Graph Deep Learning Based Hashtag Recommender for Reels on Social Media

Sriya Balineni
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Artificial Intelligence and Robotics Commons, and the Databases and Information Systems Commons

Recommended Citation
Balineni, Sriya, "Graph Deep Learning Based Hashtag Recommender for Reels on Social Media" (2023). Master's Projects. 1220.
DOI: https://doi.org/10.31979/etd.7tgr-wzg4
https://scholarworks.sjsu.edu/etd_projects/1220

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.
Graph Deep Learning Based Hashtag Recommender for Reels on Social Media

A Project

Presented to
The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Sriya Balineni
May 2023
The Designated Project Committee Approves the Project Titled
Graph Deep Learning Based Hashtag Recommender for Reels on Social Media

By
Sriya Balineni

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE
SAN JOSÉ STATE UNIVERSITY
MAY 2023

Dr. William Andreopoulos  Department of Computer Science
Dr. Faranak Abri  Department of Computer Science
Mr. Manoj Desaraju  Software Engineer, Adobe
ABSTRACT

Many businesses, including Facebook, Netflix, and YouTube, rely heavily on a recommendation system. Recommendation systems are algorithms that attempt to provide consumers with relevant suggestions for items such as movies, videos, or reels (microvideos) to watch, hashtags for their posts, songs to listen to, and products to purchase. In many businesses, recommender systems are essential because they can generate enormous amounts of revenue and make the platform stand out when compared to others. Reels are a feature of the social media platforms that enable users to create and share videos of up to sixty seconds in length. Individuals, businesses, and organizations frequently use reels to display their creativity, advertise their products, and communicate with their audience. Users annotate these reels with hashtags that, in their opinion, properly depict the content of the reel. Therefore, it is important to consider user preferences of the content of the reel and their preferences of the hashtags to make relevant recommendations. In this project, we focus on providing users with hashtag recommendations for the reels they want to post, based on individual user's preferences of the content of the reel and hashtags, thus creating a personalized recommender for each user. Most methods that implement hashtag recommendations model interactions between users and hashtags or hashtags and posts, unlike this scenario, where we design a hashtag recommendation system based on users, reels, and hashtags. The dataset was built using web scraping, which downloaded the reels from TikTok. We designed a personalized hashtag recommender to recommend hashtags to users based on their previous posts, taking into account both their preferences for the content of the post and their understanding of hashtags. Our proposed graph deep learning based model outperformed existing approaches by achieving a NDCG score of 0.9156 which is significantly higher than the existing approaches.
ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to my project advisor, Dr. William Andreopoulos, who has been a constant source of support and encouragement throughout my graduate school journey. Their insights, guidance, and motivation played an important role in the completion of this project. I would like to extend my gratitude to the Division of Research and Innovation at San José State University for supporting my research under award number 23-SRA-08-028.

I would also like to thank the members of my committee Dr. Faranak Abri and Manoj Desaraju for their support and valuable feedback.

I would also like to thank God for blessing me with a loving family that constantly motivates me to pursue my goals. Lastly, I would like to thank everyone who made my journey at San José State University a memorable experience.
# TABLE OF CONTENTS

1. Introduction .......................................................................................................................... 1  
   1.1. Motivation ....................................................................................................................... 4  
   1.2. Problem Statement ......................................................................................................... 5  
2. Background Work ................................................................................................................. 6  
   2.1. Hashtag Recommendation .............................................................................................. 6  
       2.1.1. Content-Based Methods ......................................................................................... 7  
       2.1.2. Collaborative Filtering Methods ........................................................................... 8  
   2.2. Graph Convolutional Networks ..................................................................................... 10  
3. Methodology .......................................................................................................................... 13  
   3.1. Dataset Creation ............................................................................................................ 14  
   3.2. Dataset Preprocessing ................................................................................................... 15  
   3.3. Graphs construction ....................................................................................................... 16  
   3.4. Feature Engineering ....................................................................................................... 18  
   3.5. Graph Neural Network Architecture ............................................................................ 19  
   3.6. Model Training and Validation ..................................................................................... 23  
4. Evaluation and Results ........................................................................................................... 24  
   4.1. Evaluation Metrics ........................................................................................................ 24  
   4.2. Feature Analysis ............................................................................................................ 27  
       4.2.1. Video, Audio and Text Features ............................................................................. 28  
       4.2.2. Video and Audio Features ..................................................................................... 30  
       4.2.3. Video and Text Features ...................................................................................... 33  
       4.2.4. Audio and Text Features ...................................................................................... 36  
       4.2.5. Video Features ...................................................................................................... 38  
       4.2.6. Audio Features ...................................................................................................... 40  
       4.2.7. Text Features ....................................................................................................... 43  
   4.3. Performance Comparison with Existing Approaches .................................................. 47  
5. Conclusion and Future Work ................................................................................................. 53  
References ............................................................................................................................... 55
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.</td>
<td>Image of a beach annotated with #salty</td>
<td>1</td>
</tr>
<tr>
<td>Figure 2.</td>
<td>Image of an appetizer annotated with #salty</td>
<td>2</td>
</tr>
<tr>
<td>Figure 3.</td>
<td>Model Framework</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4.</td>
<td>Graph Structure</td>
<td>16</td>
</tr>
<tr>
<td>Figure 5.</td>
<td>Example Graph</td>
<td>17</td>
</tr>
<tr>
<td>Figure 6.</td>
<td>Graph Cardinality</td>
<td>17</td>
</tr>
<tr>
<td>Figure 7.</td>
<td>An Example to illustrate edge list</td>
<td>21</td>
</tr>
<tr>
<td>Figure 8.</td>
<td>Precision@5 for video, text and audio features</td>
<td>29</td>
</tr>
<tr>
<td>Figure 9.</td>
<td>Recall@5 for video, text, and audio features</td>
<td>29</td>
</tr>
<tr>
<td>Figure 10.</td>
<td>NDCG@5 for video, text and audio features</td>
<td>30</td>
</tr>
<tr>
<td>Figure 11.</td>
<td>Precision@5 for video and audio features</td>
<td>31</td>
</tr>
<tr>
<td>Figure 12.</td>
<td>Recall@5 for video and audio features</td>
<td>32</td>
</tr>
<tr>
<td>Figure 13.</td>
<td>NDCG@5 for video and audio features</td>
<td>32</td>
</tr>
<tr>
<td>Figure 14.</td>
<td>Precision@5 for video and text features</td>
<td>34</td>
</tr>
<tr>
<td>Figure 15.</td>
<td>Recall@5 for video and text features</td>
<td>34</td>
</tr>
<tr>
<td>Figure 16.</td>
<td>NDCG@5 for video and text features</td>
<td>35</td>
</tr>
<tr>
<td>Figure 17.</td>
<td>Precision@5 for audio and text features</td>
<td>36</td>
</tr>
<tr>
<td>Figure 18.</td>
<td>Recall@5 for audio and text features</td>
<td>37</td>
</tr>
<tr>
<td>Figure 19.</td>
<td>NDCG@5 for audio and text features</td>
<td>37</td>
</tr>
<tr>
<td>Figure 20.</td>
<td>Precision@5 for video features</td>
<td>39</td>
</tr>
<tr>
<td>Figure 21.</td>
<td>Recall@5 for video features</td>
<td>39</td>
</tr>
<tr>
<td>Figure 22.</td>
<td>NDCG@5 for video features</td>
<td>40</td>
</tr>
<tr>
<td>Figure 23.</td>
<td>Precision@5 for audio features</td>
<td>41</td>
</tr>
<tr>
<td>Figure 24.</td>
<td>Recall@5 for audio features</td>
<td>42</td>
</tr>
<tr>
<td>Figure 25.</td>
<td>NDCG@5 for audio features</td>
<td>42</td>
</tr>
<tr>
<td>Figure 26.</td>
<td>Precision@5 for text features</td>
<td>44</td>
</tr>
<tr>
<td>Figure 27.</td>
<td>Recall@5 for text features</td>
<td>44</td>
</tr>
<tr>
<td>Figure 28.</td>
<td>NDCG@5 for text features</td>
<td>45</td>
</tr>
<tr>
<td>Figure 29.</td>
<td>Precision@5 of PHR</td>
<td>49</td>
</tr>
</tbody>
</table>
Figure 30. Recall@5 of PHR ................................................................. 49
Figure 31. NDCG@5 of PHR ............................................................... 50
Figure 32. Precision@5 of LOGO ......................................................... 51
Figure 33. Recall@5 of LOGO .............................................................. 51
Figure 34. NDCG@5 of LOGO .............................................................. 52
LIST OF TABLES

