Enhanced Multi-granularity Image Noise Filtering for Better Multi-modal Keyphrase Generation

Suhang Wu[1*], **Wengyi Zhan**[1*], **Xudong Li**[1*], **Xin Li**[1*]

[1]Xiamen University

{38120231150165,36920231153211}-ai, {31520231154282,23020231154204}-xxxy,

Abstract

Multi-modal keyphrase generation aims to produce a set of keyphrases that represent the core points of the input text-image pair. In this regard, dominant methods mainly focus on multi-modal fusion for keyphrase generation. Nevertheless, there are still exists drawbacks. For example, the input text and image are often not perfectly matched, and thus the image may introduce noise into the model. To address these limitations, in this paper, we propose a novel multi-modal keyphrase generation model, which can effectively filter image noise. In our model, we compute both an image-text matching score and image region-text correlation scores concurrently to facilitate multi-granularity image noise filtering. Specifically, we incorporate correlation scores between image regions and ground-truth keyphrases to enhance the calculation of the aforementioned correlation scores. To demonstrate the effectiveness of our model, we conduct several groups of experiments on the benchmark dataset.

Introduction

With the growth of social platforms, users increasingly express views via multi-modal data, including text and images. Multi-modal keyphrase generation, which derives keyphrases from such data, has become essential, as illustrated in Figure 1. This method, distinct from traditional text-only approaches (Meng et al. 2017; Ye et al. 2021), harnesses both text and image for superior keyphrase extraction, gaining traction in opinion mining and content recommendation.

In pursuing this task, initial research posited that hashtags encapsulate vital information in multi-media content, leading to their treatment as keyphrases (Zhang et al. 2017, 2019). As such, multi-modal keyphrase generation is often framed as a hashtag recommendation endeavor. Predominantly, these works employ a co-attention network to merge textual and visual tweet data for hashtag suggestions (Zhang et al. 2017, 2019). Wang et al. (2020) initially utilizes Optical Character Recognition (OCR) to discern optical characters from images, followed by an image captioning model to ascertain implicit image semantics. To optimize multi-modal data integration, they implement a multi-modal multi-head attention mechanism, capturing semantic interplay between modalities.

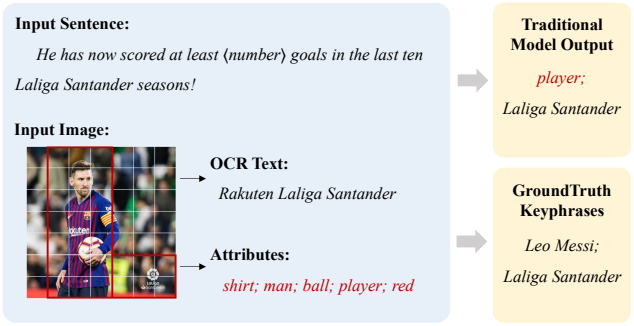

Figure 1: An example of multi-modal keyphrase generation. We can observe that certain image regions may be irrelevant to the sentence, consequently leading to the introduction of noise into the model.

Despite their achievements, the aforementioned studies exhibit limitations. They predominantly focus on multi-modal fusion, often overlooking potential discrepancies between text and image in social multi-media content. Even within pertinent text-image pairs, certain image regions may not align with the text. As depicted in Figure 1, regions highlighted by red boxes correlate closely with the text, whereas others exhibit lesser relevance. Such incongruities can introduce noise, potentially compromising model efficacy. Thus, optimally leveraging images remains a pressing challenge in multi-modal keyphrase generation.

In this study, we will introduce a multi-modal keyphrase generation model equipped with image noise filtering. Overall, our model includes three modules: 1) Multi-modal feature encoding module, which learns the representations of the input text and image, respectively; 2) Image noise filtering module that conducts multi granularity image noise filtering to generate a better image representation; 3) Keyphrase generation module, which generates each keyphrase in the form of a sequence. Note that we incorporate the Image Noise Filtering module into the original model. The Image Noise Filtering module introduces a multi-granularity noise filtering mechanism, which can be roughly categorized into coarse-grained and fine-grained fil-

*These authors contributed equally.

tering.

