JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, XXX-XXX (2022)
DOI: 10.6688/JISE.20220X_38(X).00XX

# Generative Adversarial Networks for Tag Recommendation based on Multimodal and Group Article Features

[1]DUEN-REN LIU, [2,*]CHIN-HUI LAI, [3]YANG HUANG, [4]SHU-TING CHANG
*[1,3,4]Institute of Information Management,*
*National Yang Ming Chiao Tung University*
*Hsinchu 300, Taiwan*

*[2]Department of Information Management*
*Chung Yuan Christian University*
*Taoyuan City 320, Taiwan*
*E-mail: [1]dliu@nycu.edu.tw, [2]chlai@cycu.edu.tw, [3]yang.huang@nycu.edu.tw,*
*[4]stchang.mg08@nycu.edu.tw*

Tags play an important role in helping people find their preferred content. However, tagging is a labor-intensive task; therefore, tag recommendation has gained increasing attention. Most of the extant literature on automated tag recommendation systems mainly considers a single type of data, such as tags based on the article text or image annotation tasks. Very few studies have demonstrated the ability to increase the quality of tag recommendation by considering both textual and visual data. Moreover, similar articles are likely to share some common tags, and article titles are strongly correlated to the tags, which have not been considered by existing studies. In this paper, we propose a novel tag recommendation method based on Generative Adversarial Networks (GAN) considering Multimodal data and Group Article Features. Our proposed method takes the individual article, image, group articles, as well as title information into account and applies a novel co-attention mechanism to extract the relevant and important latent features for successful tagging. Moreover, we develop a GAN-based model to make use of the powerful ability of adversarial learning and effective predictions of the tags. The experimental results show that our proposed method outperforms various representative methods.

***Keywords:*** Recommendation Systems, Tag Recommendation, Multimodal Feature Extraction, Convolutional Neural Network, Co-Attention Mechanism, Generative Adversarial Networks

## 1. INTRODUCTION

To efficiently find information that meets user needs from the massive network data, tagging [1] and marking media information, including articles, images, or videos through a tagging system, has become an important trend in online media research. Tags usually summarize the important content of media information with keywords, and can effectively support information retrieval (IR) services [2], content recommendation, etc., to improve the efficiency of users' searching and browsing experience on the Internet. Most media websites use automated tag recommendations to improve the tag quality of their articles and enhance the possibility of articles being searched to gain more exposure to websites; therefore, automated tag recommendation has become a crucial research topic.

Various recommendation methods, including collaborative filtering [3] and content-based approaches [4-7], are utilized for tag recommendation. Deep learning methods like

[+] Corresponding author. Dr. Chin-Hui Lai, chlai@cycu.edu.tw

RNN, LSTM, and CNN are extensively used in tag recommendation systems [8-11]. Additionally, the attention-based model [11] captures interactive relationships among images, text, and tags, aiding in tag recommendation for multimodal data. These models can also integrate with Latent Dirichlet Allocation (LDA) topic models through co-attention mechanisms [12]. Recently, GANs [13] have gained popularity in the deep learning field. However, research on GAN-based tag recommendation [8, 14-17], particularly in multi-label classification, remains limited. Quintanilla et al. [8] applied GAN to improve image label recommendation quality, mainly focusing on image annotation without considering other related data or the influence of group content on tag recommendations.

Most extant studies on automated tag recommendation technology only consider a single type of data, such as articles tagging based on text features [18] or image annotations [15]. Sparse literature [19, 20] considers multiple types of data and has proved their effectiveness. Therefore, our proposed method will consider the multimodal information (image and text) of an article. In addition, the relevant literature only uses individual articles and their tags to train the tag recommendation model. Nevertheless, using only the individual tag data of articles may be insufficient to achieve a thorough understanding of the data. To solve this issue, we gathered similar articles into a group and made use of the group content since we observed that similar articles are likely to have some common tags. Group information can assist individual articles in seeking to learn similar or popular tags appearing in group articles. Furthermore, since tags would sometimes appear in titles, we also consider title information to strengthen feature extraction.

In this paper, we propose a novel hybrid GAN-based tag recommendation method that considers multimodal data such as images, article content, title, and group article content to strengthen the article feature analysis and extraction. There is no article tag recommendation method considering both group and title information in the relevant literature. The proposed method will combine multimodal data through the attention mechanism [21]. By reweighting each element in the data, it makes neural network model learning more flexible and helps to capture important information. Our study uses the collaborative attention (co-attention) mechanism [22] to effectively extract the features and interactive relationships of multiple types of data through neural networks, and further adopts adversarial learning to optimize tag generation. Despite the success of GAN, there are few related studies on tag recommendation based on GANs. Our model adopts a novel co-attention mechanism to effectively combine multimodal data, including text, image, group, and title information, in the generator. Our aim is to generate realistic tag predictions through competitive learning of GANs; to meet this objective, we developed a novel competitive learning framework for GANs to enhance the accuracy of tag recommendations.

We implemented the proposed method on a media website, NiuNews, and the effectiveness was evaluated and compared. The results demonstrated that our method outperforms several representative methods in the tag recommendation field. The developed tag recommendation method helps the platform to tag articles more efficiently, reduce labor costs effectively, and enable users to find their favorite articles faster through tags, thereby improving platform adhesion, and exhibiting practical application value and contribution.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of related work, and section 3 explains our proposed recommendation approach. Section 4 describes and analyzes the experiment and evaluation results. The final section concludes the research and discusses future work.

# 2. RELATED WORKS

This section introduces related work and the methods adopted in our work.

## 2.1 Tag Recommendation Systems

Tag recommendation systems include: Collaborative Filtering (CF), Content-Based Filtering (CBF), and Hybrid Filtering. The collaborative filtering method [3, 23, 24] constructs tag recommendation models based on the similarity of the user's historical tagging behavior (user-based) or the similarity between items (item-based). The content-based filtering method [4-7, 25] extracts features for item content and recommends tags for new content through similarity between contents and records. The Labeled LDA (Latent Dirichlet Allocation) model [5] is proposed to extend LDA by restricting the correspondence between tags and potential topics to recommend tags.

Hybrid filtering [26, 27] is a method of combining the above two concepts to overcome their respective shortages. A hybrid tag recommender system is designed by combining collaborative filtering with content-based methods [28]; the usage patterns and time characteristics of tags are considered for generating recommendations [29].

## 2.2 Deep Learning Approaches for Tag Recommendation

Several related studies have combined deep learning and tag recommendation. For example, the attention-based convolutional neural network (CNN) model [9] expresses topic tag recommendations as a multi-label classification task that takes the content of the post as input, extracts the feature vector related to the article through the CNN and attention mechanism, and finally outputs the probability of the recommended label. The tag recommendation model based on the LSTM [10] also considers tag recommendation as a classification task, a novel RNN model that learns the representation vector of Twitter content to recommend tags. Related literature [18] proposes organizing article content through a temporal neural network to recommend relevant tags. The model uses the encoder and attention mechanism to model text semantic features through RNN. Besides, the decoder is used to process the correlation of tags through the predicting path.

## 2.3 Tag Recommendation Based on a Multimodal Fusion Framework

In addition to text content, recent online articles often contain other types of data. Some studies [30, 31] use multimodal data to improve the recommendation models. Related literature [11] proposes a multimodal neural network model based on the attention mechanism to capture the potential interactions between images, texts, and tags in microblogs to recommend related tags.

In addition, related research [32] proposed a tag recommendation method based on a memory-enhanced parallel collaborative attention model for photo-sharing services. The model uses content modules to simultaneously model images and texts; it introduces external memory and adopts a parallel collaborative attention mechanism to extract image features and text features.

## 2.4 Generative Adversarial Networks for Tag Recommender Systems

The Generative Adversarial Network (GAN) was proposed by Goodfellow et al. [13]. Although GAN has achieved good results in many applications, there are few related studies on the topic of multi-label classification [8, 15-17]. Related research on text tag recommendation includes a multi-label classification model based on adversarial learning, while image label recommendation includes using adversarial learning architecture to strengthen the generation of label probability distribution to improve image label prediction [17, 33]. It also includes the automatic image annotation model based on the concept of imitating human annotations and the GAN architecture [15], to produce tags that can denote the image content more appropriately. Huang, et al. [34] use the attention method and the adversarial network to learn the common representation vector of multimodal data and apply it to tag recommendation. A personalized image tag recommendation based on GANs was proposed by Quintanilla et al. [8]. Their model uses adversarial learning architecture to learn and predict the probability distribution of tags similar to those generated by users.

