A Multi-granularity Retrieval System for Natural Language-based Vehicle Retrieval

Jiacheng Zhang[1,2‡*] Xiangru Lin[1*] Minyue Jiang[1] Yue Yu[1] Chenting Gong[1]
Wei Zhang[1] Xiao Tan[1] Yingying Li[1] Errui Ding[1] Guanbin Li[2†]
[1]Department of Computer Vision Technology (VIS), Baidu Inc., China
{linxiangru,jiangminyue,yuyue15,gongchenting,zhangwei99,tanxiao01,liyingying05,dingerrui}@baidu.com
[2]Sun Yat-sen University, China
zhangjch58@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn

Abstract

*We focus on the task of the Natural language-based vehicle track retrieval of the 6th AI City Challenge. Performing target vehicle retrieval using natural language descriptions is a comprehensive task, requiring a model to first understand the semantics of the language and vision modalities and then match them to generate accurate retrieval results. However, this task involves the following challenges: (1) the ambiguity of the natural language descriptions towards a target vehicle; (2) the matching between the linguistic semantics of the language descriptions and the corresponding static and dynamic properties of the target vehicle; (3) the shortage of the annotated language and target vehicle pairs. Obviously, focusing on solving a subset of the problems cannot generate a robust retrieval model. Therefore, we propose a multi-granularity retrieval system to solve this task, consisting of three main modules: (1) Language parsing module that aims to obtain the fine-grained vehicle attributes (e.g. color, type and motion) from the language descriptions; (2) Language-augmented multi-query vehicle track retrieval module that serves as our baseline model to incorporate information from multiple imperfect queries; (3) Target vehicle attributes enhancement module that explicitly fuses the static and dynamic properties of the target vehicle to generate the final retrieval results. Our system has achieved the **1**st place on the 6th AI City Challenge, yielding a strong performance on the private test set.*

1. Introduction

Vehicle track retrieval has been an important part of applying AI to improve the efficiency of operations in city environments. Previous works typically focus on building vehicle retrieval systems that are image-to-image matching based using image modality only, which aims to retrieve all instances of a particular vehicle identity from a gallery set of vehicle images that are captured under diverse traffic cameras. The AI City Challenge has made a step further and formulates a new task that incorporates a language modality, called Natural language-based vehicle track retrieval. Different from the image-to-image style vehicle retrieval, also known as vehicle re-identification, this task aims to retrieve single-camera tracks of vehicles that are consistent with a natural language query describing the static and dynamic properties of the target vehicles. This task is inherently challenging as it requires a retrieval model to first grasp the semantics of the language and vision modalities and then match them to generate accurate retrieval results. Concretely, the challenges can be summarized as follows: (1) at the language side, for a given target vehicle track, the corresponding language descriptions in a query set can sometimes be of low-quality, too general, ambiguous, or even conflict, as shown in Fig. 1(a), which adds noise to both training and evaluation of the retrieval model. (2) at the vi-

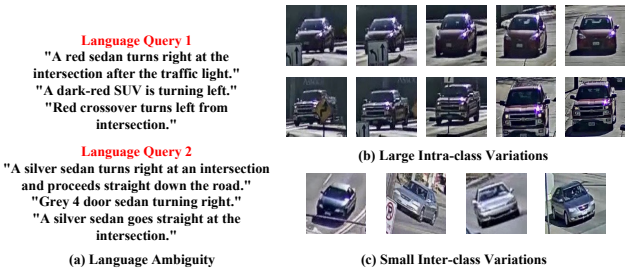

Figure 1. Overview of the challenges in the dataset.(a) Language ambiguity commonly existed in the language query set. (b) The intra-class variations of the vehicle images in a vehicle track. (c) The inter-class variations of different vehicle tracks.

*Equally-contributed authors. ‡Work done during internship at Baidu.
†Corresponding author.

sion side, vehicles with different identities show small inter-class variations. As presented in Fig. 1(b) and (c), they often share same static properties (e.g. color and type) or dynamic properties (e.g. motion patterns), which significantly increases the difficulties of the cross-modal matching. (3) the training set provided consists of only 2155 annotated language-vehicle pairs, which is insufficient for training a robust retrieval model. Obviously, a retrieval model that solves only a subset of the aforementioned problems cannot generate desired retrieval results.

Performing accurate target vehicle retrieval using natural language descriptions is a comprehensive task. In this paper, we present a multi-granularity retrieval system consisting of three main modules that tackle the aforementioned challenges from three perspectives. Concretely, (1) We introduce a **Language Parsing** module to obtain the fine-grained vehicle attributes information and formulate corresponding vehicle attributes labels to serve as extra supervisory signals for later cross-modal matching. The motivation lies in the fact that a language query sentence typically contains a detailed description of the target vehicle's color, type, and motion direction patterns. Unlike previous works [16], we add an extra motion direction parser to extract motion direction words. Besides, we further aggregate all parsed vehicle attribute words of a language query via word frequency voting to generate extra supervisory signals. (2) To tackle the language ambiguity problem commonly existed in the language query set, we present a **Language-augmented Multi-query Vehicle Retrieval** module that serves as our baseline model to incorporate information from multiple imperfect language descriptions in the query set. This is significantly different from current works which typically focus on experimenting under the single-query setting (that is retrieving target vehicle given a single text description as input). The motivation is that imperfect language descriptions could contain partial information of a target vehicle and aggregating multiple imperfect language descriptions of a target vehicle completes the language feature descriptor for the target vehicle, which is robust to the language ambiguity and benefits the cross-modal matching. Besides, following [1], we further augment the language query set with BaiduNLP library [17] to incorporate more imperfect language sentences. (3) Although the baseline model exhibits competitive retrieval performance, the small inter-class variations and large intra-class variations of the target vehicles pose a great challenge for the model to obtain robust retrieval results, which emphasize the necessity of a post processing step. Thus, we propose a **Target Vehicle Attributes Enhancement** module that further refines the retrieval results by explicitly fusing the static and dynamic properties of the target vehicles. Experiments results indicate that our enhancement module significantly boosts the final retrieval performance.