Table 1.  Relevance scores of hashtags .............................................................. 26
Table 2.  Hashtag recommendations ................................................................. 27
Table 3.  Summary of model performance for video, audio, text features ............. 30
Table 4.  Summary of model performance for video and audio features ............... 33
Table 5.  Summary of model performance for video and text features ................. 35
Table 6.  Summary of model performance for audio and text features ................. 38
Table 7.  Summary of model performance for video features ............................. 40
Table 8.  Summary of model performance for audio features ............................. 43
Table 9.  Summary of model performance for text features ............................... 45
Table 10. Summary of feature analysis ............................................................... 46
Table 11. Summary of the performance comparison .......................................... 48
1. INTRODUCTION

With the growing prevalence of social media platforms such as TikTok, YouTube, Twitter, and Instagram, hashtags have become an essential tool for posting and finding content. Hashtags play a crucial role in content discovery and user interaction. They serve as an effective tool for users to categorize and search for relevant content, as well as provide insights into trending topics and user interests. However, with the increasing amount of content on social media platforms, recommending customized and relevant hashtags to users has become increasingly crucial to enhance the user experience and promote meaningful interactions.

A teenager may use the hashtag, #tea to discuss celebrity gossip, while a middle-aged individual may use the same hashtag to describe the beverage, chai tea. Figure 1 and Figure 2 also illustrate how #salty is used to annotate a photo of a beach and a shrimp appetizer. These examples show the stark difference in how two individuals can use the same hashtag for distinct posts, thus making user understanding of hashtag semantics, user preferences of hashtags an important aspect of recommending relevant hashtags for users.

![Figure 1](image1.jpg)

Figure 1. Image of a beach annotated with #salty
A user may upload a photo of their garden using hashtags such as #gardening and #greenery to annotate the post. However, users with different interests may be attracted to different aspects of the post and annotate the same photo with completely different hashtags. For instance, a user interested in sustainability may annotate the photo of the garden with hashtags like #sustainableliving and #ecofriendly, whereas a user interested in relaxation could be more interested in hashtags like #meditation and #selfcare. This illustrates that user preference of content of the post is equally important in making relevant recommendations to users. Therefore, utilizing user preferences of content and hashtags could help generate accurate and relevant personalized hashtag recommendations.

Numerous prior studies [1, 2, 3, 4] approached hashtag recommendation as a classification problem. However, hashtags can have multiple meanings depending on the person
and the post; therefore, hashtag recommendation as a classification problem may not reflect this complexity, resulting in incorrect or inappropriate recommendations.

A personalized hashtag recommender for users can increase user engagement by recommending hashtags that correspond with the users interests. It can further enhance user satisfaction, content visibility, and the user experience. A personalized hashtag recommender could offer users a more seamless experience by eliminating the need to manually search for hashtags, thereby saving time. To address this issue, we propose a personalized hashtag recommender system that provides users with customized hashtag recommendations to annotate the reels they are going to post.

This report presents a novel approach to personalized hashtag recommendation that leverages the rich relationships between users, reels, and hashtags. Our project aims to develop a personalized hashtag recommender system that accurately suggests hashtags based on the user's preferences, the content of the reels, and the associations among users, reels, and hashtags.

This work's primary contribution is the development of a graph-based model that includes three types of nodes: users, reels, and hashtags. This unique framework enables a more comprehensive depiction of the complex connections present between users, reels, and hashtags. In addition, we use graph convolutional networks with GATConv, SAGEConv, and GCNConv layers to incorporate attention mechanisms, and we also extract video, audio, and text features from the reels in order to generate personalized and content-aware hashtag recommendations.

In this report, section 2 provides a literature review of relevant works in hashtag recommendation and graph neural networks. Section 3 details the methodology, including the construction of the graph, feature engineering techniques, architecture, and implementation of our model. Section 4 presents the experiments and results, including a discussion of feature
analysis and a comparison of our model's performance with existing methods, and Section 5 concludes the report with a summary of our contributions and suggestions for future work.

Through the development of this personalized hashtag recommender system, we aim to improve the user experience by providing accurate and relevant hashtag recommendations that cater to individual preferences and interests while accounting for the complex relationships between users, videos, and hashtags.

1.1 Motivation

Massive growth in user-generated content has resulted from the extensive prevalence of social media platforms like TikTok, YouTube, and Instagram. Every day, billions of videos are uploaded, making it increasingly difficult for users to locate the content that interests them. To address this issue, social media platforms have introduced personalized hashtag recommendation systems that recommend hashtags based on users preferences or their viewing history. There is significant research on hashtag recommendations for content in the format of text and images [5, 6, 7, 8]. However, there is a lack of research on hashtag recommendation and personalized hashtag recommendation for micro-video (reel) content. Moreover, the majority of present works on personalized hashtag recommendation representations [9, 10, 11] only examine users and the posts they create or users and the hashtags they use to annotate the posts. The prior does not account for the user's preference for hashtags, and the latter does not consider the post content or the user's preference for reels. Moreover, the majority of existing hashtag recommendation approaches depend either on content-based methods, which primarily rely on the content of the posts [9, 12, 13], or collaborative filtering techniques that take user interactions into account and recommend hashtags based on the user's pattern of hashtag usage, ignoring the content of the
posts [14, 15, 16, 17, 18, 19, 20, 21]. Both of these approaches fail to completely exploit the complex relationships between users, reels, and hashtags.

To overcome these limitations, we propose a hybrid graph-based deep learning method for recommending personalized hashtags for reels using the TikTok dataset. Our method employs a comprehensive graph representation comprising three node categories, namely users, reels, and hashtags, to depict the intricate interrelationships between these entities. In addition, we extract video, audio, and text features to create a comprehensive feature set that depicts the unique characteristics of each reel, which assists in determining the user's preference of the reel. By using this approach, we aim to provide personalized hashtag recommendations to reels that consider the user's preferences of the content and hashtags. This can potentially improve the user experience by enabling users to discover relevant content more efficiently and by saving their time.

1.2 Problem Statement

The primary objective of this project is to develop a personalized hashtag recommender system for reels (micro-videos), which is a hybrid of content-based and collaborative filtering techniques. We aim to design and implement a graph deep learning based hybrid model that includes three types of nodes, users, videos, and hashtags, which will help the model learn about the underlying relationships between users, posts made by them, and the hashtags they use to annotate the posts. Our goal is to generate accurate, personalized, and content-aware hashtag recommendations by combining methods of feature engineering for extracting video, audio, and text features with graph neural networks consisting of multiple convolutional layers. The problem can further be subdivided into the following subproblems:
1. Creating graphs with user, video, and hashtag nodes to effectively represent the connections between them.

