Graph Neural Networks with Social Media E-Commerce*

Sonia Zeng (1006683082), Bonnie Luo (1006762905), Sherry Li (1006922332)

April 14, 2023

Abstract

This project utilizes Graph Neural Networks (GNNs) to analyze the possible connections between hashtags found in social media posts, with the aim of predicting future product and social media trends for e-commerce businesses. A dataset of 150 hashtags was collected from a particular brand (i.e., Nike) on Instagram and used to train the model, resulting in a graph with 3024 edges and a final accuracy of 68.14%. The model can be applied to any brand to identify potentially popular hashtags that are yet to be discovered, thereby providing inspiration for product development for the next commercial trend. To enhance the model's efficiency and effectiveness, future work will involve incorporating edge features to the GNN and conducting data analysis across multiple platforms and brands. Last but not least, the ethical implications of the project were also evaluated from various perspectives, including marketing, sustainability, and ethics, to remind companies to use the framework responsibly and promote healthy and sustainable purchasing behaviour.

1 Introduction

Social media has become an increasingly essential tool for e-commerce brands, where they can understand relationships between different products, develop products catering to new trends, and eventually gain a competitive advantage in the market [1]. Traditionally, companies rely on the experience and intuition of industrial analysts or influencers for trend prediction. While these methods may be effective, their performance in trend identification is poorer than data-driven methods using statistical models or machine learning [2].

This project aims to take full advantage of social media posts about products, from both companies and users, to analyze potential connections between *hashtags* using Graph Neural Networks (GNNs), in which way to predict future product and social media trends. By analyzing the connection between hashtags used in the existing posts and other hashtags, we aim to help e-commerce businesses identify new directions for innovations, indicated by those promising new hashtags.

2 Method Overview

2.1 Data Collection

The data for this project was obtained from Instagram, a social media platform that is popular among younger users who frequently post user experiences on the platform [3]. To obtain the

*All scripts of this project can be found in this repository.

initial set of hashtags, we scraped the hashtags used by a selected brand account as the source of our database. We then collected other related hashtags in our domain of interest to form a pool of hashtags. Part of the data becomes the training set, which goes through the data processing as illustrated in Fig. 1. Those training data form a connected graph, where nodes represent hashtags and edges represent connections between hashtags (Fig. 6), as the training input to the model.

2.2 Link Prediction

To predict potential connections between hashtags, we implemented *negative sampling* by treating existing edges as positive examples and sampling a set of non-existing samples as negative examples (Fig. 5). We then divided the subgraphs of positive and negative examples into training, validation, and testing sets. These sets were fed into a GNN for further link prediction.

The GNN was selected as the primary link prediction algorithm due to its ability to analyze graph-structural data, which is the natural representation of data in an e-commerce social network. Unlike other machine learning frameworks, GNN naturally integrates node information as well as topological structure, making it more powerful in learning graph representations [4]. In this way, the existence of an edge between two arbitrary nodes in a graph can be predicted.

Furthermore, nodes represent Instagram hashtags, while *node features* and *edge features* can be used to indicate the occurrence frequency of each hashtag, and the co-occurrence frequency of a particular pair of hashtags. These features can thus be passed down to the model for the computation of pair-wise scores as discussed in Section 5.

2.3 Application

The trained GNN-based model can be used to predict future products and social media trends. For a relatively sparse network between hashtags that have already been used by the brand and potential new hashtags, the model could help predict linking that is not already present in the network. Eventually, the model produces new potential hashtags as shown in Model Application in Fig. 1.

3 Prior Work

3.1 GNNs

The primary prior work used in the project is GNNs, which uses graphs, a non-Euclidean data structure for machine learning [5], for node classification, link prediction, and clustering tasks.

A graph is denoted as $G = \{V, E\}$, where $|V| = N$ is the number of nodes in the graph, and $|E| = N^e$ is the total number of edges with e representing the number of edges for one node, assuming all nodes have the same degree. There are several types of graphs:

- **Directed/Undirected graphs**: In a directed graph, the connections between the nodes have a specific direction, whereas in an undirected graph, the connections are bidirectional. In this sense, the direction of connections indicates causal relationship between variables, or the order of events, while undirected connections contain correlation information.
- **Homogeneous/Heterogeneous graphs**: A homogeneous graph is a graph where all nodes belong to the same type or category, while the nodes in a heterogeneous graph are different. Homogeneous graphs are often used in the context of graph-based classification or regression tasks, where the goal is to predict a target variable based on the attributes of the nodes