To sum up, this paper has the following contributions:

- We introduce a multi-granularity retrieval system to tackle the natural language-based vehicle retrieval task in the 6th AI City Challenge, which systematically solves the noisy cross-modal matching problem from different perspectives.
- We propose a language-augmented multi-query retrieval module to incorporate multiple imperfect language descriptions to form a complete and robust language embedding that alleviates the language ambiguity problem commonly existed in the challenge.
- We devise a target vehicle attributes enhancement module that refines the retrieval results by forcing the predicted static and dynamic properties of target vehicles to match the parsed language descriptions.
- Experiments show that our system achieves **1**st place on the private set of the challenge.

2. Related Work

2.1. Natural Language based Video Retrieval

Natural language based video retrieval task has attracted increasing attention in recent years. In order to establish a connection between text and video, early works [6, 10, 11, 15] mainly focus on extracting representative features from both video and text data. These works utilize a textual feature extractor such as ERNIE [21],Word2Vec [14], LSTM [7] to encode the language and employ a robust network such as VIT [5] to extract visual feature. Recently, large-scale pretrained vision-language models [12, 18, 27] have achieved impressive performance in video retrieval task. CLIP4Clip [13] explores a way to transfer the knowledge of the pretrained model to video-language retrieval in an end-to-end manner. CLIPBERT [9] employs sparse sampling to enable affordable end to end learning for video-language tasks. Wang et.al [25] further propose a new framework which uses multiple queries as inputs to generate more accurate results instead of simply combining similarity outputs of multiple queries from previous single-query trained models. However, different from the traditional video retrieval task, vehicle retrieval task is essentially an instance-level retrieval task that requires a model to have a better understanding of traffic scenes and vehicle attributes. Base on these characteristics, our method systematically solves the problem from multiple aspects and explicitly integrate the properties of the target vehicle with the language description.

2.2. Vehicle Re-identification

Vehicle Re-identification targets at retrieving the query from a big gallery. However, it is a challenging task due to many factors, such as occlusion and wide variety of appearance under different environments. Most of the research

efforts are dedicated to the design of network architecture as well as loss functions. For the network architecture design, PAMTRI [23] proposes pose-aware multi-task for vehicle re-Identification. And it embeds multi-task learning pipeline including keypoints, heatmaps and segments. Wang et.al. [24] proposed an orientation invariant feature embedding module and a spatial-temporal regularization module for vehicle re-identification framework. For the loss functions, different implementations of triplet loss [8] are provided under an extensive evaluation. Our task is different in that the aim of our task is to perform cross-modal matching between the language and vision modalities.

3. Method

3.1. Overview

We aim to present an overall solution for the natural language-based vehicle retrieval task. To carry out the task, we systematically analyze the challenges of the task and propose a multi-granularity retrieval system, as presented in Fig. 2. In general, our system can be decomposed into three main modules: the Language Parsing module, the Language-augmented Multi-query Vehicle Retrieval module, and the Target Vehicle Attributes Enhancement module. These modules tackle the challenges of the task from different granularities and altogether they achieve a remarkable performance on the benchmark dataset. In the following sections, we first introduce the details of the Language Parsing module; then, we present our Multi-query vehicle retrieval baseline model; finally, we illustrate the Target Vehicle Attributes Enhancement module.

3.2. Language Parsing Module

According to the task setting, language descriptions often contain rich descriptive information about the static and/or dynamic properties of the target vehicle. Specifically, it mainly includes the type of the target vehicle (pickup, truck, car), the color of the target vehicle (silver, red, black), and the motion direction of the target vehicle (go straight, turn left, turn right, etc.). We observed that most language queries have a similar language structure like *main subject + action + (optional other subject + action)*, which is organized around some core verbs. This motivates us to apply SRL to parse the queries. SRL [20] is a task which aims to get the semantic role of other parts in the sentence for each verb, including the Agent of the action, the Patient of the action, etc. In this task, we utilize a SRL tool to parse the language query to extract the vehicle attributes (type, color, motion) of the target vehicle.

