Image Clustering using Topic Modelling

Rahul kumar(21111069) Nishant kiran valvi(21111407)
Sandeep saket(21111055) Rakesh(21111051)

rahulkumar21@iitk.ac.in, nkiranv21@iitk.ac.in
sandeeps21@iitk.ac.in, rakesh21@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

We can't go on a trip without taking hundreds of photographs. With the rapid growth in technology, it is easier to take pictures and can store them in large. As a result we ended up taking similar images multiple times till we get a perfect shot. Manually going through such huge collections of images and clustering into one is a tedious job. Our problem statement is inspired from the same problem. The main goal of this report is to predict the topics of images and then combine them into one cluster based on topics.

1 Introduction

Topic modelling is a technique used to extract the hidden topics from a large volume of text. There are several algorithms used for topic modelling such as Latent Dirichlet Allocation(LDA), Non-Negative Matrix Factorization(NMF), Latent Semantic Analysis(LSA), etc. It is easier to extract topics from textual information and can predict topics for new set of texts. However, in our problem statement the major challenge is to extract the topics from visual context like images. To solve this problem we need to involve both the text summarization and the image processing technique to extract good quality of topics. This report explains the steps involved in combining both of these processing techniques to uncover the themes from images.

The following report is written as follows: Section 2 includes related work in this area. Section 3 includes proposed idea and framework of this project. Section 5 includes the final results. Section 6 includes conclusion and future work of the proposed idea.

2 Related Work

Most image clustering that have been used previously which is based on timestamps and content of the image as a partitioning criteria. Sharma et al. [2015] proposed a way of summarizing a given collection of photographs to represent a distinct set of representative images. In this paper, authors modified the existing Latent Dirichlet Allocation technique, a generative probabilistic model, to partition the images from a 'Bag of words' representation created using Scale Invariant Feature Transform (SIFT) vectors and then clustering these vectors into bins. Zhu et al. [2018] proposed a novel attention mechanism, called topic-guided attention, which integrates image topics in the attention model as a guiding information to help select the most important image features. After going through these papers, we have implemented transfer learning based image clustering technique using topic modelling. Li et al. [2003] and Yang and Wang [2009] talked about image clustering in details.

3 Proposed Idea

The Latent Dirichlet Allocation(LDA) algorithm extracts a set of topics from each text document

in the corpus. Using this algorithm, a set of keywords is to learn from the documents collection in the corpus. These set of keywords called as topics which represents each document in terms of probability.

The LDA alogrithm uses dirichlet priors for the document-topic and word-topic distributions. It follows dirichlet distributions in particular. Here's how the LDA model looks like in Fig. 1

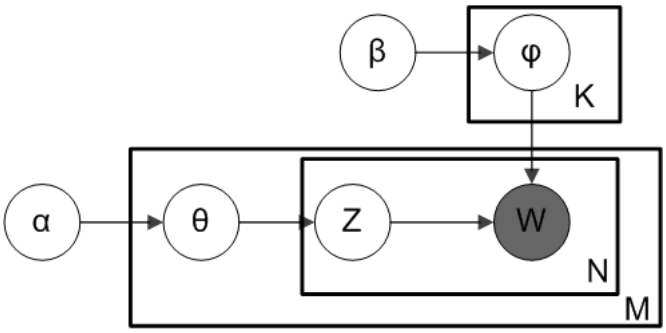

Figure 1: LDA Model

From a dirichlet distribution Dir(α) we randomly draw sample which represent the topics distributions of all topics of a particular document. This topic distribution is θ in the model. From θ, we select a topic Z based on the distributions. Later, we define another distribution Dir(β) for word distribution i.e ϕ. Given a topic Z we randomly draw a sample of keyword W from the word distributions. Finally, we try to generate same document by maximizing the product of topic distribution and word distribution.

In our case, we have used image captions as documents and tried to extract topics from all the captions. Based on topics extracted we assigned one topic to each images in the datasets.

