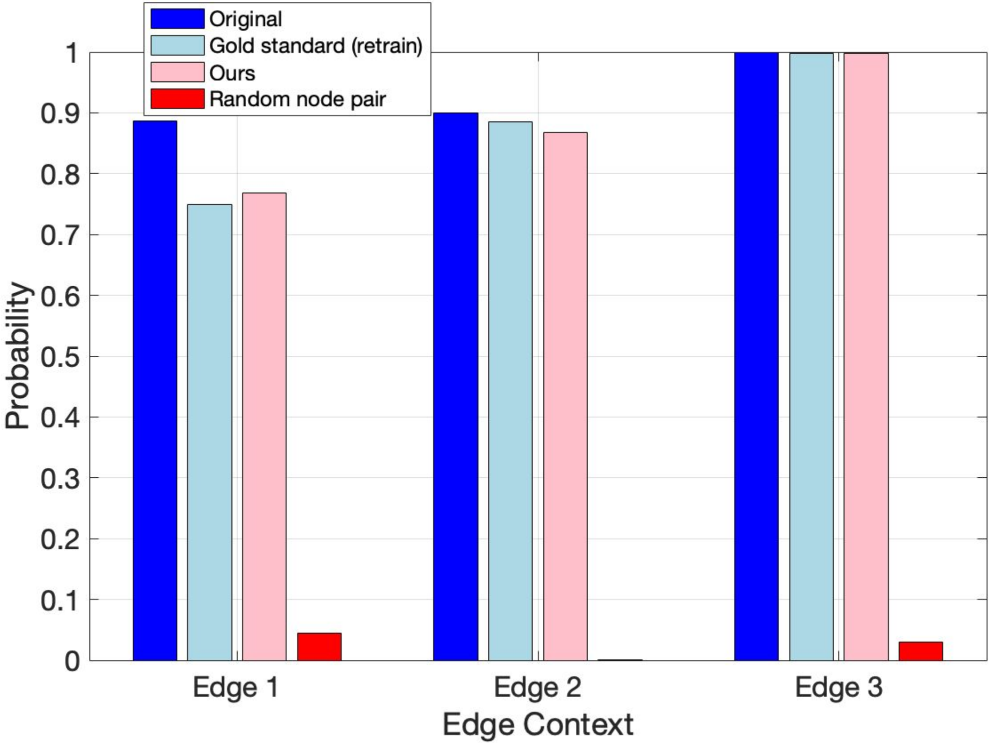
New Edge-Unlearning Model

- Train the GCN model and Edge Predictor model with entire Cora Dataset
- *(u, v)* is an edge to be unlearned, let the predicted probability be p
- We set the new value of p as $c*p$ for some constant $c < 1$, and do back propagation to modify the parameters of the GCN (while freezing the parameters of the Edge Predictor)

Resulting Probabilities

3 edges before unlearning	Gold standard probability of the 3 edges after unlearning	3 random non-existent edge probabilities
0.9387, 0.9096, 0.9993	0.8777, 0.6747, 0.9974	0.0006, 0.0010, 0.0158

Main Results On Edge Unlearning

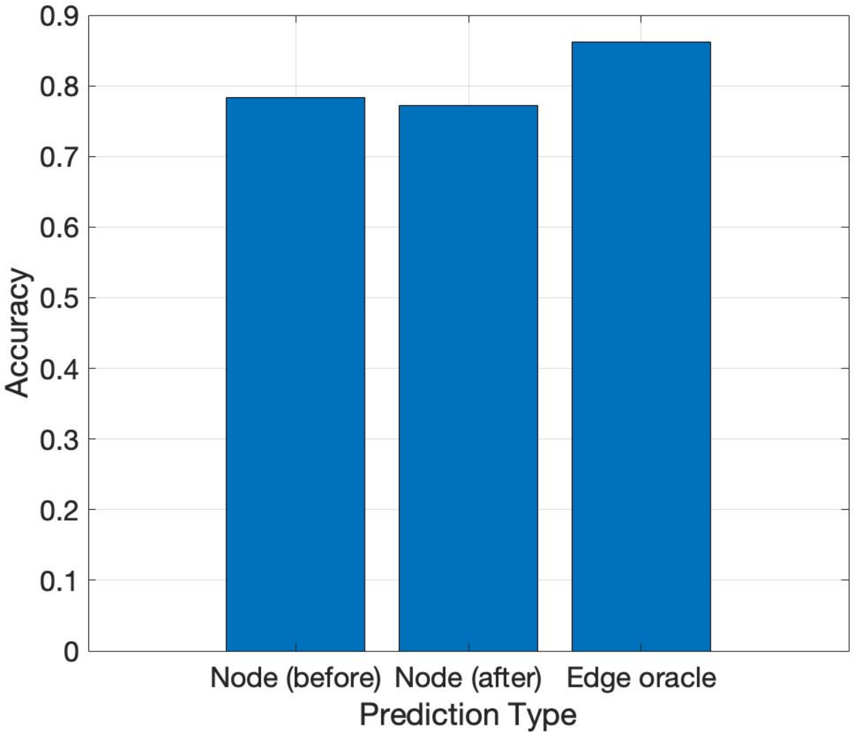


Predicting edge probability between the vertices of 3 unlearned edges

Edge probabilities in original model are close to 1, our methods get similar probabilities to gold standard

Edge probabilities of nonexistent edges are close to 0

Accuracy before and after edge unlearning



Node label predictions before and after edge unlearning

Accuracy of trained edge oracle predicting edge probabilities

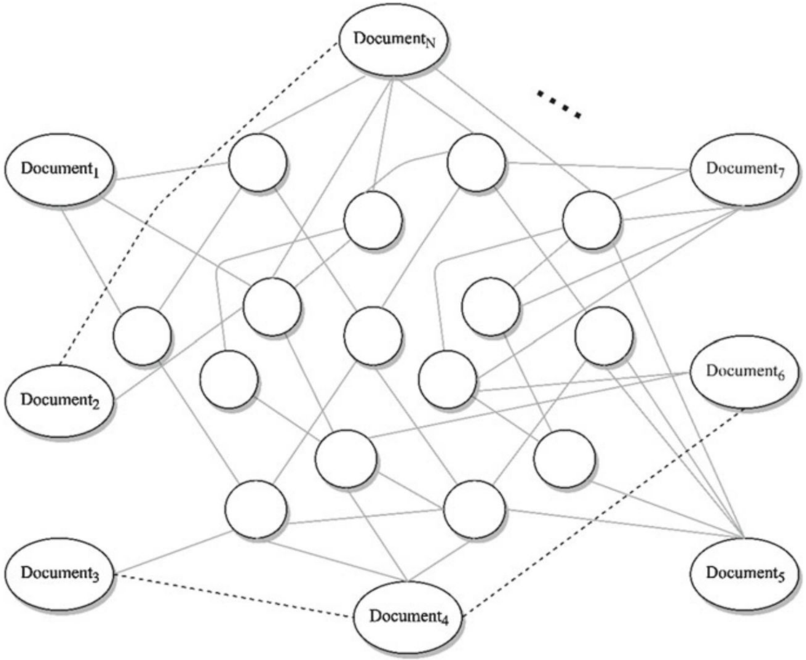
Graph Unlearning Mechanisms

- Edge unlearning makes sense for social network graphs
- In contrast, we can consider heterogeneous graphs for document classification