Specifically, our language parsing module is mainly divided into 3 steps: data pre-processing, statistics analysis and extraction. First, in order to ensure the langauge queries can be parsed correctly and efficiently, we performed the following preprocessing operations: (1) Spell check. We first detect misspelled words in queries, and then use the Levenshtein distance metric to find the corresponding correct word with the shortest edit distance in our predefined dictionary and replace it. (2) Word Conversion. SRL takes verbs as the core to analyze different semantic roles in sentence. However, due to the ambiguity of natural language itself, some words can be used as both verbs and nouns, such as turn and drive, which confuse the SRL tool. Therefore, in order to parse the verbs in the sentence correctly, we replace all these verbs with their corresponding past tense.

After the data pre-processing step, we apply the SRL tool to parse the original language queries, and collect the main vehicle attributes words, including the type of the target vehicle, the color of the target vehicle and the motion direction of the target vehicle. Given the word frequency statistics of these attribute words, we keep the words with the highest frequency, and build up three vehicle attribute word dictionaries regarding vehicles color,type and motion direction. In general, the generated dictionaries contain 8 colors, 6 vehicle types and 4 motion states. Then, to obtain extra vehicle attributes supervisory signals, for a given language query set containing three language descriptions, we count the word frequency statistics of the vehicle attributes related words against predefined vehicle attributes dictionaries. A specific vehicle attribute label is extracted if the word frequency of the corresponding vehicle attribute related words is larger than one. Note that we generate two types of labels for each vehicle attribute, a multi-label format and a one-hot label format. The multi-label format is generated simply by thresholding word frequency that is larger than one while the one-hot label format is obtained by taking the maximum word frequency.

Finally, we get the vehicle color (L_{color}), type (L_{type}) and motion direction $L_{direction}$ labels for each language query set from the corresponding semantic role, such as ARG1-4 for color and type labels, ARG1,ARG2 and ARGM-DIR for motion labels, as presented in Fig. 3. In our experiment, we use these extra supervisory information during both cross-modal training and target vehicle attribute enhancement. The difference is that multi-label format is adopted during cross-modal training to combat the language ambiguity problem and the one-hot label format is used in the enhancement module to serve as a strict constraint for retrieval refinement.

3.3. Language Augmented Multi-query Vehicle Retrieval Module

Problem Formulation. Formally, given a vehicle track database $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ and a language query q_i of vehicle track v_i, the goal of natural language-based vehicle retrieval is to successfully retrieve v_i from $\mathcal{V}$ based on q_i. However, as mentioned in previous sections, given a

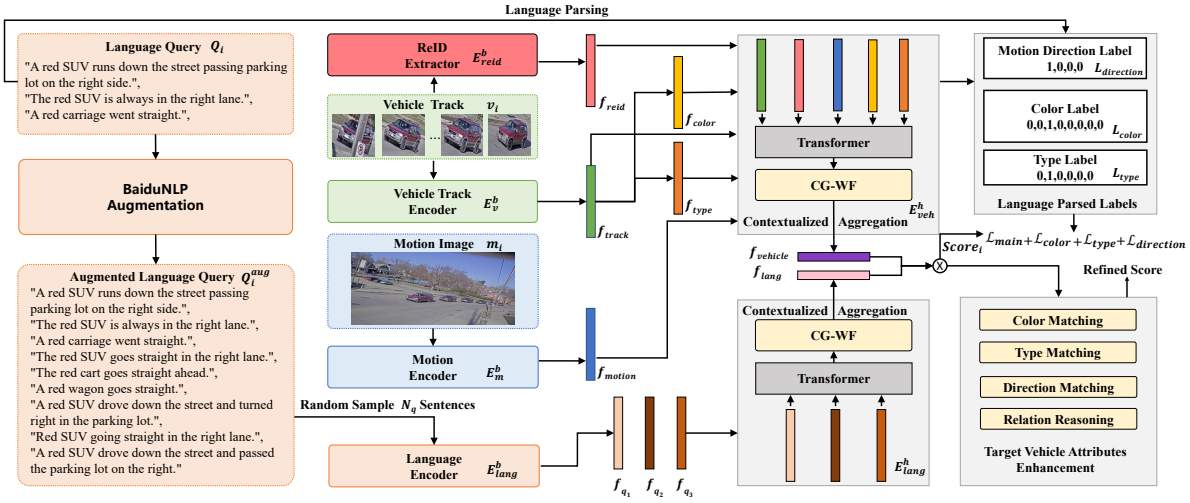

Figure 2. Overview of our method. First, Language Parsing module parses all the language sentences q_i of a language query set Q_i to obtain the fine-grained vehicle attributes information, which are then transformed to extra supervisory signals through words frequency voting. Then, BaiduNLP library is applied to augment the language query set Q_i^{aug} to include more imperfect language sentences. A multi-query vehicle track retrieval model $\mathcal{M}^{multi}(\cdot)$ is constructed with vehicle track images v_i, sampled N_q language sentences Q_i^{aug}, and a motion image m_i as inputs. The Motion $E^b_m(\cdot)$ and Vehicle Track $E^b_v(\cdot)$ encoder have the same network architecture (Spatial-Temporal Transformer Encoder [3]), producing corresponding vehicle track feature f_{track} and motion feature f_{motion}. ReID feature extractor $E^b_{reid}(\cdot)$ is also utilized to extract robust vehicle track features f_{reid}. Note that the ReID feature extractor is fixed during training. The vehicle Track feature f_{track} is further strengthened by the parsed color L_{color} and type L_{type} labels to obtain fine-grained vehicle color feature f_{color} and vehicle type feature f_{type}. Then, a Contextualized Aggregation module $E^h_{veh}(\cdot)$ is applied to re-weights all the vehicle related features to obtain the final vehicle embedding $f_{vehicle}$. Similarly, the sampled N_q language sentences in a language query set are forwarded to a language encoder $E^b_{lang}(\cdot)$, which are then fused by another Contextualized Aggregation module $E^h_{lang}(\cdot)$ to generate final language embedding f_{lang}. Then, the cross-modal matching similarity is calculated by a simple dot product between $f_{vehicle}$ and f_{lang}. After the training, the similarity score is further refined by the Target Vehicle Attributes Enhancement module.

