Tracked Vehicle Retrieval using Natural Language Descriptions

Aishwarya Jadhav
anjadhav@cs.cmu.edu

Benny Jiang
xinhaoji@cs.cmu.edu

Vivek Sourabh
vsourabh@cs.cmu.edu

Xiyang Hu
xiyanghu@cmu.edu

Eric Huang
jiahuihu@andrew.cmu.edu

Abstract

We propose a model integrating vision and language features for the natural language based vehicle retrieval task. Our model learns paired vision-text similarity using various text and visual features extracted from the tracks to minimize the InfoNCE and Triplet Losses. We introduce several video encoder backbones to extract the spatio-temporal features from the track videos. Further, we use transformer based language models to extract language embeddings. By using different NLP data augmentation techniques, we generate additional natural language descriptions to boost our model performance. We test multiple combinations of features extracted from the vision and text data. We could observe that the more features are added to the model, the better MRR and Recall we achieve. We also propose a two-stage model where we first extract candidates based on local features, then we do further retrieval based on global motion features. Compared with the baseline, our new video embedding methods show significant improvement in both MRR and top Recall@10. The results of our final multimodal model exceed the baseline by 8 percent in MRR and more than 10 percent in Recall. Our code and results are available at github.com/vivek10p14/11775Proj.

1. Introduction

Vehicle Retrieval is crucial task for important aspects of a smart city such as efficient traffic management, etc. With time as the number of vehicles on the road increases, the amount of data collected/stored increases. This calls for an efficient way to fuse different modalities of data, to create a system that is able to efficiently retrieve useful information as per the users request.

We present a framework to retrieve vehicle tracks from a given set of videos using a natural language query. Taking a set of videos and a natural language query as input, the aim of the framework is to generate a ranked list of videos that match the given natural language description. The query would be composed of a static and a dynamic part. The static part of the query is aimed to represent the information about the vehicle such as size, color and so on. Whereas, the dynamic part of the query would include information such as vehicle motion and its relation to the other/vehicles and environment.

2. Literature Review

Natural language-based video retrieval aims to search a specific video matching the given language description from a large number of candidate videos. For this multimodal task, many previous approaches aim to separately extract representations of natural language queries and videos and estimate the similarity between these pairs to find the top-n most similar videos to retrieve[1; 6; 16; 19]. Usually, the similarity metric can be an MLP learned to estimate the closeness between language and video embeddings. Similarity can also be calculated using simpler distance metrics [27] like Euclidean or Cosine distance, if the query and video embeddings are projected to a joint semantic space. The initial research on cross-modal retrieval focused on image-text retrieval, and video-text retrieval[6; 19] was studied subsequently

Most of these works employ language encoders such as Word2Vec [18], LSTM [12] to extract query embeddings. [22] uses Glove embeddings. Some approaches such as [1] employ transformer based approaches such as BERT [5] or RoBERTa [17] to extract the textual features. For the extraction of video representations, prior works have used Two-Stream Network [20], C3D [3], S3D [26], etc. Transformer-based approaches for video feature extraction are also quite popular in recent years [2; 21]. Other works also employ additional multimodal features such as audio and [7; 8]. Additional approaches utilized moment retrieval [9; 11] and object retrieval [4] for further context.

3. Dataset

Our dataset is from the 2022 AI City Challenge Track 2: Tracked-Vehicle Retrieval by Natural Language Descriptions. Specifically, the training dataset contains 2,155 vehicle tracks. Every vehicle track has been annotated with 3 natural language descriptions of the target. For the test dataset, there are 184 tracks of candidate target vehicles, and 184 queries with each of them containing three natural language descriptions of the vehicle target.