In the tag recommendation studies based on the GAN, image tagging is mostly used, while there is less research focus on article tagging. Therefore, the GANs still have great research value in article tagging tasks.

## 3. PROPOSED APPROACH

### 3.1 Overview

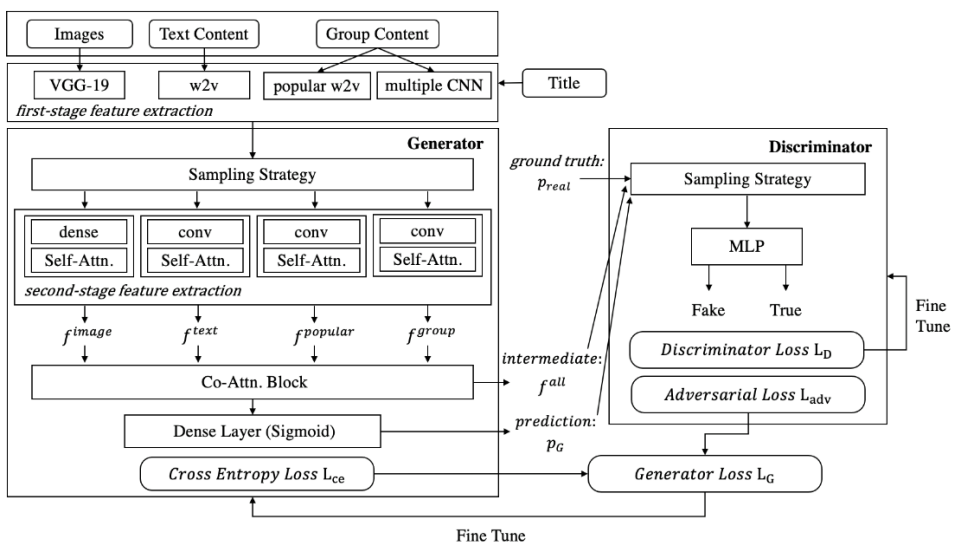


Fig. 1. Overview of the Proposed Recommendation Method

We propose a novel tag recommendation method based on multimodal data features, group article tag feature learning, and GAN. The overall framework of the proposed method is shown in Fig. 1. The steps of our study mainly include the feature extraction method combining clustering and multimodal data with CNNs, using a novel collaborative attention mechanism to learn weights, and constructing a tag prediction model with adversarial learning architecture. The proposed article tag recommendation method includes (1)

two-stage feature extraction models for multimodal data; (2) combining images, individual articles, titles, and group article features to generate integrated article feature representation vectors to enhance the feature analysis and extraction of article content; (3) constructing a new tag recommendation model based on integrated article features and GAN architecture. The competitive learning framework helps to learn the distribution of tag probabilities based on multimodal data and group article characteristics.

Since only using a single article for multi-label classification of tag recommendations may cause the predicted tags to be uncommon or not sufficiently representative of the target article due to the narrow data scope, our method considers three additional data types to make up for the shortcomings of traditional recommendation methods. We observed that similar articles often have similar tags and that tags sometimes appear in the title. Therefore, grouping articles and extracting group article features based on group article tags, as well as title information, can overcome the insufficiency of single article features. In addition, related literature has used multimodal data (images and articles) to improve the accuracy of recommendations and has proven efficiency; thus, our method also incorporates the visual data in the article into feature learning. The proposed method uses four independent CNNs to extract the hidden features of individual article images, individual articles, group popular words, and group articles. Additionally, article titles are incorporated to enhance the feature learning of our textual data, including individual and group articles. The tag recommendation model is implemented by adversarial learning architecture and multi-label classification. First, we used four types of data with pre-trained first-stage feature extraction models to produce primary latent feature vectors. The samples to be trained are input into the second-stage feature extraction models in the generator to produce four latent feature vectors. After finishing the feature extraction, the co-attention mechanism is adopted to adjust the weights of latent features to learn the importance of different types of data for tag recommendation and further predict the classification results. The discriminator uses a multi-layer neural network to perform binary classification to determine whether the input sample is real or fake. The generator network optimizes the multi-label classification prediction through the classification loss function and the reward from the discriminator so that it can produce more accurate labels. The discriminator network will be optimized through the binary classification loss so that it can distinguish between real and fake data. Finally, once the generator and discriminator training converge, the generator can be used to generate accurate tags for the recommendation.

### 3.2 Data Preprocessing and Primary Feature Extraction

At this stage, we pre-processed the news articles and pre-trained three different modality first-stage feature extraction models to extract the primary feature latent vectors. We used the popular architecture handling visual tasks, VGG-19 [35], as our first-stage image feature extraction model. We also pre-trained the word2vec model for all the textual data. For the group article features, a separate CNN model was trained for each group, designed as a multi-label classifier, and trained independently.

(A) Visual Data Preprocessing and Visual Feature Extraction

In the data preprocessing step, we resize all images into 224×224 as input of the pre-trained model. In vision applications, CNN has achieved great success, and many pre-

trained models have achieved excellent results. In this study, we constructed our visual feature extraction model using one of the most popular models, the VGG-19 model and pre-trained it using the ImageNet dataset. We only took the model structure before the last output layer to extract the primary visual latent vector $pf_d^{image}$ for each article $d$.

(B) Textual Data Preprocessing and Textual Feature Extraction

In this section, we first removed stop words in our articles and used Jieba to do Chinese word segmentation. To extract our primary text feature, we pre-trained a word2vec model with 354,158 Chinese wiki documents and text in our dataset. Once the training was finished, the word2vec model could generate the word embedding vector to represent each word. We adopted the word2vec model to produce both article and title latent vectors and concatenated them to be our primary text latent vector $pf_d^{text}$ for each article $d$.

(C) Group Data Preprocessing and Group Feature Extraction

In this part, we first used LDA (Latent Dirichlet Allocation) to group articles. LDA can efficiently classify articles into N topics (groups). Since similar content has a higher probability of being assigned to the same topic, the topic assigned to each article can be seen as its group. Since we observed that articles in the same group often consist of some common words, we posited that words frequently appearing in one group should carry important group information. Thus, we took the top 100 frequent words in each group and produced the primary group popular word latent feature vectors through the word2vec model pre-trained in 3.2.2. The primary group popular word latent feature vector $pf_d^{gpop}$ for article $d$ is the same for the articles in the same group.

Besides, to learn the hidden characteristics in each group, we trained different CNN feature extraction models for each group independently. Since the features extracted represent the group articles' common pattern, we used "group pattern features" $f^{group}$ to denote these features in the following parts. The group CNN models have the same architecture and are trained as a multi-label classification task. The difference between the models is the data to be trained; each model only takes the article data that belong to the corresponding group, where article data are concatenated with article word embedding and title word embedding. We fed the article latent vector $pf_d^{text}$ to the corresponding group CNN model to extract the group's primary pattern feature vector $pf_d^{group}$, as shown in Eq. (1). To pre-train each primary group pattern feature CNN model, we input the group pattern feature vector $pf_d^{group}$ of each article $d$ in the group into the fully connected layer and used activation function σ (sigmoid) to obtain the classification prediction result $p_d^{group}$, which is shown in Eq. (2). We took the corresponding real tags of each article $d$ to be the ground truth $y^d$, and calculated the binary cross entropy loss $(BCE)$ with our predicted tags probability $p_{d,i}^{group}$ of tag *i* for the same article, as shown in Eq. (3). We used this loss function to update our models. Once the model training process was completed, we extracted the vector before the last layer to be our primary group pattern feature $pf_d^{group}$ for article $d$:

$$pf_d^{group} = Group - \mathrm{CNN}(pf_d^{text}), \tag{1}$$

$$p_d^{group} = \sigma\left(W_g \cdot pf_d^{group} + b_g\right), \tag{2}$$

$$BCE(p_d^{group}) = -\frac{1}{Q}\sum_{i=1}^{Q}[y_i^d \log(p_{d,i}^{group}) + (1 - y_i^d)\log(1 - p_{d,i}^{group})], \quad (3)$$

where $Q$ represents the number of tags in the tag set, and $y_i^d$ signifies whether article d has been labeled by tag $i$. An overview of pre-training of the first-stage group pattern feature extraction models is shown in Fig. 2. As mentioned above, we have N group pattern feature extraction models. Each model is trained by the article content as well as the title of the articles that belong to that group.