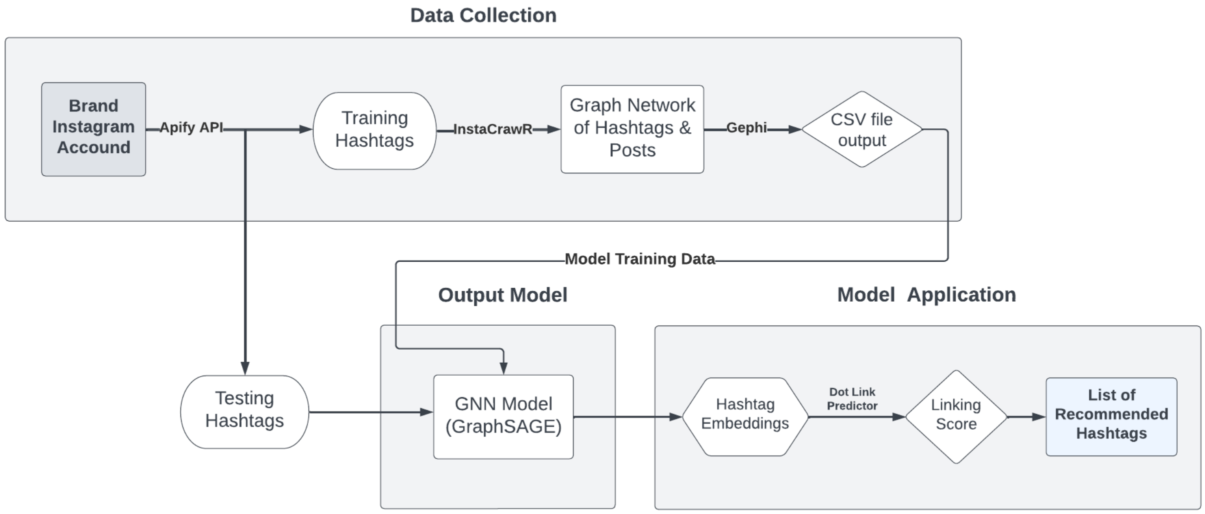


Figure 1: The overall workflow of the GNN-based Social Media E-Commerce Recommender. The pool of hashtags collected from the brand's Instagram account is split into training set and testing sets. The training set is further processed to be graph inputs, and then CSV files, used for training the GNN model. Whereas the testing set is used to take advantage of the output model to generate hashtag embedding, link prediction score between hashtags, and eventually, the list of recommended hashtags for the brand.

and edges. In contrast, heterogeneous graphs are often used in the context of graph-based recommendation systems, where the goal is to recommend items or products to users-based on their preferences or past behaviour.

For our purpose of the project, we decided to use **undirected homogeneous** graph to represent hashtags that appear in the same post, which have more correlation relation over causation.

The GraphSAGE convolution can also be used for our project, which is a general *inductive* framework that generates node embeddings for previously unseen data by sampling and aggregating neighbouring node's features [6]. Compared to other embedding methods using matrix factorization, GraphSAGE utilizes nodes features (e.g., node degree, node centrality, node eigenvector similarity, etc.) to generate inductive embeddings for new nodes with a set of aggregation functions which are capable of aggregating information from nodes located at various distances or depths from the given node (Fig. 2). As for the aggregation calculations, three methods have been proposed:

- Mean aggregator: take the mean value of the embeddings of the node and its neighbors
- LSTM aggregator: permutes the nodes randomly as the nodes are not in sequence
- Pooling aggregator: an element-wise pooling function on the neighboring nodes.

They have also designed an unsupervised loss function for GraphSAGE, which allows it to be trained without the requirement of task-specific supervision, following prior research on generating node embeddings.

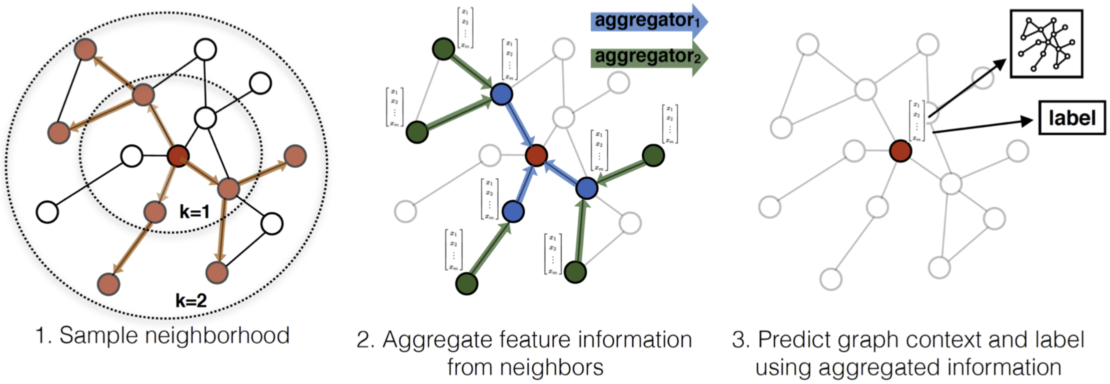


Figure 2: GraphSAGE's approach to sample and aggregate neighbor nodes' features with 3 steps