There are some noises in the training and testing dataset, which brings some obstacles to our model. In the query dataset, we observe that some descriptions of the same query conflict with each other. For example, in a test query, one description is "A black SUV turned right at the intersection.", and another description is "A black SUV turns left at an intersection.". The two descriptions of the same target conflict in terms of the direction the SUV turns. One more example is: "A black sedan stops." versus "A black sedan turns left.". The two descriptions have some discrepancies in the sedan's movement. In addition, we also observe that there are several tracks with exactly the same three language descriptions. For example, the four different tracks (`e25719da-ea73-44b5-94a8-3da35bc39e9e`, `c5522d2d-37c7-4d61-8f4d-86d785490fd9`, `4f3a93fb-1089-470e-aba6-b89dab3da082`, `e974d968-ccb9-4f6c-9af3-fb4f390e355b`), have the same following three natural language descriptions: "A white pickup truck waits for the light and then passes the intersection." "A white truck stops at the intersection before driving straight." "A white pickup stop after another white vehicle."

4. Baseline Model

4.1. Baseline Model Reproduction

In our project, we are using the winner system from AI City Natural Language Vehicle Retrieval last year [1] as our baseline model. This model calculates the relevance between text description and vehicle tracks with the following features.

1. For text description, the system generates sentence embeddings from BERT[5] or RoBERTa[17].
2. For vehicle image crops from the video tracks, the model applies pretrained visual encoding models such as EfficientNet[24] to generate crop image embeddings.
3. The system also combine vehicle crops in different frames into a same background image to generate a motion image, and then use the same pretrained visual model to generate motion image embeddings.

The model includes MLPs to project each feature embedding into the same dimension. During training, the model optimizes on Symmetric InfoNCE Loss[25] between text feature project and crop image feature projection, and the same loss function between another text feature project and motion image feature projection. There are different configurations regarding the type of pretrained model, whether using motion image, and whether adding a classification loss during training. In our project, we reproduce and trained the best single model configuration(RoBERTa large + EfficientNet-b3 + motion image + no classification) reported by the authors on our dataset.

As in Table 2, we compared our retrained model with the authors' pretrained weights. Our reproduced checkpoint is slightly worse, and there are two possible reasons. 1) The baseline authors had more than 2400 tracks of training data, while we have about 1900 training tracks. 2) Due to budget and time limit, we trained our model for **32 epochs for 48 hours on a g5.2xlarge instance**. while the original model was trained for more than 40 epochs with 4 GPUs. In a word, we have reproduced a reasonable baseline that we can further improve in this project.

4.2. Error Analysis

We did an error analysis of the baseline model. We listed the queries and tracks on which the model performs worst, and found the potential reason for bad retrieval. Here we list three worst retrieval example in Figure 1. For each of them, the correct track is ranked after 100 out of 207. Having taken a closer look at the negative examples. We find three potential reasons that the model doesn't perform well in some cases.

1. In all three samples, we see that there are descriptions that tell the vehicles and their relationship with other vehicles. For example, "behind another black SUV", "followed by another red SUV", but in the motion images and crops, these information is actually filtered out.
2. Some of the images, vehicles, and descriptions are quite similar. For example, Sample 1 and Sample 2 are almost the same. If the model doesn't take surrounding vehicles into account, it will not differentiate these two tracks.
3. As mentioned in the previous section, there are noises in the descriptions. For example, "followed by" should be "following" in Sample 2 and 3.

5. Our Approach

We tried to implement two different approaches to achieve improvements over the baseline. The first approach involves addition of new features to the baseline model. Essentially, we introduce a video encoder backbone to extract the spatial temporal features from the track videos. The figure 3 below

Model	MRR	Recall@5	Recall@10
Reproduced Model	0.2034	0.2705	0.4348
Pre-trained Weights	0.2213	0.3236	0.4879

Table 1. Baseline Model Validation Performance.

shows the overall architecture of this approach. In the second approach (figure 4), we break down the ranking and retrieval pipeline into 2 stages: One for the local features and the next one for global motion features. For both of these approaches, we also use various augmentation techniques to generate more training data.

In this section we first go over the various augmentation techniques we employed. Next, we give a description of the two approaches we implemented.

5.1. Data Augmentation for Natural Language Queries

In the baseline method, they applied data augmentation on natural language descriptions by emphasizing the subject in the sentence in the beginning and adding a new sentence with the subjects, as shown in Figure 2. They used Spacy library to implement this method. Based on our error analysis, we find that emphasis on subjects of descriptions is not enough, and many tracks even have same subjects, so we finetuned a BART[15] seq2seq model to paraphrase one of the original natural language descriptions as an extra description. More specifically, we followed the following steps.