2. Extraction of video, audio, and text features from the reels to learn the reel representation and user preference in the reels.

3. Designing a graph neural network architecture capable of learning from graphs and utilizing the features extracted from the reels for personalized hashtag recommendation.

4. Evaluating the performance of the proposed model using real-world datasets and suitable performance metrics, and then comparing the results to existing methodologies in the field.

By handling these subproblems, we will be able to create a personalized hashtag recommender system that could overcome the shortcomings of existing methods and provide users with relevant hashtag recommendations that align with their preferences, thereby enhancing the user experience on social media platforms.

2. BACKGROUND WORK

In this section, we review the studies most relevant to our work, which are hashtag recommendation and graph convolutional networks.

2.1 Hashtag Recommendation

Hashtag recommendation models can be classified into two main approaches called content-based methods and collaborative methods.
2.1.1 Content-Based Methods

These methods focus on the relationship between hashtags and the content of the post. They evaluate the content's features and make hashtag recommendations based on how strongly the two items are related. Examples of content-based methods include the use of machine learning algorithms like Support Vector Machines (SVM), Naïve Bayes, or deep learning models like Recurrent Neural Networks (RNNs) or Transformer models for hashtag prediction based on the extracted features. Other techniques include text analysis, semantic analysis, and topic modeling.

Gong et al., [12] demonstrate a Convolutional Neural Network (CNN)-based hashtag recommendation system that learns the representation of text content and predicts relevant hashtags for a given post. An attention mechanism is also used to help the model focus on the most relevant words and phrases for predicting hashtags. Sedhai et al., [9] propose a content-based hashtag recommendation system based on text analysis for tweets that contain hyperlinks. The authors use a combination of hyperlink features, text features, and a learning-to-rank technique to recommend hashtags for tweets. The proposed design involves extracting relevant words from the tweet and the linked web page. The authors then utilize these keywords to generate a list of potential hashtags. The candidate set is further refined by selecting only those hashtags that are relevant to the tweet's and linked page's content.

The actual content of the post is the main focus of the above approaches. The user preferences of the post and hashtags are not taken into account. There are a lot of different instances in which users will have varied reactions to the content of the post. For instance, User A, who is a teenager, and User B, who is a middle-aged adult, both find themselves watching the same video on the most recent developments in technological trends. User A, the teenager, could
be more interested in the newest cell phones, gaming consoles, and social networking platforms. On the other hand, User B, an adult, would be more interested in technological developments relating to productivity tools, home automation, and security systems. User A may propose hashtags such as #SmartphoneTrends, #GamingConsoles, or #NewSocialMediaApps when asked to recommend hashtags for the video, whereas User B could be more inclined to use hashtags like #HomeAutomation, #SecurityTech, or #ProductivityTools. In this particular example, the fact that the two users are of different ages and have had various life experiences contributes to their different preferences of the video's content, which in turn influences the hashtags they choose to annotate the posts.

Veit et al., [13] distinguish between self-expression and visual content in hashtag supervision for image classification. The authors propose a method that combines hashtags and the visual content of the image to enhance image representations. They believe that hashtags fall into two categories: content-related hashtags, which characterize the visual content of the image, and self-expression hashtags, which reflect users' personal preferences, opinions, or sentiments. This work, however, centers solely on user preference for hashtags and disregards user preference of the post itself. Using graph-based representations and user-specific post content and hashtags, our model would recommend personalized hashtags to each user, taking into account user preference on both the reel and hashtags.

2.1.2 Collaborative Filtering Methods

These approaches recommend hashtags by leveraging the association between hashtags and users. They are predicated on the assumption that individuals who have used similar hashtags in the past will do so again. Two categories of collaborative filtering approaches include user-based collaborative filtering and item-based collaborative filtering.
The user-based collaborative filtering method recommends hashtags based on the user's similarity to other users. It identifies users who have previously used similar hashtags and suggests hashtags that these users have used but the target user has not. For instance, if both User A and User B have frequently used hashtags such as #MachineLearning, #DataScience, and #AI, and User B has also used #DeepLearning, the algorithm may recommend #DeepLearning for User A's next post.

Item-based collaborative filtering is a recommendation system based on the similarity of objects. If a user likes item A and item B is similar to item A, the user is likely to enjoy item B as well. Various measures, including cosine similarity and the Pearson correlation coefficient, can be used to assess the similarity of two items.

Torres et al., [14] developed a Twitter hashtag recommendation system using collaborative filtering techniques, including user-based and item-based approaches. The authors generated recommendations by accumulating and preprocessing Twitter data, constructing a user-item matrix, and measuring user similarity with cosine similarity and the Pearson correlation coefficient. User-based filtering concentrated on recommending hashtags frequently used by users with similar features, whereas item-based filtering considered hashtags frequently used alongside the target user's hashtags.

Chen et al., [15] and Khabiri et al., [16] propose a hashtag recommendation system based on collaborative filtering that leverages text similarity. To measure the similarity between texts, they use various text similarity functions, such as Jaccard similarity, cosine similarity, and dice similarity. The method recommends hashtags based on the similarity between the target tweet and other tweets containing particular hashtags.
As every user perceives content differently, Salakhutdinov et al., [17], Alvari et al., [18], and Kumar et al., [19] present a probabilistic matrix factorization (PMF) approach to collaborative filtering that models hashtags and user interactions. This technique has not completely utilized the post content, which contains valuable information about user preferences and hashtag semantics.

Collaborative methods assume that the relationships between users and posts or identifiers are linear, which is ineffective because it disregards the post's actual content; it merely captures user-specific hashtag usage patterns while disregarding user preference and the semantics of hashtags. Although hashtags and social media users can number in the millions, the majority of users only interact with a small subset of them. Due to the dearth of interaction data, collaborative filtering algorithms have difficulty precisely identifying similar users or hashtags. Our model is not a collaborative filtering based method as we don't consider similarity between neither users nor hashtags.

2.2 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are a form of deep learning model that works on graph-structured data. They are an enhancement to Convolutional Neural Networks (CNNs), which have proven to be highly effective at processing grid-like inputs like images. CNNs are best adapted for data with regular structures, whereas GCNs can manage graphs with irregular structures. Graphs are a versatile data format that can represent complex interactions between entities. In a graph, entities are depicted as nodes, and edges between nodes represent the interactions between nodes. GCNs operate by learning to aggregate and transform the characteristics of neighboring nodes in a network, taking into consideration both the local structure of the graph and the characteristics of the nodes. This is also referred to as "message
passing" and "neighborhood aggregation." During this process, the model learns to identify significant patterns and relationships in the network, which can then be applied to the tasks at hand, including node classification, link prediction, and graph classification. Multiple layers are utilized in a GCN to transfer data over varying graph distances. Each GCN layer performs a graph convolution operation that combines a node's characteristics with those of its neighbors, followed by a non-linear activation function (such as ReLU). By layering many layers, a GCN can discover increasingly complex patterns and relationships in a graph.

Wang et al., [20] present the Neural Graph Collaborative Filtering (NGCF) model, a graph-based recommendation system that employs the user-item interaction graph for collaborative filtering. The NGCF model learns user and item (hashtag) representations using GCNs to capture high-order connectivities in the interaction graph. Graph Convolutional Matrix Completion (GCMC) is an approach that Berg et al., [21] present for collaborative filtering-based recommendation systems to capture both local and global structure in the user-item interaction graph. The GCMC model employs graph convolutional networks and matrix completion methods to recommend hashtags.