To investigate the effectiveness of our model, we will conduct several groups of experiments on the benchmark dataset.

Related Works

Keyphrase Generation. The task of keyphrase generation has received sustained attention in recent years. The commonly-used models for keyphrase generation can be roughly classified into extraction and generation approaches. Early studies mainly focus on using statistical models to perform keyphrase extraction (Salton and Buckley 1988; El-Beltagy and Rafea 2009). With the rapid development of deep learning, a number of neural network based models have been proposed for keyphrase generation. Generally, the frameworks for keyphrase generation can be divided into three categories: 1) One2one (Chen et al. 2018). This category splits a training instance into multiple pairs, each consisting of the input text and only one corresponding keyphrase. During inference, it adopts beam search to produce candidate phrases and then selects the top-K ranked ones as the final keyphrases. 2) One2seq (Yuan et al. 2020), which concatenates all keyphrases in a given order as a training instance. During inference, the model outputs all keyphrases as a sequence. 3) One2set (Ye et al. 2021). In this category, the generation of keyphrases is modeled as a generation task of a keyphrase set, where keyphrases are individually generated in parallel.

Multi-modal Fusion. How to effectively fuse multi-modal information is always a hot research topic. Dominant approaches can be roughly classified into the following three categories (Zhang et al. 2020): 1) simple operations such as concatenation (Anastasopoulos, Kumar, and Liao 2019), weighted sum with scalar weights (Pérez-Rúa et al. 2019) and progressive exploration decision fusion (Liu et al. 2017; Pérez-Rúa, Baccouche, and Pateux 2018); 2) bilinear pooling (Kim, Jun, and Zhang 2018; Ben-Younes et al. 2019); 3) attention-based methods, such as symmetric attention mechanisms (Zhao, Liu, and Lu 2021), dual attention networks (Nam, Ha, and Kim 2017), dynamic gated aggregation mechanisms (Chen et al. 2022), and dynamic parameter prediction networks (Noh, Seo, and Han 2016).

Particularly, some studies concentrate on multi-modal fusion in the presence of image noise. For example, Sun et al. (2020) present a pre-trained multi-modal model based on relationship inference and visual attention. Typically, it contains a gated unit that adjusts the weights of visual features during fusion based on the image-text matching score. Yu et al. (2022) put forward a coarse-to-fine image-target matching model for the target-oriented (aspect-based) multi-modal sentiment classification task. With extra manually labeled data, they explore two supervised tasks to capture the image-target matching relations for multi-modal fusion. Ye et al. (2022) construct a cross-modal relation-aware attention module, which is equipped with a mask matrix based on the relevance of text and image regions. This matrix conducts noise filtering during the self-attention process, improving the performance of multi-modal machine translation.

Methodology

Before elaborating on our model, we first briefly introduce the formulation of this task. Given a text-image pair (X_S, X_I) of the dataset D, multi-modal keyphrase generation aims to predict a keyphrase set $\mathcal{Y}$. Following (Meng et al. 2017), we replicate the original input pair multiple times to ensure that each input pair is associated with one keyphrase, forming a triplet set $\{(X_S, X_I, y)\}$, where $y \in \mathcal{Y}$. In the subsequent subsections, we first give a description of the architecture of our model, and then describe details of the model training.

Model Architecture

Figure 2 illustrates the overview of our model. Overall, our model includes four modules: 1) *Multi-modal feature encoding module* learning the representations of the input text and image, respectively; 2) *Image noise filtering module* that conducts multi-granularity image noise filtering to generate a better image representation; 3) *Keyphrase classification module* that fuses the filtered image and text representations and then performs keyphrase classification; 4) *Keyphrase generation module*, which is based on a pointer network and generates each keyphrase in the form of a sequence. These modules are described in detail in the following.