We have used 2 different image captions datasets 'Flickr8k' and 'Microsoft COCO(8k)' to build a topic detection model for images. Using Latent Dirichlet Allocation(LDA), we extracted the topics from the vocabulary of caption data and reported the performance on 3 different pretrained models VGG16, VGG19 model with some fine tuning to extract the patterns from the images. Then we trained the model to predict the topics for the given images and combined them into a cluster based on predicted topics. Figure 2 shows complete framework of the project.

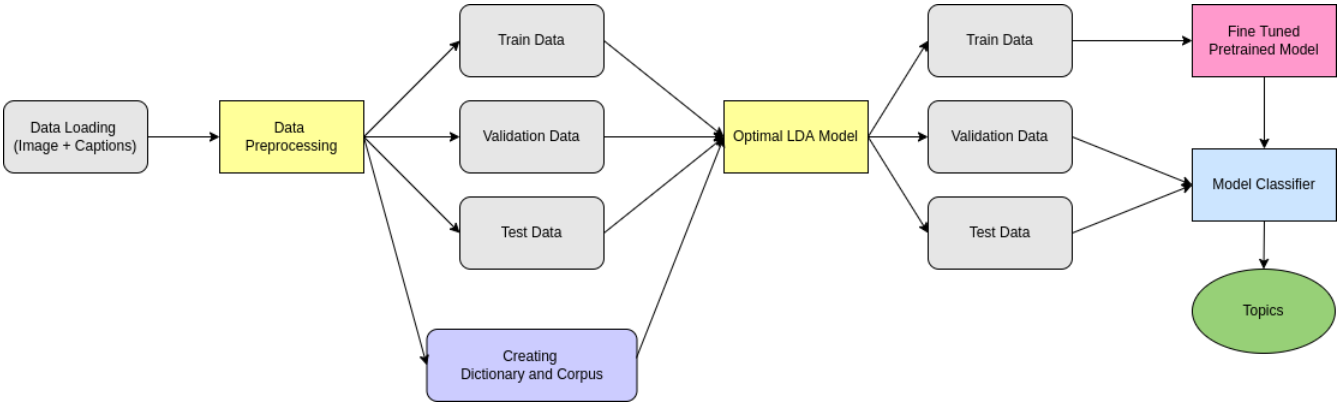

Figure 2: Proposed Framework

4 Methodolgy

4.1 Data Cleaning

It includes all the data preprocessing steps i.e keyword extraction, removal of stopping words, keyword lemmatization that contain only specific pos-tags like NOUN, ADJ, VERB, and ADV to improve the topic detection performance and removal of all the special characters. After doing all the steps, datasets looks like fig. 3.