The Model

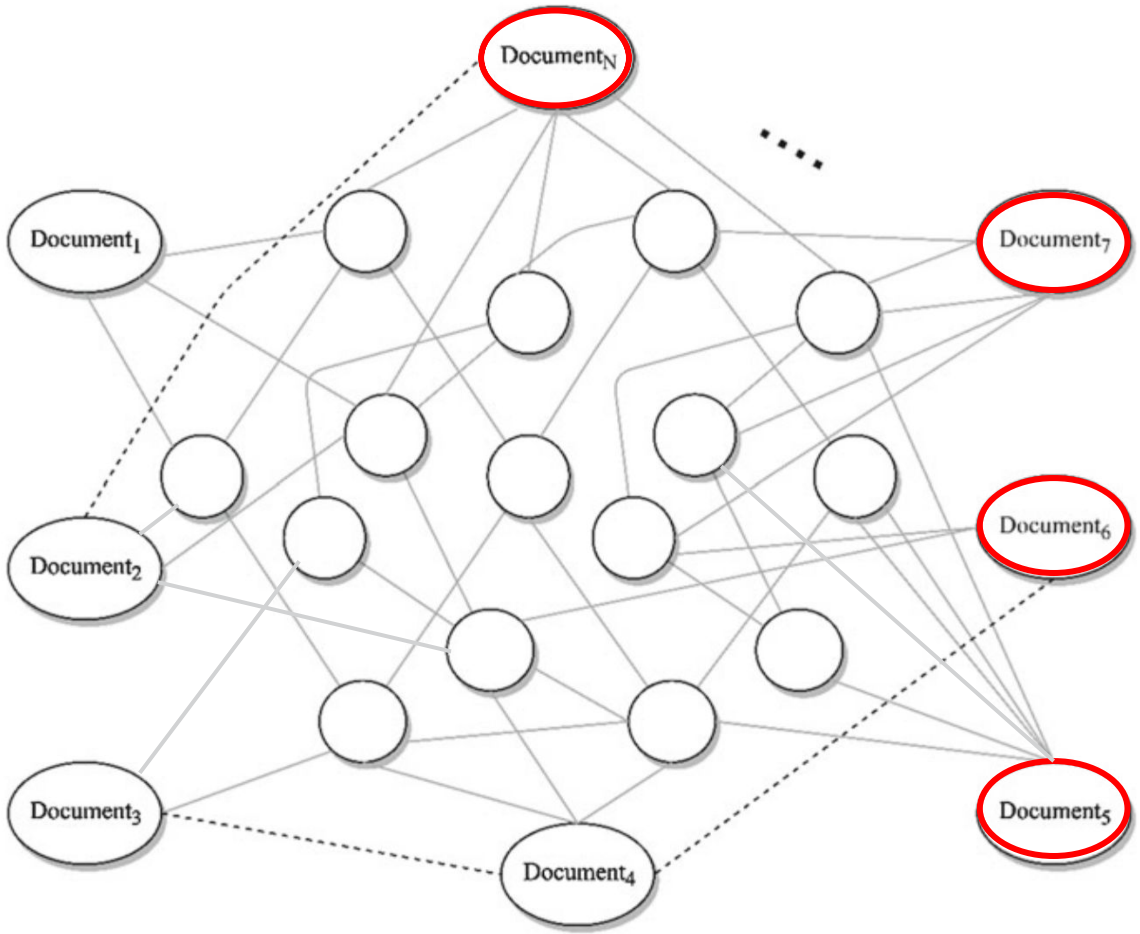
- “Graph of Docs” Model
  - Nodes: Documents and key words
  - Edges: “CONNECTS” “INCLUDES” “SIMILAR”
- Documents are split into “similarity subgraphs”