Multi-modal Feature Encoding Module

This module contains an image sub-encoder and a text sub-encoder, extracting visual features and textual features respectively. To provide this module with more information for better keyphrase generation, we first preprocess the input image to get the OCR information contained in the image.

Specifically, we use the commonly-used PaddleOCR[1] to extract the explicit optical characters (e.g., slogans) from the image. This additional OCR textual information can serve as semantic anchors to facilitate cross-modal semantic alignment, thus leading to better keyphrase generations. To facilitate the subsequent descriptions, we denote the extracted OCR text as X_O. Then, the original input text and OCR text are sequentially concatenated and fed to the text sub-encoder. Meanwhile, the input image is encoded by the image sub-encoder.

Text sub-encoder. To distinguish X_O from the original input text X_S, we insert a delimited $\langle\text{seq}\rangle$ tokens to indicate the beginning positions of X_O, obtaining the concatenated input of text modality: $X_T = X_S\langle\text{seq}\rangle X_O$. Then, we feed X_T into the text sub-encoder, which is based on Bi-GRU, learning the token-level semantic representations of X_T:

$$\mathrm{H}_T = \text{Bi-GRU}(X_{emb}), \tag{1}$$

where $\mathrm{H}_T \in \mathbb{R}^{|X_T| \times d_1}$, d_1 denotes the hidden state dimension, and X_{emb} is the embedding sequence of X_T. Here we use the sum of word embedding and type embedding to represent each token. Besides, we use the pre-trained Glove (Pennington, Socher, and Manning 2014) word embedding to initialize the input word embedding, and randomly initialize the type embedding. Finally, we obtain a global vector

[1] https://github.com/PaddlePaddle/PaddleOCR

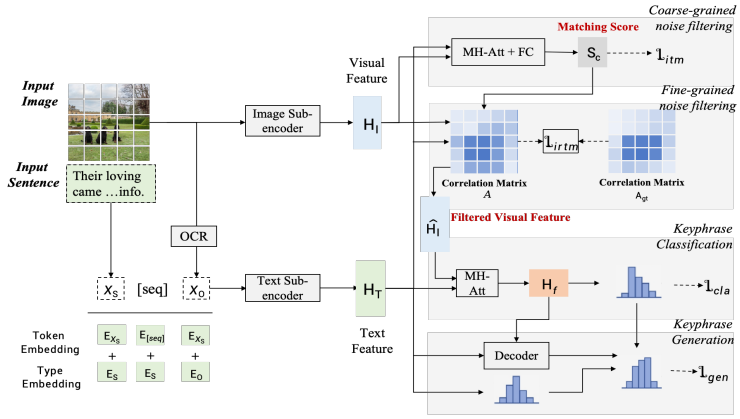

Figure 2: The overall architecture of our model, including multi-modal feature encoding module, image noise filtering module, keyphrase classification module and keyphrase generation module.

representation of text modality via max-pooling operation: $\mathrm{M}_T = \text{Max-pooling}(\mathrm{H}_T)$.

Image sub-encoder. Following common practice (Sun et al. 2020), we employ the pre-trained model VGG19 (Simonyan and Zisserman 2015) to extract the visual features of each input image. Concretely, we first resize each image to 224×224 pixels and feed it to the VGG19 model. The last-layer output is a $7 \times 7 \times 512$-dimensional vector containing 49 local spatial region features for each image. That is, the visual feature of each region is represented as a 512-dimensional vector. To further use these visual features, we perform flattening and linear projection on these visual features:

$$\mathrm{H}_I = \text{flatten}(\text{VGG19}(X_I))W_I + b_I, \tag{2}$$

where $\mathrm{H}_I \in \mathbb{R}^{49 \times d_1}$ and $\text{flatten}(\cdot)$ is a function reshaping the $7 \times 7 \times 512$-dimensional vector to a 49×512-dimensional one. Additionally, $W_I \in \mathbb{R}^{49 \times d_1}$ and $b_I \in \mathbb{R}^{49 \times d_1}$ are learnable parameter matrices.