	image_id	caption	caption_lemmatized
0	/kaggle/working/coco_datasets/391895.jpg	A man with a red helmet on a small moped on a dirt road. Man riding a motor bike on a dirt road on the countryside. A man riding on the back of a motorcycle. A dirt path with a young person on a motor bike rests to the foreground of a verdant area with a bridge and a background of cloud-wreathed mountains. A man in a red shirt and a red hat is on a motorcycle on a hill side.	[man, red, helmet, small, mope, dirt, man, ride, motor, countryside, man, ride, motorcycle, dirt, path, young, person, motor, bike, rest, foreground, verdant, area, bridge, background, cloudwreathe, mountain, man, red, shirt, red, hill, side]
1	/kaggle/working/coco_datasets/522418.jpg	A woman wearing a net on her head cutting a cake. A woman cutting a large white sheet cake. A woman wearing a hair net cutting a large sheet cake. there is a woman that is cutting a white cake A woman marking a cake with the back of a chef's knife.	[woman, wear, net, head, cut, cake, woman, cut, large, white, sheet, cake, woman, wear, hair, net, cut, large, sheet, cake, woman, cut, white, cake, woman, mark, cake, chef, knife]
2	/kaggle/working/coco_datasets/184613.jpg	A child holding a flowered umbrella and petting a yak. A young man holding an umbrella next to a herd of cattle. a young boy barefoot holding an umbrella touching the horn of a cow A young boy with an umbrella who is touching the horn of a cow. A boy holding an umbrella while standing next to livestock.	[child, hold, flower, umbrella, pet, young, man, hold, umbrella, next, herd, cattle, young, boy, barefoot, hold, umbrella, touch, horn, cow, young, boy, umbrella, touch, horn, cow, boy, hold, umbrella, stand, next, livestock]
3	/kaggle/working/coco_datasets/318219.jpg	A young boy standing in front of a computer keyboard. a little boy wearing headphones and looking at a computer monitor He is listening intently to the computer at school. A young boy stares up at the computer monitor. a young kid with head phones on using a computer	[young, boy, stand, front, computer, keyboard, little, boy, wear, headphone, look, computer, monitor, listen, intently, computer, school, young, boy, stare, computer, monitor, young, kid, head, phone, use, computer]
4	/kaggle/working/coco_datasets/554625.jpg	a boy wearing headphones using one computer in a long row of computers A little boy with earphones on listening to something. A group of people sitting at desk using computers. Children sitting at computer stations on a long table. A small child wearing headphones plays on the computer.	[boy, wear, headphone, use, computer, long, row, computer, little, boy, earphone, listen, group, people, sit, desk, use, computer, child, sit, computer, station, long, table, small, child, wear, headphone, play, computer]

Figure 3: Preprocessed datasets

4.2 Finding an optimal number of topics for LDA

Given a set of lemmatized keywords, we need to find an optimal number of topics for the image captions. This will also help in generalization of topics for unseen images. The first step is to create a dictionary of keywords mapped with an id and also need to prepare corpus that contains term-document frequency table which are required for LDA model to process.

In the next step, we iterated through the list of several number topics ranging from 10 to 100 and built the LDA model for each number of topics using Gensim's LDAMulticore class. In order to identify the optimal number of topics we have used coherence score. Coherence score is nothing but giving a score to single topic by measuring the degree of semantic similarity between high scoring words in the topic and the keywords in the topics will support each other. Thus, a coherent score can be interpreted in a context that covers all or most of the facts. An ideal LDA model should have higher coherence scores.

```
Num Topics = 10  has Coherence Value of 0.2671
Num Topics = 15  has Coherence Value of 0.2949
Num Topics = 20  has Coherence Value of 0.2667
Num Topics = 25  has Coherence Value of 0.2476
Num Topics = 30  has Coherence Value of 0.2577
Num Topics = 35  has Coherence Value of 0.2572
Num Topics = 40  has Coherence Value of 0.2532
Num Topics = 45  has Coherence Value of 0.2555
Num Topics = 50  has Coherence Value of 0.2542
Num Topics = 55  has Coherence Value of 0.2794
Num Topics = 60  has Coherence Value of 0.2593
Num Topics = 65  has Coherence Value of 0.2608
Num Topics = 70  has Coherence Value of 0.2582
Num Topics = 75  has Coherence Value of 0.2647
Num Topics = 80  has Coherence Value of 0.2653
Num Topics = 85  has Coherence Value of 0.2689
Num Topics = 90  has Coherence Value of 0.2625
Num Topics = 95  has Coherence Value of 0.2612
```