  → determines labels

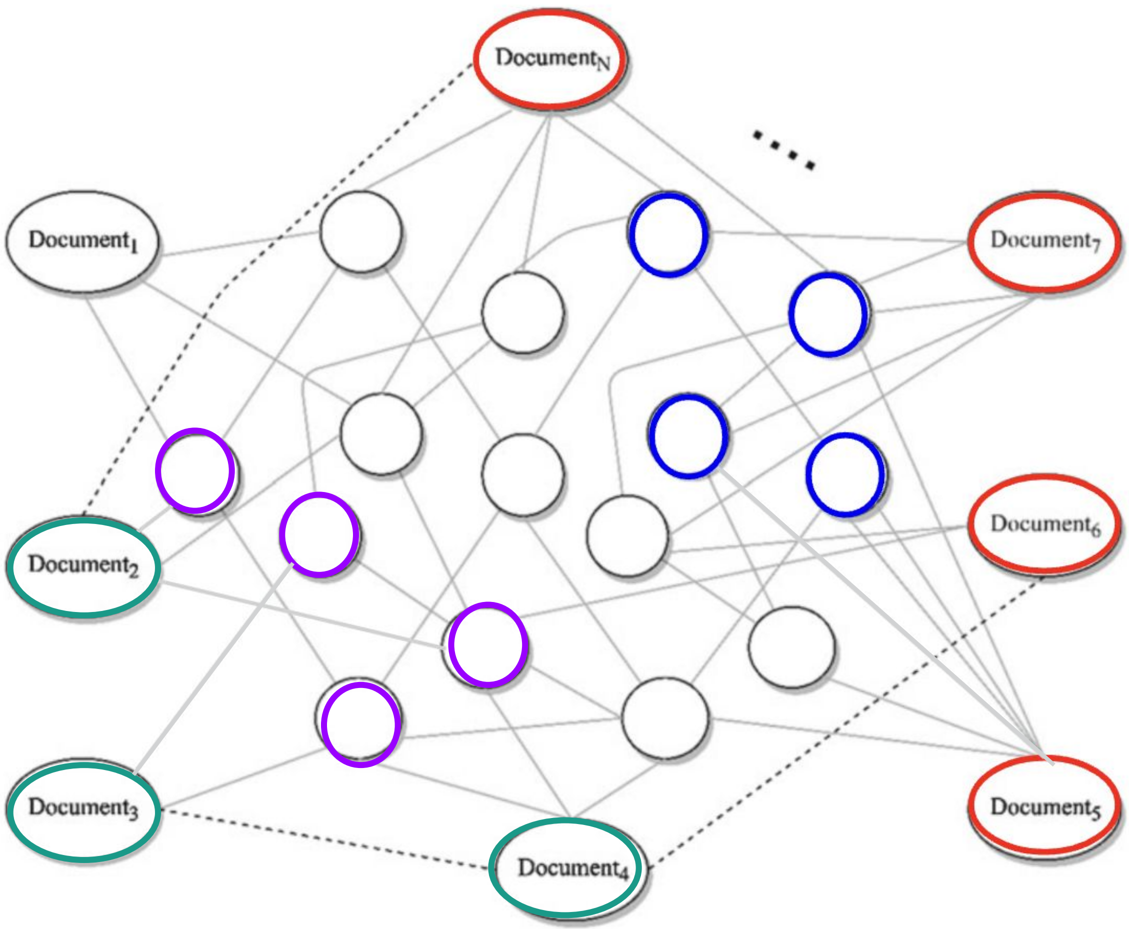


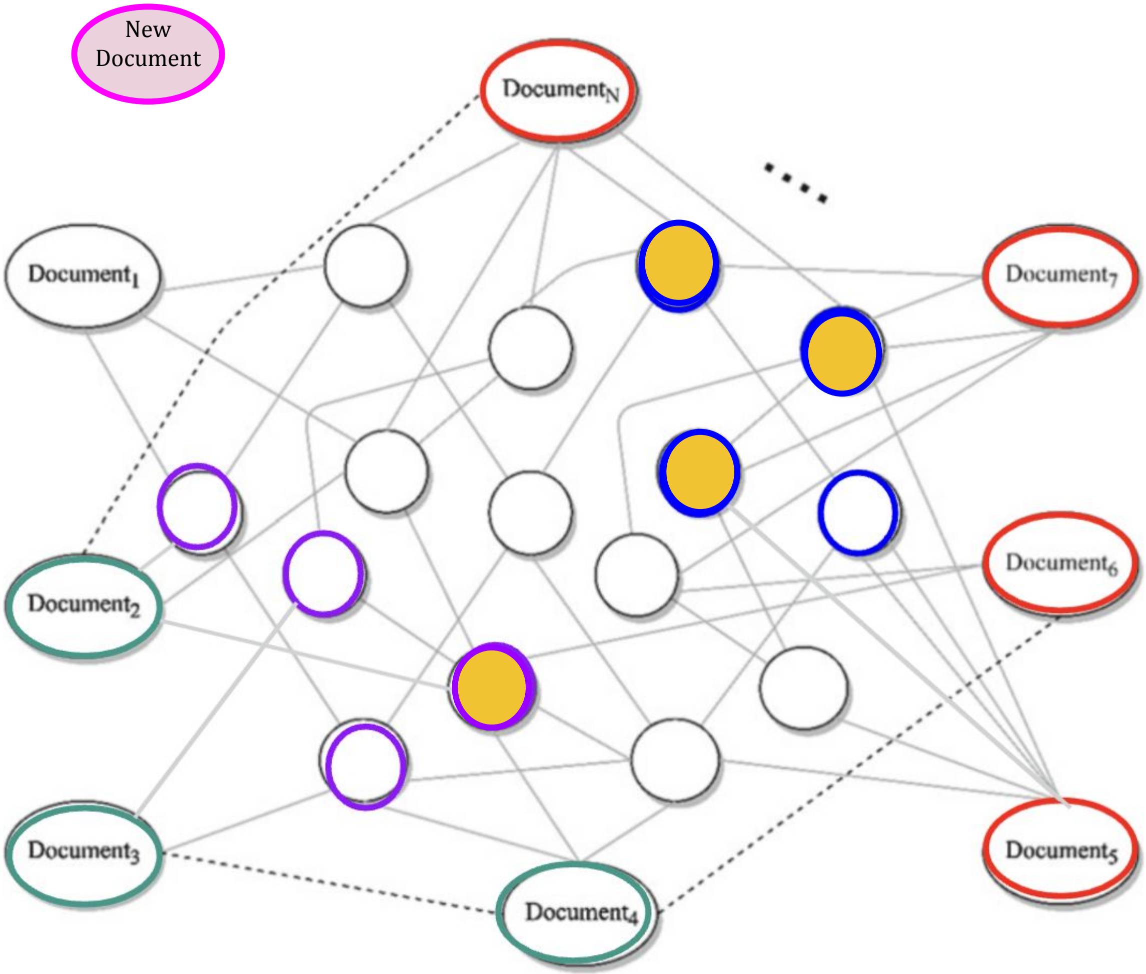
The Model

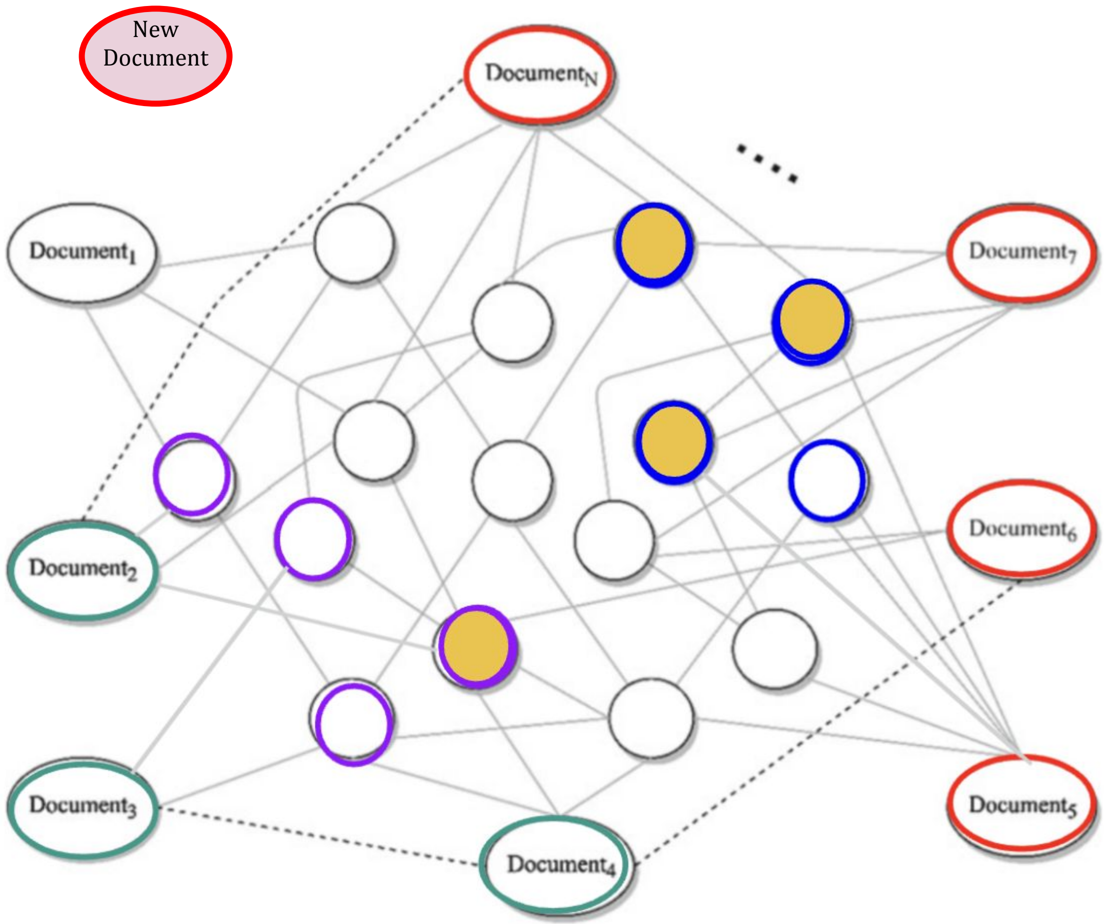
- 20 newsgroup dataset
- Word Importance and Document Connections
- Document Similarity Subgraph:
  - Uses similarity scores to create a subgraph where documents are connected via similarity relationships
- Communities of Similar Documents:
  - Creates categories of documents via Louvain algorithm
- Rank the importance of each class for each word node