Image Noise Filtering Module

In this module, we explore two cross-modal matching strategies to filter the noise of each input image, obtaining a filtered image representation. Via the combined effect of the two cross-modal matching strategies, this module may help the model focus on key regions for keyphrase generation while avoiding the interference of image noise.

Image-text matching. Using this strategy, we obtain a score indicating the semantic matching degree between the whole image and the input text. Specifically, we first use a multi-head cross-attention function to the fusion representation H_c:

$$\mathrm{H}_c = \text{MultiHead}(\mathrm{M}_T, \mathrm{H}_I, \mathrm{H}_I), \tag{3}$$

where MultiHead(*) is a multi-head cross-attention function, the global textual feature M_T is used as the query, and the visual feature H_I works as the key and value.

On the top of H_c, we stack a fully connected (FC) layer to perform image-text matching, where a matching score s_c is acquired and then used in conjunction with the subsequent image region-text matching strategy to filter image noise.

Image region-text matching This strategy is used to filter the irrelevant regions of the input image. To achieve this, we first project the visual feature H_I and the global textual feature M_T into a shared semantic space: $\overline{\mathrm{H}}_T = W_T \mathrm{M}_T$, $\overline{\mathrm{H}}_I = W_I \mathrm{H}_I$, facilitating the subsequent calculation of their semantic correlation. Here, H_I represents the flattened representation of 7×7 image region features, while W_* are learnable parameter matrices.

Subsequently, we calculate the image region-text correlation matrix A as fellow:

$$\mathrm{A} = \text{FFN}\left(\frac{(\overline{\mathrm{H}}_T) \cdot (\overline{\mathrm{H}}_I)^\top}{\sqrt{d_2}} + J \times s_c\right), \tag{4}$$

where FFN(*) is a feedforward network, d_2 is the dimension of vector representation in the shared semantic space, the element A_{l_1,l_2} indicates the semantic matching score between the input text and the $(7 \times l_1 + l_2)$-th image region, and J is an all-ones matrix. Note that we use the above-mentioned image-text matching score s_c to smooth the matrix A.

Lastly, we use a Sigmoid function to produce a filtered image representation $\widehat{\mathrm{H}}_I$:

$$\widehat{\mathrm{H}}_I = \text{Sigmoid}(\mathrm{A}) \odot \mathrm{H}_I, \tag{5}$$

where $\odot$ is the element-wise multiplication.

Keyphrase Classification Module

Following (Wang et al. 2020), we also regard each keyphrase in training data as a discrete label and directly use a classifier to predict keyphrases.

Concretely, we first use a multi-head cross-attention to effectively fuse the filtered visual and textual features, and then use an FFN with residual connection and layer normalization to obtain a fused vector H_f:

$$\mathrm{H}_f = \text{FFN}(\text{MultiHead}(\mathrm{M}_T, \widehat{\mathrm{H}}_I, \widehat{\mathrm{H}}_I)), \tag{6}$$

where the global textual feature M_T is used as the query and the filtered visual feature $\hat{\mathrm{H}}_I$ serves as key and value.

Finally, on the basis of H_f, we construct a classifier based on a two-layer multi-layer perception (MLP) to produce a keyphrase distribution d_{cla} as follows:

$$d_{cla} = \text{Softmax}(\text{MLP}(\mathrm{H}_f)). \tag{7}$$

Keyphrase Generation Module

As implemented in (Wang et al. 2020), we introduce the pointer network (Gu et al. 2016) to generate each keyphrase y as a sequence. Typically, by equipping with an extended copy mechanism, this module models the token-level generation probability $p(y_j)$ at each timestep j as the weighted sum of two types of probabilities:

Prediction probability $p_p(y_j)$. To model this probability, we update the decoder hidden state s_j as follows:

$$s_j = \text{GRU}(y_{j-1}, s_{j-1}, c_j), \tag{8}$$

$$c_j = \sum_{i=1}^{|X_T|} \alpha_{j,i} h_i, \tag{9}$$

$$\alpha_{j,i} = \text{Softmax}(V_\alpha^\top \tanh(W_\alpha[s_j; h_i])), \tag{10}$$

where y_{j-1} is the output at timestep j-1, c_j is the context vector, $\alpha_{j,i}$ is the normalized weight that measures the compatibility between s_j and h_i, V_α and V_α are learnable parameter matrices.

Next, we further introduce the fusion vector H_f to produce a token distribution $p_p(y_j)$ as follows:

$$p_p(y_j) = \text{Softmax}(W_p[y_{j-1}; s_j; c_j + \mathrm{H}_f]), \tag{11}$$

where W_p is a learnable parameter matrix.

Copy probability $P_c(y_j)$. To generate better keyphrases, we also adopt an extended copy mechanism to simultaneously leverage the words of concatenated input text X_T and the classifier predictions d_{cla}.

Specifically, we first retrieve the top-5 classifier predictions and transform each prediction into a sequence of words $\mathbf{w} = w_1, \ldots, w_{|\mathbf{w}|}$. Afterwards, we use a softmax function to normalize the corresponding classification logits into word-level distributions $\{\beta_k\}_{k=1}^{|\mathbf{w}|}$. Finally, we define the copy probability $p_c(y_j)$ as

$$p_c(y_j) = \lambda_c \cdot \sum_{i:x_i=y_j}^{|X_T|} \alpha_{j,i} + (1-\lambda_c) \cdot \sum_{k:w_k=y_j}^{|\mathbf{w}|} \beta_k, \tag{12}$$

where λ_c is a hyper-parameter used to decide whether to copy from the concatenated input text or the classification predictions.

With the above two kinds of probabilities, we obtain the generation probability $p(y_j)$ as follows:

$$p(y_j) = \lambda p_p(y_j) + (1-\lambda) p_c(y_j), \tag{13}$$

$$\lambda = \text{Sigmoid}(W_\lambda[y_{j-1}; s_j; c_j + \mathrm{H}_f]), \tag{14}$$

where λ is a soft switch and W_λ is a learnable parameter matrix.

Training Framework

We propose a two-stage training framework to train our model.

Stage 1. During this stage, we first pre-train the multi-modal feature encoding module, image noise filtering module and keyphrase classification module. To this end, we define the following training objective involving three loss items:

$$\mathcal{L}_1 = \mathcal{L}_{itm} + \mathcal{L}_{irtm} + \mathcal{L}_{cla}, \tag{15}$$

where $\mathcal{L}_{itm}$, $\mathcal{L}_{irtm}$, $\mathcal{L}_{cla}$ are loss items proposed for three tasks. We will describe in detail these three losses, respectively.

The loss item for image-text matching: $\mathcal{L}_{itm}$. As described in previously, we introduce an image-text matching task to perform coarse-granularity image noise filtering. Given an additional dataset $D_{itm} = \{(X_T, X_I)\}$, we define the following cross-entropy loss:

$$\mathcal{L}_{itm} = -\sum_{(X_T, X_I) \in D_{itm}} \log(p_{itm}(X_T, X_I)), \tag{16}$$

where $p_{itm}(*)$ is the probability of correct classification.