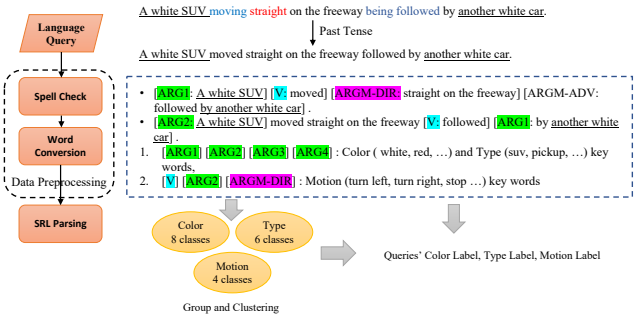

Figure 3. Overview of the Language Parsing Module. First we perform data preprocessing. Then, we take the SRL tool to parse the language queries and group the words to produce the vehicle attribute word dictionaries. Finally, we take the corresponding keywords to generate final vehicle attributes labels.

language query set containing multiple language query sentences, each language query sentence q_i is often imperfect and the query set suffers from the language ambiguity problem. Thus, instead of performing single-query vehicle retrieval, we adopt multi-query retrieval, where we define a language query set $Q_i = \{q_i^1, q_i^2, ..., q_i^k\}$ for a vehicle track v_i, which contains multiple imperfect language descriptions. Here we formulate the language-based vehicle retrieval as follows,

$$\forall j, j \neq i, \mathcal{M}^{multi}(Q_i, v_i) > \mathcal{M}^{multi}(Q_i, v_j) \quad (1)$$

where $\mathcal{M}^{multi}(\cdot)$ denotes the learned retrieval model and our goal is to retrieve a target vehicle track v_i from the vehicle track database $\mathcal{V}$ based on a language query set $Q_i = \{q_i^1, q_i^2, ..., q_i^k\}$. $\mathcal{M}^{multi}(\cdot) \in R$ reflects how the language query set matches the vehicle track. Ideally, as illustrated in equation 1, a perfect retrieval would score the matching language-vehicle pair higher than non-matching pairs. Then we denote the similarity score of language-vehicle pair as,

$$\mathcal{M}^{multi}(Q, v) = S(E_{lang}(Q), E_{veh}(v)) \quad (2)$$

where $E_{lang}(\cdot)$ denotes the language encoder network which maps the input query set to a high dimensional vector $E_{lang}(Q) \in R^m$. $E_{veh}(\cdot)$ denotes the vehicle track encoder network, which maps the input vehicle track to the same embedding space $E_{veh}(v) \in R^m$. $S(\cdot)$ is the scoring function which measures the similarity between $E_{lang}(Q)$ and $E_{veh}(v)$. Here we choose cosine similarity as $S(\cdot)$.

Overall Vehicle Track Encoder $E_{veh}(\cdot)$. It can be decomposed into three parts: a vehicle track encoder $E_v^b(\cdot)$, a vehicle motion encoder $E_m^b(\cdot)$, and a vehicle contextualized aggregation encoder $E_{veh}^h(\cdot)$. Following [3,25], we implement $E_v^b(\cdot)$ and $E_m^b(\cdot)$ as a spatial-temporal transformer encoder network that share the same network architecture. They are initialized with the same pre-trained transformer encoder weights and the only difference is that the inputs to $E_{veh}^h(\cdot)$ is a video-based vehicle track images and the inputs to $E_m^b(\cdot)$ is a single image-based vehicle motion image. Note that they both are leanred end-to-end. The motivation of this separate encoders design lies in the fact that the motion image focuses on the global context information and dynamic properties of a target vehicle while the vehicle track video represents local static and dynamic properties of a target vehicle. For a given vehicle track v_i and its corresponding motion image m_i generated by following [1], the motion encoder generates $f_{motion} = E_m^b(m_i)$ and the vehicle track encoder produces $f_{track} = E_v^b(v_i)$. To further constrain the vehicle track embedding f_{track}, we forward f_{track} through two different embedding layers (Fully-connected layers) to extract the color embedding f_{color} and the type embedding f_{type}, which are learned under the supervisory signal from L_{color} and L_{type} with the loss function denoted as $\mathcal{L}_{color}$ and $\mathcal{L}_{type}$. Similarly, we constrain f_{motion} with $L_{direction}$, the loss function of which is denoted as $\mathcal{L}_{direction}$. Furthermore, following previous works [1, 16], we add a Re-identification (ReID) feature extractor to further extract discriminative vehicle track embedding feature $f_{reid} = E_{reid}^b(v_i)$. In order to effectively combine multiple feature embeddings, we choose to adopt a contextualized aggregation network that is based on transformer attention network $E_{veh}^h(\cdot)$ to fuse the aforementioned vehicle related embeddings via $f_{vehicle} = E_{veh}^h([f_{track}, f_{reid}, f_{color}, f_{type}, f_{motion}])$.

