Visual Question Answering with Graph Matching and Reasoning

Anonymous ICCV submission

Paper ID ****

Abstract

*Visual Question Answering (*VQA*) is of great significance in offering people convenience: one can raise a question for details of objects, or high-level understanding about the scene, over an image. This paper proposes a novel method to address the* VQA *problem. In contrast to prior works, our method that targets single scene* VQA*, replies on graph-based techniques and involves reasoning. In a nutshell, our approach is centered on three graphs. The first graph, referred to as inference graph G_I, is constructed via learning over labeled data. The other two graphs, referred to as query graph Q and entity-attribute graph G_{EA}, are generated from natural language query Q_{nl} and image Img, that are issued from users, respectively. As G_{EA} often does not take sufficient information to answer Q, we develop techniques to infer missing information of G_{EA} with G_I. Based on G_{EA} and Q, we provide techniques to find matches of Q in G_{EA}, as the answer of Q_{nl} in Img. Unlike commonly used* VQA *methods that are based on end-to-end neural networks, our graph-based method shows well-designed reasoning capability, and thus is highly interpretable. We also create a dataset on soccer match (Soccer-VQA) with rich annotations. The experimental results show that our approach outperforms the state-of-the-art method and has high potential for future investigation.*

1. Introduction

In recent years, visual query answering (VQA) has received significant attention [11, 15, 7] as it involves multidisciplinary research, *e.g*. natural language understanding, visual information retrieving and multi-modal reasoning. The task of VQA is to find an answer to a query Q_{nl} based on the content of an image. There are a variety of applications of VQA, *e.g*. surveillance video understanding, visual commentator robot, *etc*. Solving VQA problems usually requires high level reasoning from the content of an image.

Example 1: ADD AN EXAMPLE! □

This example suggests that we leverage graph-based method to resolve the VQA problem. While to do this, several questions have to be settled. (1) How to represent image and query with graphs? (2) How to infer crucial information when G_{EA} constructed from image is insufficient? (3) How to find answers from graphs with G_{EA}?

Contributions. In contrast to a majority of deep learning based VQA techniques, which lacks of necessary reasoning and thus performs poorly, our approach divides VQA tasks into three parts, and incorporates reinforcement learning and reasoning for each subtask. The main contributions of the paper are as follow.

(1) We propose new approaches for visual tasks based on reinforcement learning. Given an image and a question, our approach only identifies those objects that are related to users' questions rather than the complete set of objects along with their attributes. This substantially improves performance of object detection

(2) We propose approaches to answering visual questions with graph-based techniques. More specifically, we first construct an entity-attribute graph from a given image; we then train a classifier to infer missing information that are crucial for answering queries; we finally provide methods to answer queries with graph pattern matching.

3) We conduct extensive experimental studies to verify the performance of our method. We find that X, Y, and Z.

2. Related Work

We categorize related work into following three parts.

Visual query answering. Current VQA approaches are mainly based on deep neural works. [27] introduces a spatial attention mechanism similar to the model for image captioning. Instead of computing the attention vector iteratively, [23] obtains a global spatial attention weights vector which is then used to generate a new image embedding. [26] proposed to model the visual attention as a multivariate distribution over a grid-structured conditional random field on image regions, thus multiple regions can be selected at the same time. This attention mechanism is called structured multivariate attention in [26]. There has been many other improvements to the standard deep learning method, *e.g*. [6] utilized Multimodal Compact Bilinear (MCB) pooling to efficiently and expressively combine multimodal features. Another interesting idea is the implementation of Neural Module Networks [1, 8], which decomposes queries into their linguistic substructures, and uses these structures to dynamically instantiate module networks. [17] proposed to build graph over scene objects and question words. The visual graph is similar to ours, but the query graph differs. Note that the method [17] proposed is still a neural network based method as the structured representations are fed into a recurrent network to form the final embedding and the answer is again inferred by a classifier.

Query Oriented Visual Tasks. Peixi, please add Reinforcement learning based object detection here. Explores the environment to acquire supervision 1. Reinforcement driven information acquisition in non-deterministic environments 2. RL in negvitation 3. RL in other visual+language field

Reinforcement learning in vqa 2. Learning to Reason: End-to-End Module Networks for Visual Question Answering Reinforcement learning inference time and accuracy trade-off. Feature based on low, middle and high level makes it better preserves information in the process, and less probability falling into local minimum.