Document$_N$
Document$_1$
Document$_7$
Document$_2$
Document$_6$
Document$_3$
Document$_4$
Document$_5$

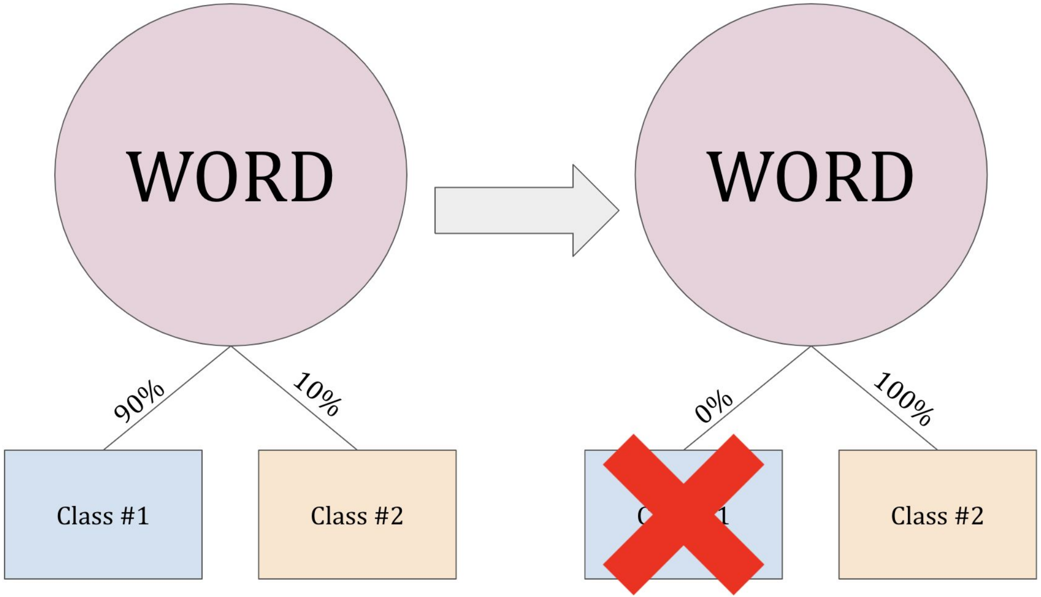

DocumentN
Document1
Document7
Document2
Document6
Document3
Document4
Document5

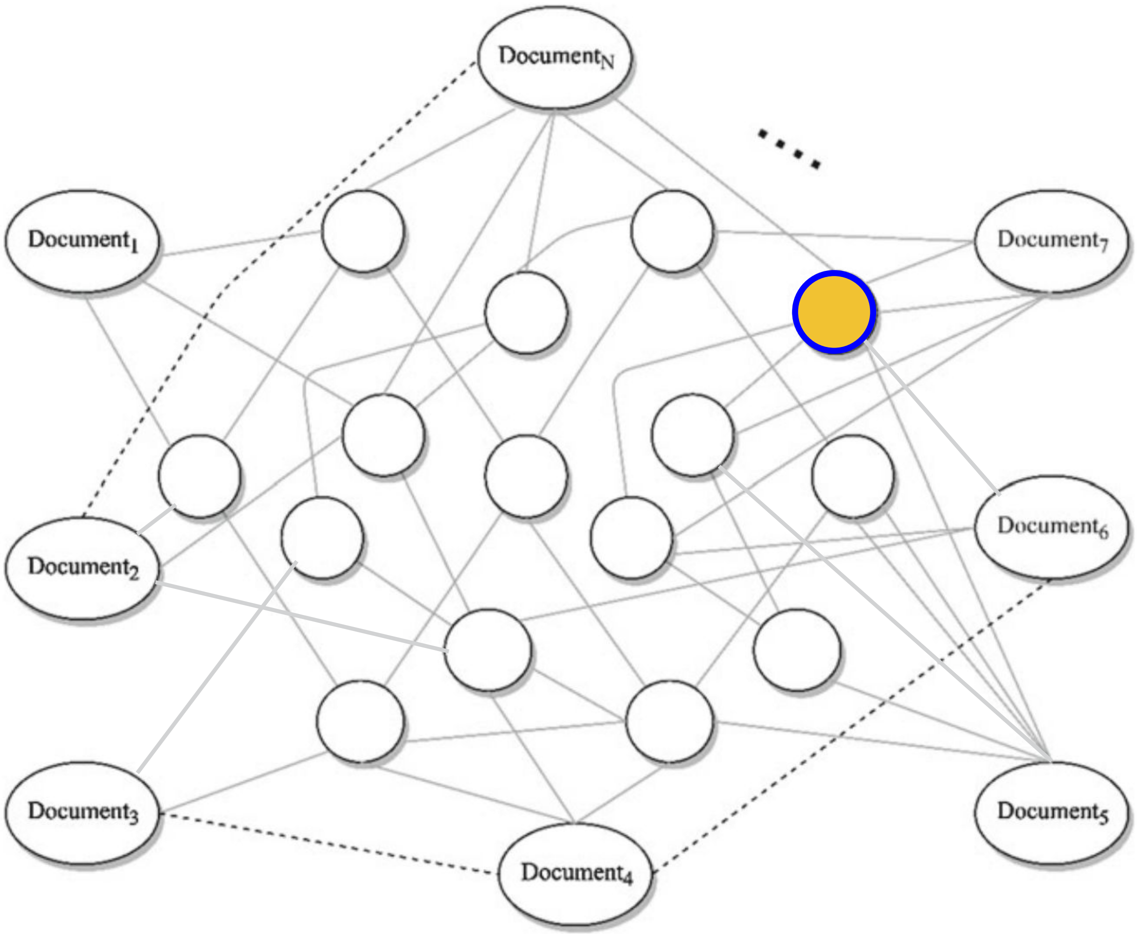

New Document
Document$_N$
Document$_1$
Document$_7$
Document$_2$
Document$_6$
Document$_3$
Document$_4$
Document$_5$


New Document
$Document_N$
$Document_1$
$Document_7$
$Document_2$
$Document_6$
$Document_3$
$Document_4$
$Document_5$