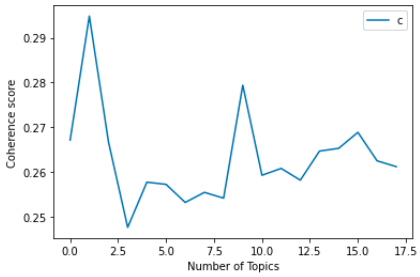

Figure 4: Finding optimal LDA model

```
(0, '0.040*"pizza" + 0.037*"people" + 0.036*"kite" + 0.035*"fly" + 0.032*"sit" + 0.027*"group" + 0.022*"frisbee" + 0.021*"table" + 0.015*"computer" + 0.014*"child"')
(1, '0.029*"train" + 0.026*"bathroom" + 0.026*"white" + 0.021*"sink" + 0.015*"people" + 0.015*"sit" + 0.014*"room" + 0.013*"large" + 0.012*"black" + 0.012*"bear"')
(2, '0.043*"man" + 0.041*"baseball" + 0.028*"hold" + 0.017*"bird" + 0.017*"water" + 0.017*"girl" + 0.016*"stand" + 0.016*"bat" + 0.015*"player" + 0.013*"beach"')
(3, '0.031*"motorcycle" + 0.023*"park" + 0.021*"soccer" + 0.020*"field" + 0.020*"man" + 0.014*"sit" + 0.013*"grass" + 0.013*"ball" + 0.012*"player" + 0.012*"scissor"')
(4, '0.053*"cat" + 0.026*"bed" + 0.019*"fly" + 0.018*"sky" + 0.017*"sit" + 0.017*"lay" + 0.017*"man" + 0.014*"stand" + 0.014*"blue" + 0.013*"cut"')
(5, '0.047*"sign" + 0.031*"train" + 0.028*"street" + 0.022*"sit" + 0.015*"red" + 0.014*"stop" + 0.013*"park" + 0.013*"large" + 0.012*"bed" + 0.012*"next"')
(6, '0.041*"bus" + 0.026*"sit" + 0.022*"dog" + 0.021*"stand" + 0.019*"grass" + 0.017*"bear" + 0.016*"next" + 0.016*"street" + 0.015*"field" + 0.014*"zebra"')
(7, '0.036*"room" + 0.031*"chair" + 0.030*"sit" + 0.025*"living" + 0.025*"table" + 0.022*"cat" + 0.016*"fire" + 0.014*"couch" + 0.014*"hydrant" + 0.013*"fly"')
(8, '0.061*"man" + 0.036*"tennis" + 0.028*"sit" + 0.024*"stand" + 0.023*"table" + 0.019*"plate" + 0.014*"white" + 0.013*"court" + 0.013*"dog" + 0.012*"food"')
(9, '0.174*"sit" + 0.079*"woman" + 0.076*"closely" + 0.073*"man" + 0.062*"girl" + 0.059*"put" + 0.054*"together" + 0.049*"look" + 0.041*"couple" + 0.030*"female"')
(10, '0.027*"man" + 0.027*"kitchen" + 0.022*"horse" + 0.022*"white" + 0.013*"girl" + 0.013*"sit" + 0.011*"next" + 0.011*"room" + 0.010*"park" + 0.009*"water"')
(11, '0.043*"people" + 0.029*"street" + 0.026*"horse" + 0.022*"parking" + 0.021*"ride" + 0.020*"sit" + 0.019*"man" + 0.015*"light" + 0.013*"meter" + 0.013*"table"')
(12, '0.204*"subway" + 0.137*"train" + 0.084*"sit" + 0.064*"boy" + 0.054*"people" + 0.050*"male" + 0.049*"arm" + 0.046*"female" + 0.040*"couple" + 0.034*"look"')
(13, '0.040*"toilet" + 0.034*"bathroom" + 0.033*"plate" + 0.027*"elephant" + 0.020*"sit" + 0.020*"white" + 0.019*"small" + 0.016*"stand" + 0.013*"pizza" + 0.012*"sink"')
(14, '0.053*"man" + 0.025*"ski" + 0.023*"snow" + 0.020*"ride" + 0.019*"person" + 0.016*"woman" + 0.012*"hold" + 0.011*"motorcycle" + 0.011*"stand" + 0.011*"skier"')
Optimal Number of Topics : 15

Perplexity:  -6.595191362404969

Coherence Score:  0.29118921286298716
```

Figure 5: Topics' keywords

4.3 Predict Topics for Caption data

Since we already have an optimal LDA model. Now we used this model to predict topics for caption data. Later we obtained dominant topic based on the percentage contribution of each words in the topics and added top 10 keywords of a topic for each image in the datasets.