The embedding generation algorithm used by GraphSAGE is introduced in the Algorithm below as in the original paper [6]. In a graph of $\mathcal{G}(\mathcal{V}, \mathcal{E})$, all nodes have their corresponding features $\mathbf{x}_v$ as the input for the GraphSAGE algorithm. By sampling a fixed size of neighbour nodes, different but uniform samples are obtained per batch and aggregated to a single vector $\mathbf{h}^{k-1}_{\mathcal{N}(u)}$. In this step, AGGREGATE $_k, \forall k \in \{1, \dots, K\}$ stands for the K aggregator functions which aggregate information from neighbours.

After this, the GraphSAGE algorithm concatenates the current node's vector representation $\mathbf{h}^{k-1}_v$ with that of its neighbours $\mathbf{h}^{k-1}_{\mathcal{N}(u)}$, before computing a dot product with a set of weight matrices $\mathbf{W}^k, \forall k \in \{1, \dots, K\}$, which propagates information between different layers of the model, and passing to a nonlinear activation function σ. Consequently, nodes can collect information from further nodes and obtain a label with all aggregated features.

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\textsc{aggregate}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \to 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}^0_v \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \textsc{aggregate}_k(\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}^k_v \leftarrow \sigma\left(\mathbf{W}^k \cdot \textsc{concat}(\mathbf{h}^{k-1}_v, \mathbf{h}^k_{\mathcal{N}(v)})\right)$
  end
  $\mathbf{h}^k_v \leftarrow \mathbf{h}^k_v / \|\mathbf{h}^k_v\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}^K_v, \forall v \in \mathcal{V}$

Figure 3: Pseudo code for GraphSAGE's algorithm to generate embedding.

3.2 Apify API

We can construct a dataset containing hashtags relevant to our domain via Apify API, which returns a specific number of posts with the targeted hashtag in the JSON or CSV formats. The Apify API can also be accessed using python by import the `apify-client` PyPI package. With our own scripts of obtaining and adjusting the dataset, Apify can be well integrated into our pipeline, while ensuring the dataset is inclusive and general enough.

3.3 InstaCrawlR

InstaCrawlR is a GitHub repository of four scripts, including jsonReader, hashtagExtractor, graphCreator, and g2gephi. By using these scripts, we can

- download the latest posts for any designated hashtag,
- export files that shows post ID, URL, number of likes, account ID, post data and location,
- automatically extract associated hashtags and download images from the posts,
- collect related hashtags and compute frequency,
- generate and export a graph that displays the connections between primary and related hashtags.

For example, when we want to evaluate customer feedback from a company on Instagram, the primary input for InstaCrawlR will be all hashtags that this company's brands have used. After this, a social network analysis will be run to scrape all other related hashtags and build up the hashtag database. Note that as we want to build an undirected (a.k.a. bidirectional) graph, we will set this up specifically with InstaCrawIR and use the results for the next step.

However, the functionality of scraping posts are now no longer available, as a result of Instagram's restriction on API use to prevent spamming behaviour and data exploitation [7]. Thus, we decided to use Apify API to reach the same purpose.

4 Data Processing

4.1 Scraping Posts by Hashtags

Hashtag is a very convenient way for brands to label products on Instagram, infer user preferences user and product relationships, and create customer groupings for a product. This project aims to help brands that are in domains where Instagram is used heavily in propagating and advertising products, where analyzing hashtags becomes meaningful. The Apify API was used to scrape hashtags that are related to the target brand.

We aim to collect hashtags that are (1) related to the target brand's products, and (2) may help the brand gain new perspectives, by satisfying these two requirements. The first requirement is satisfied by using the hashtags that are used in the target brand's Instagram posts as the initial query, to branch out and search for other related hashtags. To satisfy the second requirement, we increase the "depth" of each branch of search, such that we might reach some "interesting" hashtags that may be related to the target brand more subtly, which provides more insight into the next prospective innovation for the brand. The hashtag collection workflow is illustrated in graph Fig. 4.

