Contrastive Learning for Natural Language-Based Vehicle Retrieval

Tam Minh Nguyen[1], Quang Huu Pham[1], Linh Bao Doan[1],
Hoang Viet Trinh[1], Viet-Anh Nguyen[1], Viet-Hoang Phan[2]
[1]AI Research Team
R&D Lab, Sun Asterisk Inc.
[2]Hanoi University of Science and Technology
nguyen.minh.tam-b@sun-asterisk.com

Abstract

AI City Challenge 2021 Task 5: The Natural Language-Based Vehicle Tracking is a Natural Language-based Vehicle Retrieval task, which requires retrieving a single-camera track using a set of three natural language descriptions of the specific targets. In this paper, we present our methods to tackle the difficulties of the provided task. Experiments with our approaches on the competitive dataset from AICity Challenge 2021 show that our techniques achieve Mean Reciprocal Rank score of 0.1701 on the public test dataset and 0.1571 on the private test dataset.

1. Introduction

Natural Language Processing plays a pivotal role within the linguistics field and their applications are widely used in specific tasks. However, the information in real world scenarios often comes in complex modalities. Data type containing visual perception, such as images, can be associated with text explanation or description. Simultaneously, texts can supplement images to encode richer semantic features. By enhancing image representation with text supervision, computer systems can achieve better comprehension at multiple domain data.

CityFlow-NL [11], a Natural Language-based Vehicle Retrieval task, provides multi camera tracking views of various vehicles with Natural Language (NL) query in sequence. These information is retrieved in a series of bounding boxes on each camera view of the object. This dataset opens up different research possibilities in applying multi-modal learning to perform the collaborative representation of multiple modalities. With this dataset, authors propose two foundation tasks: the Vehicle Retrieval by NL task and Vehicle Tracking by NL task with the purpose of providing future studies on the multi-camera multi-target tracking by NL description. In this paper, we concentrate on resolving the Vehicle Retrieval task presented in AI City Challenge 2021[1].

The dataset for this track is constructed upon the CityFlow Benchmark[29] by annotating vehicles with descriptions. This dataset contains over 2,000 tracks of vehicles with three unique NL descriptions under each. In particular, 530 unique vehicle tracks together with 530 query sets each with three descriptions are selected for this challenge.

The main contributions of our paper are stated as follows:

- We propose a unified model with Bi-directional InfoNCE Loss [13] to learn the vehicle and textual representations for the text-vehicle retrieval task.
- Our data processing techniques are efficient and straightforward to enhance the matching of text queries and vehicles.
- We utilize CLIP [24], a multi-modal pre-trained model, to obtain better semantic representations of both images and texts.
- This paper also make inquires about Hard Negative Mining for text and images. Despite their ineffectiveness in this task, further investigation for causes and lessons will take place.

The rest of the paper is organized as follows. Section 2 review related work briefly. We present baseline of our work in section 3. Experimental results of this Track are reported and discussed in Section 4. Final conclusion are summarized in the last section.

[1]https://www.aicitychallenge.org/

Figure 1. Vehicle movement is indicated by a green line

2. Related work

2.1. Metric learning

Metric learning aims to learn a function that reduces the distance between similar samples while pushes uncorrelated samples away. There has been numerous of losses for Metric learning that maximize distance to the negative sample while minimize distance to the positive one. One of the most effective losses is Contrastive Loss [5, 12] which calculates similarity of pairwise samples. Triplet Loss [3, 6, 32], on the other hand, constructs a triplet including a query, a positive and a negative examples. InfoNCE Loss [30] is a type of Contrastive Loss in which similarity is calculated on one positive example and k negative examples, allowing it to estimate mutual information between examples.

2.2. Vision-Language multimodal

Video-Text multimodal. With emerging sources of videos with both machine generated and human annotated scripts, multi-modal is gaining much attention for representation learning. [9] proposes a dual networks for video retrieval by text. This encodes video and text into two separated latent and concept space, enhancing representation quality and interpretability. [27] utilizes BERT-like architecture to learn cross-modal context, achieving remarkable results on video captioning. [18] sparsely samples frames from the video and fuses their representations with text embedding using Transformers.

Image-Text multimodal. Image-Text multimodal modeling is a crucial component in many tasks such as visual question answering [16, 2, 38], visual dialog [33, 22] and text-image retrieval [23, 36]. Recent literature for representation learning relies on Transformers [28, 19, 4, 17]. In [28], three different encoders are combined to connect visual and language information then trained on five tasks, leading to state-of-the-art results. Instead of random masking, [4] utilizes conditional masking on four pre-training tasks. This helps the model to quickly adapt to a new downstream task. In this work, we solve video-text retrieval problem by image-text multimodal modeling with a simgle frame.