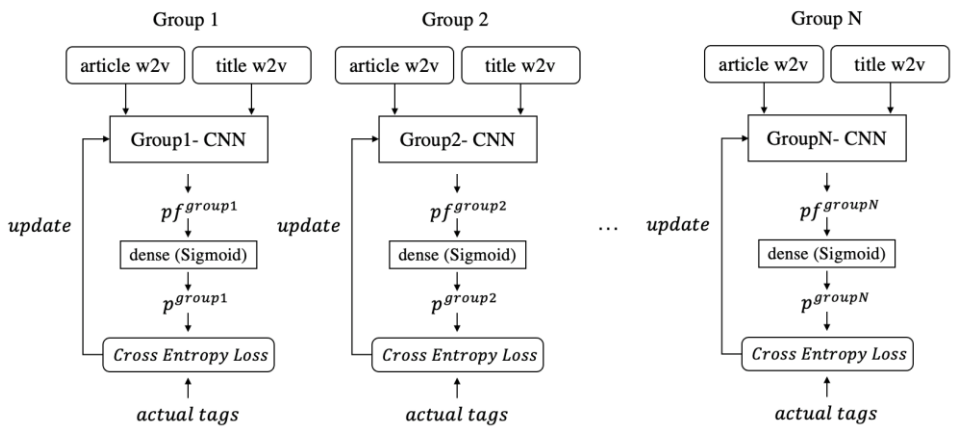


Fig. 2. An Overview of the First-Stage Group Pattern Feature Extraction Models Pre-training

### 3.3 Generative Adversarial Network

Our study developed a reinforcement learning architecture based on the generative adversarial network (GAN) for tag recommendation. GAN includes two competing learning models: the Generator and the Discriminator. In our case, the generator was used to generate the tag prediction probability of the article, and the discriminator was used to distinguish whether the probability was derived from training data or generated by the Generator. The proposed model constructs the convolutional networks as the second-stage feature extraction neural network in the generator and uses the attention mechanism to combine visual, textual, and group features to generate integral article feature representation vectors to enhance the feature analysis and extraction of article content. The generator architecture is shown in Fig. 3.

(A) Generator

The generator in our model adopts four different kinds of data: individual article images, individual articles with titles, group popular words, and group article patterns. It takes the primary features extracted in 0, which are $pf^{image}$, $pf^{text}$, $pf^{gpop}$, and $pf^{group}$ in Fig. 3, as inputs. Then the generator selects samples through a designed sampling strategy that we can increase the weight of samples that have better generation performance to improve the training effect of the generator.

The selected samples are processed into the four second-stage feature extraction model to extract the latent features of four types of data denoted as $f^{image}$, $f^{text}$, $f^{gpop}$, and $f^{group}$ in Fig. 3. Then the co-attention mechanism is adopted to integrate the latent

features to generate a comprehensive feature vector of the article, and finally generate a predicted tag probability through the fully connected layer.

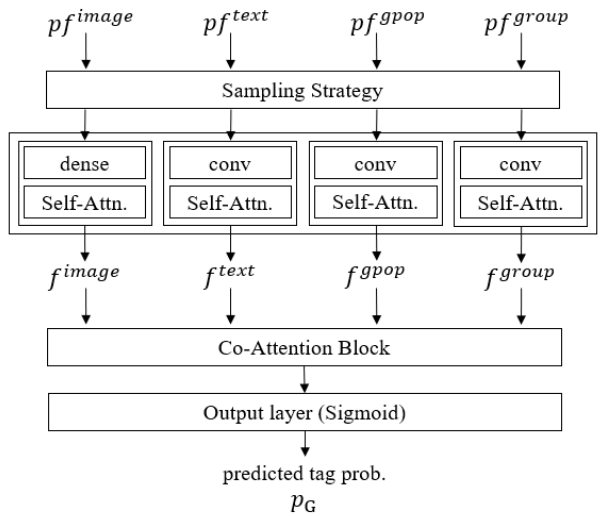


Fig. 3. Generator Model Architecture

Second-Stage News Features Extraction

After we sampled the first-stage features, we fed these features to our second-stage extraction models. For visual features, we used a fully connected layer to map the features into the same dimension of the textual feature vectors and also added a self-attention layer to learn the relationship between different parts in our visual feature vector, as shown in Eq. (4).

$$f_d^{image} = \text{self-attn}\left[\tanh\left(W_t \cdot pf_d^{image} + b_t\right)\right] \tag{4}$$

Our feature extraction models are mainly composed of CNN to extract the textual latent features of the data. We apply three convolution layers with different kernel sizes to extract second-stage feature vectors for individual articles, group popular words, and group articles. A self-attention layer is added to the three feature extractors, as shown in Eqs. (5) ~ (7):

$$f_d^{text} = \text{self-attn}[t\text{-CNN}(pf_d^{text})] \tag{5}$$

$$f_d^{gpop} = \text{self-attn}\left[p\text{-CNN}\left(pf_d^{gpop}\right)\right] \tag{6}$$

$$f_d^{group} = \text{self-attn}\left[g\text{-CNN}\left(pf_d^{group}\right)\right], \tag{7}$$

where $pf_d^{text}$, $pf_d^{gpop}$, $pf_d^{group}$ denote the primary latent feature vectors of individual article content, group popular words, and group article patterns of article $d$, respectively.

Self-attention [21] can capture important information by considering the neighborhood context. In our second feature extraction models, we added the self-attention mechanism to all of them before we generated our latent feature vectors. The self-attention is implemented as follows:

$$h_{t,t'} = \tanh\left(W_t x_t^T + W_{t'} x_{t'}^T + b_t\right) \tag{8}$$

$$\alpha_{t,t'} = \sigma\left(W_\alpha h_{t,t'} + b_\alpha\right) \tag{9}$$

$$l_t = \sum_{t'} \alpha_{t,t'} x_{t'} \tag{10}$$

The attention weight $\alpha_{t,t'}$ captures the relationship between the input features $x_t$ and $x_{t'}$ at regions/timesteps $t$ and $t'$, respectively. σ is the element-wise sigmoid function. The attentive feature representation $l_t$ for each region/timestep $t$ is the weighted sum over the other context features. In our case, we split the visual features into different regions and the textual features into different timesteps to replace $x_t$.

Integrating News Features with Co-attention Mechanism

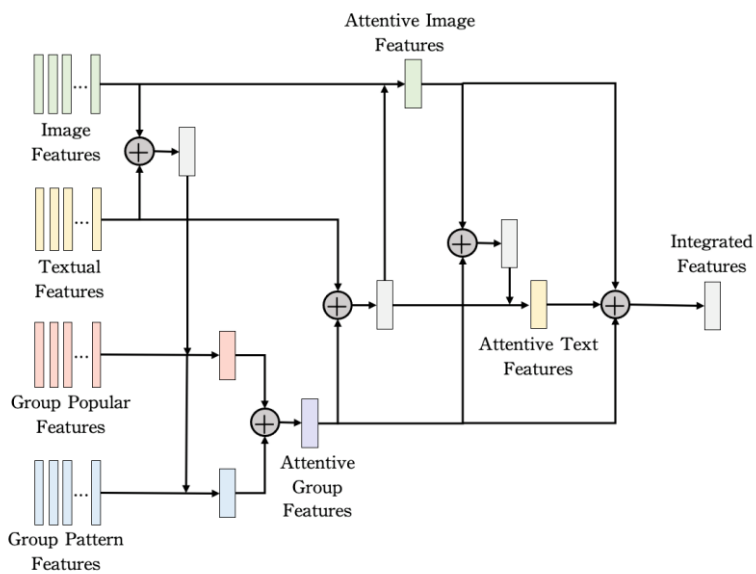


Fig. 4. Co-attention Block

After obtaining all the latent feature vectors: $f^{image}$, $f^{text}$, $f^{gpop}$, and $f^{group}$, we adopted our innovative co-attention structure to integrate them. We introduced the idea of co-attention presented in [34] and made some modifications to make it applicable in our case and improve the effect. Since individual data seems to be more important than group data, we first use individual image and text data to guide our group's popular words and group pattern feature. Besides, the text often contains more information than the image does, so we attend to the image before the text. Fig. 4 shows our co-attention block.