The loss item for image region-text matching: $\mathcal{L}_{irtm}$. As mentioned above, for each training text-image pair $(X_S, X_I, y) \in D$, we introduce a correlation matrix A to perform fine-granularity image noise filtering. To accurately model A, we encode the ground-truth keyphrases and calculate the correlation score between each region of the input image and ground-truth keyphrases according to Equation 4, forming a correlation matrix A_{gt}. Afterwards, we use A_{gt} as supervisory signals to train A by introducing a MSE(Mean Squared Error) loss to minimize their divergence:

$$\mathcal{L}_{irtm} = \sum_{(X_S, X_I) \in D} \text{MSE}(\mathrm{A}, \mathrm{A}_{gt}). \tag{17}$$

The loss item for keyphrase classification: $\mathcal{L}_{cla}$. To train the previously-mentioned keyphrase classifier, we define the following standard cross-entropy loss:

$$\mathcal{L}_{cla} = -\sum_{(X_S, X_I, y) \in D} \log(d_{cla}), \tag{18}$$

where d_{cla} denotes the predictions of keyphrase classification, defined as Equation 7.

Stage 2. In this stage, we optimize the model for the keyphrase generation task. Following common practice (Meng et al. 2017), we design $\mathcal{L}_{gen}$ as a token-level cross-entropy loss:

$$\mathcal{L}_{gen} = \sum_{(X_S, X_I, y) \in D} \sum_{j=1}^{|y|} \log(p(y_j)). \tag{19}$$

EXPERIMENT

Benchmark Datasets

In our experiments, we use two datasets. One is the TRC dataset[2], which is used to train the model via the image-text matching task. The other is the dataset for multi-modal keyphrase generation collected by Wang et al. (2020). This dataset includes 53,701 English tweets, each of which comprises a distinct text-image pair, with user-annotated hashtags serving as keyphrases.

[2]https://github.com/danielpreotiuc/text-image-relationship/

Implementation Details

To ensure fair comparisons, in the experiments, we use the setting used in (Wang et al. 2020) which is our most important baseline. Specifically, we select the top 45K most frequent words as the vocabulary for keyphrase generation and 4,262 keyphrases of the training data as candidate ones in the classifier. When constructing our encoder and decoder, we initialize the input word embeddings with 200-dimensional GloVe (Pennington, Socher, and Manning 2014) ones, and set their hidden state dimensions as 300. To encode the input image, we use 49 grid-level VGG features, where each grid is represented as a 512-dimensional vector. During training, we use Adam (Kingma and Ba 2015) to optimize the model, with an initial learning rate of 10^{-3}. Additionally, we perform dropout(Srivastava et al. 2014) with a rate of 0.1 to enhance the robustness of our model. Particularly, we employ early stopping to stop the model training according to the performance on the validation dataset. During inference, we apply beam search with a beam size of 10 to produce a ranked list of keyphrases. We conduct the experiments repeat five times using different random seeds, and report the averaged results.

Metrics Following previous studies (Meng et al. 2017; Wang et al. 2020), we use the commonly-used macro-average F1@K to evaluate the model performance, where K is 1 or 3. Besides, as implemented in (Chen et al. 2019), we measure the keyphrase orders with the mean average precision (MAP) for the top-5 predictions.

Baseline

In our study, we compare various baseline models categorized into three main groups: Image-only models, Text-only models, and Text-image models. In the Image-only category, we have the **VGG** model, using a pre-trained VGG encoder (Anderson et al. 2018), and the **BUTD** model, which employs a bottom-up attention mechanism (Anderson et al. 2018). For Text-only models, we consider classification-based models like **AVG**, **MAX**, and **TMN** (Zeng et al. 2018), as well as generation-based models including **ATT** (Bahdanau, Cho, and Bengio 2015), **COPY** (See, Liu, and Manning 2017) and **TOPIC** (Wang et al. 2019). Lastly, in the Text-image category, we evaluate **CO-ATT** (Zhang et al. 2017), **BAN** (Kim, Jun, and Zhang 2018) and **M³H-ATT** (Wang et al. 2020), with the latter achieving a strong performance in multi-modal keyphrase generation.

Main Results

Table 1 shows the performance of our model and baselines on the dataset collected by Wang et al. (2020). Here we can obtain the following conclusions:

First, our model surpasses all baselines in terms of all metrics. Specifically, our model outperforms M^3H-ATT by 1.2 points in terms of F1@1, 0.8 points in terms of F1@3, and 1.1 points in terms of MAP@5. Note that M^3H-ATT is one of the high-performance models in multi-modal keyphrase generation. This result strongly confirms the effectiveness of our model.