Fan et al., [22] construct a graph that includes user, hashtag, and social nodes, as well as edges indicating interactions between users and items and social relationships between users. They employed an attention mechanism that learned the significance of different relationships. Secondly, a gating mechanism is utilized to propagate information selectively between the nodes to balance the significance of various kinds of nodes in the recommendation process. This approach utilized three types of nodes, but the content of the posts is ignored, which would help understand the user preference in the post to recommend relevant hashtags.
During training, GCNs require maintaining the entire graph structure and feature data in memory. This can be challenging for large graphs because the memory requirements may exceed the hardware's capacity. By learning inductive representations that can generate node representations for unseen nodes, inductive learning methods are designed to manage large-scale graphs. This makes training and inference on large graphs more efficient. Hamilton et al., [23] present GraphSAGE, a large-scale inductive representation learning method that can generate embeddings for previously unknown nodes. GraphSAGE employs aggregator functions such as mean, long short-term memory (LSTM), and max-pooling to collect neighborhood-specific data. The method successively applies aggregator functions to a neighborhood of a fixed size surrounding the target node. This allows GraphSAGE to efficiently scale.

Users and items (hashtags) are typically the two categories of nodes used to depict the user-item interaction in GCN graphs. However, our proposed model operates on graphs with three node types: users, reels, and hashtags. The graph depicts a more detailed representation of the connection between users, reels, and hashtags by incorporating three categories of nodes. This additional data enables our model to understand the fundamental patterns of user preferences, the content of the reel, and hashtag usage with greater accuracy. The proposed model considers the content of videos when recommending hashtags by incorporating reel nodes and their associated features. This is an enhancement over models that rely solely on user-item interactions, which may not completely capture a hashtag's relevance to a video's content. By incorporating user-specific data, our model is able to generate hashtag suggestions that are more tailored to the preferences and tendencies of individual users.
3. METHODOLOGY

In this section, we describe the steps in designing our model with the main goal of recommending personalized hashtags to every user, as shown in Figure 3.

![Model Framework](image)

Figure 3. Model Framework

The entire process can be divided into six steps. They are as follows:

1. Building the dataset
2. Preprocessing the data
3. Constructing the graphs

4. Feature engineering

5. Developing a graph neural network architecture

6. Training the model and validation

### 3.1 Dataset Creation

In order to begin the process of constructing the dataset, we began by obtaining an Excel document from Kaggle. This sheet had information on a variety of users, including the URLs of reels they had posted on TikTok as well as the hashtags they had used to annotate those reels. Each reel was annotated with two hashtags. We built a web scraper with Python's Selenium package to obtain the actual video files that corresponded to the URLs. Using Selenium, we are able to explore and interact with a website called SnapTik that belongs to a third party in an effective and time-saving manner. The SnapTik website allows users to download TikTok reels using the reel's URL. Our web scraper was designed with the following steps:

1. Reading the Excel sheet and extracting the reel URLs and user and hashtag data associated with them

2. Initializing an instance of Selenium WebDriver to automate browser interactions

3. Looping through the reel URLs in the Excel sheet and navigating to the SnapTik website

4. Pasting the URL into the input field on the SnapTik website and submitting the form

5. Waiting for the video download link to become available and then downloading the reel to a specified folder on our local machine.
6. Storing the downloaded reel information, including the local file path, user data, and hashtags, in a structured format for further processing

We were able to compile a dataset of TikTok reels along with the corresponding user and hashtag information by automating the download process using Selenium. This dataset served as the foundation for our graph construction, feature engineering, and eventual development of our personalized hashtag recommender system. We were able to download 1080 reels using the above mechanism. Ten reels that were present in the Excel sheet but had been deleted by users were unable to be downloaded.

3.2 Dataset Preprocessing

In the data preprocessing stage, we took several steps to clean and refine the dataset. These steps included:

1. Removing users with only one reel as our goal is to construct a model that can effectively learn from the user's posting history. We needed at least two videos per user for training and validating the model.

2. Filtering out any records that had no audio or video content to ensure our dataset contained relevant data for extracting features from the reels. This step eliminated noise and irrelevant data points from our dataset.

3. Excluding posts without hashtags, as it is essential to have at least one hashtag associated with each reel for the model to learn user preferences on hashtags.

4. Eliminating redundant data to ensure that our dataset consisted only of unique graph representations and did not introduce any biases or overfitting.

We were left with 300 reels uploaded by 97 distinct users and 260 unique hashtags used by the users to annotate the reels.
3.3 Graphs Construction

In this section, we describe the process of creating a comprehensive graph representation to model the relationships between users, reels, and hashtags. We build heterogeneous graphs where there are multiple types of nodes and each edge connects two nodes [24]. We have a heterogeneous graph $G = (U, R, H, E)$, as shown in Figure 4, where $U$ represents the set of user nodes, $R$ represents the set of reel nodes, $H$ represents the set of hashtag nodes, and $E$ represents the set of edges that connect the nodes in the graph. There are edges between the user nodes and reel nodes to indicate that a user has uploaded a corresponding reel. Similarly, there are edges between user nodes and hashtag nodes to show that a user has annotated a reel they have posted with the corresponding hashtag. Finally, there are edges between the reel nodes and hashtag nodes to indicate that a specific hashtag has been used to annotate a particular reel.

![Graph Structure](image)

Figure 4. Graph Structure

In graph theory, the degree of a node in a graph refers to the number of connections that the node has to other nodes in the graph. For example, in Figure 5, the degree of the user node Joe is 3, the degree of the reel uploaded by Joe is 3, and the degree of #hiit and #workout is 2.
The cardinality of the graph is illustrated in Figure 6. According to the dataset, a user could have uploaded n reels. Each reel is annotated by two hashtags, and as the same hashtags could be used to annotate multiple reels, each user will have used k hashtags, where k is the number of unique hashtags used by the user.

We built functions that, given the input data, generate edge sets that represent the relationships between users, reels, and hashtags. In addition, user-hashtag and reel-user-hashtag connections were separated for further processing. We constructed the graph using the derived
edge sets from the input data. Using the PyTorch library, these edge sets were then transformed into tensors. As we construct the edges between users, videos, and hashtags, the nodes of the graph become implicitly represented.

### 3.4. Feature Engineering

Feature engineering enables the model to interpret and understand different aspects of data. We extracted video features, audio features, and text features from the reels, and hashtags used to annotate the corresponding reels. This thorough feature set enables the model to collect vital data across multiple modalities. The four main steps in our feature engineering are as follows:

1. **Video feature extraction:** By isolating keyframes at 30-frame intervals, we captured the most representative frames from each video. These keyframes provide a condensed representation of the video content that can be utilized to extract video features. To extract features from the keyframes, we used ResNet50, a pre-trained deep learning model. This method allowed us to gain a comprehensive comprehension of the video's visual content.

2. **Audio feature extraction:** We separated the audio from each video using Pydub, a Python library. Then, we extracted audio features using Librosa, a prominent audio analysis library. We calculated Mel-frequency cepstral coefficients (MFCCs) for every audio segment and used their mean values as audio features.

3. **Text feature extraction:** The hashtags associated with each video were used to extract features of text. We used the TfidfVectorizer to convert the hashtags into a numeric representation that quantifies the significance of each hashtag within the context of the entire dataset.
4. Feature combination: Finally, we combined the video, audio, and text features of each reel to create an extensive feature set. This combined set of features enabled our recommendation model to utilize data from all three modalities, resulting in more relevant personalized hashtag recommendations.