We separate the alternating co-attention into three phases. In phase 1, the new group

representation is generated by comparing the original image and textual features to the group-related features including group popular features and group pattern features separately. We first add up the mean of the image features and the mean of text features to derive our individual information. Then we calculate the attention weights for different parts of the group popular features and group pattern features according to the individual information. Take group popular features for example. We use a single $tanh$ layer to learn the relationship between group popular information and individual information and a following $softmax$ layer to produce the attention weight. Lastly, we weigh different parts in our original group's popular features $f^{gpop}$ and sum them up to generate the attentive group popular representation vector $Af^{gpop}$. The calculating details are shown in Eq. (11).

$$\begin{aligned} h_p &= \vartheta\left(W_{f^{gpop}} \cdot f^{gpop} \odot W_{f^{it}} \cdot (v^{image} + v^{text})\right), \\ a_p &= \text{softmax}\left(W_{h_p} \cdot h_p + b_p\right), \\ Af^{gpop} &= \sum a_p \cdot f^{gpop}, \end{aligned} \tag{11}$$

where $f^{gpop} \in \mathbb{R}^{e \times T}$; $v^{image}, v^{text} \in \mathbb{R}^d$ represent the mean of $f^{image}$ and $f^{text}$ respectively; $e$ is the embedding size; $T$ is the maximal length of each article; $+$ denotes the element-wise addition; $\odot$ denotes the operation of concatenating each column of the 2d array and the 1d vector; $\vartheta$ is the $tanh$ activation function. The generated vector $Af^{gpop}$ also has the dimension of $e$, that is, $Af^{gpop} \in \mathbb{R}^e$.

We also generate the attentive group pattern representation vector $Af^{group}$ through the same process, as shown in Eq. (12).

$$\begin{aligned} h_g &= \vartheta\left(W_{f^{group}} \cdot f^{group} \odot W_{f^{it}} \cdot (v^{image} + v^{text})\right), \\ a_g &= \text{softmax}\left(W_{h_g} \cdot h_g + b_g\right), \\ Af^{group} &= \sum a_g \cdot f^{group}, \end{aligned} \tag{12}$$

Then we combine the attentive group popular representation vector $Af^{gpop}$ and group pattern representation vector $Af^{group}$ to generate the attentive group representation vector $f^{pop-grp}$ by simply summing them up with balance weight $\gamma$ and $\omega$, as shown in Eq. (13), where $f^{pop-grp} \in \mathbb{R}^e$:

$$f^{pop-grp} = \gamma \cdot Af^{gpop} + \omega \cdot Af^{group} \tag{13}$$

In phase 2, we use the original text feature vector and our new group feature vector to guide the image features. This approach is similar to what we did in phase 1; the weights are calculated for different regions in the image vector based on the mutual influence between image information and text-group information. Then we derive the weighted sum of the image region vector as the attentive image representation vector $Af^{image}$, as shown in Eq. (14):

$$h_i = \vartheta\left(W_{f^{image}} \cdot f^{image} \odot W_{f^{tg}} \cdot (v^{text} + f^{pop-grp})\right),$$
$$a_i = \text{softmax}(W_{h_i} \cdot h_i + b_i), \quad (14)$$
$$Af^{image} = \sum a_i \cdot f^{image},$$

where $f^{image} \in \mathbb{R}^{e \times I}$, $e$ is the embedding size, and $I$ is the number of regions in an image. Other symbols have the same meanings as mentioned above. The generated vector $Af^{image}$ also has the dimension of $e$, that is, $Af^{image} \in \mathbb{R}^e$.

In phase 3, we use the attentive image feature vector and new group feature vector to guide the text features. This approach is also similar to the above, that is the weights are calculated for different sequence parts in the text vector based on the mutual influence between text information and the image-group information. Then we can derive the weighted sum of the text sequence vector as the attentive text representation vector $Af^{text}$, as shown in Eq. (15):

$$h_t = \vartheta\left(W_{f^{text}} \cdot f^{text} \odot W_{f^{ig}} \cdot (Af^{image} + f^{pop-grp})\right),$$
$$a_t = \text{softmax}(W_{h_t} \cdot h_t + b_t), \quad (15)$$
$$Af^{text} = \sum a_t \cdot f^{text},$$

where $f^{text} \in \mathbb{R}^{e \times T}$ and $T$ is the maximal length of each article. Other symbols have the same meanings mentioned above. The generated vector $Af^{text}$ also has the dimension of $e$, that is, $Af^{text} \in \mathbb{R}^e$.

Finally, we combine all the attentive representation vectors via Eq. (16) to generate our integrated feature vector $f^{all}$ for later use:

$$f^{all} = f^{pop-grp} + Af^{image} + Af^{text}, \quad (16)$$

where $f^{pop-grp}$, $Af^{image}$, and $Af^{text}$ denote the attentive representation of group, image, and textual features, respectively.

Tag Probability Generation and Generator Model Updating

After the generator G generates the article integration feature $f^{all}$ of all the sample data via Eq. (16), we take the feature $f_d^{all}$ for every single article $d$ to predict its generated tag probability $p_G^d$ through a fully connected layer, as shown in Eq. (17). The update of the parameters in the generator is based on the classification performance and reward of the discriminator. The loss function mainly includes the binary cross-entropy loss function $L_{ce}$ (Eq. (18)) of multi-label classification and the reward loss function $L_{adv}$ (Eq. (19)) of the discriminator. $D(p_G^d)$ is the probability that the discriminator distinguishes whether $p_G^d$ is true or not. If the tag probability generated by the generator can easily fool the discriminator, this reward function can be minimized. Therefore, the generator will try to generate a tag probability that is close to the pattern of the real probability. In this way, the generator can generate accurate tags. The loss function $L_G$ of the generator is shown in Eq. (20):

$$p_G^d = \sigma(W_{all} \cdot f_d^{all} + b_{all}) \tag{17}$$

$$L_{ce} = -\frac{1}{|S_G|} \cdot \frac{1}{Q} \sum_{d \in S_G} \sum_{i=1}^{Q} [y_i^d \log(p_{G,i}^d) + (1 - y_i^d) \log(1 - p_{G,i}^d)] \tag{18}$$

$$L_{adv} = -\frac{1}{|S_G|} \sum_{d \in S_G} \log\left(1 - D(p_G^d)\right) \tag{19}$$

$$L_G = \alpha \times L_{ce} + \beta \times L_{adv} \tag{20}$$

$S_G$ is the sample set of the generator; Q represents the number of tags in the tag set; $i$ represents the order of tags; $d$ represents the order of articles; $y_i^d$ represents whether the tag $i$ is used to tag article $d$; $p_{G,i}^d$ is the probability of tag $i$ generated by the generator when predicting article $d$. The generator model, including feature extraction models and co-attention architecture, is updated by backpropagation using the loss function $L_G$. After multiple iterations of the generative adversarial learning, the training of the discriminator and generator both converge.

(B) Discriminator

The objective of discriminator D is to distinguish between the predicted distribution by generator G and the real distribution; it also gives the reward to the generator according to the discriminating result each time. The method employed here mainly refers to related literature [17] that generates tag probability and real tag probability as the input of the discriminator. In our study, we also concatenate the tag probability with our integrated feature $f^{all}$ as a condition to guide the discriminator to make better distinctions. We use a multilayer perceptron (MLP) as the discriminator architecture, as shown in Fig. 5.

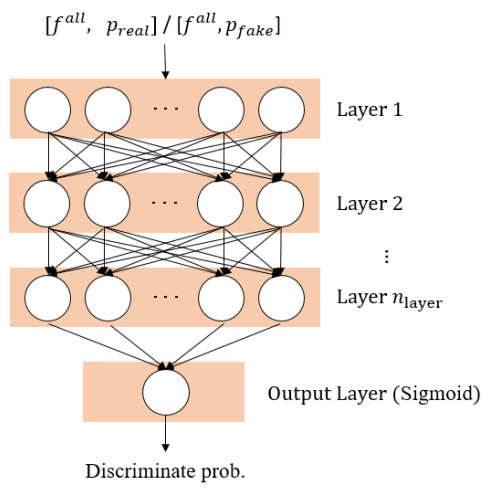


Fig. 5. Discriminator Model Architecture

The input of D is the integrated feature $f^{all}$ concatenating with the real tag probability distribution $p_{real}$ or the generated tag probability distribution $p_G$. After the multi-

layer neural networks and the single node of the last sigmoid layer, the discriminator outputs the probability that the input is recognized as true. A closer value to 1 represents the input is closer to the real tag probability; conversely, the closer the value is to 0, the more likely the input tag probability was generated by the generator.