Graph-based query answering. Query answering has been extensively studied for graph data. In a nutshell, this work includes two aspects: query understanding, and query evaluation. We next review previous work on two aspects.

(1) Queries expressed with natural languages are very user-friendly, but nontrivial to understand. Typically, they need to be structured before issuing over *e.g*. search engine, knowledge graph, since structured queries are more expressive. There exist a host of works that based on query logs, human interaction and neural network, respectively. [14] leverages query logs to train a classifier, based on which structured queries are generated. [25] propose an approach to generate the structured queries through talking between the data (*i.e*. the knowledge graph) and the user. [24] introduced how to generate a core inferential chain from a query with convolutional neural networks. As we only cope with a set of fixed queries, hence, we defer the topic of query understanding to another paper, and focus primarily on the query evaluation.

(2) To evaluate queries on graphs, a typical method is graph pattern matching. There has been a host of work on graph pattern matching, *e.g*. techniques for finding exact matches [3, 19], inexact matches [28, 18], and evaluating SPARQL queries on RDF data [20]. Our work differs from the prior work in the following: (1) we integrate arithmetical and set operations in the query graph, and (2) we develop technique to infer missing values for query answering.

3. Overview of the Approach

We start from representations of images and questions, followed by the overview of our approach.

3.1. Representation of Images and Questions

We use the same representations as [22]. To make the paper self-contained, we cite them as follows (rephrased).

Entity-Attribute Graphs. Entities are typically defined as objects or concepts that exist in the real world. An entity often carries attributes, that describe features of the entity.

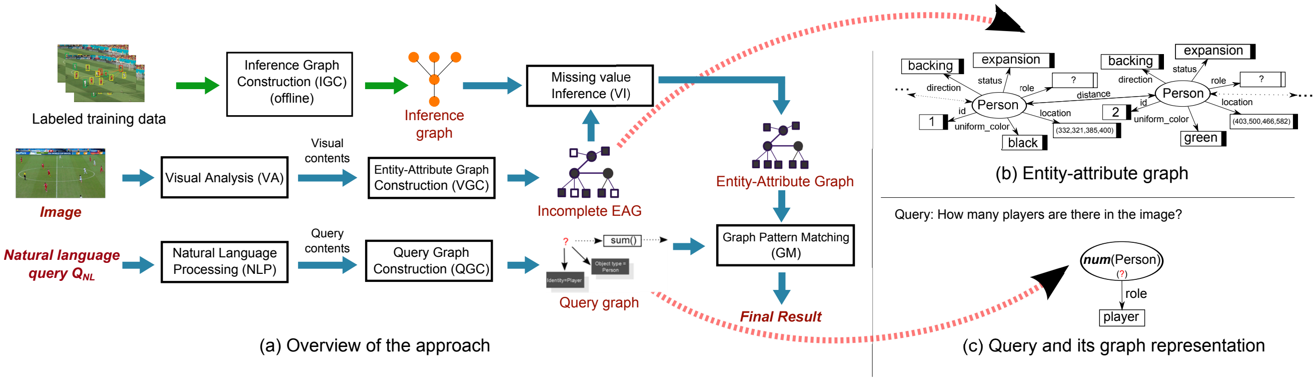
Assume a set $\mathcal{E}$ of entities, a set $\mathcal{D}$ of values, a set $\mathcal{P}$ of predicates indicating attributes of entities and a set Θ of types. Each entity e in $\mathcal{E}$ has a *unique ID* and a *type* in Θ.

An *entity-attribute* graph, denoted as EAG, is a set of triples $t = (s, p, o)$, where *subject* s is an entity in $\mathcal{E}$, p is a *predicate* in $\mathcal{P}$, and *object* o is either an entity in $\mathcal{E}$ or a value d in $\mathcal{D}$. It can be represented as a directed edge-labeled graph $G_{EA} = (V, E)$, such that (a) V is the set of nodes consisting of s and o for each triple $t = (s, p, o)$; and (b) there is an edge in E from s to o labeled by p for each triple $t = (s, p, o)$.