3.5 Graph Neural Network Architecture

This section provides a summary of our Graph Neural Network architecture, layers, message transmission mechanism, initialization, and forward pass. In graph neural networks, each node in the graph has a feature vector that reflects the node's attributes. This feature vector is known as the node representation or node embedding. By representing each node in the graph with a feature vector, the graph neural network will learn to propagate information between nodes that are connected in the graph. This enables the model to capture the complex relationships and interactions between nodes, such as user preferences, the content of reels, and the semantics of various hashtags. Node representations are learned by updating the initial feature vectors of each node through several iterations of message passing between neighboring nodes or by using the node's local information itself. In our model, the user node's representation is learned by aggregating the message from its neighboring hashtag and reel nodes. Similarly, a hashtag's representation is learned based on the message from its neighboring user and reel nodes. The representation of the node reel is the features (video, audio, and text) of the reel.

One obvious problem with this method is the noise in the messages sent by the reels to the user node since the user is only interested in particular areas of the reel that would be the deciding factors of the hashtags. We have leveraged an attention mechanism to overcome the noise and redundancy in the messages sent by reels. Attention mechanisms are a deep learning approach that enables you to focus on specific portions of input data while ignoring irrelevant or
noisy parts. It works by giving weights to various input elements or components depending on their value and relevance to the task. This enables the model to efficiently filter out noise and focus solely on the most valuable pieces of input, resulting in improved model performance. Utilizing the attention mechanisms, the user node, with the help of corresponding hashtags used to annotate the reel, will be able to focus on the relevant part of the message sent by the reel node as the key moments of a reel based on the user's interests and preferences are used to annotate the post (reel) using hashtags.

The user representation is combined with the video and hashtag representations to generate user-specific representations of both videos and hashtags for the purpose of calculating relevance scores. By combining user-specific reel representation and user-specific hashtag representation, we can determine the relevance score of a hashtag the user has used to annotate a reel. Given a set of hashtags, reels, and user information, the model generates relevance scores for each hashtag and outputs the top k recommended items in descending relevance score order for the corresponding reel.

Our model accepts video features, an edge list, and the mapping of videos to feature indices as input parameters. The edge list is used to define the local neighbors of each node to enable message passing between adjacent nodes. For instance, the edge list of the graph in Figure 7 would be [(Joe, #fitness), (Joe, #cardio), (Joe, Reel), (Reel, #fitness), (Reel, #cardio)], as the graph is not directional, (Joe, #fitness) is equivalent to (#fitness, Joe). Therefore, there is only one entry for each edge to avoid redundancy.
Below are the layers in our model, GDLHR (Graph Deep Learning Based Hashtag Recommender):

1. Linear layer

   This is a feedforward neural network layer that uses weights and biases to apply a linear transformation to input features. It is employed to transform video features into the same dimension as latent features.

2. GATConv layer

   In GNNs, Graph Attention Network (GAT) layers are used to model the attention mechanism. The attention mechanism permits the model to evaluate the contribution of each neighbor, allowing it to prioritize the most relevant information [5, 22, 25].

3. SAGEConv layer

   The main goal of GraphSAGE is to learn a function that generates node representations by sampling a fixed-size neighborhood of each node and averaging the features of the sampled nodes. Utilizing sampling techniques to maintain a neighborhood of a fixed size
enables efficient inductive learning. Various aggregate functions, such as mean, max, or LSTM, could be used to aggregate neighborhood information. Mean based aggregation gave us the maximum performance [23].

4. GCNConv layer

Graph Convolutional Network (GCN) layers learn node representation by aggregating data from immediate neighbors using a first-order approximation of spectral graph convolutions, which allow the model to learn meaningful representations efficiently [26].

5. Linear layers

These layers are used to transform the concatenated representations of users - reels and user - hashtags to generate the final embeddings.

We utilized Xavier normalization to initialize the weights of the GATConv, SAGEConv, and GCNConv layers. In the forward pass of our model, random hashtags are sampled from the set of hashtags to enable contrastive learning for the model to be able to differentiate between the correct hashtag and a random hashtag. The forward pass then applies the linear layer to reel features in order to match the dimension of latent features and concatenates user-hashtag representations with transformed video feature representations to produce a combined representation. It then normalizes the resulting embeddings to prevent magnitude from affecting the learning of the model. The node representations are passed through GATConv, SAGEConv, and GCNConv layers with ReLU activation functions for non-linearity. The model then computes user-specific reel representations, user-specific correct hashtag representations, and user-specific random hashtag representations to capture the relationship between users, videos, and hashtags. Finally, the relevance score of positive hashtags is calculated by taking the dot product of user-specific video representation and user-specific positive hashtag representation.
Similarly, the relevance score of the random hashtag is calculated by taking the dot product of user-specific video representation and user-specific random hashtag embeddings. We have used the Adam optimizer and defined a loss function, which is a negative log-likelihood loss with sigmoid activation.

3.6 Model Training and Validation

The primary objective of the training phase was to optimize the model's parameters and weights to minimize the loss function without overfitting the data. The hold-out validation set was utilized to assess the model's generalization performance. The dataset is partitioned into training, validation, and testing sets. The training set is used to update the model weights, the validation set is used to monitor the model's performance during training and the testing set is used to evaluate the model's performance on previously unseen data.

To prepare the data for training, we used PyTorch DataLoaders to provide the model with mini-batch input data and labels during each training iteration. We tried different parameter settings, such as latent feature dimensions of [32, 64, 128], batch sizes of [12, 24, 48], and learning rates of [0.00001, 0.0001, 0.001, 0.01, 0.1]. The best results came from using a batch size of 12, a latent feature dimension of 32, and a learning rate of 0.001. We have trained our model for 250 epochs.

During the training phase, the model was fed samples of input features and correct and random hashtags. The model's forward function was then used to generate relevance scores for each of the correct and random hashtags. These scores were then used to compute the loss function, which assists the model in distinguishing between correct and random hashtags. To validate the model, we utilized a validation set that was not utilized for training. This validation set consisted of a 20% random sample of the entire dataset's data points. This set was used to
evaluate the performance of the model on unseen data by comparing the top 5 hashtags recommended by the model to the relevant hashtags associated with each reel. Common metrics in recommendation tasks like precision, recall, and NDCG were used to evaluate the model's performance.

4. EVALUATION AND RESULTS

In this section, we evaluate the performance of our personalized hashtag recommendation system and present the results of our experiments. The purpose of our experiments is to evaluate the effect of various feature combinations on our model and compare its performance to that of a few existing methods. Generally, a recommender system's performance is measured by its ability to recommend the top $k$ items that a user is most likely to be interested in. This is to ensure that the system can select a subset of the most relevant items to recommend to the user because, in the vast majority of real-world situations, it is impractical to recommend every possible item to a user.

4.1 Evaluation Metrics

Precision is a frequently employed evaluation metric for recommender systems. It measures the proportion of relevant items recommended by the system relative to the total number of items recommended. In other words, it determines the percentage of relevant user recommendations. It is calculated by dividing the number of relevant items within the top $k$ recommended items by the top $k$ recommendations. For example, if a recommendation system suggests 10 items to a user and only five of those items are relevant, the precision@5 score would be 0.5 (5/10). A higher precision@$k$ score indicates that the system provides more relevant recommendations to the user. Precision@$k$ is a useful metric for evaluating
recommendation systems because it takes into account both the relevance and quantity of items suggested. It is particularly beneficial when the goal is to provide the user with a small number of highly relevant recommendations, which is the case in most real-world scenarios.