Discriminating Between Real and Generated Tag Probability

In our model, a multilayer perceptron (MLP) was adopted to construct a discriminator. Through the multi-layer activation function, the discriminator can finally identify the reality of the tag probability. If the output probability is closer to 1, the discriminator considers the sample to be more real. Conversely, the discriminator concludes that the sample is more likely to be generated by the generator. The probability of each tag in ground truth is either 0 or 1, that is, in determining whether it is used to label the sample article. However, the tag generated by the generator is a probability value between 0-1, which makes it too easy for the discriminator to find the difference and identify the reality. Therefore, we introduce a modification method referred to [8] to adjust the input $p_{truth}^{d}$ of the real label to $p_{real}^{d}$, so that the discriminator can be effectively trained. The conversion method is shown in Eq. (21):

$$p_{real}^{d} = \left(p_{truth}^{d} \times \text{random}(\eta,\ 1)\right) + \text{random}(0,\ \varepsilon) \tag{21}$$

The values contained in $p_{truth}^{d}$ are all 0 or 1, which represents whether the tag is used to label the sampled article $d$. $\eta$ is the critical value, and $\varepsilon$ is the upper bound of the random function. Assuming that $\eta$ = 0.25 and $\varepsilon$ = 0.001, through Eq. (21), the probability of a tag originally used for labeling being converted to a probability value is between 0.25 and 1.0, and the probability of a tag not used for labeling being converted to a probability value is between 0.0 and 0.001. Through this conversion, the difficulty of the discriminator's recognition can be increased, further achieving a better training effect.

Discriminator Model Updating

The discriminator uses the binary cross entropy to calculate the loss function $L_D$. This result can be achieved by minimizing the objective function shown in Eq. (22), which is used to evaluate the quality of discrimination between the real tag probability $p_{real}^{d}$ and the generated tag probability $p_{G}^{d}$ of article $d$:

$$\begin{aligned} L_D = & -\frac{1}{|S_{real}|} \sum_{d \in S_{real}} \log D\left(\left[f_d^{all},\ p_{real}^{d}\right]\right) - \\ & \frac{1}{|S_G|} \sum_{d \in S_G} \log\left(1 - D\left(\left[f_d^{all},\ p_{G}^{d}\right]\right)\right), \end{aligned} \tag{22}$$

where $f_d^{all}$ denotes the article integrated feature of a single article $d$ derived from Eq. (16); $D([f_d^{all},\ p_{real}^{d}])$ and $D([f_d^{all},\ p_{G}^{d}])$ are the probability of discriminating $p_{real}^{d}$ and $p_{G}^{d}$ to the real, respectively. $S_{real}$ is the set of real samples, while $S_G$ is the set of samples generated by the generator. To minimize the loss function, the discriminator should let $D\left(\left[f_d^{all},\ p_{real}^{d}\right]\right)$; the probability of discriminating $p_{real}^{d}$ as real is when it is as close to 1 as possible. Conversely, for the probability of discriminating $p_{G}^{d}$ as real,

$D([f_d^{all}, p_G^d])$, it should be closer to 0. After generative adversarial training for many iterations, the discriminator and generator's training process will converge. At that time, $D([f_d^{all}, p_{real}^d])$ and $D([f_d^{all}, p_G^d])$ should approach 0.5, which means that the generated tag has successfully confused the discriminator.

### 3.4 Top-K Recommendation

The generator and the discriminator are trained alternatively until convergence, at which time the discriminator is no longer able to discriminate whether the sample is real or fake and the generator can generate tag probability whose distribution is close to the true probability.

Finally, our main purpose was to recommend tags relevant to the new articles. Hence, once we obtained the article's information, we fed it into our trained generator to generate the corresponding tag probability. Tags with higher probability indicate that they are more relevant to the news and have a higher possibility of being chosen by the user to tag the article; therefore, we selected the top K tags ranked by the predicted tag probability to recommend.

## 4. EXPERIMENT AND EVALUATION

### 4.1 Dataset

Our experiments were conducted on a real-world news dataset, obtained from a website named NiusNews (https://www.niusnews.com/). We collected 9,530 articles. After filtering out the articles with no image and the low frequency (appearing less than 15 times in all articles) tags, the experiment dataset contains 5,796 articles, 42,804 vocabularies, and 266 tags. Each article in the dataset involves posted time, title, text content, image, channel ID, author ID, and tags. We randomly selected 80% of our data as the training set (4,643 articles), and the other 20% as the testing set (1,153 articles).

### 4.2 Experimental Setting

After numerous tests, we determined our parameters as described below. For images in our dataset, we resized them into 224x224 and then fed them into a VGG-19 net pretrained by the dataset. We adopted the output of the last pooling layer as the image features. For text contents, we first used Jieba to eliminate stop words and tokenize them. Then we trained our word2vec model by Chinese wiki documents and our documents to generate text feature vectors, whose embedding size was set to be 200 while the max length of every article was 50. For the number of groups, we tried 10, 15, 20 and ultimately determined to use 15 groups since this yielded the best performance. For adversarial learning, we set the D step as two iterations and the G step as one iteration; that is, the generator would train double that of the discriminator. We used Adam as our optimizer. The learning rate of the generator and discriminator was set to 0.0008 and 0.00001, respectively. The batch size was set to 64.

To evaluate our performance, we adopted six metrics: hit ratio (Hit), precision (P), recall (R), F1-score (F1), Normalized Discounted Cumulative Gain (NDCG), and Mean

Average Precision (MAP). The hit ratio represents the percentage of correct recommendations (which has at least one tag that matches the ground truth) among all the recommendations. Precision indicates the percentage of correct recommended tags in our recommended list. Recall denotes the percentage of correct recommended tags in the actual tags. Lastly, the F1-score could be used to observe the balance between precision and recall. NDCG and MAP focus on the ranked position of correct tags in the recommendation list.

### 4.3 Evaluation

(A) Ablation Study

Some ablation experiments were designed to prove the efficiency of newly added components in our model. In Table 1, we first compared our proposed method with all deconstructed methods of our model when recommending 5 tags. The effectiveness of every element we added to our model was clearly shown in the table since the proposed method outperformed all deconstructed methods under different metrics. We show the effectiveness of each element in the following section.

**Table 1. Ablation study showing the comparison of performance under top-5 recommendation**

| Methods | Hit@5 | P@5 | R@5 | F1@5 | NDCG@5 | MAP@5 |
|---|---|---|---|---|---|---|
| Text-CNN | 0.732697 | 0.259081 | 0.596102 | 0.33302 | 0.546746 | 0.487988 |
| Image-CNN | 0.467823 | 0.167875 | 0.347595 | 0.209822 | 0.305591 | 0.262801 |
| Group-CNN | 0.626366 | 0.222515 | 0.488704 | 0.281876 | 0.443929 | 0.388872 |
| CoA | 0.802602 | 0.289679 | 0.67901 | 0.37516 | 0.610386 | 0.546541 |
| CoA+Group | 0.802775 | 0.292281 | 0.684986 | 0.37863 | 0.613572 | 0.549906 |
| CoA+Group+Title | 0.817693 | 0.298213 | 0.700233 | 0.38647 | 0.629304 | 0.565877 |
| CoA+GAN | 0.808846 | 0.293356 | 0.687871 | 0.379846 | 0.616825 | 0.552877 |
| CoA+Group+GAN | 0.809714 | 0.29412 | 0.691911 | 0.381234 | 0.6213 | 0.558182 |
| CoA+Group+Title+GAN | 0.822203 | 0.298664 | 0.706188 | 0.387771 | 0.63302 | 0.569553 |
| **CoA+Group+Title+CGAN(proposed)** | **0.833131** | **0.302515** | **0.715027** | **0.392805** | **0.643685** | **0.579663** |