An image can be represented as an EAG with detected objects and obvious attributes. This can be achieved via a few visual tasks. While EAG generated directly after image processing is often incomplete, *i.e*. it may miss some crucial information to answer queries. We hence refer to *entity-attribute graphs* with incomplete information as *incomplete entity-attribute graphs*, and associate nodes with white rectangles, to indicate the missing value of an entity or attribute in EAG. Figure **??**(b) is an *incomplete entity-attribute graph*, in which square nodes representing person roles are associated with white rectangle.

Query Graphs. A query graph $Q(u_o)$ is a set of triples (s_Q, p_Q, o_Q), where s_Q is either a variable z or a function $f(z)$ taking z as parameter, o_Q is one of a value d or z or $f(z)$, and p_Q is a predicate in $\mathcal{P}$. Here function $f(z)$ is defined by users, and variable z has one of three forms: (a) *entity variable* y, to map to an entity, (b) *value variable* $y*$, to map to a value, and (c) *wildcard* $\_y$, to map to an entity. Here s_Q can be either y or $\_y$, while o_Q can be y, $y*$ or $\_y$. Entity variables and wildcard carry a *type*, denoting the type of entities they represent.

A query graph can also be represented as a graph such that two variables are represented as the same node if they have the same name of y, $y*$ or $\_y$; similarly for functions $f(z)$ and values d. We assume *w.l.o.g*. that $Q(x)$ is connected, *i.e*. there exists an undirected path between u_o and each node in $Q(u_o)$. In particular, u_o is a designated node in $Q(u_o)$, denoting the query focus and labeled by "?". Take Fig. 1(c) as example. It depicts a query graph that is generated from query "*How many players are there in the im-*



(a) Overview of the approach

(b) Entity-attribute graph

(c) Query and its graph representation

Figure 1: Overview of our approach, Entity Attribute Graph and Queries

age?". Note that the "query focus" u_o carries a function num () that calculates the total number of *person* entities with *role* "player".

3.2. Approach Overview

Along the same lines as representations for images and questions, and graph pattern matching for question answering, raised in [22], we propose a comprehensive approach as modeling of the VQA problem.

Figure 1(a) presents the overview of our approach. As can be seen, our approach revolves around three graphs: entity-attribute graph, query graph and inference graph. The generation of entity-attribute graph G_{EA} follows three steps. Module VA conducts the first step, *i.e.* image processing, and outputs all the detected objects along with their attributes. Using visual contents produced in step one, module VGA constructs an *incomplete* EAG. In the last step, module VI takes inference graph and *incomplete* EAG as inputs, infer missing information with G_I, and outputs an updated EAG for query answering. The inference graph G_I is used to infer missing values of an *incomplete* EAG. and constructed by module IGC over training data. As is query-independent, G_I is constructed offline, which warrants the efficiency of our approach. As the other part of input, natural language query Q_{nl} needs to be structured for query evaluation. To this end, Q_{NL} is first parsed via our NLP module, and then structured by module QGC. After $Q(u_o)$ and G_{EA} are generated, our approach employs module GM for matching computation, and returns final result.

As some modules employ existing techniques, to emphasize our novelty, we will elaborate modules VA and VGA in Section **??**, modules IGC and VI in Section **??**, and module GM in Section **??** with more details.

4. Query Oriented Visual Tasks

In this section, we introduce how we do visual tasks that are in connection with questions.

4.1. Visual Processing

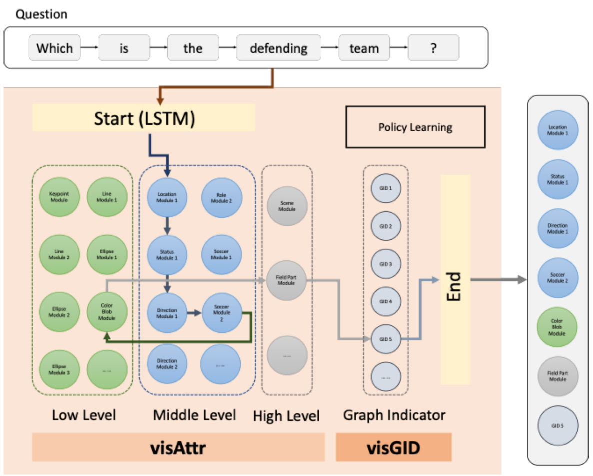


Figure 2: Overview of our approach

In our approach, we build a structure which selects subtasks to form a policy which is based on queries. For instance, with the input *which is the defending team?*, the system first predicts the corresponding steps are *Human Module*, *Gesture Module*, *Direction Module*, *Soccer Module*, *Color Blob Module*, *Field Part Module* and *Graph Indicator*. Guided by such action sequence, the image features are extracted by operating relevant vision tasks. An overview is shown in Figure 2.