Another common metric used to evaluate the effectiveness of a recommendation system is recall. Recall is the proportion of relevant items recommended by the recommender relative to the total number of relevant items. Unlike precision, recall takes into account all relevant items, not just those that were recommended. Recall@k measures the recall of the top k recommendations. It indicates the percentage of all relevant items identified in the top k recommendations. A high recall@k indicates that the recommendation system is able to locate a significant number of relevant items among the top-k recommendations. For instance, if a user has interacted with three items in the past and the model recommends only two of the three relevant items in the top five recommended items, the recall@5 would be 0.66 (2/3). A high recall@k score indicates that the model is effective at recommending pertinent items within the top k items to users, whereas a low score indicates that the model is ineffective at recommending relevant items. Similar to precision@k, recall@k can be calculated for each individual user and then averaged across all users to evaluate the system's overall effectiveness. It is crucial to recognize that recall and precision frequently compete with one another, implying that enhancing one may come at the expense of the other. Consequently, a compromise must be struck between the two metrics based on the objectives of the recommender system.

Normalized Discounted Cumulative Gain (NDCG) is another prominent metric for evaluating ranking algorithms in information retrieval and recommendation systems. It analyzes both the item's relevance and its position on the ranked list to assess the ranking quality of the recommendation list. NDCG@k is a variant of NDCG in which only the top k recommendation
list elements are considered. DCG@k represents the discounted cumulative gain for the initial k recommended items according to their position in the recommendation list generated by the recommendation system, while IDCG@k represents the ideal discounted cumulative gain for the initial k items, which is the DCG@k for the items in the ideal recommendation list with the items sorted in descending order based on their relevance score. NDCG@k is the ratio of DCG@k and IDCG@k. DCG@k is calculated as the sum of the relevance scores of the recommended items that have been discounted using a logarithmic function based on their list positions. DCG@k is given by

$$\text{DCG@k} = \sum \left( \frac{\text{relevance_score}_i}{\log_2(i+1)} \right) \text{ for } i = 1 \text{ to } k$$

IDCG@k represents the utmost DCG@k for a given set of recommended items. IDCG@k is given by

$$\text{IDCG@k} = \sum \left( \frac{\text{relevance_score}_i}{\log_2(i+1)} \right) \text{ for } i = 1 \text{ to } k$$

Suppose a recommender suggests the top three hashtags from a set of seven hashtags based on the relevance scores. Table 1 shows the relevance scores of all seven hashtags.

Table 1

<table>
<thead>
<tr>
<th>Hashtag</th>
<th>Relevance Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>#bussin</td>
<td>5</td>
</tr>
<tr>
<td>#barbiecore</td>
<td>4</td>
</tr>
<tr>
<td>#ootd</td>
<td>2</td>
</tr>
<tr>
<td>#datefit</td>
<td>2</td>
</tr>
<tr>
<td>#dapper</td>
<td>3</td>
</tr>
<tr>
<td>#rock</td>
<td>0</td>
</tr>
<tr>
<td>#eyecandy</td>
<td>5</td>
</tr>
</tbody>
</table>
If the recommender's top 3 recommended hashtags are in the following order.

Table 2

<table>
<thead>
<tr>
<th>Hashtag</th>
<th>Relevance Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>#eyecandy</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>#barbiecore</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>#bussin</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

DCG@3 for the above recommendation list is given by

$$DCG@3 = (5 / \log2(2)) + (4 / \log2(3)) + (5 / \log2(4)) = 5.8616$$

IDCG@3 for the ideal recommendation list where #bussin would have been placed above #barbiecore is given by

$$IDCG@3 = (5 / \log2(2)) + (5 / \log2(3)) + (4 / \log2(4)) = 7.6724$$

Therefore, the NDCG@3 for the given recommendations would be 0.7639. A high NDCG value suggests the recommendation system is providing the user with accurate and pertinent recommendations. It indicates that the items recommended are extremely relevant to the user's preferences and interests.

4.2 Feature Analysis

We conducted experiments with various combinations of video, audio, and text features in order to determine the effect of various feature combinations on our model's performance. Specifically, we trained and evaluated our model with the following combinations of features:

1. Video, audio, and text features
2. Video and audio features

3. Video and text features

4. Audio and text features

5. Video features

6. Audio features

7. Text features

4.2.1 Video, Audio, and Text Features:

The model obtained a precision@5 value of 0.3581, recall@5 of 0.8952 and NDCG@5 of 0.9156. Precision@5 of 0.3581 indicates that the model is capable of recommending pertinent hashtags to users, with more than 36% of the recommended hashtags being relevant to the user's preferences. The value indicates that the model can propose pertinent hashtags to users with precision. In addition, the model's recall@5 value of 0.8952 indicates that it is capable of identifying almost 90% of all relevant hashtags that could have been recommended. This indicates that the model can effectively accommodate a substantial proportion of user preferences. The NDCG@5 score of 0.9156 demonstrates that the model's recommendations are highly relevant and align with the user's preferences. This is because NDCG considers the importance of the recommended items in addition to their position on the list of recommendations. A close proximity to one indicates that the recommended items are highly relevant and positioned highly on the list.

Given that there are only two positive hashtags and five negative hashtags, making it challenging to recommend pertinent hashtags to the user, the model's performance is particularly impressive. With such a limited number of positive hashtags, the utmost precision@5 achievable is only 0.4, which is not significantly higher than the model's precision value of 0.3581. The results indicate that the personalized hashtag recommender system is able to identify and recommend
relevant hashtags to users, even in situations with few positive examples. Precision@5, recall@5, and NDCG@5 of the model using video, audio, and text features over the epochs are shown in the figures below.

Figure 8. Precision@5 for video, text and audio features

Figure 9. Recall@5 for video, text, and audio features
4.2.2 Video and Audio Features:

The model achieved a precision@5 of 0.3505, recall@5 of 0.8762, and NDCG@5 of 0.8855 using video and audio features. Recall@5 and NDCG@5 demonstrate a minor decrease in performance when compared to results derived using three features (video, audio, and text).
Precision@5 remains relatively stable, but the decline in recall@5 and NDCG@5 suggests that text characteristics may play an important role in hashtag identification. However, the stable precision@5 indicates that the model can still recognize the vast majority of relevant identifiers when only video and audio features are used. Overall, these results indicate that using video and audio features without text features may continue to result in acceptable performance. Precision@5, recall@5, and NDCG@5 of the model using video and audio features over the epochs are shown in the figures below.