Language Encoder $E_{lang}(\cdot)$. For a language query set Q_i, in training, we first sample a subset from the set consisting of N_q language sentences. For instance, we set $N_q = 3$. Then, we forward these language sentences to the language encoder $\{f_{q1}, f_{q2}, f_{q3}\} = E_{lang}^b(Q_i = \{q_i^1, q_i^2, q_i^3\})$. To combat the language ambiguity problem discussed previously, we adopt a contextualized aggregation network based on transformer that combines $\{f_{q1}, f_{q2}, f_{q3}\}$ to f_{lang} via $f_{lang} = E_{lang}^h([f_{q1}, f_{q2}, f_{q3}])$. Following [25], we choose the distilbert as the base structure of the $E_{lang}^b(\cdot)$. In inference, since the aggregation network $E_{lang}^h(\cdot)$ can dynamically fuses any number of language sentences in a query set, we thus set $N_q = 3$ to incorporate all language sentences in the test language query set.

Language Augmentation. In order to enhance the model robustness and take better use of the capacity of multi-query retrieval model $\mathcal{M}^{multi}(\cdot)$, we propose to further integrate the language augmentation strategy [2] to expand the training language query set from Q to Q^{aug}. Specifically, we collect all the language queries in the training dataset of Track2, then translate the language descriptions into Chinese and back-translate them afterwards.

Cross Modal Matching. The final cross modal matching between a language query set Q_i and a vehicle track v_i is performed by $score_i = S(f_{vehicle}, f_{lang})$ where $S(\cdot)$ is a cosine similarity function.

Total Loss functions. For a batch of N language-vehicle pairs, it consists of $N \times N$ possible sample pairs. The total loss function is defined as follows,

$$\mathcal{L} = \mathcal{L}_{main} + \mathcal{L}_{color} + \mathcal{L}_{type} + \mathcal{L}_{direction} \tag{3}$$

where $\mathcal{L}_{main}$ is the symmetric InfoNCE loss that is defined as,

$$\mathcal{L}_{t2i} = \frac{1}{N}\sum_{i=1}^{N} -log\frac{exp(S(f_{vehicle}^i, f_{lang}^i)/\tau)}{\sum_{j=1}^{N} exp(S(f_{vehicle}^j, f_{lang}^j)/\tau)} \tag{4}$$

$$\mathcal{L}_{i2t} = \frac{1}{N}\sum_{i=1}^{N} -log\frac{exp(S(f_{lang}^i, f_{vehicle}^i)/\tau)}{\sum_{j=1}^{N} exp(S(f_{lang}^j, f_{vehicle}^j)/\tau)} \tag{5}$$

$$\mathcal{L}_{main} = \mathcal{L}_{t2i} + \mathcal{L}_{i2t} \tag{6}$$

and $\mathcal{L}_{color}$, $\mathcal{L}_{type}$, and $\mathcal{L}_{direction}$ are standard multi-label classification loss.

Inference. At inference stage, we first adopt the average model soup strategy [26] that averages several models trained with different hyperparameters. Then, the averaged model are utilized to generate the retrieval results for the test set denoted as M_{model} for later enhancement.

3.4. Target Vehicle Attributes Enhancement Module

Since we do not explicitly align the static and dynamic vehicle properties of the target vehicle between the two modalities during the cross-modal training, we propose a post-processing enhancement module to explicitly align these properties of the target vehicle between the two modalities. The implementation of the enhancement strategies presented in this section is essentially carried out by re-weighting the retrieval results generated by our baseline model introduced in previous section.

Target Vehicle Attribute Predictors. To explicitly learn classifiers that individually predict different vehicle attributes, we formulate three separate classification networks focusing on vehicle color $C_{color}(\cdot)$, vehicle type $C_{type}(\cdot)$, and vehicle motion direction $C_{direction}(\cdot)$ respectively. To train the color $C_{color}(\cdot)$ and the type $C_{type}(\cdot)$ classifiers, we first crop the vehicle track images from the vehicle track dataset $\mathcal{V}$ and generate corresponding one-hot format label

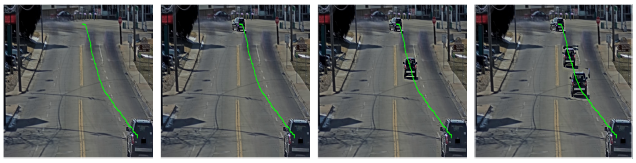

Figure 4. The training examples of motion direction classifier $C_{direction}(\cdot)$. We draw the trajectory of the target vehicle on the corresponding background image. In order to make the movement state more clear, we uniformly sample 1 to 4 cropped vehicle track images and paste them on the trajectory with each image serving as a training example for our motion classifier.

for vehicle track color and type attributes as mentioned in the Language Parsing module. With these training set, the learned $C_{color}(\cdot)$ and $C_{type}(\cdot)$ could individually predict vehicle color and type attributes given a vehicle track. For the motion direction classifier$C_{direction}(\cdot)$, we draw the vehicle motion trajectory on the motion image m_i and use the generated motion label L_{motion} to train the classifier. An example training image is presented in Fig. 4.