1. Data Preparation: From original dataset, each track has three descriptions, so they very likely carry the same meaning, but there are sentences for same track that are very different. Therefore, we selected pairs of descriptions of same vehicles with cosine similarity of RoBERTa sentence embeddings over 0.9 as training data. There are 1500 pairs in total.

2. Finetuning: We then finetuned BART model with a prefix "paraphrase".

3. Augmentation By Paraphrase: Paraphrase the third of the descriptions for each vehicle that is to augment natural language queries.

4. Augmentation By Subject: We continued the method of baseline. For any NL query set, we extract noun phrase denoting the subject vehicle attributes. These phrases are prepended to the queries. Extracted subjects together also form a 5th query.

An example of NLP augmentation has been shown in the Figure 2

5.2. Using Spatial Temporal Visual Features

The baseline model uses image level features to encode the motion of a vehicle in a given track. Although, this approach provides encouraging results, we believe that using image level features to encode motion information might lead to some information loss and therefore propose the use of a video level feature extraction model, to capture the global motion of a subject vehicle. [10] shows that although using clip based data for training is computationally more expensive, if the spatial-temporal relations in a clip are captured effectively it can prove to be extremely beneficial. We use the R(2+1)D-34 model architecture pre-trained on IG-65M dataset to extract the spatio-temporal features. Although, there is a domain mismatch between the data used for pre-training the above mentioned model and the task at hand, [10] points out, that models pre-trained on such large volumes of data tend to corroborate transfer learning. Building on this hypothesis, we experiment with different versions of the IG-65M model trained with different clip sizes, and find out the each of these feature types helps us improve upon the baseline model performance.

By error analysis, we found that baseline system missed temporal movement and street view evidence and hence used video embeddings.We incorporated R(2+1)D-34 model pre-trained on the IG65-M dataset as our visual backbone and tried model training with different configurations. The original IG-65M model provides multiple versions trained on clip sizes 32 and 8 respectively. Using these models we generate features for different clip sizes (the model generates a feature vector for every clip) and aggregate those features to generate a single video level feature. We use the model pre-trained on the clip size 32 to generate features for clip size 16 and 1, and use model pre-trained on clip size 8, to generate features for clip size 8.

5.3. Multi-Stage Retrieval

Our queries consist of description of the vehicle attributes as well as a description of the surroundings and motion of the vehicle. Based on this observation, we decided to divide the end-to-end pipeline of the baseline model into 2 stages.

5.3.1 Stage 1: Local feature-based candidate extraction

In this stage, we just focus on the local attributes of the vehicle without any consideration of its environment or motion

Sample 1, ranked 166, descriptions:
"A silver SUV runs straight down the highway behind another black SUV."
"A gray hatchback runs down on the left lane of the road."
"A silver hatchback continues straight until reaching a group of cars."

(a) motion image

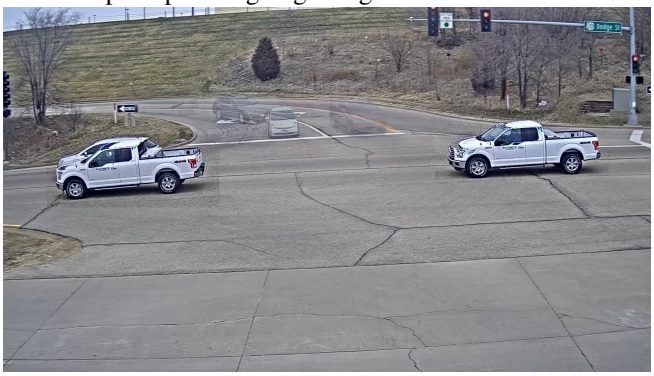
(b) crop image, in red box

Sample 2, ranked 102, descriptions:
"A silver pickup drives down a busy road."
"A gray SUV runs down the street followed by another red SUV."
"White SUV going straight on road."