![Precision@5 over epochs - Video and audio features](image.png)

Figure 11. Precision@5 for video and audio features
Figure 12. Recall@5 for video and audio features

Figure 13. NDCG@5 for video and audio features
Below table summarizes the model's performance at corresponding echos

TABLE 4

SUMMARY OF MODEL PERFORMANCE FOR VIDEO AND AUDIO FEATURES

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>9.8864</td>
<td>0.32</td>
<td>0.8</td>
<td>0.8082</td>
</tr>
<tr>
<td>99</td>
<td>6.9453</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8498</td>
</tr>
<tr>
<td>149</td>
<td>14.2014</td>
<td>0.3238</td>
<td>0.8095</td>
<td>0.8261</td>
</tr>
<tr>
<td>199</td>
<td>5.0117</td>
<td>0.3276</td>
<td>0.8190</td>
<td>0.8340</td>
</tr>
<tr>
<td>249</td>
<td>7.2831</td>
<td>0.3505</td>
<td>0.8762</td>
<td>0.8855</td>
</tr>
</tbody>
</table>

4.2.3 Video and Text Features

The model achieved a precision@5 of 0.3486, recall@5 of 0.8714, and NDCG@5 of 0.8760. When using only video and text features, precision, recall, and NDCG are marginally lower than when using all three features. Precision@5, recall@5, and NDCG@5 of the model using video and audio characteristics over the epochs are shown in the figures below.
Figure 14. Precision@5 for video and text features

Figure 15. Recall@5 for video and text feature
Figure 16. NDCG@5 for video and text features

Below table summarizes the model's performance at corresponding echos

TABLE 5
SUMMARY OF MODEL PERFORMANCE FOR VIDEO AND TEXT FEATURES

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>11.5243</td>
<td>0.3085</td>
<td>0.7714</td>
<td>0.8028</td>
</tr>
<tr>
<td>99</td>
<td>8.3053</td>
<td>0.3238</td>
<td>0.8095</td>
<td>0.8231</td>
</tr>
<tr>
<td>149</td>
<td>6.5373</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8622</td>
</tr>
<tr>
<td>199</td>
<td>5.1996</td>
<td>0.3390</td>
<td>0.8476</td>
<td>0.8703</td>
</tr>
<tr>
<td>249</td>
<td>4.8751</td>
<td>0.3486</td>
<td>0.8714</td>
<td>0.8760</td>
</tr>
</tbody>
</table>
4.2.4 Audio and Text Features

Our model attained precision@5 value of 0.3333, recall@5 value of 0.8333, and NDCG@5 value of 0.8485 using audio and text features. The model performs slightly poorly when only audio and text features are utilized as opposed to all three (video, audio, and text) features. This indicates that the video feature contributes significantly to the model's enhanced performance. Precision@5, recall@5, and NDCG@5 of the model using audio and text characteristics over the epochs are shown in the figures below.

![Precision@5 over epochs - Audio and text features](image)

Figure 17. Precision@5 for audio and text features
Figure 18. Recall@5 for audio and text features

Figure 19. NDCG@5 for audio and text features
Below table summarizes the model's performance at corresponding echos

TABLE 6
SUMMARY OF MODEL PERFORMANCE FOR AUDIO AND TEXT FEATURES

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>11.1162</td>
<td>0.3180</td>
<td>0.7952</td>
<td>0.8088</td>
</tr>
<tr>
<td>99</td>
<td>7.5169</td>
<td>0.32</td>
<td>0.8</td>
<td>0.7772</td>
</tr>
<tr>
<td>149</td>
<td>6.1341</td>
<td>0.3314</td>
<td>0.8285</td>
<td>0.8240</td>
</tr>
<tr>
<td>199</td>
<td>3.9165</td>
<td>0.3295</td>
<td>0.8238</td>
<td>0.8469</td>
</tr>
<tr>
<td>249</td>
<td>3.7084</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8485</td>
</tr>
</tbody>
</table>

4.2.5 Video Features

The performance of the model using only video features is quite similar to when all three features are used, with precision@5 remaining the same at 0.3581 and recall@5 and NDCG@5 being only slightly lower at 0.8952 and 0.9015, respectively. This indicates that video attributes can be a sufficient indicator of relevant hashtags for recommendation purposes. Precision@5, recall@5, and NDCG@5 of the model using video features over the epochs are shown in the figures below.
Figure 20. Precision@5 for video features

Figure 21. Recall@5 for video features
Below table summarizes the model's performance at corresponding echos

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>13.1257</td>
<td>0.3085</td>
<td>0.7714</td>
<td>0.7931</td>
</tr>
<tr>
<td>99</td>
<td>6.9695</td>
<td>0.3219</td>
<td>0.8047</td>
<td>0.8360</td>
</tr>
<tr>
<td>149</td>
<td>7.3713</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8446</td>
</tr>
<tr>
<td>199</td>
<td>6.9180</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8641</td>
</tr>
<tr>
<td>249</td>
<td>3.7948</td>
<td>0.3581</td>
<td>0.8952</td>
<td>0.9015</td>
</tr>
</tbody>
</table>

4.2.6 Audio Features

Compared to video features, the model's performance leveraging only audio features is slightly worse. Precision@5 and NDCG@5 are 0.3371 and 0.8491, respectively, while recall@5 is 0.8429. The graphs below reveal that the model's improvement over the echos is not as
seamless compared to other feature combinations, indicating that the model requires additional training. This suggests that leveraging only audio features may not be as useful for personalized hashtag recommendations for reels, and they would require more training to be reliable. Precision@5, recall@5, and NDCG@5 of the model leveraging only audio features over the epochs are shown in the figures below.

![Figure 23. Precision@5 for audio features](image-url)
Figure 24. Recall@5 for audio features

Figure 25. NDCG@5 for audio features
Below table summarizes the model's performance at corresponding echos

**TABLE 8**

**SUMMARY OF MODEL PERFORMANCE FOR AUDIO FEATURES**

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>11.479</td>
<td>0.3219</td>
<td>0.8047</td>
<td>0.8177</td>
</tr>
<tr>
<td>99</td>
<td>7.0017</td>
<td>0.3276</td>
<td>0.8190</td>
<td>0.8151</td>
</tr>
<tr>
<td>149</td>
<td>7.1204</td>
<td>0.32</td>
<td>0.8</td>
<td>0.8126</td>
</tr>
<tr>
<td>199</td>
<td>6.2190</td>
<td>0.3238</td>
<td>0.8395</td>
<td>0.8350</td>
</tr>
<tr>
<td>249</td>
<td>6.2936</td>
<td>0.3371</td>
<td>0.8429</td>
<td>0.8491</td>
</tr>
</tbody>
</table>

**4.2.7 Text Features**

The model achieved a precision@5 of 0.3333, recall@5 of 0.8333, and NDCG@5 of 0.8788. The performance of the model with only text features is weaker than its performance with video and audio features or video and text features, but it is still superior to its performance with audio features alone. NDCG@5 is greater than that of the audio features model, indicating that the order of the recommended hashtags is better when text features are used. The background music or tracks used in TikTok videos may or may not be associated with the content or hashtags used to annotate the reel, resulting in noise in the feature representation. This may make it more difficult for the model to convey the importance of hashtags based solely on their audio properties. Although audio features are less useful on their own, when combined with video features, they can assist in making more accurate recommendations. Precision@5, recall@5, and NDCG@5 of the model leveraging only text features over the epochs are shown in the figures below.
Figure 26. Precision@5 for text features

Figure 27. Recall@5 for text features
Below table summarizes the model's performance at corresponding echos

TABLE 9

SUMMARY OF MODEL PERFORMANCE FOR TEXT FEATURES

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Average_loss</th>
<th>Testing_precision</th>
<th>Testing_recall</th>
<th>Testing_NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>9.5062</td>
<td>0.3257</td>
<td>0.8142</td>
<td>0.8482</td>
</tr>
<tr>
<td>99</td>
<td>8.5956</td>
<td>0.3409</td>
<td>0.8523</td>
<td>0.8677</td>
</tr>
<tr>
<td>149</td>
<td>5.7385</td>
<td>0.3428</td>
<td>0.8571</td>
<td>0.8691</td>
</tr>
<tr>
<td>199</td>
<td>6.0783</td>
<td>0.3312</td>
<td>0.8310</td>
<td>0.8568</td>
</tr>
<tr>
<td>249</td>
<td>4.7662</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8788</td>
</tr>
</tbody>
</table>

The dataset provided two positive hashtags for each post made by an individual user, and we have sampled five negative hashtags from the set of all possible hashtags in the dataset for every post made by an individual user. Therefore, the expected maximum precision and recall for the top 5 recommendations are given by
Maximum precision@5 = 2/5 = 0.4
Maximum recall@5 = 2/2 = 1.0

Table 10 gives a detailed summary of precision@5, recall@5, NDCG@5 for different combinations of video, audio, and text features.