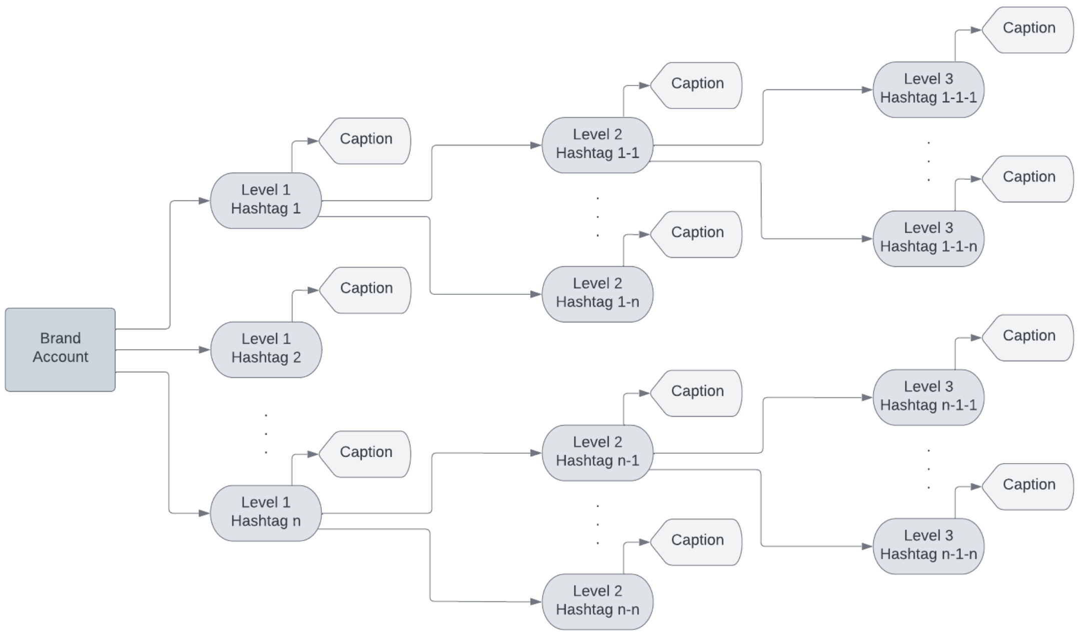


Figure 4: An illustration of collecting multiple levels of hashtags from the target brand's account.

4.2 Preprocessing Data using Python Scripts

We also pre-process the dataset obtained from the last step using Python to generate a form that can be passed down to R scripts and become a graph input. We randomly selected 50 level-3 hashtags to be passed down to create graphs. Level-3 hashtags are not too far down the branches from initial account hashtags, so they are still closely related to the brand, as well as to each other, which is important for maintaining the accuracy of the model. On the other hand, level-3 hashtags contain more insightful information as they are further branched out, which is important for providing new perspectives. We collect the information of all posts that mentioned at least one of these hashtags, including post ID, URL, and post caption. These hashtags and the related post information are passed down as the source hashtags as illustrated in step 1 of Stage 1 in Fig. 5.

4.3 Creating Graphs using R Scripts

After the previous two steps, the clean data are organized in the .csv file format with columns "ID, URL, Text". The open-source GitHub repository [7] provides some basic R scripts to help build connections between different hashtags and compute the weight of each edge (i.e., the frequency of two hashtags' co-occurrence).

hashExtractor.R scans through all the post texts and extracts related hashtags by tokenizing them. It also captures the frequency of each hashtag for later usage and organizes the hashtags with their frequency in .csv format. Note that hashtags with all degrees are preserved (even hashtags that may only appear once in the entire network), such that any novel information and insights introduced by rare hashtags are maintained in the network. Then, Stage 1 of the entire process is finished.

graphCreator.R reads from the .csv file generated from the previous script and generates an edge matrix. Then it calculates the degree of each hashtag (i.e number of connections to other hashtags). To improve the link prediction of the hashtags that have a lower frequency, the ones with a lower degree will be more focused. Graphs will then be generated using `induced.subgraph` function as shown in Fig. 6.