For these three classifiers, we adopt EfficientNetB3 [22] network as the backbone network accompanied by a corresponding classification head. We adopt cross-entropy loss function as the training loss function.

Retrieval Enhancement With Re-weighting. With the learned color, type, and motion direction predictors, we design a simple yet effective re-weighting strategy to fuse them with the retrieval matrix generated by our baseline model. Specifically, for each language query Q_i, we check all the candidate vehicle tracks in $\mathcal{V}$. If the predicted vehicle attributes of this candidate vehicle track is consistent with the vehicle attributes label parsed from the query, we increase this pair's similarity score, otherwise, we decrease the similarity score. This strategy can be formulated as,

$$M_{ij}^{attri} = \begin{cases} s_{attri}, & L_i^{attri} == P_j^{attri} \\ -s_{attri}, & otherwise \end{cases} \tag{7}$$

where $attri$ stands for the attributes of vehicles including $color$,$type$ and $direction$. M is a $N \times N$ score matrix which measures the consistency of target vehicle attributes and language description. L_i is the property label obtained by parsing the language queries Q_i, P_j is the prediction made by classification models of track v_j. s_{attri} is the penalty coefficient, s_{color}, s_{type}, $s_{direction}$ can be different according to the importance of different properties. The final target vehicle score matrix M_{tgt} is:

$$M_{tgt} = M_{tgt}^{color} + M_{tgt}^{type} + M_{tgt}^{direction} \tag{8}$$

According to our experiments, this strategy greatly boost our retrieval performance.

(a) (b) (c)

Figure 5. Overview of the relation module. (a) detection results (b) trajectory mask filtering. yellow area stands for trajectory mask (c) related vehicle. red box represents front vehicle, green box represents target vehicle, blue box represents back vehicle..

3.5. Related Vehicle Attribute Enhancement Module

In addition to the target vehicle attributes, some sentences also provide us informative clues about properties of related vehicles such as 'following red sedan', 'behind another white vehicle'. To exploit these valuable information, we propose a Related Vehicle Attribute Enhancement Module which greatly boost the final performance of the retrieval task. This module extracts information about related vehicles both from natural languages and video frames.

First of all, in order to obtain the potential related vehicle locations, we use object detection framework Cascade-RCNN [4] which is trained by AICITY22-track1 training data to detect all vehicles in each frame, the result is shown in Fig. 5(a). Since we only concern about the related cars which are the vehicles in front of or behind the target vehicle in this task, we filter out the irrelevant vehicles, for example, vehicles in opposite lanes. We assume that the related vehicle and the target vehicle have similar trajectories and denote the trajectory mask of target vehicle as T, the n-th detected bounding box area of vehicle as b_n. For each track, T is obtained by merging the bounding box area of target vehicle in each frame. After that, we compute the intersection over union(IOU) between each (b_n, T) pair and filter out the bounding boxes whose IOU(b_n, T) is lower than the threshold δ, Fig. 5(b) shows the filtering results. After filtering, we calculate the L2 distance between target vehicle and related vehicles and seek out vehicles in front of and behind the target vehicle based on such distance. Specifically, for a frame t, the distances are calculated by following equation:

$$D_n^t = \sqrt{(x_n^t - x_{tgt}^t)^2 + (y_n^t - y_{tgt}^t)^2} \tag{9}$$

where D_n^t is the L2 distance between the i-th detected bounding box of frame t b_n^t and the target bounding box b_{tgt}^t provided by track information, x_n^t, y_n^t are the center coordinates of b_n^t. Similarly, x_{tgt}^t and y_{tgt}^t are the center coordinates of b_{tgt}^t.

Obviously, smallest D_n^t appears when box b_n^t and b_{tgt}^t are the same vehicle and this kind of box is represented in green in the Fig. 5(c). After finding the detected target box, we divide the remaining boxes into two categories: boxes

in front of target B_{front} and boxes in back of target B_{back}. Since the target vehicle is moving forward, front vehicles will be closer to the target vehicle in the next frame, on the contrary, if the vehicle is behind the target, the distance D_n^t will getting larger. This can be formulated as follows:

$$\begin{cases} b_n^t \in B_{front}, & D_n^t > D_n^{t+1} \\ b_n^t \in B_{back}, & otherwise \end{cases} \tag{10}$$

Then we can get the front vehicle and back vehicle bounding boxes by finding the minimum D_n^t in B_{front} and B_{back} respectively, these two kind of boxes are also drawn in red and blue in Fig. 5(c).

After obtaining the language relation information extracted by language parsing module and track relation information generated by above procedure, we employ the re-weighting strategy mentioned in section 3.4 to generate score matrices M_{front} and M_{back} for front vehicle and back vehicle respectively, thus the final similarity matrix M_{final} becomes:

$$M_{front} = M_{front}^{color} + M_{front}^{type} \tag{11}$$

$$M_{back} = M_{back}^{color} + M_{back}^{type} \tag{12}$$

$$M_{final} = M_{model} + M_{tgt} + M_{front} + M_{back} \tag{13}$$

where M_{model} is the similarity matrix calculated by $\mathcal{M}^{multi}(\cdot)$.