2.3. Contrastive Language Image Pretraining

CLIP (Contrastive Language Image Pretraining) [24] is a new multimodal network of OpenAI trained on various (image, text) pairs. The model is rated for performance on more than 30 different existing computer vision datasets, spanning tasks for instance OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification. CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training. The text encoder is a modified Transformer [31] architecture with the 63M-parameter, 12-layer 512-wide model and 8 attention heads beside the image encoder use Visual Transformer [34].

3. Methodology

3.1. Dataset and Data Preprocessing

The data for this challenge is from many traffic cameras, which are synchronized in time from a city in the United States as well as from the state highway in Iowa. The organizer's original dataset for this task includes 2,498 tracks of vehicles with three unique NL descriptions each for training. Each annotation describes medium vehicle type and color, motions like turns, and relationships to other vehicles in the scene. The test data comprises 530 unique vehicle tracks and 530 query sets, each annotated with three NL descriptions. The vehicle information in each track contains image frames and the vehicle's bounding box in each frame.

Investigating the CityFlow-NL Dataset thoroughly, we endeavour to apply several data processing techniques. All images are resized to a pre-fixed size and normalized. We separated the main object, which is an individual car, by cropping it out of the last original image in the sequence of frames. Cropped images are more concentrated on major subjects, which are better for learning the object representation. Separated original images are treated as background images. Background images are preferable for learning the whole context.

To learn the vehicle's directional movement, we utilize its bounding boxes' top-left coordinates (referred to as a box point), in each frame t, as its location information of each time step t. We inject this sequential information to the background image by continuously draw lines connecting these box points. As shown in Figure 1, the green line represents the direction of the vehicle's movement. This aim of learning the vehicle's movement implicitly has a drawback.

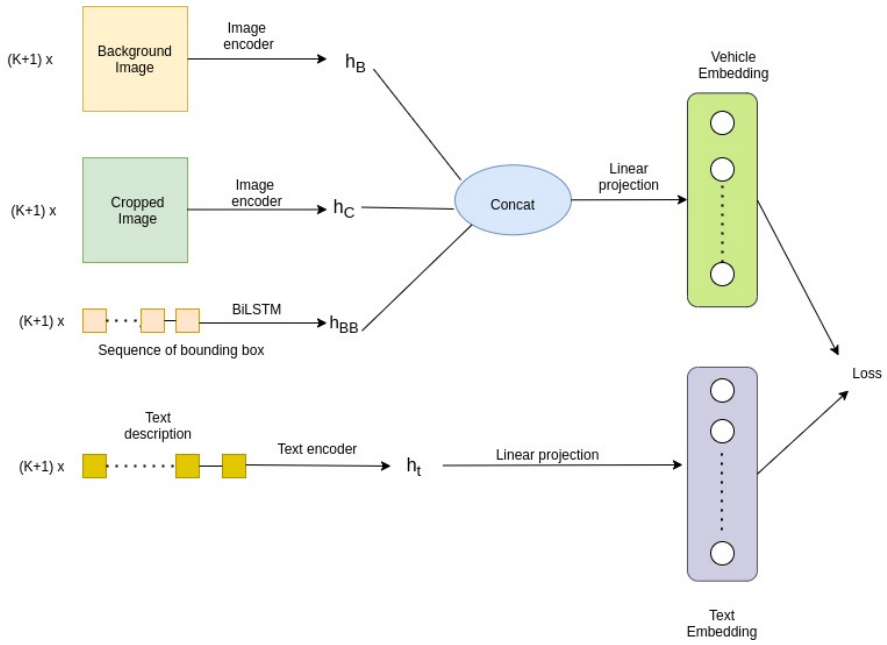

Figure 2. The architecture of our base model. In visual branch, we extract background image feature for global context and cropped object image feature for subject understanding. Since only a single frame is used, movement of the vehicle is learned from a sequence of its bounding boxes. Features of the visual branch are combined with text embeddings for further metric learning.

As it is unable to capture multi-stage movement such as an object stops then turns right and vice versa (e.g., "A gray sedan stops at the intersection for a while and turns right along the street."). Hence, to learn the object's motion directly, we use a Bi-directional Long Short Term Memory network[15] to extract this information from a sequence of box points coordinates.

3.2. Contrastive Learning with Deep Multi-input model

3.2.1 Data Sampling

At training time, for each text description in each track, we extract vehicle information (each contains background, cropped images and a sequence of box points) in the same track to form a positive vehicle-text pair. We denote the track containing the positive pair as anchor track, while its text description, vehicle are anchor text, anchor vehicle, respectively. One thing notice here is that the background image of vehicle information is randomly selected from the track's frames. We randomly sample k negative texts for the anchor vehicle, and the vehicle described by those texts are chosen as negatives for the anchor text.