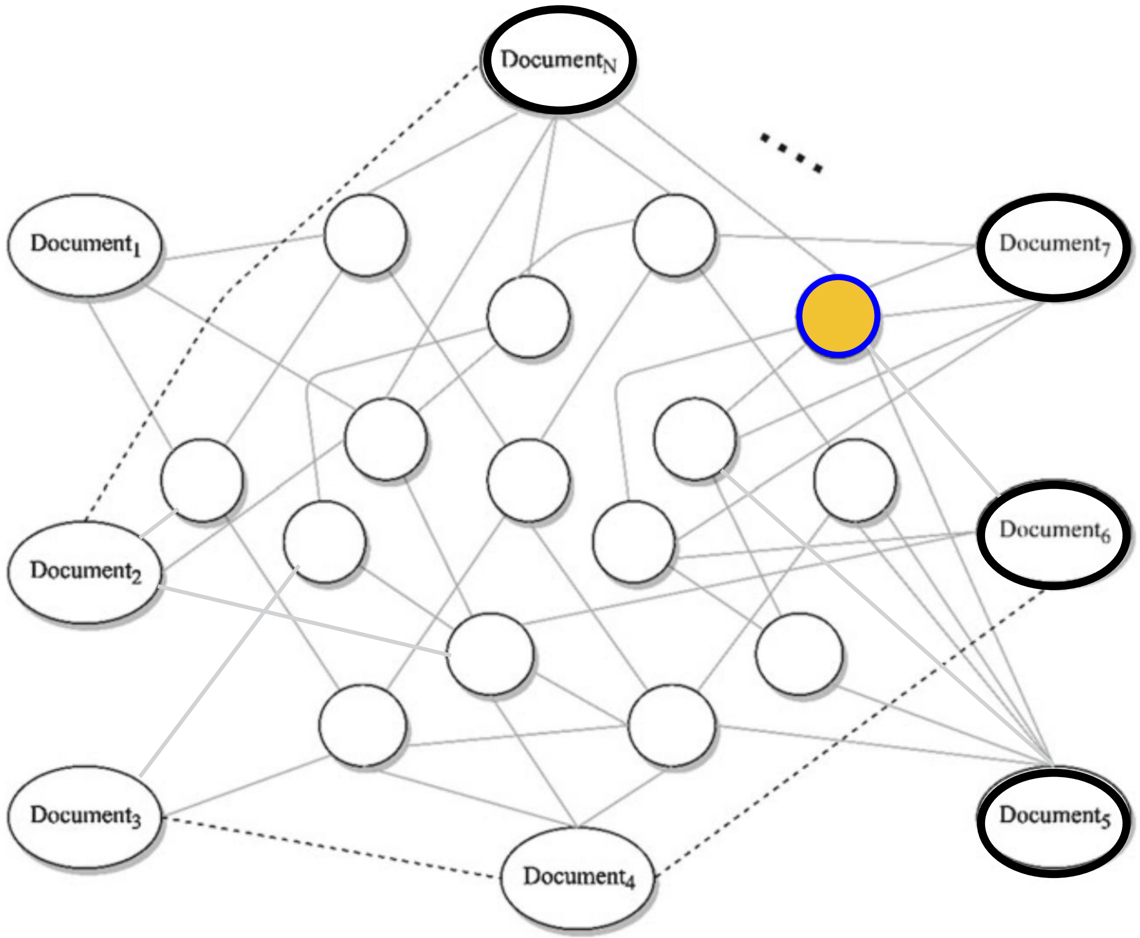
Model Modifications

- For the class being removed, remove that importance from each all of the words

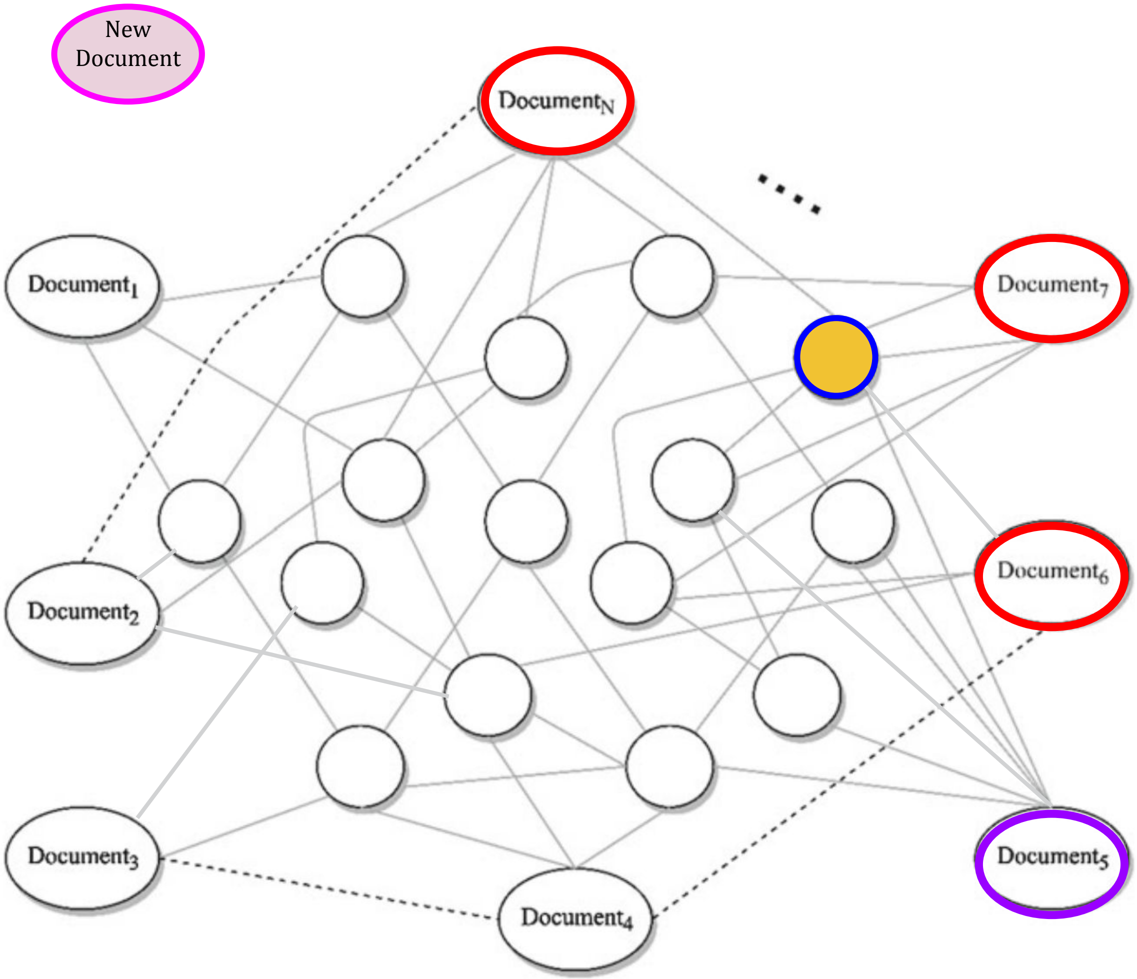




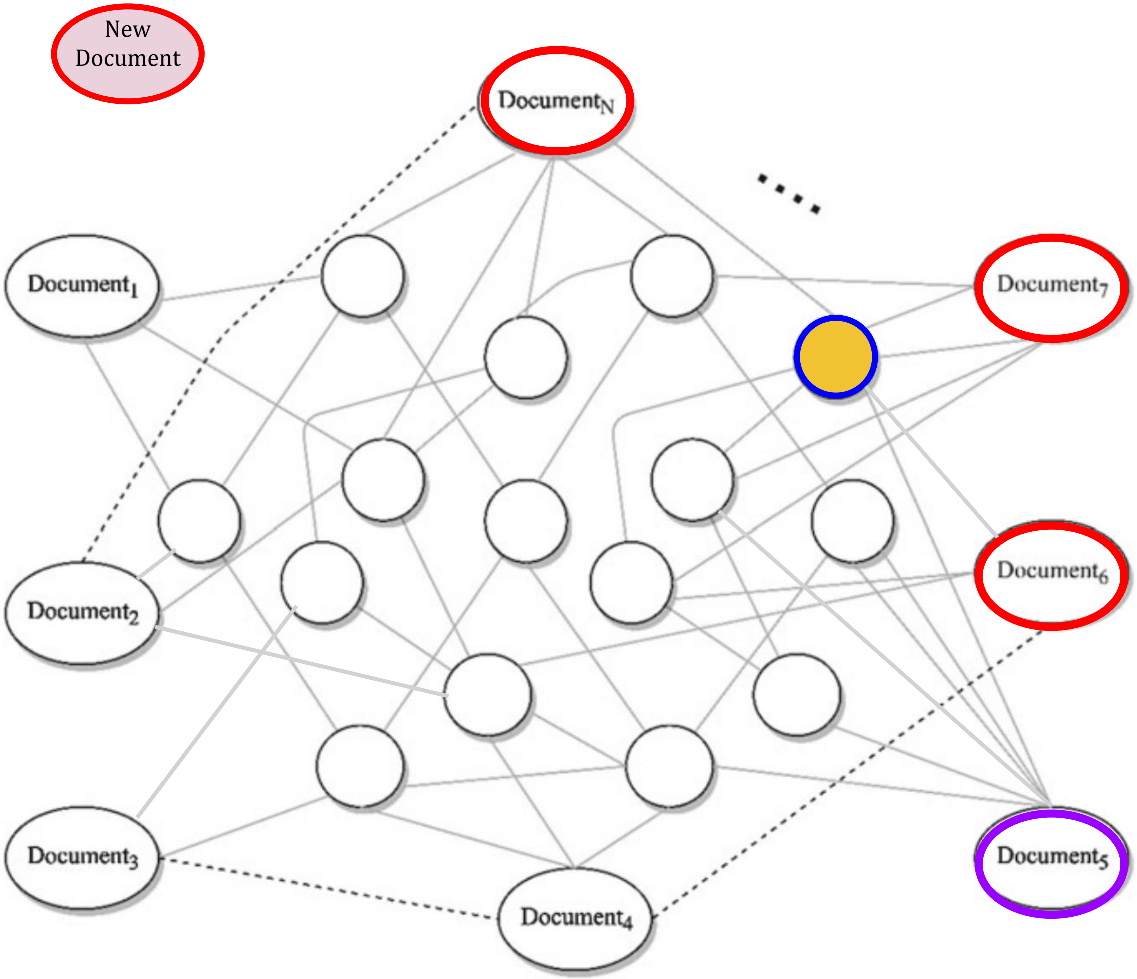
Document$_N$