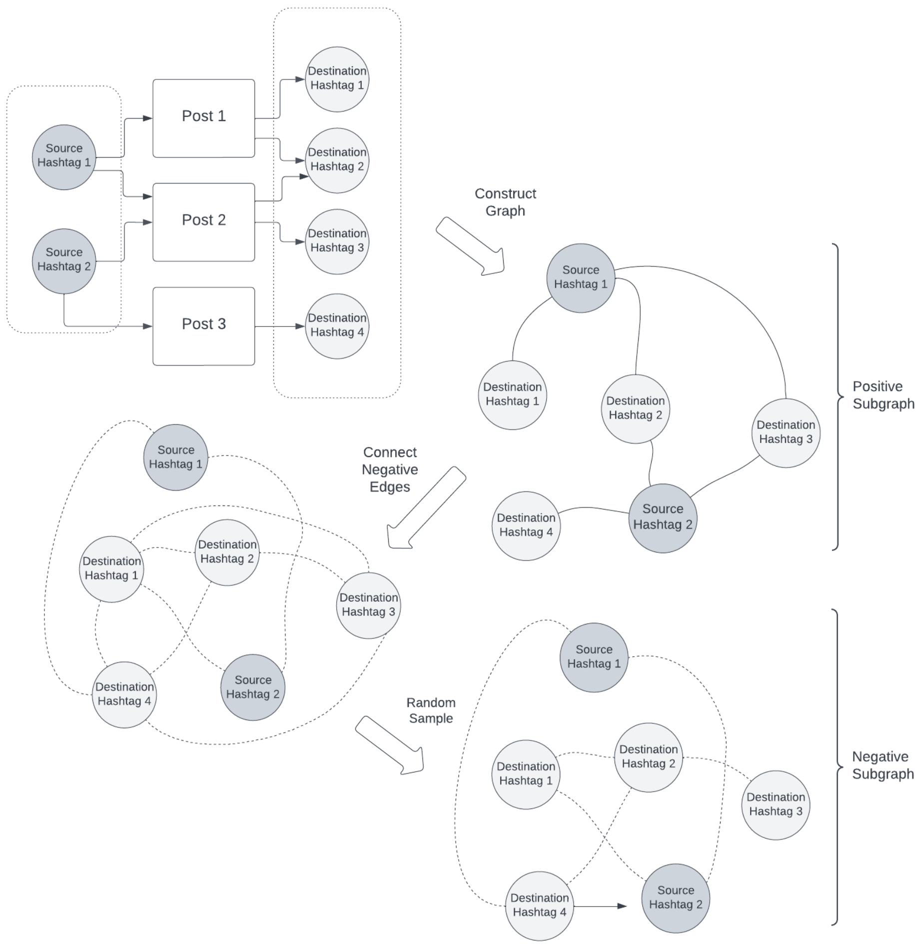


Figure 5: Data gathering and processing pipeline, including four stages which are introduced in Section 4.2 and Section 4.3. Stage 1, 2, 3 and 4 are ordered as what the arrow points to.

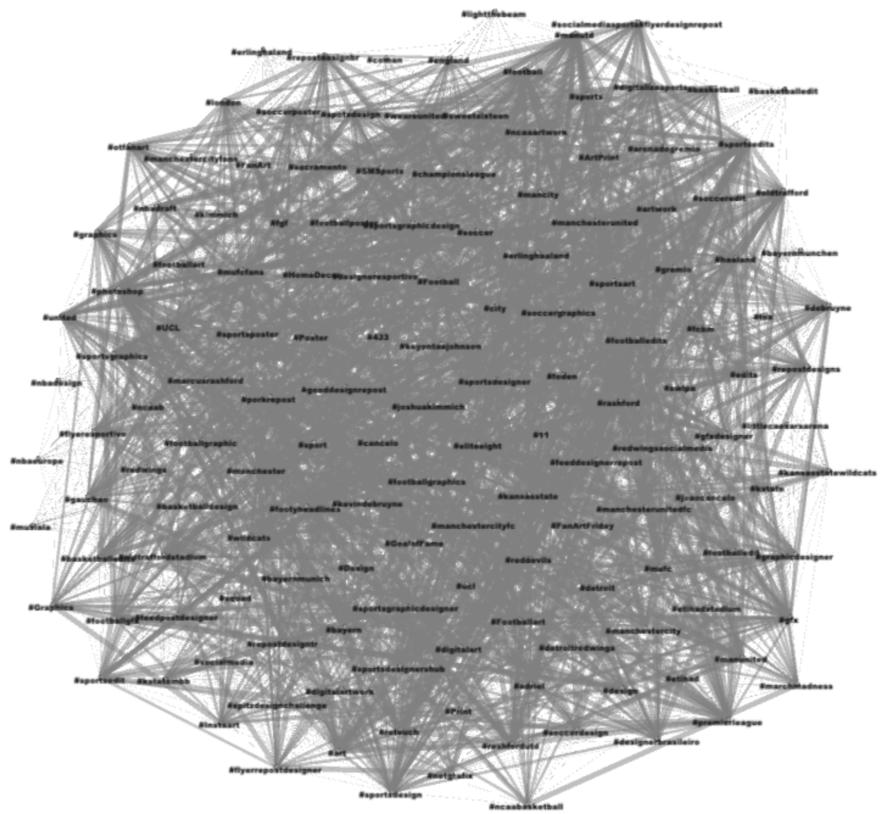


Figure 6: The generated graph with dark grey edges representing strongly related edges and light grey edges representing weakly related edges.

4.4 Subgraphs

With the edge information exported from the R scripts, an original graph of social networks is obtained, which is also called a *positive* subgraph (Stage 2 of Fig. 5) containing all existing edges. A fully connected graph can thus be generated by permuting all nodes in the graph as demonstrated in Stage 3. The non-exisiting edges are named as *negative* edges, by randomly sampling which and reaching to the same number of positive edges, a negative subgraph is acquired (Stage 4). Now, the positive and negative subgraphs can be passed to the GNN model as inputs.