4.4 Model Building

We have used transfer learning technique to extract features. As our task is similar to the object detection task so we can use pre-trained model to extract image features and apply on our use case.

For this we have used pre-trained VGG16 and VGG19 model for image processing which is trained on Imagenet datasets. We fine tuned both the pre-trained model, removed the top layer and added two Dense layer with 2056 units and 1024 units with two dropout layer with 0.5 percentage. Figure 6 shows the VGG16 model architecture.

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000
_________________________________________________________________
dense (Dense)                (None, 2056)              2058056
_________________________________________________________________
dropout (Dropout)            (None, 2056)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              2106368
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                10250
=================================================================
Total params: 142,532,218
Trainable params: 4,174,674
Non-trainable params: 138,357,544
```

Figure 6: VGG16 Model architecture

4.5 Model Training

We trained the model for 100 epochs with a batch size 64. Figure 1 shows the hyperparameters used while training the model.

Hyperparameters	
Epochs	100
Batch Size	64
Optimizer	Adam
learning rate	0.01
Loss Function	CrossEntropyLoss

Table 1: Hyperparameters used for Transfer-Learning

4.6 Model Evaluation

For model evaluation, we have used Bilingual Evaluation Understudy Score popularly known as BLEU score which is a good metric for evaluating a generated topics to a reference topics. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0.

5 Results

Figure 7 shows the topics generated on Flickr8k datasets whereas fig. 8 shows the topics generated on Coco datasets. Table 2 shows the final BLEU score of both Flickr8k and Coco datasets on two different pre-trained models i.e VGG16 & VGG19. In the table 2 we can see, BLEU scores are decreasing as topics number increases which is kind of obvious because there will be more classes and model will become too complex to predict topics for new set of images. Figure 9 shows the topics which model generated after training it where the bold topics are having higher weight value and varying as per weight value.