(c) motion image

(d) crop image, in red box

Sample 3, ranked 100, descriptions:
"A white pickup truck runs down the street followed by another white vehicle."
"A white pickup drives down a road."
"A white pickup truck going straight down the street."



(e) motion image

(f) crop image, in red box

Figure 1. Negative Sample Outputs of Baseline Model.

"A mid-sided blue sedan goes straight through an intersection behind a blue vehicle."
"A black sedan keeping straight down the street followed by another black vehicle."
"A black sedan goes down the street after a blue sedan."

"A mid-sided blue sedan. A mid-sided blue sedan goes straight through ……. "
"A black sedan. A black sedan keeping straight down the street followed ……"
"A black sedan. A black sedan goes down the street after a blue sedan. "
"A black car. A black car runs down the road following another blue vehicle." (paraphrase)
"A mid-sided blue sedan. A black sedan. A black sedan. A black car. "

Figure 2. Example of Description Augmentation in Baseline Method.

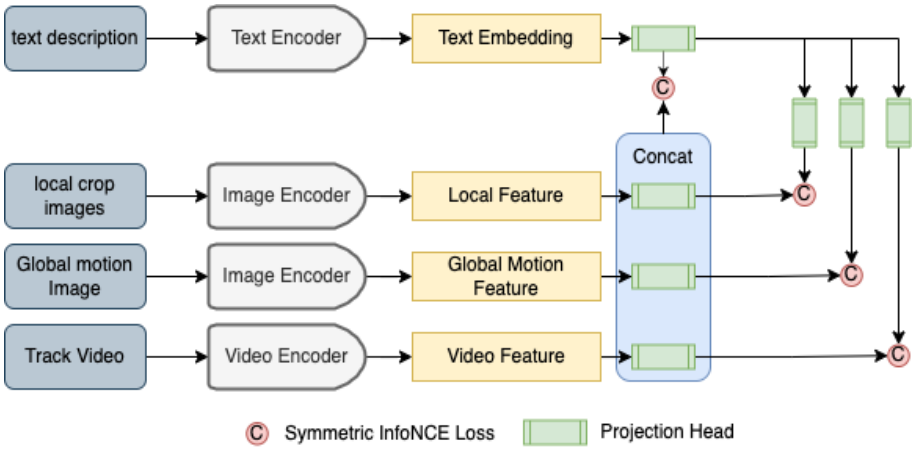


Figure 3. Diagram of Our Retrieval System, with Video Encoding Features

attributes. For this, we extract the subject phrases from our queries which contain descriptions of the vehicle attributes. For example, if the query is "A blue mid-size sedan turns right at an intersection", we extract "a blue mid-sized sedan" from this query as the subject phrase. For the visual features, we extract crops of the target vehicle from each of the tracks. We then fine-tune the text-encoder and the local image encoder portions of the baseline model to extract embeddings for the subject phrase and the local image crops. The output of stage 1 is a list of all candidate tracks that contain the car that matches the query subject description as determined by the Cosine Similarty between these text and image embeddings.

We fine-tune the RoBERTa model [17] for text encodings. For the local image crops we fine-tune the ReID pipeline based on SEResNeXt50 [14] from the baseline system. We use the Triplet loss [13] to learn multi-modal embedding space by jointly training an image encoder and text encoder to maximize the cosine similarity of the text embeddings (anchor) with the positive image crop and minimize it with the negative image crop. Furthermore, we introduce the instance loss to learn the instance-level features. We model the problem as a muli-label classification/similarity problem. For a given subject-phrase query, there will be multiple tracks that contain the vehicle that matches the description. Hence, we use the BCE loss with Sigmoid activations for the multi-label classification projection heads that provide inputs to the Instance Loss. Similarly, the local image embeddings from multiple tracks may have high similarity with the query embedings and we generate appropriate positive-negative pairs while training.