Table 1: Performance comparison for multi-modal keyphrase generation task. $*$ indicates the results are directly cited from (Wang et al. 2020).

Models	F1@1	F1@3	MAP@5
Image-only models			
VGG*	15.69	13.67	19.70
BUTD* (Anderson et al. 2018)	20.02	16.97	24.73
Text-only models			
AVG*	35.96	27.59	41.84
MAX*	38.33	28.84	44.15
TMN* (Zeng et al. 2018)	40.33	30.07	46.28
ATT* (Bahdanau, Cho, and Bengio 2015)	38.36	27.83	43.35
COPY* (See, Liu, and Manning 2017)	42.10	29.91	46.94
TOPIC* (Wang et al. 2019)	43.17	30.73	48.07
Text-image models			
CO-ATT* (Zhang et al. 2017)	42.12	31.55	48.39
BAN* (Kim, Jun, and Zhang 2018)	38.73	29.68	45.03
M^3H-ATT* (Wang et al. 2020)	47.06	33.11	52.07
Our text-image model			
Our Model *w/o* $\mathcal{L}_{itm}$	47.76	33.66	52.94
Our Model *w/o* $\mathcal{L}_{irtm}$	48.03	33.46	52.88
Our Model	**48.19**	**33.86**	**53.28**

Second, the multi-modal models outperform both image-only and text-only models, echoing the results reported in (Wang et al. 2020). Our model exhibits superior performance compared to all the text-only keyphrase generation methods. Thus, we confirm that the complementarity between image and text enables our model to effectively capture crucial information for multi-modal keyphrase generation.

Third, the text-only models perform better than the image-only ones, showing that each input text provides more cues than the input image, and therefore can contribute more to keyphrase generation. For this result, we speculate that the inferior performance of image-only models may be attributed to the feature sparsity and noise in the input image, making it challenging for models to acquire effective features.

Finally, we also conduct ablation studies to analyze our model. Our model *w/o* $\mathcal{L}_{itm}$ conducts coarse-granularity image noise filtering without supervisory information and causes a performance decline. Our model *w/o* $\mathcal{L}_{irtm}$ does not use the correlation matrix A_{gt} between ground-truth keyphrases and image regions to supervise the training of the correlation matrix. The performance degradation confirms our hypothesis that the correlation matrix A_{gt} can guide the model to focus on key regions.

CONCLUSION

In this paper, we perform image-text matching and image region-text matching to effectively filter image noise. We conduct several groups of experiments on the commonly-used dataset. Experimental results and in-depth analyses verify the effectiveness of our model.

References

Anastasopoulos, A.; Kumar, S.; and Liao, H. 2019. Neural Language Modeling with Visual Features. *CoRR*, abs/1903.02930.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR 2018*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*.

Ben-Younes, H.; Cadène, R.; Thome, N.; and Cord, M. 2019. BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection. In *EAAI 2019*.

Chen, J.; Zhang, X.; Wu, Y.; Yan, Z.; and Li, Z. 2018. Keyphrase Generation with Correlation Constraints. In *ACL 2018*.

Chen, W.; Chan, H. P.; Li, P.; Bing, L.; and King, I. 2019. An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction. In *NAACL 2019*.

Chen, X.; Zhang, N.; Li, L.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; Si, L.; and Chen, H. 2022. Good Visual Guidance Make A Better Extractor: Hierarchical Visual Prefix for Multimodal Entity and Relation Extraction. In *Findings of NAACL 2022*.

El-Beltagy, S. R.; and Rafea, A. A. 2009. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Inf. Syst.*, 34(1): 132–144.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. K. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACl 2016*.