Topics	Flickr8k		COCO	
	VGG16 (BLEU)	VGG19 (BLEU)	VGG16 (BLEU)	VGG19 (BLEU)
10	0.839	0.834	0.818	0.817
20	0.766	0.764	0.803	0.798
30	0.725	0.707	0.768	0.782
40	0.744	0.752	0.758	0.761
50	0.707	0.694	0.782	0.794
60	0.702	0.697	0.792	0.798
70	0.673	0.668	0.764	0.778
80	0.656	0.648	0.773	0.767
90	0.686	0.682	0.768	0.776

Table 2: Results

6 Conclusion

As we can see performance on COCO datasets are better than Flickr8k datasets. Since we have considered only 8K samples from COCO due to resource constraint. The performance will further improve if we consider more number of samples. For future work, we can build multi-modal deep

Figure 7: Topics generated on Flickr8k

Figure 8: Topics generated on COCO

learning model where we can also include images captions while training the model. We can use pre-trained transformer BERT model to get caption representations and imagenet based pre-trained model to extract images features and then train single model including both the features. This will improve the accuracy further and will also improve the quality of the topics generated.

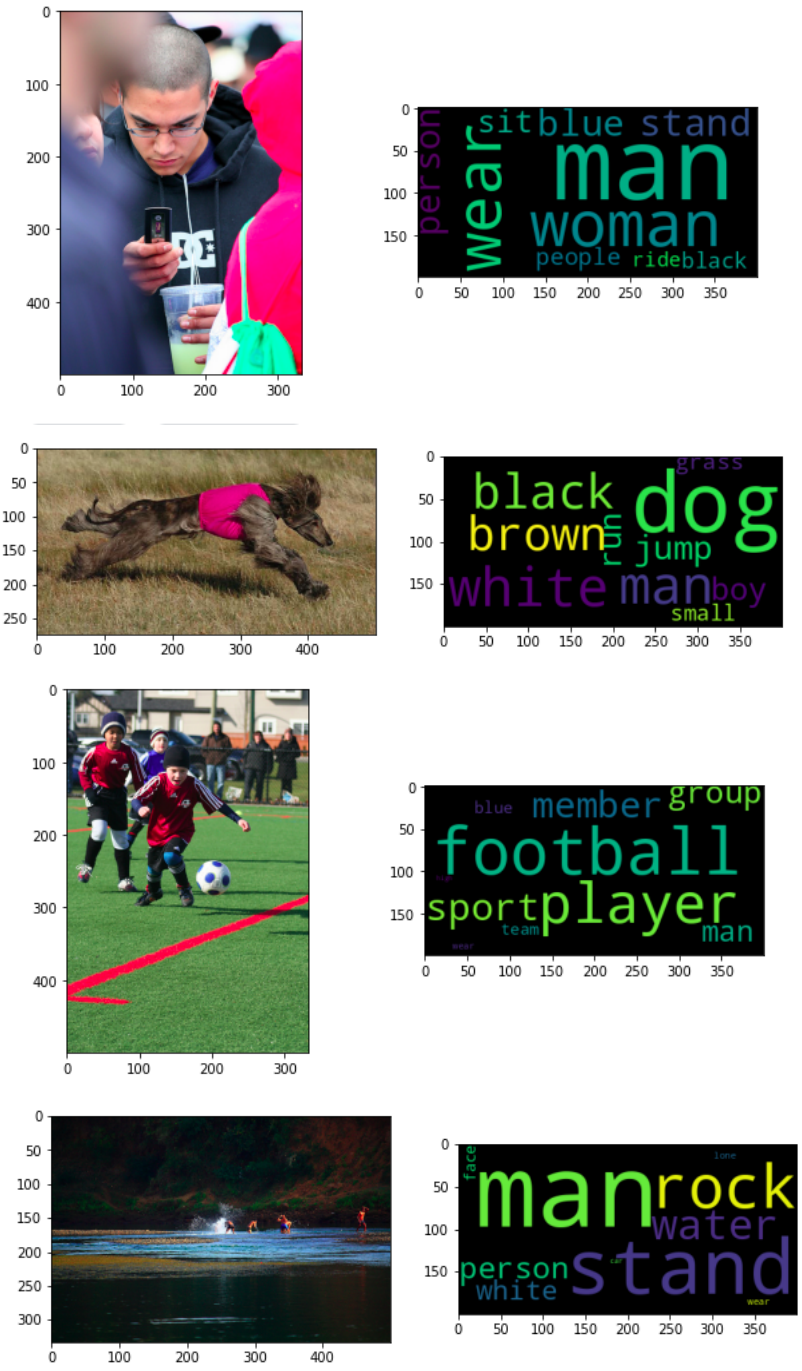

Figure 9: Generated Topics

References

Jun Li, Joo Hwee Lim, and Qi Tian. Automatic summarization for personal digital photos. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 3, pages 1536–1540. IEEE, 2003. URL https://ieeexplore.ieee.org/abstract/document/1292724.

Vasu Sharma, Akshay Kumar, Nishant Agrawal, Puneet Singh, and Rajat Kulshreshtha. Image summarization using topic modelling. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 226–231, 2015. doi: 10.1109/ICSIPA.2015.7412194. URL https://ieeexplore.ieee.org/document/7412194.

Heng Yang and Qing Wang. Grouping and summarizing scene images from web collections. In *International Symposium on Visual Computing*, pages 315–324. Springer, 2009. URL https://link.springer.com/chapter/10.1007/978-3-642-10520-3_29.

Zhihao Zhu, Zhan Xue, and Zejian Yuan. Topic-guided attention for image captioning, 2018. URL https://arxiv.org/abs/1807.03514.