At inference time, we use the last frame of the track as the background image.

3.2.2 Model

Each input example contains $2k + 1$ (vehicle, text description) pairs. While one is positive, the others are image-to-text and text-to-image negative pairs. We denote the track containing the positive pair as anchor track, while its text description, vehicle are anchor text, anchor vehicle, respectively. Background, cropped images, and the sequence of the object's bounding boxes are referred to as vehicle information.

Figure 2 demonstrates our baseline model. Both background and vehicle cropped images are encoded by a share-weights Convolution Neural Network backbone [1], pre-trained on Image Net dataset [7], to extract visual representations of the vehicle and its context.

Additionally, the vehicle's movement, represented by its sequence of box points, is not only explicitly learned by a BiLSTM but also implicitly extracted from the background image representation. Finally, we combine background and cropped images and sequential positional representations by concatenation. This unified embedding is linearly projected onto a lower-dimensional space to obtain the final vehicle representation.

On the other hand, the text description embedding is encoded by a pre-trained contextual language representation (we use BERT [8], pre-trained on the general domain, for our baseline). The $< CLS >$ token embedding of the BERT backbone's layer is linearly transformed to attain the final text representation.

To utilize the adaptability of pre-trained CLIP model, we replace the pre-trained models in backbone by CLIP model's vision and text encoders. The comparison of using these pre-trained models is discussed in the Experiment Section.

3.2.3 Bi-directional InfoNCE and Marginal Triplet Loss

To learn the representation of vehicles and textual descriptions, we minimize two loss functions: text-to-vehicle and vehicle-to-text InfoNCE loss, as we refer to as bi-directional InFoNCE loss for short [37]. The effectiveness of using bidirectional instead of only either uni-directional loss for this task is discussed in the Experiments section.

- (v_i, s_i): positive vehicle
- v_i: anchor vehicle
- S_i: anchor text
- $v_{k's}$: negative vehicles
- M: mini-batch size
- λ: the importance coefficient which weights uni-directional lossé

Text-to-Vehicle InfoNCE loss:

$$\ell^{(s\to v)} = -\frac{1}{M}\sum_{i=1}^{M}\log\left(\frac{e^{g(s_i,v_i)/\tau}}{\sum_{k=1}^{K} e^{g(s_i,v_k)/\tau}}\right)$$

Vehicle-to-Text InfoNCE loss:

$$\ell^{(v\to s)} = -\frac{1}{M}\sum_{i=1}^{M}\log\left(\frac{e^{g(v_i,s_i)/\tau}}{\sum_{k=1}^{K} e^{g(v_i,s_k)/\tau}}\right)$$

Bi-directional InfoNCE loss:

$$\ell = \lambda\ell^{(s\to v)} + (1-\lambda)\ell^{(v\to s)}$$

$$\text{Where } g(s,v) = \frac{s^\top v}{\|s\|\|v\|}$$

Besides, we also utilize the Marginal Triplet Loss, both bi-directional and uni-directional, to compare with the InfoNCE loss.

Marginal Triplet loss (MTL):

$$\ell\,(a,p,n) = \max\{d(a,p) - d(a,n) + \text{magin}, 0\}$$

Where: $d(x,y) = \|x-y\|^2$

Text-to-Vehicle MTL:

$$\ell^{(s\to v)} = \frac{1}{M}\sum_{i=1}^{M}\ell\,(s_i, v_i, v_k)$$

Vehicle-to-Text MTL:

$$\ell^{(v\to s)} = \frac{1}{M}\sum_{i=1}^{M}\ell\,(v_i, s_i, s_k)$$

Bi-directional MTL:

$$\ell = \lambda\ell^{(s\to v)} + (1-\lambda)\ell^{(v\to s)}$$

3.2.4 Hard Negative Mining for image and text

For hard negative vehicle mining, we notice that the same video camera tracks multiple vehicles. Therefore, for anchor text in track, we use vehicle information of other tracks in the same video camera as negative examples to pair with the anchor text, because they track different objects while sharing the similar backgrounds. We suspect that with very different backgrounds (easy negative examples), the text query easily matches the right vehicle, resulting in small-to-zero loss and gradients. In contrast, similar backgrounds tracking different objects creates false positive examples, forcing the model to learn harder to detect true positive pairs.