Document$_1$
Document$_7$
Document$_2$
Document$_6$
Document$_3$
Document$_5$
Document$_4$



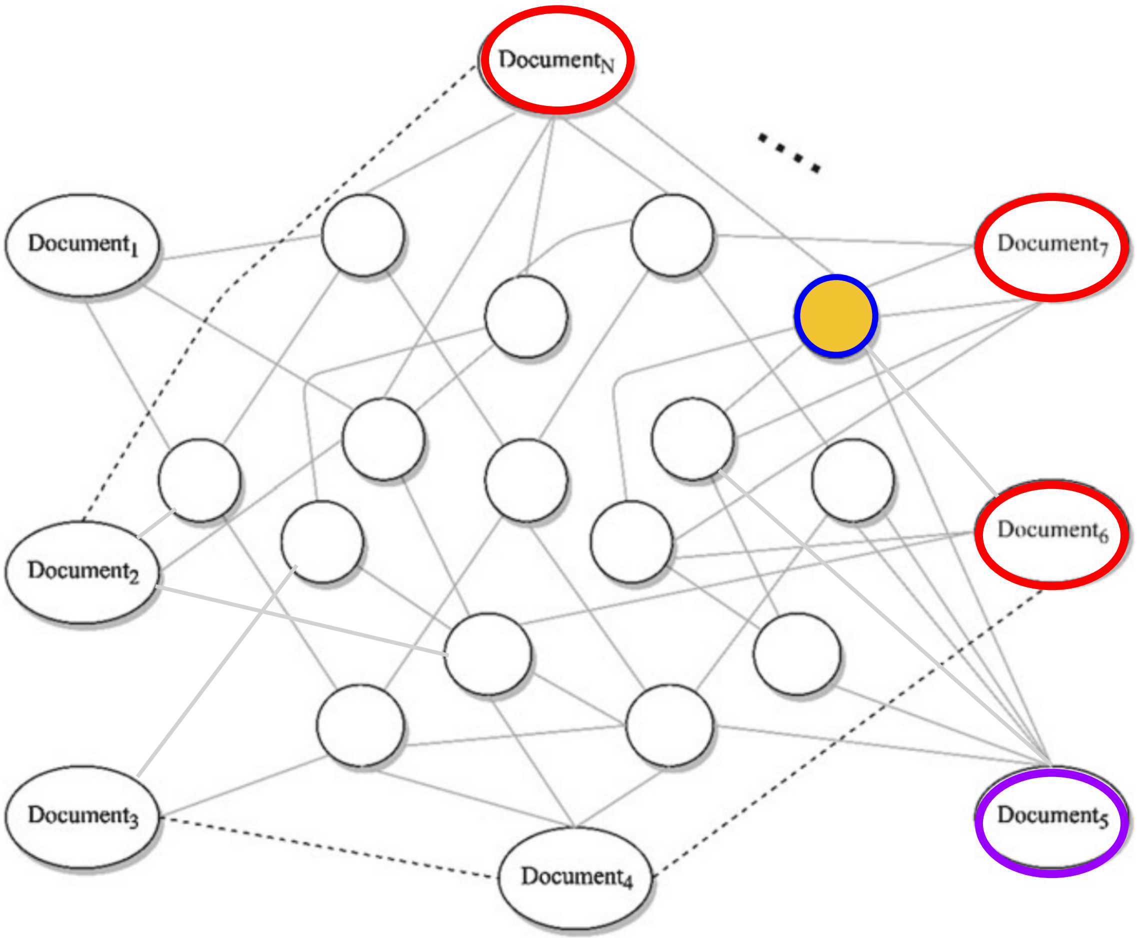
$Document_N$
$Document_1$
$Document_7$
$Document_2$
$Document_6$
$Document_3$
$Document_4$
$Document_5$