5 Model

5.1 Model Implementation

According to the paper, the mean operator can be seen as a "skip connection" as it does not concatenate the node's previous layer with the current layer of the neighboring nodes, it is proved to improve the performance of the model [8]. The project will use the mean aggregator to update the embeddings of the nodes according to the formula:

$$h_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(h_v^{k-1} \cup h_u^{k-1}, \forall u \in \text{N}(v))) \tag{1}$$

where h is the embeddings, v is the updated node, k represents the k-th layer, $N(\cdot)$ are the connected nodes.

Word2Vec is used to represent the feature vector of each node (hashtag) in the graph. The hashtag names are long truncated words, thus they are first passed into a `segment-hashtag` function, and then `glove-twitter-25` library maps each segmented word to a vector. Lastly, averaging all vectors for each hashtag provides the overall feature vector of the hashtag.

After the graph inputs pass through two SAGEConv layers and an activation function (Fig. 7), the model outputs the node embeddings for the entire graph, such that the node information of all sampled neighbours (h_u^{k-1}) and its own embedding from last iteration (h_v^{k-1}) are aggregated to individual nodes as in Fig. 2 which is denoted as h_v^k.

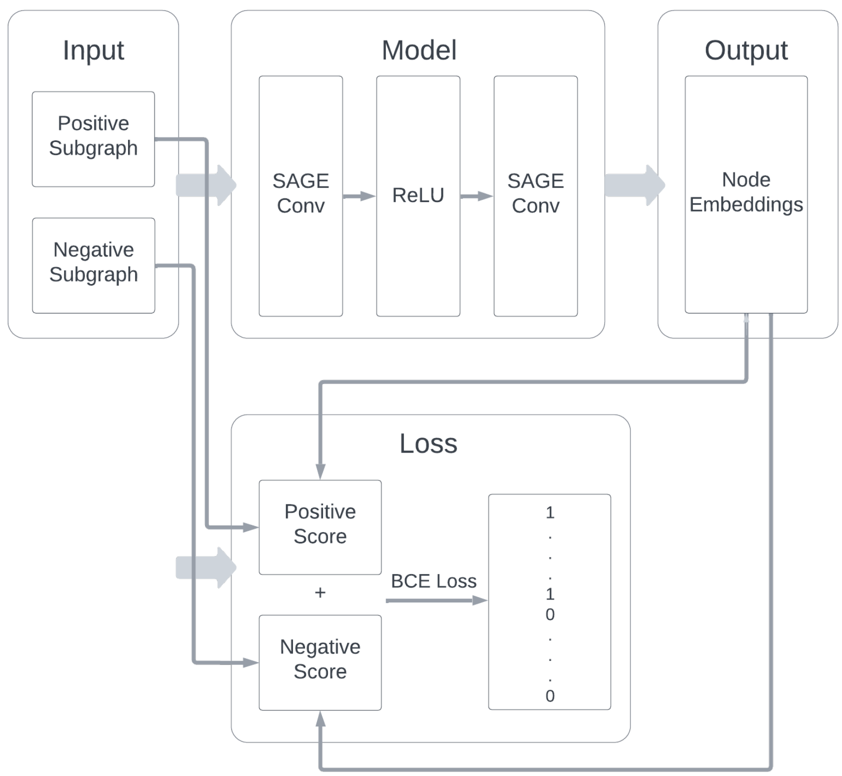


Figure 7: The model architecture used in the paper.

5.2 Model Training

The **objective of the training process** is to build a model that maximizes the similarity between the node embeddings learned from the model and the original node features in the dataset. After sending the positive and negative subgraphs into the model and passing through two GraphSage submodules (Fig. 7), the model will output the predicted node embeddings. To calculate the loss, the function of `DotPredictor` is used to compute the edge embeddings from the node embeddings and compared them with the ground truth label (connected edges are labeled as 1 and unconnected edges are labeled as 0).

Following the guidelines outlined in Deep Learning Tuning Playbook [9], we tuned the hyper-parameters of our model to obtain optimal performance. For instance, we first adjusted our model architecture by changing the number of ConvSage layers as shown in Fig. 8 (Left). As the accuracy of using 3 or more ConvSage layers decreases the accuracy, we decided to keep the architecture unchanged. We also evaluated the performance of different activation functions (Middle) and decided to use `relu` by comparing the final accuracy of Train Set 1 and 6 (Fig. 9).

More importantly, we reflected on the dataset input to our model. At first, we only passed nodes with a number of connections less than 80, since the final goal is to predict connections between the nodes that are not related. By keeping only the nodes with fewer connections, the model might be able to have a better estimation for the final link prediction. However, we later found out that training in the entire dataset would let the model understand the graph features better and make more accurate predictions (Right).

Following this, we also changed the number of epoch that the model should run from 50 to 30, which turned out that the model performance is improved. Together with the other findings above, we plotted the figure below (Fig. 9).