4.2. Visual Processing

Multi-layer LSTM with Attention

The task here is to predict the most suitable action module sequence by given questions and performance

We would like to predict the most suitable reasoning structure tailored to each question. For an input question q such as What object is next to the t Figure 4

visAttr	Level	Sub-task	Descriptions
	Low	Keypoint Module	To get detailed information of the soccer field. $F_{keypoint}$
		Line Module	
		Ellipse Module	
		Color Blob Module	To detect blobs of different color among a region(whole soccer field or a small bounding box).
	Middle	Location Module	To detect the location of the person. $P_{location}$
		Status Module	To get the person's gesture P_{status} (standing, moving, expansion).
		Direction Module	To get whether a person is facing the goal or not. $P_{direction}$
		Uniform Module	To get the uniform color of the person $P_{uniform}$
		Soccer Module	To detect the location of the soccer. $S_{location}$
	High	Field Part Module	To detect which part of the soccer field is there in the image F_{part} .
visGID		Graph ID Indicator	To indicate which type of graph will be used in the following process.

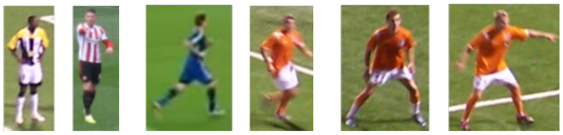
Table 1: Visual Task Pool

Visual Task Pool

First, the question features $w_{q,d}$ are extracted from questions by a Long short-term memory (LSTM) and comes into the step of visual task selection (VTS). VTS is guided by the question feature, selecting target visual tasks from the task pool by Monte Carlo learning. The task pool is demonstrated in Table 1.

The pool is constructed by two parts, the first one is *visAttr* which aims to discover the attribute of people, soccer, field and scene [22], while the other one *visGId* is an indicator showing the current question belongs to which graph type. There are three levels of *visAttr*: low, middle and high, which represents different difficulty degree of the vision tasks.

For each vision task, the approach is not fixed, one task can be achieved by different methods with variance in time and accuracy. For instance, object detection based methods, like Faster R-CNN [16], R-FCN [4], SSD [10] or skeleton keypoints detection based method like [2] and [21] can be implemented as a set of status module methods, because all of them is able to localize and distinguish people who are moving, standing or with an expansion gesture (Figure 3).



(a) Standing (b) Moving (c) Expansion

Figure 3: Person status.

Time and Accuracy Term

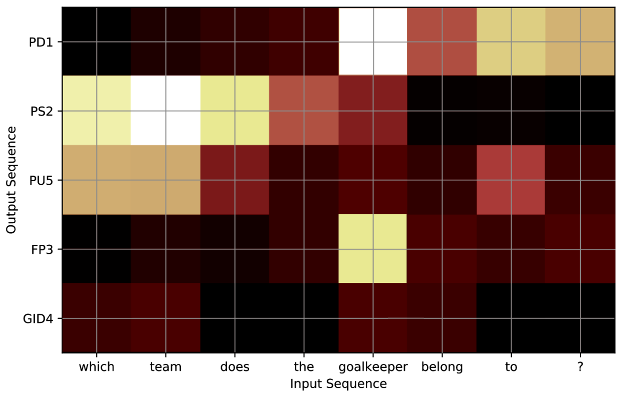


Figure 4: Attention Mechanism in Query

VQA is a topic in real-time domain, which leads that the answering time cannot be directly ignored, meanwhile, the accuracy still plays a significant role. To better balance these two, we proposed time and accuracy terms in loss function of training process.

Monte Carlo Methods

4.3. Construction of EAG

After objects that are related to questions are identified, we construct a graph structure, denoted as EAG, along the same line as [22].

Example 2: ADD AN EXAMPLE TO ILLUSTRATE PROGRESS IF NECESSARY! □

5. Reasoning

An *incomplete* EAG is often not able to provide query answers due to missing values of some hidden attributes. This motivates us to develop methods to infer values of hidden attributes. Below, we present modules IGC and IM, which are responsible for inference graph construction and missing value inference, respectively.