Effectiveness of Multimodal Data and Co-attention Mechanism

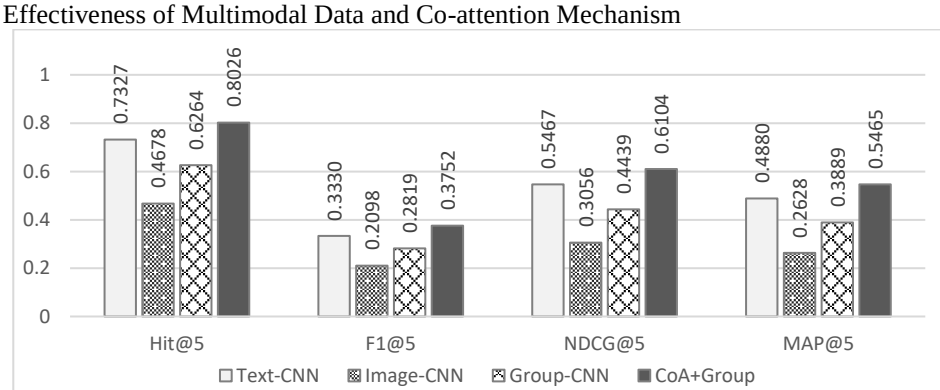


Fig. 6. Effect of considering multimodal data

In Fig. 6, we compare the model using multimodal data (CoA+Group) with the models using only one type of data (Text-CNN, Image-CNN, Group-CNN). We maintain the same basis of the model to focus only on the effects of data information. The results show that CoA+Group outperforms the other methods, demonstrating the effectiveness of considering more than one type of data. We can also see that the model using text as input has better performance than using image or group data as input, which shows that text content might provide more important information for tag recommendation compared to the image and group content.

Effectiveness of considering title information

In Fig. 7, we compare the model that considers title information (CoA+Group+Title, CoA+Group+Title+GAN) with those that do not (CoA+Group, CoA+Group+GAN). The results show a slight improvement when adding title information to our input data, demonstrating that the title information can help improve the recommendation effect to some extent.

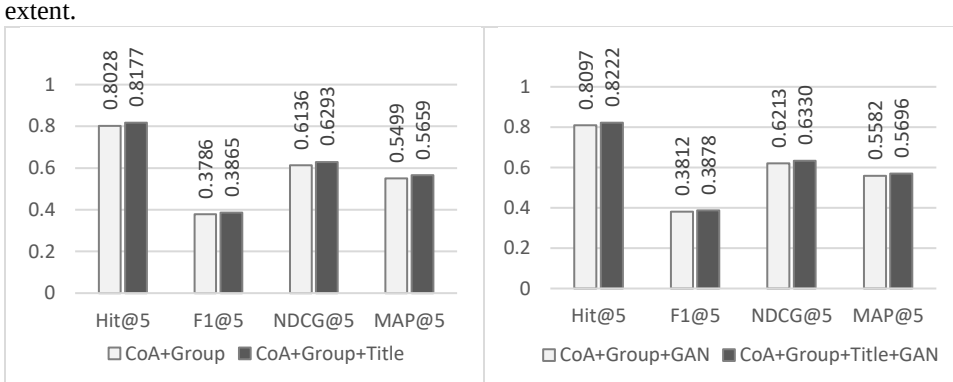


Fig. 7. Effect of considering title

Effectiveness of Adversarial Learning

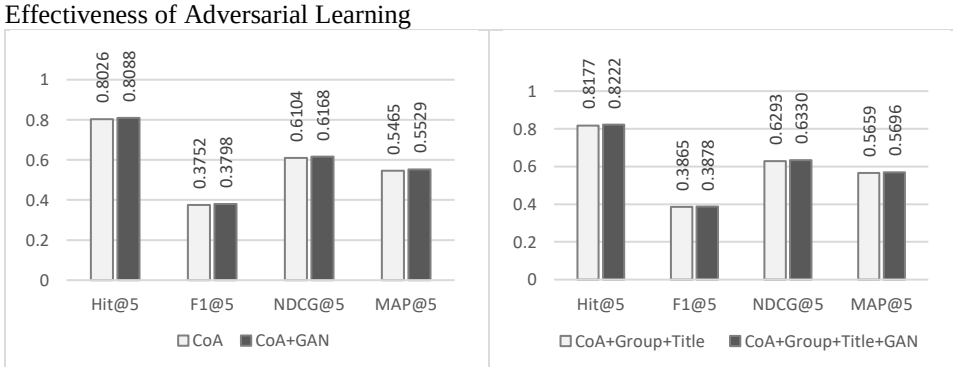


Fig. 8. Effect of adversarial learning

In Fig. 8, we compare the model that adopts adversarial learning (CoA+GAN, CoA+Group+Title+GAN) with those that do not (CoA, CoA+Group+Title). Since the generator already has a strong ability itself, adding the discriminator in the models only slightly improves the performance.

Effectiveness of Condition in GAN

In Fig. 9, we compare the model that adds integrated features as conditions in the discriminator (CoA+Group+Title+CGAN) with those that do not (CoA+Group+Title+GAN). The results show the improvement of the model by adding conditions in the discriminator, demonstrating that adding conditions to the discriminator can boost the generator's production of tag probabilities that are more similar to the real tag probabilities.

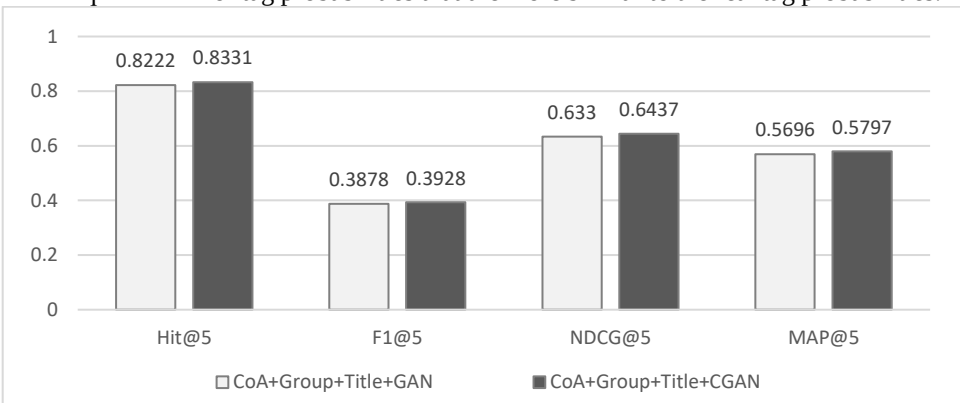


Fig. 9. Effect of condition on the discriminator

(B) Comparison with Other Methods

We compared the results of our proposed method to some baselines and other existing methods.

- **SVM:** A discriminative model proposed by [36]. We implemented a multi-label classifier using the MultiOutputClassifier function provided by Scikit-learn. For its input, we took the sum of pre-trained word embedding as the word feature.
- **Tag2Word:** A content-based tag recommendation method proposed by [25]. The authors considered the tag-content co-occurrence and designed a generative model to generate words according to the tag-word distribution.
- **iTag:** A deep neural network tag recommendation model based on seq2seq proposed by [18]. Their input sequences are textual content, and the output sequences are the tags. The model consists of three main elements: textual content modeling, tag correlation, and content-tag co-occurrence.
- **ABC:** An attention-based CNN model proposed by [9] for hashtag recommendation; it also treated the hashtag recommendation as a multi-label classification problem. The model performed feature extraction by global and local attention channels, targeting the whole document or the important words, respectively.
- **TLSTM:** A topical attention-based LSTM model proposed by [12] for hashtag recommendation. It incorporates LDA topic distribution into LSTM to combine the word embeddings and topic vectors through an attention mechanism.
- **CoA:** The co-attention hashtag recommendation model was proposed by [19]. It leverages the co-attention approach to model the interactions between the textual and visual information of microblogs and then recommends hashtags relevant to the multimodal information.

In this part, we reproduce the existing models above and try to maintain the conditions

of the models as closely as possible to make a fair comparison. We also use our pre-trained word2vec model to produce the word embeddings in all the models as we did in our proposed model.