5.3.2 Stage 2: Global Motion-based retrieval

In this stage we aim to retrieve the correct track from all the candidate tracks retrieved by stage 1. The correct track should match the global environment and motion descriptions present in the query along with containing the appropriate vehicle. Since, all our candidate tracks contain vehicles that match the local descriptions in the query, we can replace these descriptions from the query by a new token: "vehicle". For the earlier query example, it will be transformed to "vehicle turns right at an intersection". We believe this transformation will aid the model to focus on the environment and motion details of the vehicle in the stage 2. For the visual features, similar to the baseline model, we use global motion images that contain the averaged background and crops of the vehicle at various positions throughout the track. We train the text and image encoders on the sets of transformed queries and the motion images of the candidates to retrieve the final track. Figure 4 shows the overall architecture of this

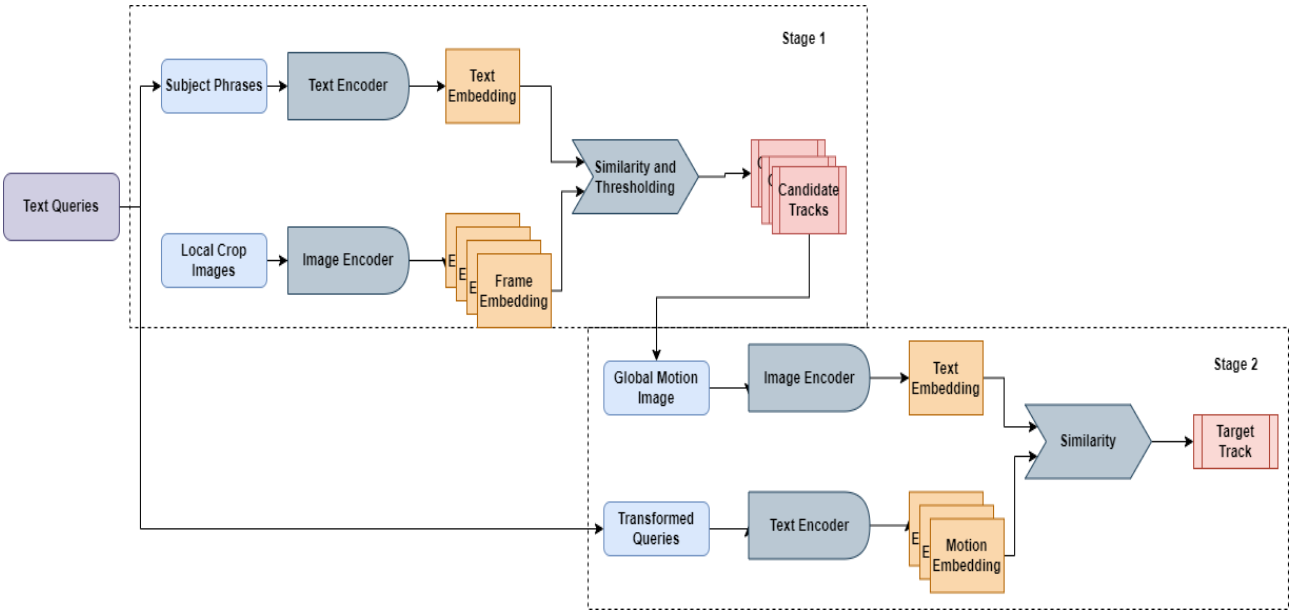


Figure 4. Architecture of the Multi-Stage Model

pipeline.

For this pipeline we use RoBERTa [17] for the text encodings as before. For the visual backbone we use the EfficientNet B3 model [23]. While training, we try to maximize the Cosine Similarity between the query embedding and the correct track embedding while minimizing this similarity between the query and all the other candidate track embeddings. We do this by training the model to minimize the Triplet Loss[13] implemented using the negative cosine similarity as the distance metric.

5.3.3 Retrieving and Ranking Tracks

During the inference, we average all the frame features of the target in each track as track visual features for stage 1 visual embedding. Similarly the query embeddings are also averaged fro both stages. Then, in stage 1, for every query, we calculate the Cosine Similarity for all the track embeddings and pick the tracks having similarity > 0 with the query. For stage 2, we again compute the similarity between the transformed and averaged query embeddings and the candidate track motion image embeddings to rank the tracks according to their scores and retrieve the one with the maximum similarity score.

6. Experiments and Result Analysis

The Table 2 display the experiments and corresponding results. The evaluation metrics we used are Mean Reciprocal Rank(MRR) and Recall@n.