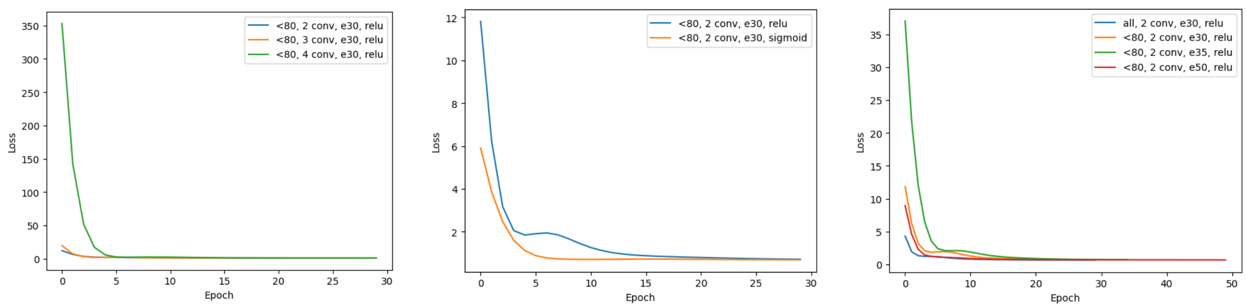
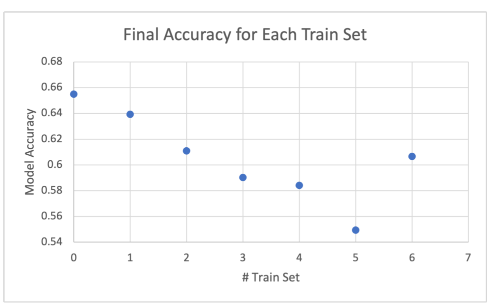


Figure 8: Loss Curves for Different Training Sets: *left* figure fine tunes on the number of layers; *middle* figure tunes on the activation function; *right* figure fine tunes on the epoch number and dataset size



Train Set	Data	# ConvSage	Epoch	Activation
0	all	2	30	Relu
1	partial	2	30	Relu
2	partial	2	35	Relu
3	partial	2	50	Relu
4	partial	3	30	Relu
5	partial	4	30	Relu
6	partial	2	30	Sigmoid

Figure 9: Final Accuracy for each train set with their hyperparameter settings

According to the above statistics, three conclusions are made:

- an epoch number of 30 is better than 35 and 50.
- using 2 ConvSage layers is better than using 3 or more ConvSage layers.
- using all datasets is better than filtering out the hashtags with more connections.

6 Result

We used a particular brand, Nike, on Instagram as our initial query, then scraped 10 users for each brand, resulting in 150 total hashtags and 3024 edges.

After training the model with this dataset over 30 epochs, the final test accuracy is 68.14 %. Compared to the ratio in the original dataset (50% of connected edges and 50% of unconnected edges), the model can predict the link between some unrelated nodes and therefore can be used to detect the future trend.

In a practical sense, a brand may build a network of all used hashtags in its existing posts. Then, new hashtags may be introduced to the built network for analyzing connections with the used hashtags. Negative edges with high predicted probability point to the hashtags that can be potentially popular and trigger a new trend, but are yet discovered as demonstrated below. Consequently, companies can develop products catering to this new trend indicated by those promising hashtags or send posts with these hashtags to improve brand popularity and gain competitive advantages.

Source	Destination
#NikeFC	#socceredit
#AF1JoinForces	#designerbrasileiro
#OurBeautifulGame	#london
#nike	#sportsdesign
#FeelYourAll	#rashford
#Nike50	#premierleague
#NikeForward	#gauchao
#JustDoIt	#rashford
#Yardrunners	#eliteeight
#LebronXX	#footballgraphic

7 Future Work

7.1 Incorporate Edge Features

The current model only takes in the `Word2Vec` embedding of each hashtag as its node feature. However, the edge features could be also added to the graph network as weights on each edge, to help the model better understand the relationship and possible linkage between hashtags. For instance, we may use the occurring frequency of each hashtag pairs as the feature of the edge connecting these hashtags.

7.2 Multi-Platform Data Analysis

At the current stage, we have only been scraping data from Instagram, which might have caused our data to be biased or under-representative since there are other major social media platforms that are heavily used for promoting e-commerce products. For example, Facebook and YouTube has more monthly active users than Instagram [10]. The contents on Facebook and YouTube are also labeled by keywords that are similar to the function of hashtags in Instagram, so we may gather effective data in the form of keywords similarly.

On the other hand, the brand's official website is another important source of data. Since the official website is more product-centric, it is a convenient place to collect customer feedback regarding specific products, e.g. comments under each product listing. By doing semantic analysis on the comments, we may determine the emotions associated with each product, which can be embedded as a node feature for training our GNN model.