**Table 2. Performance comparison of existing methods under top-2 recommendation**

| Methods | Hit@2 | P@2 | R@2 | F1@2 | NDCG@2 | MAP@2 |
|---|---|---|---|---|---|---|
| SVM | 0.060538 | 0.034085 | 0.02339 | 0.025807 | 0.039082 | 0.033348 |
| Tag2Word | 0.060997 | 0.030498 | 0.019418 | 0.021916 | 0.027756 | 0.019759 |
| iTag | 0.392 | 0.272 | 0.318667 | 0.29349 | 0.289629 | 0.263333 |
| ABC | 0.576236 | 0.387598 | 0.38269 | 0.351917 | 0.473377 | 0.444579 |
| TLSTM | 0.654987 | 0.442671 | 0.449604 | 0.408147 | 0.545877 | 0.515048 |
| CoA | 0.678404 | 0.452645 | 0.462528 | 0.419158 | 0.56202 | 0.528968 |
| **Proposed** | **0.707199** | **0.476409** | **0.491746** | **0.443473** | **0.594232** | **0.562012** |

**Table 3. Performance comparison of existing methods under top-5 recommendation**

| Methods | Hit@5 | P@5 | R@5 | F1@5 | NDCG@5 | MAP@5 |
|---|---|---|---|---|---|---|
| SVM | 0.08621 | 0.022689 | 0.037878 | 0.026392 | 0.037933 | 0.027214 |
| Tag2Word | 0.121993 | 0.024914 | 0.038272 | 0.028034 | 0.032281 | 0.01757 |
| iTag | 0.466667 | 0.151289 | 0.336622 | 0.208756 | 0.252346 | 0.200283 |
| ABC | 0.708586 | 0.254536 | 0.581117 | 0.326619 | 0.517055 | 0.456908 |
| TLSTM | 0.77693 | 0.284441 | 0.660191 | 0.366793 | 0.59122 | 0.529245 |
| CoA | 0.802602 | 0.289679 | 0.67901 | 0.37516 | 0.610386 | 0.546541 |
| **Proposed** | **0.833131** | **0.302515** | **0.715027** | **0.392805** | **0.643685** | **0.579663** |

**Table 4. Performance comparison of existing methods under top-10 recommendation**

| Methods | Hit@10 | P@10 | R@10 | F1@10 | NDCG@10 | MAP@10 |
|---|---|---|---|---|---|---|
| SVM | 0.108586 | 0.014814 | 0.049338 | 0.021551 | 0.042632 | 0.029061 |
| Tag2Word | 0.167182 | 0.017887 | 0.061923 | 0.026042 | 0.040609 | 0.020139 |
| iTag | 0.397333 | 0.075733 | 0.296337 | 0.120636 | 0.189970 | 0.178076 |
| ABC | 0.802428 | 0.154709 | 0.707942 | 0.239343 | 0.564515 | 0.483583 |
| TLSTM | 0.856028 | 0.168465 | 0.777936 | 0.261019 | 0.635502 | 0.555527 |
| CoA | 0.879445 | 0.170633 | 0.795267 | 0.26507 | 0.653932 | 0.571854 |
| **Proposed** | **0.901821** | **0.176496** | **0.826672** | **0.274462** | **0.685848** | **0.605088** |

We evaluate all methods with the metrics we mentioned in 0 under different numbers of recommended tags. The results of comparison under the top 2, 5, and 10 recommendations are listed in Tables 2, 3, and 4, respectively. We interpret the results in the following discussion: First, it shows that the methods adopting deep learning models generate much better results than those using traditional methods like SVM and Tag2Word. Second, the methods using the attention mechanism to integrate other information, including ABC, TLSTM, CoA, and our proposed method, can achieve even better performance, which demonstrates the effectiveness of the attention mechanism. Third, methods considering multimodal data, including CoA and our proposed method, have better performance than those using only one type of data, which demonstrates that multiple types of data can benefit the task. Finally, the results show that our proposed model outperforms the other methods, demonstrating that the elements added to our method, including adversarial learning architecture and additional information like group features and titles, can bring benefits to

our model.

We achieved at least 3.5% F1-score improvement, 4.9% NDCG improvement, and 5.8% MAP improvement compared to the existing methods under different numbers of recommended tags. We also plotted the curves of different metrics in Fig. 10. Each point of a curve represents the number of the recommended tags. We can see that our proposed method always yields better performance than the others, even when higher numbers decrease the precision and F1-score, demonstrating the effectiveness and stability of our method. The improvement of our proposed method can be credited to two aspects: for the data aspect, we combined the group features and title information with the article and image information through a novel co-attention mechanism. For the model aspect, we used CGAN as our core architecture to assist the model training.

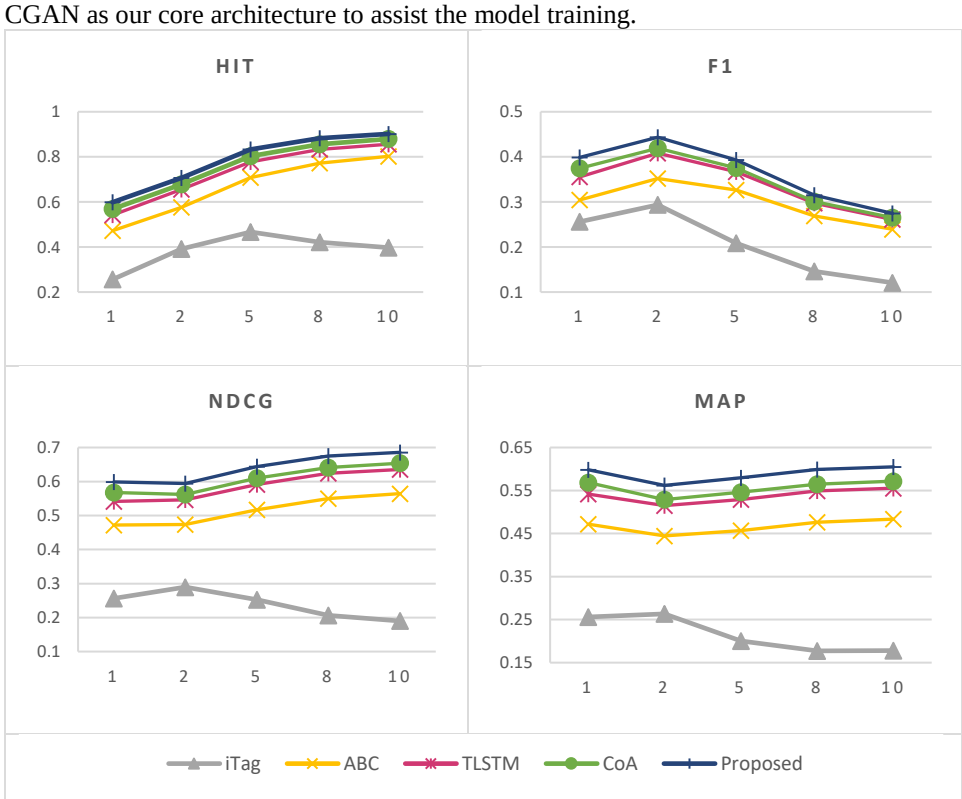


Fig. 10. Performance comparison of existing methods with different amounts of recommended tags

## 5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel GAN-based tag recommendation model considering multimodal data. For the feature extraction, we applied two-stage feature extraction models to explore the important information in the multimodal data, including text, image, group, and title content. Group and title information helped the model to better analyze the article content to recommend appropriate tags, which was not considered in other existing models. We then adopted a novel co-attention architecture to integrate all the features and

obtain an integral representation of all the types of data. Our tag recommendation model is based on a Generative Adversarial Network (GAN); through the competitive learning of the generator and the discriminator, the generator can learn to produce tags that are more similar to real tags.

Several experiments were conducted on a dataset collected from a news website NiusNews to evaluate our proposed model. The results show that our proposed model can enhance the recommendation result and obtain better performance than all the methods. The results of the ablation study also demonstrate the effectiveness of every element we added to our model. The tags generated by our model can help users find news articles similar to their preferred article efficiently with a significant chance to extend their dwell time on the news website. Our model also has great potential in other practical applications, such as generating metadata like tags or labels for any type of data, and can also be applied to multiple kinds of data.

In the future, we will work on finding solutions to generate new tags that did not appear in past articles. Since we model the tag recommendation as a classification problem, it can only predict the tags that have appeared in the history, which limits the diversity of tags. To overcome this limitation, we can give greater consideration to the co-occurrence of tag content or tag titles in the future to generate new tags in addition to the tags in our tags pool. Moreover, we can combine topic-related articles and article-related popular search words to augment our data and further enhance the effectiveness of the tag recommendations.

## ACKNOWLEDGEMENT

This research was supported by the National Science and Technology Council of Taiwan under grant number: NSTC112-2410-HA49-015-MY2 and NSTC 112-2410-H-033-016.