**TABLE 10**
SUMMARY OF FEATURE ANALYSIS

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision@5</th>
<th>Recall@5</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video, Audio, Text</td>
<td>0.3581</td>
<td>0.8952</td>
<td>0.9156</td>
</tr>
<tr>
<td>Video, Audio</td>
<td>0.3505</td>
<td>0.8762</td>
<td>0.8855</td>
</tr>
<tr>
<td>Video, Text</td>
<td>0.3486</td>
<td>0.8714</td>
<td>0.8760</td>
</tr>
<tr>
<td>Audio, Text</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8485</td>
</tr>
<tr>
<td>Video</td>
<td>0.3581</td>
<td>0.8952</td>
<td>0.9015</td>
</tr>
<tr>
<td>Audio</td>
<td>0.3371</td>
<td>0.8429</td>
<td>0.8491</td>
</tr>
<tr>
<td>Text</td>
<td>0.3333</td>
<td>0.8333</td>
<td>0.8788</td>
</tr>
</tbody>
</table>

The highest values for each of the three metrics were achieved by the combination of all three features (video, audio, and text), indicating that using all three features together produces the most effective recommendations. The performance of the combination of the video and audio features was also satisfactory, but it was inferior to that of the combination of all three features. This demonstrates that audio features are more effective at generating recommendations when combined with video features. However, text features are also quite essential, as they outperform
audio features when trained without video features. Using audio or text features alone resulted in a lower level of performance compared to using video features, indicating that video features play a more significant role in the process of generating recommendations. Overall, NDCG@5 was the most sensitive to changes in the evaluated feature combinations, whereas precision@5 and recall@5 had very similar patterns. It suggests that the NDCG@k is a more reliable metric for measuring the effectiveness of the recommendation system.

4.3 Performance Comparison with Existing Approaches

Wei et al., [25] proposed a model for personalized hashtag recommendation (PHR), which is based on a graph neural network that considers user, microvideo, and hashtag information. In addition to incorporating information from the video and user profiles, the model employs an attention mechanism to determine the relative importance of various elements in the final recommendation.

Cao et al., [27] designed a hashtag recommendation model (LOGO) that makes personalized suggestions by incorporating visual, audio, and textual data modalities. The model learns to incorporate data from each modality using parallel LSTMs, and an attention mechanism enables the model to focus on the features that are most important to each user. Table 11 compares the performance of our model GDLHR to that of PHR and LOGO.

We tested the existing approaches using our dataset and not the datasets included in the paper itself as we did not find any works that utilized TikTok dataset for hashtag recommendation.
### TABLE 11

**SUMMARY OF THE PERFORMANCE COMPARISON USING OUR DATASET**

<table>
<thead>
<tr>
<th>Metrics</th>
<th>PHR</th>
<th>LOGO</th>
<th>GDLHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision@5</td>
<td>0.3238</td>
<td>0.3124</td>
<td>0.3581</td>
</tr>
<tr>
<td>Recall@5</td>
<td>0.8095</td>
<td>0.7810</td>
<td>0.8952</td>
</tr>
<tr>
<td>NDCG@5</td>
<td>0.8202</td>
<td>0.7710</td>
<td>0.9156</td>
</tr>
</tbody>
</table>

On comparing PHR and GDLHR it can be concluded that GDLHR outperforms PHR in all three metrics. Precision@5 computes the proportion of the top five recommended hashtags that are accurately recommended. GDLHR has a higher precision@5 rating of 0.3581 than PHR, indicating that GDLHR suggests relevant hashtags with greater accuracy. Recall@5 determines the percentage of pertinent hashtags suggested in the top five results. GDLHR's recall@5 score of 0.8952 is higher than PHR's recall@5 score of 0.8095, indicating that GDLHR is more effective at identifying relevant hashtags. NDCG@5 measures the relevance of the recommended hashtags by assigning a higher score to pertinent hashtags that appear higher in the list of recommendations. GDLHR has a higher NDCG@5 score of 0.9156 than PHR, indicating that GDLHR is more effective at suggesting hashtags that are ranked higher in the recommendation list. Precision@5, recall@5, and NDCG@5 of PHR using video, audio, and text over the epochs are shown in the figures below.
Figure 29. Precision@5 of PHR

Figure 30. Recall@5 of PHR
The results also indicate that GDLHR outperforms LOGO in terms of precision@5, recall@5, and NDCG@5. This indicates that GDLHR can provide better hashtag recommendations for reels compared to LOGO. GDLHR has a higher precision, indicating that a greater proportion of the suggested hashtags are relevant to the user's preferences. Additionally, GDLHR has a higher recall, indicating that a greater proportion of pertinent hashtags are included in the recommendations. Lastly, because GDLHR has a higher NDCG, the recommended hashtags are arranged in a more useful and relevant manner. Overall, these results indicate that GDLHR is a more effective model for recommending personalized hashtags for reels. Precision@5, recall@5, and NDCG@5 of LOGO on using video, audio, and text over the epochs are shown in the figures below.
Figure 32. Precision@5 of LOGO

Figure 33. Recall@5 of LOGO
Figure 34. NDCG@5 of LOGO
5. CONCLUSION AND FUTURE WORK

We presented a personalized hashtag recommender system for reels that makes use of graph neural networks and an effective feature engineering technique. We use the relationships between users, videos, and hashtags to produce personalized hashtag recommendations for each user. In comparison to previous techniques, our model produces more accurate and relevant suggestions by adopting a graph-based representation of the data and combining information from many modalities, including video, audio, and text features.

Our method involves building a graph of users, videos, and hashtags, with edges representing the links between them. We used graph convolutional layers, such as GCNConv, GATConv, and GraphSAGEConv, to capture various elements of the graph structure and build node embeddings. We also used negative sampling to train our model and avoid overfitting famous hashtags. We tested our model on the TikTok dataset and found it to be effective in recommending relevant hashtags to users. Our results show the importance of graph-based models and feature engineering in personalized recommendation systems.

Overall, our research suggests a viable method to enhance recommendation algorithms for social media platforms, mainly micro-video based platforms like TikTok, Instagram, and YouTube, where hashtags play an important role in content discovery and virality. To improve the relevance and customization of the recommendations, future research may extend our model to include new variables such as user profiles and post descriptions. Another interesting development would be to add a fourth node, which would represent the user's relationships with their followers. Inclusion of social nodes would mean that the hashtags recommendations could also consider the followers' interests, which would increase the visibility and engagement of the
post. Collaborative filtering could also be leveraged to find the similar followers of a user to further increase the visibility and engagement of the post.
REFERENCES


APPENDIX

Source Code

The source code of our project can be viewed at https://github.com/sriyabalineni/CS298.

Experiment Setup

The project could be run on Google Colab. The scraper code needs a Chrome extension to be downloaded on the device for its execution and to download videos from TikTok. A Step by step guide to installing Selenium and Chrome extensions is available at https://selenium-python.readthedocs.io/.