Kim, J.; Jun, J.; and Zhang, B. 2018. Bilinear Attention Networks. In *NeurIPS 2018*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

Liu, C.; Zoph, B.; Shlens, J.; Hua, W.; Li, L.; Fei-Fei, L.; Yuille, A. L.; Huang, J.; and Murphy, K. 2017. Progressive Neural Architecture Search. *CoRR*, abs/1712.00559.

Meng, R.; Zhao, S.; Han, S.; He, D.; Brusilovsky, P.; and Chi, Y. 2017. Deep Keyphrase Generation. In Barzilay, R.; and Kan, M., eds., *ACL 2017*.

Nam, H.; Ha, J.; and Kim, J. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *CVPR 2017*.

Noh, H.; Seo, P. H.; and Han, B. 2016. Image Question Answering Using Convolutional Neural Network with Dynamic Parameter Prediction. In *CVPR 2016*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *EMNLP 2014*.

Pérez-Rúa, J.; Baccouche, M.; and Pateux, S. 2018. Efficient Progressive Neural Architecture Search. In *BMVC 2018*.

Pérez-Rúa, J.; Vielzeuf, V.; Pateux, S.; Baccouche, M.; and Jurie, F. 2019. MFAS: Multimodal Fusion Architecture Search. In *CVPR 2019*.

Salton, G.; and Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manag.*, 24(5): 513–523.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL 2017*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1): 1929–1958.

Sun, L.; Wang, J.; Su, Y.; Weng, F.; Sun, Y.; Zheng, Z.; and Chen, Y. 2020. RIVA: A Pre-trained Tweet Multimodal Model Based on Text-image Relation for Multimodal NER. In *COLING 2020*.

Wang, Y.; Li, J.; Chan, H. P.; King, I.; Lyu, M. R.; and Shi, S. 2019. Topic-Aware Neural Keyphrase Generation for Social Media Language. In *ACL 2019*.

Wang, Y.; Li, J.; Lyu, M. R.; and King, I. 2020. Cross-Media Keyphrase Prediction: A Unified Framework with Multi-Modality Multi-Head Attention and Image Wordings. In *EMNLP 2020*.

Ye, J.; Gui, T.; Luo, Y.; Xu, Y.; and Zhang, Q. 2021. One2Set: Generating Diverse Keyphrases as a Set. In *ACL 2021*.

Ye, J.; Guo, J.; Xiang, Y.; Tan, K.; and Yu, Z. 2022. Noise-robust Cross-modal Interactive Learning with Text2Image Mask for Multi-modal Neural Machine Translation. In *COLING 2022*.

Yu, J.; Wang, J.; Xia, R.; and Li, J. 2022. Targeted Multimodal Sentiment Classification based on Coarse-to-Fine Grained Image-Target Matching. In *IJCAI 2022*.

Yuan, X.; Wang, T.; Meng, R.; Thaker, K.; Brusilovsky, P.; He, D.; and Trischler, A. 2020. One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. In *ACL 2020*.

Zeng, J.; Li, J.; Song, Y.; Gao, C.; Lyu, M. R.; and King, I. 2018. Topic Memory Networks for Short Text Classification. In *ACL 2018*.

Zhang, C.; Yang, Z.; He, X.; and Deng, L. 2020. Multimodal Intelligence: Representation Learning, Information Fusion, and Applications. *IEEE J. Sel. Top. Signal Process.*, 14(3): 478–493.

Zhang, Q.; Wang, J.; Huang, H.; Huang, X.; and Gong, Y. 2017. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In *IJCAI 2017*.

Zhang, S.; Yao, Y.; Xu, F.; Tong, H.; Yan, X.; and Lu, J. 2019. Hashtag Recommendation for Photo Sharing Services. In *EAAI 2019*.

Zhao, Z.; Liu, W.; and Lu, B. 2021. Multimodal Emotion Recognition Using a Modified Dense Co-Attention Symmetric Network. In *NER 2021*, 73–76. IEEE.