To sample hard negative examples for texts, we use SBERT [25], a pre-trained multi-lingual sentence embedding model, to extract text descriptions' representations and measure their pair-wise cosine similarities. The idea is that texts which are similar to the anchor track's descriptions, while describing different vehicle types and movements, are sampled to pair with the anchor vehicle as negative pairs. However, there are exist texts describing different objects which have the same type and semantic context. For example, two texts : "A red sedan is turning right" and "A red sedan turns right" describes two different red sedans, which happen to both turning right. Hence, relying on the raw similarity scores leads to false-negative examples.

We propose a simple solution to set any "too-high" similarity score (i.e., greater than a threshold) to zero. We decided to inject randomness into the process of hard text mining by calculating the similarity score of text-tracks pairs, then collecting m most similar tracks to the text. At each iteration, we sample k negative tracks, followed by sampling one text in each of these track.

To examine the effectiveness of text and vehicle hard negative mining, we apply them individually and in combination. When mining negative vehicles, the vehicle information is taken from tracks containing negative backgrounds. The negative texts are sampled randomly from the rest of the dataset (i.e., excluding the track containing the anchor text).

In contrast, when applying hard negative text mining, we simply gather vehicle information from tracks containing negative texts. Finally, we extract negative vehicle information and text through both mining processes.

3.2.5 Inference

We use the last vehicle frame in the frame sequence to measure the similarity with the test queries as the average of the similarities between all pairs of vehicle frame in the track and Natural language description in the test queries.

$$\mathcal{S}\left(V_i, \{S_1, S_2, S_3\}_j\right) = \frac{1}{3 \cdot t} \sum_{j=1}^{3} \mathcal{S}\left(V_i, S_j\right)$$

$$\text{Where } S(s, v) = \frac{s^\top v}{\|s\|\|v\|}$$

4. Experiments and Results

4.1. Experiment Setup

4.1.1 Data preprocessing configuration

The number k of negative examples is 5 in our setting. All images are resized to have height and width of 224 and 224 respectively and normalized by the statistics of ImageNet dataset. Box points values are divided by 224 to have values in $[0, 1]$.

4.1.2 Train and validation dataset split-up

Since frames in tracks can come from the same videos but tracking different objects, randomly choosing tracks for validation set would cause data snooping phenomenom. Therefore we choose tracks from S01 of the dataset for model evaluation process and report finished results.

4.1.3 Model and losses hyper-parameters

In our baseline, we use pre-trained ResNet50[14] as background image encoder. In the sharing weights setup, the cropped image encoder is the same as background's. Otherwise, its encoder is pretrained ResNet34[14]. The text encoder is pre-trained BERT model on general domain. Experimenting with CLIP model, we use pretrained version "ViT-B/32"[10].

The embedding dimension of both vehicle information and text linear projection layers are 256, the hidden size of BiLSTM layer is 64. The temperature of InfoNCE loss is set to be 0.07, the importance coefficient of directional loss is 0.3, the margin of Marginal Triplet Loss is 0.5. In experiments with Marginal Triplet Loss, we perform L_2 normalization on each vehicle and text embedding.

4.1.4 Training hyper-parameters

We use AdamW[20] as our optimizer with learning rate of 10^{-5}. Each experiment is trained with 10 epochs. We train our model with mixed precision in all settings.

Due to limited computational resources power, we choose Google Colab[2] GPU for training. Since the maximum GPU memory of Colab is $16GB$ are available for batch size of 16.

4.2. Evaluation Metrics

To evaluate the system, we mainly use Mean Reciprocal Rank (MRR) with additional Recall@5 and Recall@10. They are standard metrics for information retrieval, which has been introduced in [21]. Given a set of queries Q, MRR is calculated as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

where $rank_i$ is the highest-relevant answer of the i-th query.

4.3. Results

Table 1 demonstrates the effectiveness of utilizing the CLIP pre-trained model as our backbone. The significant gap in using pre-trained ResNet50 and BERT model as image and text extractors instead of the CLIP encoders sheds light on the better adaptability to task-specific domains of the multi-modal pre-trained model.

Table 1. The effectiveness of using pretrained CLIP model

	MRR	**Recall 5**	**Recall 10**
Using pretrained CLIP model	**0.1778**	**0.2604**	**0.4349**
Baseline	0.0910	0.1081	0.2039

Table 2 indicates the results of applying hard negative mining techniques. Applying HNVM and HNTM individually or in combination significantly decreases the model's performance by 0.0472, 0.0314, 0.0831 respectively. We suspect that, for HNTM, too hard negative examples cause difficulties in the early stage of training. On the other hand, for HNVM, since the negative background is similar to the positive one, the model can simply ignore the background information and focus on the cropped image and the bounding boxes information.