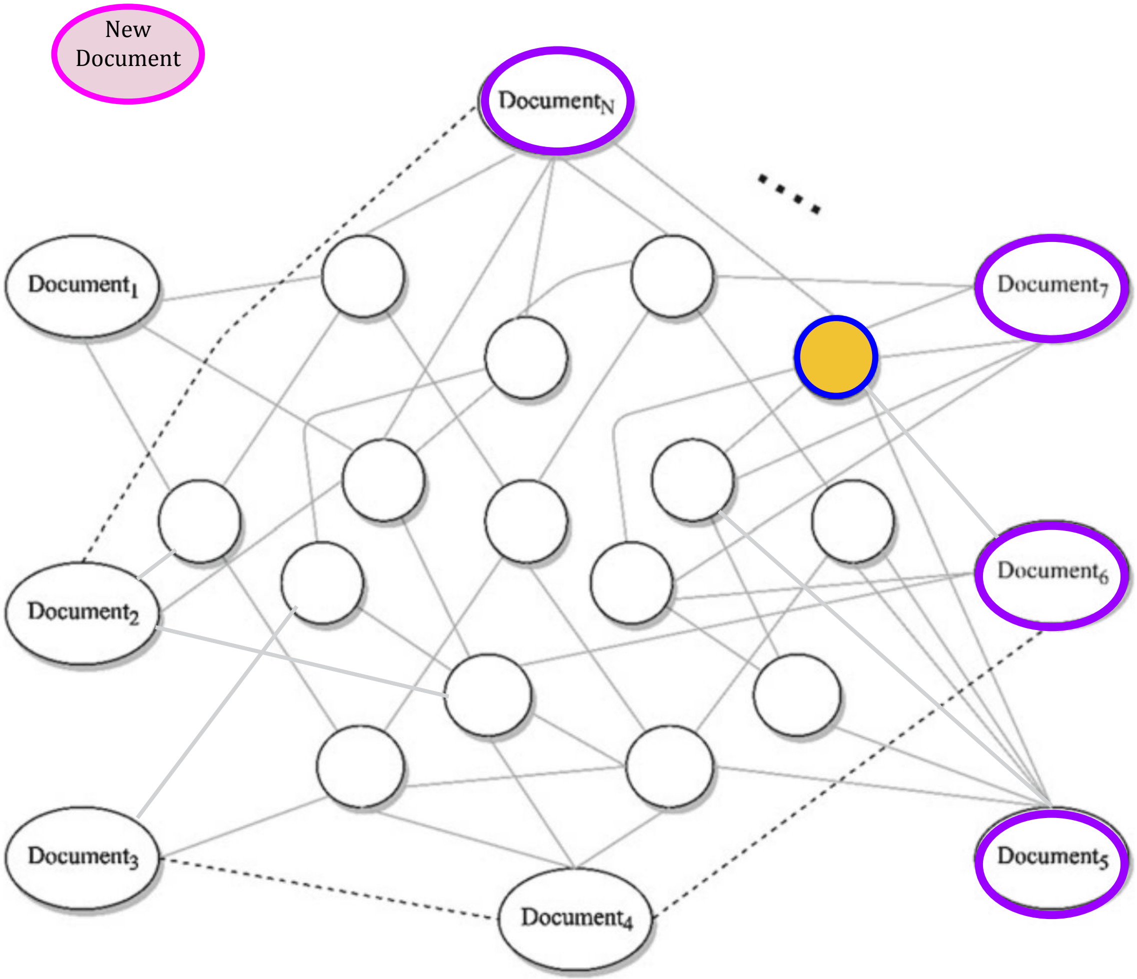
Class 1: 75%
Class 2: 25%



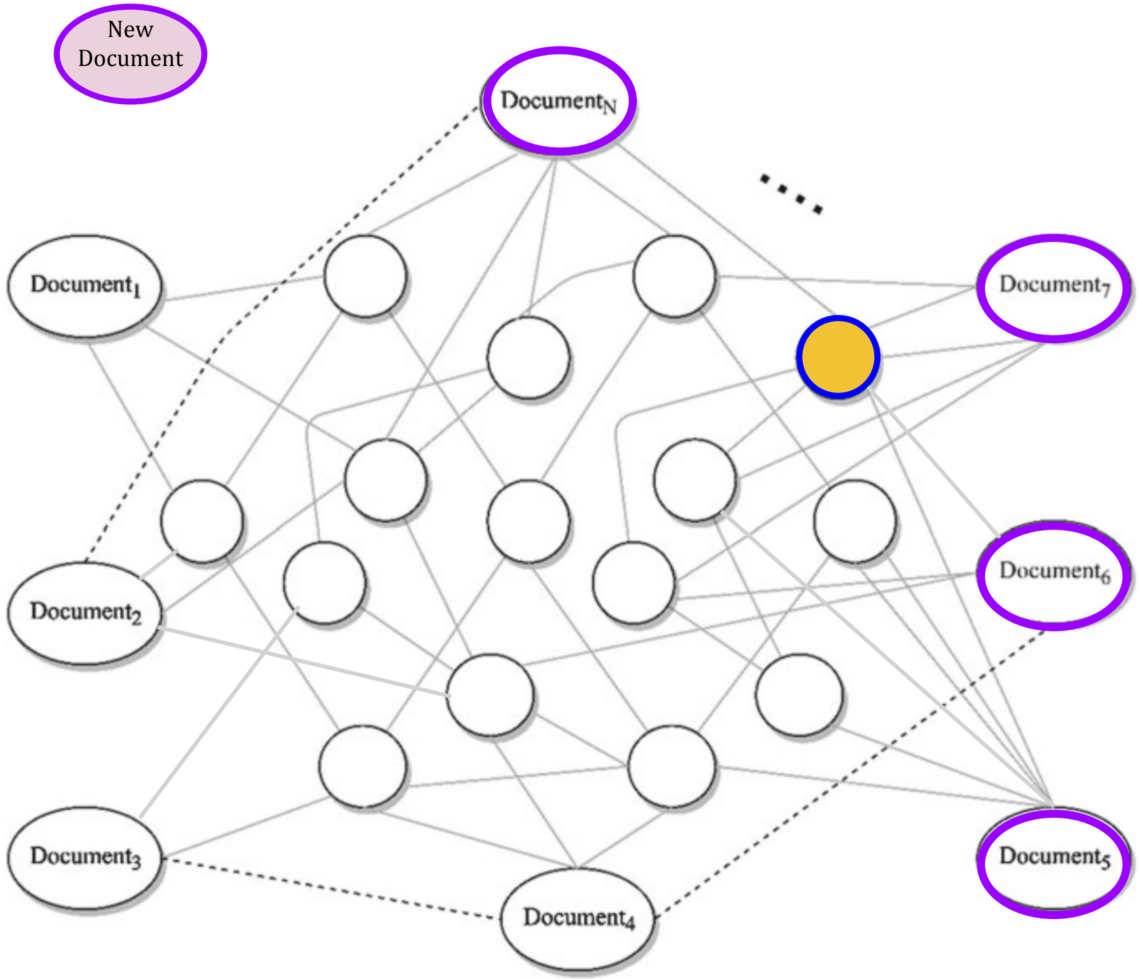
Class 1: 75%
Class 2: 25%



Class 1: 0%
Class 2: 100%



Class 1: 0%
Class 2: 100%



Class 1: 0%
Class 2: 100%

Measuring the Accuracy

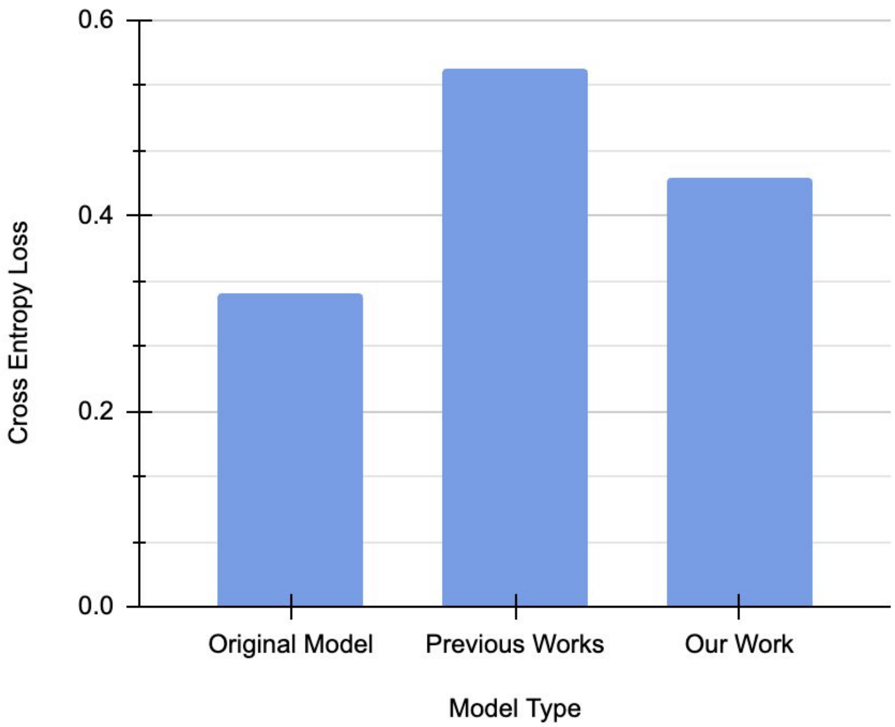
- **Cross Entropy**: Measures the difference between two probability distributions
  - Minimize the amount of entropy in the decisions for document classification for top K
  - Means documents are being classified more accurately
- We use *negative log likelihood loss* on the node label predictions for the GCN Model and binary cross entropy loss on the Edge Predictor model

True probability distribution
(one-shot)

$$H(p, q) = -\sum_{x \epsilon classes} p(x) log\ q(x)$$

Your model's predicted
probability distribution

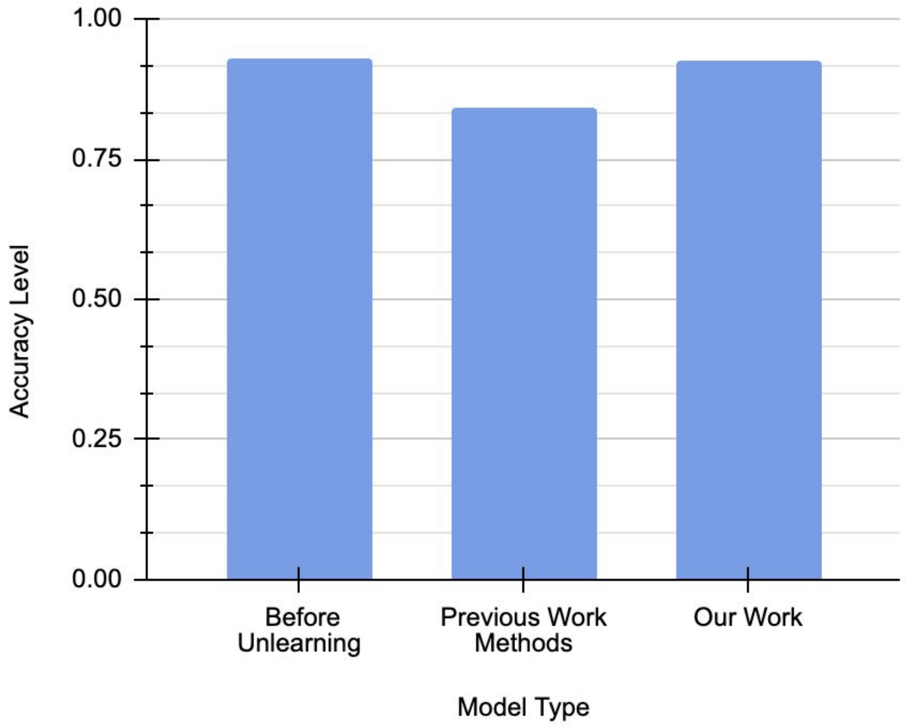
Loss Function Results On Classification Labels



We measure the cross entropy loss of the function

- Original cross entropy loss of 0.32
- Increase to 0.55 using previous methods
- Our works give 0.44

Accuracy Results On Classification Labels



We look to the accuracy of how well the model is able to predict the classes

- Original accuracy of 92.8%
- Decreases to 84.3% using current state of the art algorithms
- Our works give an accuracy of 92.55%

Future Work

- Using bigger datasets
  - Neo4j
- Unlearning edges and the classification together
  - Edges can be used to help reclassify the documents after removing a class
- Comparing the utility of unlearning the entire label with unlearning all the documents within
- Consider how unlearning an edge (say (u, v)) may affect the predicted probability of (u, w) for some other vertex w
- Consider the optimality of reorganizing word importance for classification unlearning
- Compare against previous work
- Theoretical analysis, connection with information theory and differential privacy

Acknowledgements

- Thank you to Slava Gerovitch, Srini Devadas, and the rest of MIT PRIMES for making this project possible
- Thank you to our mentor, Mayuri Sridhar, for guiding our research, and supporting us in every way
- Thank you to our parents for supporting us through the program

Bibliography

- Lei Kang, Mohamed Ali Souibgui, Fei Yang, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. “Machine Unlearning for Document Classification”. *arXiv preprint arXiv:2404.19031*, 2024.
- Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. “GNNDelete: A General Strategy for Unlearning in Graph Neural Networks”. *arXiv preprint arXiv:2302.13406*, 2024.