6.1. Experimental Results

We continue using vehicle crop images to extract local vehicle attribute-specific features using ReID backbone used in baseline. We use pretrained RoBERTa large with projection layers to extract text embeddings. Our model that incorporates video embeddings is trained for around 15 epochs as compared to the baseline model that takes around 40 epochs to converge. Even with a significant difference in training time we achieve comparable MRR scores and significantly outperform the baseline on the recall@5 and recall@10 metrics. Further, we also observe that using an ensemble of different approaches as listed in 2 leads to an improved performance across all three metrics.

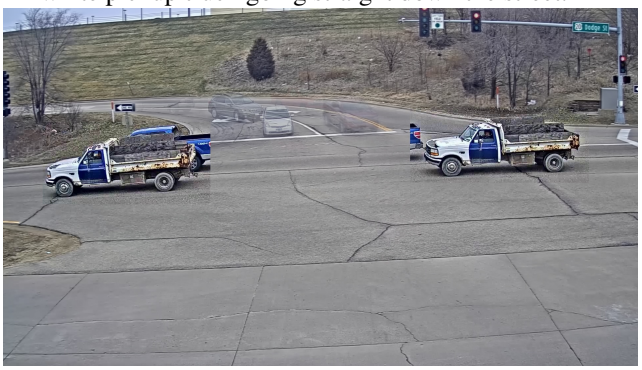
For our 2-stage model, we first independently evaluate the 2 stages and then compare it with the performance of the combined pipeline. We observe that in stage 2, since the model only considers the candidate set of tracks retrieved from stage 1, its MRR performance is better than the baseline model that considers all possible tracks in an end-to-end pipeline. The division of features (local and global) between the 2 stages enables the stage 2 to solely focus on the global features and get better retrieval performance. Given the correct set of candidate tracks, the stage 2 model achieves an MRR of 0.48. However, an evaluation of stage 1 revealed to us the bottleneck of our system. Our stage 1 model gives subpar performance in terms of the top5 (0.25) and top10 (0.33) recall while extracting the candidate tracks that may contain the vehicle described in the subject query. This results in a sub-optimal candiadate set for the stage 2 model resulting in an overall reduction in the MRR of the system as

Sample 1, ranked 113, descriptions:
"A black truck is driving on the road pulling a long trailer."
"A large black pickup truck with a trailer runs down the street."
"A large pickup truck with a trailer head straight in the right lane."
"A large pickup truck going straight down the street passing an intersection."

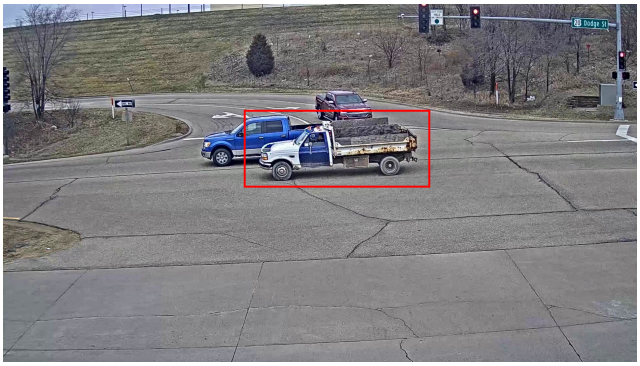
(a) motion image

(b) crop image, in red box

Sample 2, ranked 108, descriptions:
"A white pickup passes a blue pickup."
"Beige pick up truck keep straight."
"A white pickup keeps straight."
"A white pickup truck going straight down the street."



(c) motion image



(d) crop image, in red box

Sample 3, ranked 106, descriptions:
"Silver car keeps straight in a line of other cars."
"A gray sedan stops at the intersection."
"A silver sedan runs down the street."
"A silver sedan runs down the street followed by another black vehicle."

(e) motion image

(f) crop image, in red box

Figure 5. Negative Sample Outputs of Video Feature Model.