Table 5 demonstrates the effectiveness of using bi-directional InfoNCE loss instead of unidirectional one. Using only vehicle-to-text and text-to-vehicle InfoNCE loss worsens the model's performance by 0.1778 and 0.1138 MRR score respectively.

[2] https://colab.research.google.com/

Table 2. Results of applying Hard Negative Mining

	MRR	Recall 5	Recall 10
No hard negative mining	**0.1778**	**0.2604**	**0.4349**
Hard negative text mining	0.0947	0.1229	0.2285
Hard negative vehicle mining	0.1306	0.1990	0.3243
Hard negative text and negative mining	0.1464	0.2088	0.3735

Table 3. Comparison of bidirectional InfoNCE and Marginal Triplet Loss

	MRR	Recall 5	Recall 10
Bidirectional InfoNCE	**0.1778**	**0.2604**	**0.4349**
Bidirectional Marginal Triplet Loss	0.1138	0.1499	0.2801

Table 4. Comparison of share-weight and no share-weight visual models

	MRR	Recall 5	Recall 10
Share-weight	**0.1778**	**0.2604**	**0.4349**
No share-weight	0.1303	0.1916	0.3194

Table 5. The effectiveness of bidirectional loss

	MRR	Recall 5	Recall 10
Bidirectional InfoNCE	**0.1778**	**0.2604**	**0.4349**
Text-to-vehicle InfoNCE loss	0.1412	0.2064	0.3538
Vehicle-to-text NCE loss	0.1596	0.2310	0.3956

Table 6. The private test result

Rank	Team ID	Team Name	Score
1	132	Alibaba-UTS	0.1869
2	17	TimeLab	0.1613
3	36	SBUK	0.1594
4	20	SNLP	0.1571
5	147	HUST	0.1564
6	13	HCMUS	0.1560
7	53	VCA	0.1548
8	71	aiem2021	0.1364
9	87	Enablers	0.1314
10	6	Modulabs	0.1195

Table 3 indicates a significant gap when optimizing bi-directional InfoNCE loss instead of bi-directional marginal triplet loss. We suspect that this poor performance of using triplet loss is due to its optimization difficulties [35] as an inefficient negative sampling method causes the model to learn slowly.

In our experiments, we also tried learning representations of background and cropped images using two same-version CLIP model image encoders instead of sharing weights. This is inspired by our attempt to learn different information about background image (i.e., the context surrounding the vehicle) and cropped image (i.e., the vehicle properties including color, size and type). However, the result of this experiment was unfavorable shown in Table 4 and needs further investigation.

As shown in Table 6, we currently rank 4th on the private test published by the organizers.

5. Conclusion

Using contrastive learning, we propose a unified model to learn the vehicle and textual representations for the text-vehicle retrieval task. We competed in AICity Challenge 2021 and received significant results in both public test and private test.

Future work will focus on determining the effectiveness of each factor in our baseline. Besides, more investigation will be made to discover the reasons behind our Hard Negative Mining process's ineffectiveness. Additionally, we will utilize video embedding architecture such as ConvLSTM [26] and text-video retrieval architectures to enhance the learning of sequential information in videos.

Acknowledgment

This work is partially supported by ***Sun-Asterisk Inc***. We would like to thank our colleagues at ***Sun-Asterisk Inc*** for their advice and expertise. Without their support, this experiment would not have been accomplished.

References

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017. 3

[2] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering, 2019. 2

[3] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *J. Mach. Learn. Res.*, 11:1109–1135, Mar. 2010. 2

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2020. 2

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. 2

[6] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop, 2016. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 3

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 3

[9] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 5

[11] Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions, 2021. 1

[12] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006. 2

[13] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Neurips*, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. 3

[16] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa, 2020. 2

[17] Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Weak supervision helps emergence of word-object alignment and improves vision-language tasks, 2019. 2

[18] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling, 2021. 2

[19] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training, 2019. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008. 5

[22] Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline, 2020. 2

[23] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data, 2020. 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 2

[25] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. 4

[26] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015. 6

[27] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning, 2019. 2

[28] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers, 2019. 2

[29] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification, 2019. 1

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 2

[32] Jiang Wang, Yang song, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking, 2014. 2

[33] Yue Wang, Shafiq Joty, Michael R. Lyu, Irwin King, Caiming Xiong, and Steven C. H. Hoi. Vd-bert: A unified vision and dialog transformer with bert, 2020. 2

[34] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020. 2

[35] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful, 2021. 6

[36] TAN YU, Hongliang Fei, and Ping Li. Cross-probe {bert} for efficient and effective cross-modal search, 2021. 2

[37] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text, 2020. 4

[38] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa, 2019. 2