7.3 Comprehensive Analysis Across Brands

Our current solution only considers product and trend analysis for a single target brand, and thus, trains the GNN model with hashtags related to only the target brand. However, for multiple brands in the same industry, training the model using data across brands might beneficial as it can capture the industry trend more accurately and introduce new potential nodes.

8 Broader Impacts & Ethical Implications

The GNN-based framework we have developed in this project has broader implications for marketing, advertising, and sustainability. We leverage Instagram hashtags and other post features to gain insight into genuine customer feedback and social media trends. This will result in more effective marketing and advertising strategies, evnetually driving sales growth.

Furthermore, hashtags with high connection possibilities serve as inspirations for companies to develop products, with statistical evidence to support their decisions. This approach can greatly reduce energy consumption and waste production resulting from the creation of unmarketable prototypes, customer samples, and unsold or returned products [11]. Ultimately, this can help mitigate environmental challenges faced by industries such as fast fashion and cosmetics.

We have taken ethical considerations into account while developing our framework. To ensure data privacy, we strictly followed Instagram data policies [12] by scraping only public accounts and removing user-specific information during our training and testing process. As our model is used for inspiring social media and commercial trends, it is less likely to cause spamming or other issues for Instagram users.

However, misusing or overusing our model could lead to potential negative consequences. While our framework promotes sustainability by reducing the development of unpromising products, it also helps companies understand customer preferences and develop products with the potential to be popular, thus encouraging consumerism. This could result in increased energy and resource consumption for industries with high emissions and worsen the sustainability crisis. Although our model is not the direct cause of the environmental issue, it could act as a catalyst for the deteriorating process. Therefore, it is crucial to use our framework ethically and promote rational purchasing habits.

Due to the specific criteria used to gather data for our project, our dataset is biased in several ways. We only collected data from Instagram users who made public posts with captions and used at least one hashtag related to the brand. This sampling method does not ensure demographic fairness and may not represent the broader customer population, particularly those who do not use Instagram. As such, we could only account for the opinions and feedback of those who are willing and able to share their thoughts on this specific social media platform. This issues, however, can be partially alleviated with a more inclusive dataset compassing a greater population.

These ethical concerns underscore the important role that this model can play in social media E-commerce. Our framework provides an example of how to generate a meaningful dataset and train a GNN model to predict links between potential and existing hashtags. This approach can be extended to other platforms with corresponding datasets to achieve a similar purpose. It is also important to note that the model's output does not indicate *causality*, but rather a *correlation* between the existing and potential hashtags. This means that companies should view the model output as a *suggestion*, rather than a *guideline* for their product development. Ultimately, it will rely on the companies to use our model ethically and promote *healthy and sustainable* purchasing behavior among their consumer population.

9 Conclusion

This project was to forecast product and social media trends through the collection of a dataset of Instagram hashtags. Hashtags related to the field of interest were gathered and related hashtags were extracted from posts using Python and R scripts. A GNN model was then trained using the 150 total hashtags collected from a particular brand resulting in a graph with 3024 edges and a final accuracy of 68.14%. By applying the model to any particular brand, promising but yet undiscovered hashtags can be generated, serving as inspiration for the next commercial trend for companies. To improve the model's effectiveness and efficiency, future work will incorporate edge features to the GNN and implement data analysis across multiple platforms and brands. The project's broader impacts and ethical implications were also analyzed from the marketing, sustainability, and ethics perspectives, reminding companies to use the framework ethically and encourage healthy and sustainable purchasing behaviour.

References

[1] M. Singh and Gobindbir Singh. Impact of social media on e-commerce. *International Journal of Engineering and Technology(UAE)*, 7:21–26, 05 2018.

[2] Yunpeng Xiao, Yu Zhu, Weikang He, and Mengyang Huang. Influence prediction model for marketing campaigns on e-commerce platforms. *Expert Systems with Applications*, 211:118575, 2023.

[3] Amandeep Singh, Malka Halgamuge, and B. Mouss. *An Analysis of Demographic and Behavior Trends Using Social Media: Facebook, Twitter, and Instagram.* 01 2019.

[4] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, WWW '19, page 417–426, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[6] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017.

[7] Schroeder Jonas. Instacrawlr. `https://github.com/JonasSchroeder/InstaCrawlR`, 2018.

[8] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017.

[9] Deep learning tuning playbook. `https://github.com/google-research/tuning_playbook`. Accessed: 2023-04-12.

[10] Global social media statistics. `https://datareportal.com/social-media-users`.

[11] The myth of sustainable fashion. https://hbr.org/2022/01/the-myth-of-sustainable-fashion. Accessed: 2023-04-12.

[12] Data policy — instagram help center. https://help.instagram.com/155833707900388. Accessed: 2023-04-12.