Model	MRR	Recall@5	Recall@10
crop + motion background (baseline)	0.2034	0.2705	0.4348
crop + ig65m (clip 8)	0.2025	0.3140	0.4685
crop + ig65m (clip 16)	0.1944	0.3140	0.4734
crop + ig65m (clip 16) + baseline	0.2495	0.3865	0.5700
crop + ig65m (clip 1,16)	0.2702	0.4009	0.5700
crop + ig65m (clip 1,8,16) + baseline	**0.2810**	**0.4154**	**0.5942**
2-stage model (stage1 threshold: 0)	0.2261	0.3301	0.4618
2-stage model (stage1 threshold: -0.3)	0.2709	0.3995	0.5764

Table 2. MultiModal Validation Performance.

described in table 2. Here we explore different settings of the threshold similarity from stage 1 to generate the candidate set for stage 2. Future work could involve improving the performance of stage 1 retrievals to remove the bottleneck and boost the accuracy of the overall system.

6.2. Case Studies

With the same method that we analysed the baseline model, we also extract the worst predictions by our model with video features, shown in Figure 5. We have the following findings:

1. The worst cases are totally different than those of the baseline models, meaning that they are very likely having decorrelated error. The cases that the baseline model couldn't handle well are improved by video models.

2. In Sample 1 and Sample 3, the paraphrase sentence(the 4th ones) have some mismatch with the original description, making the queries inaccurate. NLP augmentation by BART brought some noise to the natural language data that lead to some bias in training and error in predictions.

3. There is also noise in video and images. For example, in Sample 1, the vehicle crop is a partial image, and in Sample 2, the car is blue and white while the query is of a white vehicle.

7. Conclusion

In this project, we proposed a vehicle track NL retrieval model leveraging vision and text encoder to learn paired similarity using Symmetric InfoNCE Loss and Triplet Loss. Compared with the baseline, our systems that combined video encoding features with text and static image features, have effectively improved all different evaluation metrics. It can be seen that MRR and Recall can also be further improved as we adopt two-stage reranking method. Our result and code are available at github.com/vivek10p14/11775Proj.

References

[1] Shuai Bai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Yi Yang, and Hongxia Yang. Connecting language and vision for natural language-based vehicle retrieval. *CoRR*, abs/2105.14897, 2021.

[2] Max Bain, Arsha Nagrani, Gul Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *arXiv preprint arXiv:2104.00650*, 2021.

[3] Joao Carreira and Andrew Zisserman. Quo vadis action recognition? a new model and the kinetics dataset. *IEEE Conf. Comput. Vis. Pattern Recog*, 2017.

[4] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee K Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Image and video to sentence matching by visual feature prediction. 2016.

[7] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. Mdmmt: Multidomain multimodal transformer for video retrieval. *arXiv preprint arXiv:2103.10699*, 2021.

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. *Eur. Conf. Comput. Vis., volume 5. Springer*, 2020.

[9] Jiyang Gaoa, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: temporal activity localization via language query. *IEEE Int. Conf. Comput. Vis.*, 2017.

[10] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition, 2019.

[11] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments

in video with natural language. *IEEE Int. Conf. Comput. Vis.*, 2017.

[12] Sepp Hochreiter and J urgen Schmidhuber. Long short-term memory. neural computation. 1997.

[13] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *CoRR*, abs/1412.6622, 2014.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *IEEE conference on computer vision and pattern recognition,*, abs/1412.6622, 2018.

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[16] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. *IEEE Conf. Comput. Vis. Pattern Recog*, 2014.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint, arxiv:1301.3781*, 2013.

[19] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkila, and Naokazu Yokoya. Learning joint representations of videos and sentences with web image search. *European Conference on Computer Vision,Springer*, 2016.

[20] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint*, 2014.

[21] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer.

[22] Ziruo Sun, Xinfang Liu, Xiaopeng Bi, Xiushan Nie, and Yilong Yin. Dun: Dual-path temporal matching network for natural language-based vehicle retrieval.

[23] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks., journal = International Conference on Machine Learning, volume = abs/1412.6622, year = 2019.

[24] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[25] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[26] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. *Eur. Conf. Comput. Vis.*, 2018.

[27] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. *Adv. Neural Inform. Process. Syst.*, 15, 2002.