4. Experiments

4.1. Dataset

Following previous works [3, 25], we use three datasets to train our retrieval system, namely CityFlowV2-NL, AICITY22-track1 and the synthetic data. Specifically, we adopt CityFlowV2-NL to train our the multi-query retrieval model $\mathcal{M}^{multi}(\cdot)$. For training the ReID model $E_{reid}^b(\cdot)$ to extract appearance features from the vehicle track, we utilize the training set of AICITY22-track1 and the synthetic data.

4.2. Evaluation Metric

Following the setting of Track2, we adopt the mean reciprocal rank (MRR) as the main evaluation metric. MRR is denoted as:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}, \tag{14}$$

where $|Q|$ is the number of language query and $rank_i$ denotes the ranking of the ground truth track for the i-th language query. We also report MRR results of Recall @5 and Recall @10.

Table 1. Comparisons between our method and other teams on the benchmark of Track2. Our results rank **1**st.

Team ID	MRR
176	**56.52%**
6	52.51%
4	47.73%
183	43.92%
91	36.11%
44	33.38%

4.3. Implementation Details

For the Multi-Query Vehicle Track Retrieval model $\mathcal{M}^{multi}(\cdot)$, following [25], we adopt the spatial-temporal transformer model proposed in [3] as the initialized weights for $E_v^b(\cdot)$ and $E_m^b(\cdot)$, which is pre-trained on the Conceptual Captions [19] and WebVid-2M [3] datasets. The language encoder $E_{lang}^b(\cdot)$ is initialized from the DistilBERT base-uncased model pre-trained on the English Wikipedia and Toronto Book Corpus. All other parts of the model are randomly initialized. $\mathcal{M}^{multi}(\cdot)$ is trained end-to-end with a batch size set to 20 and is optimized with AdamW optimizer with learning rate set to 0.00003. We train the model for 20 epochs in total with a learning rate scheduler set to Linear Warmup Cosine Annealing Learning Rate Scheduler. For $E_v^b(\cdot)$ and $E_m^b(\cdot)$, the input number frames are set to 16 and 1 accordingly and we resize the vehicle track and motion image inputs to 224×224. The multi-query number N_q is set to 4 in training and set to 3 in testing.

4.4. Comparisons with Other Teams

As shown in Tab. 1, our overall model result won the first place with the MRR of 56.52%, surpassing the second-best team by a large margin (4.01%).

4.5. Ablation Study

According to Tab. 2, our multi-query retrieval model $\mathcal{M}^{multi}(\cdot)$ achieves a strong performance of 37.05% MRR, which verifies the effectiveness of the adoption of multi-query to tackle the language ambiguity problem commonly existed in the benchmark dataset. Then with language augmentation from BaiduNLP library, we further boost the performance by 2.12% MRR, achieving 39.17%MRR. This demonstrates the language back-translation is effective. Then, we perform average-based model soup, which boost the MRR to 40.73%. Furthermore, we performed the Target Vehicle Attribute Enhancement post-processing strategy, which significantly improves the MRR by 15.79%. This suggests that although the model could learn a competitive cross-modal matching baseline with limited and noisy annotated language-vehicle pairs, yet applying explicit constraints to the alignment between the two modalities is still

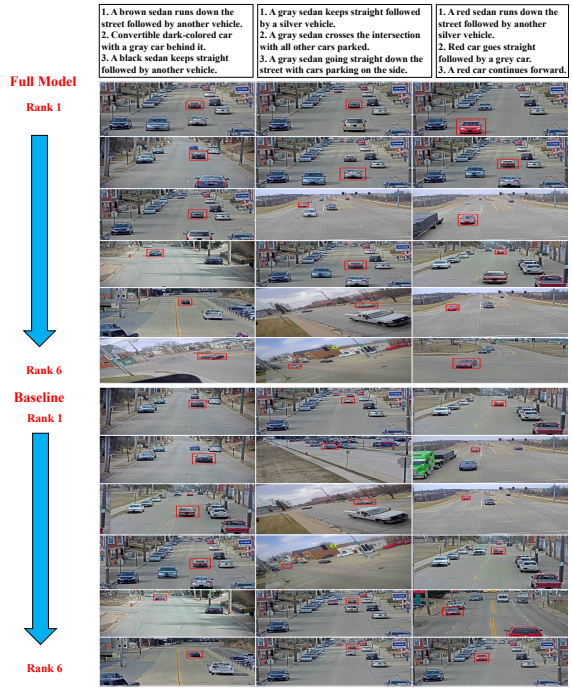

Figure 6. Qualitative Comparisons between our full model and our baseline model.

Table 2. Ablation study. Baseline is the $\mathcal{M}^{multi}(\cdot)$ model trained without language augmentation. NL Aug. represents the language augmentation through language back-translation. TVAE denotes the Target Vehicle Attribute Enhancement module. PL. denotes the pseudo label exploration.