## REFERENCES

1. E. Zheng, G. Y. Kondo, S. Zilora, and Q. Yu, "Tag-aware dynamic music recommendation," *Expert Systems with Applications,* Vol. 106, pp. 244-251, 2018.
2. D. Jeong, S. Oh, and E. Park, "DemoHash: Hashtag recommendation based on user demographic information," *Expert Systems with Applications,* Vol. 210, p. 118375, 2022.
3. X. Fang, R. Pan, G. Cao, X. He, and W. Dai, "Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin Texas, USA, 25-30 Jan 2015: AAAI, p. 29(1).
4. R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*, NY, United States, 23-25 Oct 2009, NY, United States: Association for Computing Machinery, pp. 61-68.
5. D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *In Proceedings of the*

*2009 conference on empirical methods in natural language processing*, Singapore, 6-7 Aug 2009, PA, United States: Association for Computational Linguistics, pp. 248-256.

6. A. K. Saha, R. K. Saha, and K. A. Schneider, "A discriminative model approach for suggesting tags automatically for stack overflow questions," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, California, USA, 18-19 May 2013: IEEE, pp. 73-76.
7. T. Wang, H. Wang, G. Yin, C. X. Ling, X. Li, and P. Zou, "Tag recommendation for open source software," *Frontiers of Computer Science,* Vol. 8, no. 1, pp. 69-82, 2014.
8. E. Quintanilla, Y. S. Rawat, A. Sakryukin, M. Shah, and M. Kankanhalli, "Adversarial Learning for Personalized Tag Recommendation," *IEEE Transactions on Multimedia,* Vol. 23, pp. 1083-1094, 2020.
9. Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *IJCAI-16*, NY, United States, 9-15 July 2016, California, USA: AAAI Press, pp. 2782-2788.
10. J. Li, H. Xu, X. He, J. Deng, and X. Sun, "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, 24-29 July 2016: IEEE, pp. 1570-1577.
11. Q. Yang *et al.*, "AMNN: Attention-Based Multimodal Neural Network Model for Hashtag Recommendation," *IEEE Transactions on Computational Social Systems,* Vol. 7, no. 3, pp. 768-779, 2020.
12. Y. Li, T. Liu, J. Hu, and J. Jiang, "Topical Co-Attention Networks for hashtag recommendation on microblogs," *Neurocomputing,* Vol. 331, pp. 356-365, 2019.
13. I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, Montreal, Canada, 8-13 Dec 2014: Neural Information Processing Systems Foundation, pp. 2672-2680.
14. M. G. De Oliveira, P. M. Ciarelli, and E. Oliveira, "Recommendation of programming activities by multi-label classification for a formative assessment of students," *Expert Systems with Applications,* Vol. 40, no. 16, pp. 6641-6651, 2013.
15. B. Wu, W. Chen, P. Sun, W. Liu, B. Ghanem, and S. Lyu, "Tagging like humans: Diverse and distinct image annotation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Utah, USA, 18-22 June 2018: IEEE, pp. 7967-7975.
16. R. Babbar and B. Schölkopf, "Adversarial extreme multi-label classification," *arXiv preprint arXiv:1803.01570,* 2018.
17. S. Wang, G. Peng, and Z. Zheng, "Capturing Joint Label Distribution for Multi-Label Classification through Adversarial Learning," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 32, no. 12, pp. 2310-2321, 2020.
18. S. Tang *et al.*, "An integral tag recommendation model for textual content," in *33rd AAAI Conference on Artificial Intelligence*, Hawaii, USA, 27 Jan - 1 Feb 2019, vol. 33: AAAI, pp. 5109-5116.
19. Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, "Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 19-25 Aug 2017: International Joint Conferences on Artificial Intelligence, pp. 3420-3426.
20. R. Ma, X. Qiu, Q. Zhang, X. Hu, Y.-G. Jiang, and X. Huang, "Co-attention memory

network for multimodal microblog's hashtag recommendation," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 33, no. 2, pp. 388-400, 2019.
21. A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* Vol. 30, pp. 5998-6008, 2017.
22. J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances in neural information processing systems*, Barcelona, Spain, 5-10 Dec 2016, NY, United States: Curran Associates Inc., pp. 289-297.
23. S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proceedings of the third ACM international conference on Web search and data mining*, New York, USA, 4-6 Feb 2010, New York, USA: Association for Computing Machinery, pp. 81-90.
24. Y. Zhu, X. Wu, J. Qiang, Y. Yuan, and Y. Li, "Representation learning with collaborative autoencoder for personalized recommendation," *Expert Systems with Applications,* Vol. 186, p. 115825, 2021.
25. Y. Wu, Y. Yao, F. Xu, H. Tong, and J. Lu, "Tag2word: Using tags to generate words for content based tag recommendation," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, Indiana, USA, 24-28 Oct 2016, NY, United States: Association for Computing Machinery, pp. 2287-2292.
26. R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for web service recommendation," *Expert Systems with Applications,* Vol. 110, pp. 191-205, 2018.
27. W.-J. Ye and A. J. Lee, "Mining sentiment tendencies and summaries from consumer reviews," *Information Systems and e-Business Management,* Vol. 19, no. 1, pp. 107-135, 2021.
28. P. Lops, M. De Gemmis, G. Semeraro, C. Musto, and F. Narducci, "Content-based and collaborative techniques for tag recommendation: an empirical evaluation," *Journal of Intelligent Information Systems,* Vol. 40, no. 1, pp. 41-61, 2013.
29. Y.-H. Lee, Y.-C. Cheng, and T.-H. Chu, "A Temporal Usage Pattern-based Tag Recommendation Approach," in *PACIS 2018*, Yokohama, Japan, 26-30 June 2018: Association for Information Systems, p. 221.
30. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *In Proceedings of the 28th international conference on machine learning*, Bellevue Washington, USA, 28 June - 2 July 2011, WI, United States: Omnipress, pp. 689-696.
31. X. Chen *et al.*, "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Paris, France, 21-25 July 2019, NY, United States: Association for Computing Machinery, pp. 765-774.
32. S. Zhang, Y. Yao, F. Xu, H. Tong, X. Yan, and J. Lu, "Hashtag recommendation for photo sharing services," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Hawaii, USA, 27 Jan-1 Feb 2019, vol. 33: AAAI, pp. 5805-5812.
33. C.-P. Tsai and H.-Y. Lee, "Adversarial Learning of Label Dependency: A Novel Framework for Multi-class Classification," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 12-17 May 2019: IEEE, pp. 3847-3851.
34. F. Huang, X. Zhang, and Z. Li, "Learning joint multimodal representation with

adversarial attention networks," in *Proceedings of the 26th ACM international conference on Multimedia*, Seoul, Republic of Korea, 22-26 Oct 2018, NY, United States: Association for Computing Machinery, pp. 1874-1882.
35. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.
36. S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *In Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004*, Sydney, Australia, 26-28 May 2004, Berlin, Heidelberg: Springer, pp. 22-30.

**Duen-Ren Liu (劉敦仁)** is a professor of the Institute of Information Management at the National Yang Ming Chiao Tung University of Taiwan. He received the B.S. and M.S. degrees in Computer Science from the National Taiwan University, Taiwan, in 1985 and 1987, respectively. He received the PhD degree in Computer Science from the University of Minnesota, USA, in 1995. His research interests include data mining, knowledge engineering, e-commerce and recommender systems.

**Chin-Hui Lai (賴錦慧)** is currently an associate professor of the Department of Information Management at Chung Yuan Christian University, Taiwan. She was a postdoctoral fellow of the Institute of Information Management at the National Yang Ming Chiao Tung University of Taiwan in 2010. She received the MS and the PhD degrees in the Institute of Information Management from the National Yang Ming Chiao Tung University in 2004 and 2010 respectively. Her research interests include data mining, text mining, machine learning, social network analysis, and recommender systems.


**Yang Huang (黃揚)** received the BS degree in Power Mechanical Engineering from National Tsing Hua University, Taiwan, in 2016. He received the master's degree in Institute of Management of Technology from the National Yang Ming Chiao Tung University in 2018. He is currently pursuing the Ph.D. degree with the Institute of Information Management, National Yang Ming Chiao Tung University. Taiwan. His research interests include recommender systems, collaborative learning, and data mining.

**Shu-Ting Chang (張舒婷)** received the BS degree in Information Management from National Taiwan University of Science and Technology, Taiwan. She received the MS degree in Institute of Information Management from the National Yang Ming Chiao Tung University, Taiwan. Her research interests include recommender systems, machine learning, and deep learning.