Baseline	NL Aug.	Model Soup	TVAE	PL.	MRR	Recall@5	Recall@10
✓					37.05%	57.61%	76.63%
✓	✓				39.17%	63.58%	77.17%
✓	✓	✓			40.73%	64.67%	77.17%
✓	✓	✓	✓		**56.52%**	**71.20%**	**83.15%**
✓	✓	✓	✓	✓	66.06%	82.61%	90.22%

important and has a great potential to further boost the performance, which has not been fully explored in previous works.

4.6. Qualitative Results

We visualize the ranking results of our full model and our baseline model in Fig. 6, which clearly shows the effectiveness of our proposed modules. All the top-6 ranking results are relevant to the language query descriptions.

4.7. Upper Bound

The test images are assumed to be unknown by default. However, camera deployments in a traffic transportation system are usually accessible, meaning that the exits and entrances of each movement captured by a camera could be pre-defined. Besides, the trajectories of vehicle tracks under various cameras are provided, meaning that the direction of

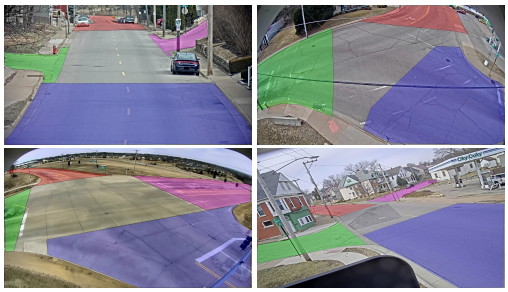

Figure 7. Upper bound performance of our system by re-training $\mathcal{C}_{direction}(\cdot)$ with pseudo labeled trajectories of vehicle tracks. We mark all the intersections in the cameras and cluster all the trajectories with simple traffic rules to assign pseudo labels to the trajectories.

these vehicle trajectories can be pseudo labeled via clustering and a set of traffic rules (priors). This motivates us to further explore the performance upper bound of our system by re-training our motion direction classifier $\mathcal{C}_{direction}(\cdot)$ in a semi-supervised manner. Concretely, for each test camera, we draw several specific intersection regions as presented in Fig. 7, and we use simple traffic rules to determine the pseudo label of each vehicle trajectory. Then, we train $\mathcal{C}_{direction}(\cdot)$ with these pseudo labeled vehicle trajectories. The network is a simple Bi-LSTM network. Note that the inputs of $\mathcal{C}_{direction}(\cdot)$ are the points embedded by a coordinate embedding layer along the vehicle motion trajectory, without any image information. Interestingly, this simple design further boosts the final performance to a new level, achieving 66.06% shown in Tab. 2.

5. Conclusion

We presented a multi-granularity retrieval system for the natural language-based vehicle retrieval task in the 6th AI City Challenge. We analyzed the dataset provided by the challenge and summarized three main problems: the language ambiguity in the language query set, the challenging intra- and inter- class variations in target vehicle tracks, and the shortage of the annotated language-vehicle pairs. To tackle these problems, we first presented a Language Parsing module to extract fine-grained vehicle attributes information from the language query and formulate extra supervisory signals for later cross-modal training. Then, we introduced a multi-query vehicle track retrieval strong baseline model that incorporates multiple imperfect language descriptions to match the described target vehicle track, which helps obtain complete and robust language embedding that benefits the overall cross-modal matching. Finally, we proposed a target vehicle attributes enhancement module to further refine the retrieval results through a set of prescribed constraints that force the predicted properties of a target vehicle to match the corresponding language descriptions. Experiments on the private set of the challenge show that our solution achieved the **1**st place.

References

[1] Shuai Bai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Hongxia Yang, and Yi Yang. Connecting language and vision for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4034–4043, June 2021. 2, 5

[2] Shuai Bai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Hongxia Yang, and Yi Yang. Connecting language and vision for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4034–4043, 2021. 5

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 4, 5, 7

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 6

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[6] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. 2

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[8] Ratnesh Kuma, Edwin Weill, Farzin Aghdasi, and Parthasarathy Sriram. Vehicle re-identification: an efficient baseline using triplet embedding. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2019. 3

[9] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 2

[10] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2657–2664, 2014. 2

[11] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 2

[12] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[13] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 2

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[15] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 19–27, 2018. 2

[16] Tien-Phat Nguyen, Ba-Thinh Tran-Le, Xuan-Dang Thai, Tam V. Nguyen, Minh N. Do, and Minh-Triet Tran. Traffic video event retrieval via text query using vehicle appearance and motion attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4165–4172, June 2021. 2, 5

[17] Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*, 2020. 2

[18] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020. 2

[19] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 7

[20] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019. 3

[21] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021. 2

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[23] Zheng Tang, Milind Naphade, Stan Birchfield, Jonathan Tremblay, William Hodge, Ratnesh Kumar, Shuo Wang, and Xiaodong Yang. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 211–220, 2019. 3

[24] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 379–387, 2017. 3

[25] Zeyu Wang, Yu Wu, Karthik Narasimhan, and Olga Russakovsky. Multi-query video retrieval. *arXiv preprint arXiv:2201.03639*, 2022. 2, 5, 7

[26] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. 2022. 5

[27] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020. 2