In our model, the inference graph is constructed using the Bayesian network. Essentially, Bayesian network is a kind of directed acyclic graph model, of which the parameters can be explicitly represented by the nodes (*i.e.*, random variables). Additionally, the parameters can be endowed with distributions (*i.e.*, priors). Using Bayesian network as inference graph leads to the resulting structure being very concise.

5.1. Inference Graph

As mentioned above, the inference graph is constructed using Bayesian network. A typical Bayesian network consists of decision and utility nodes [12]. We follow the descriptive notations used in [9] to facilitate our problem. Defined by $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ the set of N instances, each instance $\mathbf{x}^{(i)} = [x_1^{(i)}, \cdots, x_n^{(i)}]$ is the observation over n random variables: $x_1 \sim X_1, \cdots, x_n \sim X_n$. Under this assumption, a Bayesian network can be formally described by $\mathfrak{B} =< \mathcal{G}, \Theta_{\mathcal{G}} >$, where $\mathcal{G}$ is a directed acyclic graph and $\Theta_{\mathcal{G}}$ the set of parameters that can maximize the likelihood [5, 13]. The i-th node in $\mathcal{G}$ corresponds to a random variable X_i, and an edge between two connected nodes indicates the direct dependency. The symbol of $\Theta_{\mathcal{G}}$ is a parametric set that uses to quantify the dependencies within $\mathcal{G}$. Specifically, the parameters set of the i-th node associated with an observation x_i in $\Theta_{\mathcal{G}}$ can be denoted by $\theta_{x_i}|\Pi_i(\mathbf{x})$, where $\Pi_i(\mathbf{x})$ is a function which takes $\mathbf{x}$ as input, and outputs the values of attributes whose child is i. Note here that x_i is a possible value of X_i. For notational simplicity, the notation of $\theta_{x_i}|\Pi_i(\mathbf{x})$ is fully equal to $\theta_{X_i=x_i}|\Pi_i(\mathbf{x})$.

With the notations above, the unique joint probability distribution of a Bayesian network (*i.e.*, the inference graph G_i) is given by

$$P_{\mathfrak{B}}(\mathbf{x}) = \prod_{i=1}^{n} \theta_{x_i|\Pi_i(\mathbf{x})} \tag{1}$$

In our first problem, the purpose of Bayesian network is to infer the corresponding role that can be further regarded as an additional variable, *e.g.* Y (similar handling for the second one). The notation of Y is also a random variable associated with our target value with the values $y \in \mathcal{Y}$. In order to take Y into consideration, we rearrange the data $\mathcal{D}$ into another form: $\mathcal{D} = \{(y^i, \mathbf{x}^{(i)})\}_{i=1}^{N}$. Accordingly, Eq. (1) is reformulated to the following form

$$P_{\mathfrak{B}}(y|\mathbf{x}) = \frac{P_{\mathfrak{B}}(y, \mathbf{x})}{P_{\mathfrak{B}}(\mathbf{x})} = \frac{\theta_{y|\Pi_i(\mathbf{x})} \prod_{i=1}^{n} \theta_{x_i|y,\Pi_i(\mathbf{x})}}{\sum_{y' \in \mathcal{Y}} \theta_{y'|\Pi_i(\mathbf{x})} \prod_{i=1}^{n} \theta_{x_i|y',\Pi_i(\mathbf{x})}} \tag{2}$$

5.2. Learning the Inference Graph

To preserve the significance of posterior estimator $P_{\mathfrak{B}}(y|\mathbf{x})$, Naïve Bayes takes the class variables as the root, and all attributes are conditional independent when conditioned on the class [13]. This assumption leads to the following form

$$P_{\mathfrak{B}}(y|\mathbf{x}) \propto \theta_y \prod_{i=1}^{n} \theta_{x_i|y} \tag{3}$$

As can be seen here, Naïve Bayes simplifies the structure of Bayesian network. In our proposed model, the structure of Naïve Bayes is used to infer the role of detected person.

To graphically and demonstratively infer the role of detected person, Figure 5 summarizes the pipeline of inference graph G_I, which are composed of two collaborative parts: state extraction (observation) and role probability inference. To be specific, orientation, action, color uniqueness of uniform, as well as field type are firstly employed to describe the state of an unknown candidate, which are then fed into the inference graph to produce the probability of each role. And the final role is decided based on the maximum probability.

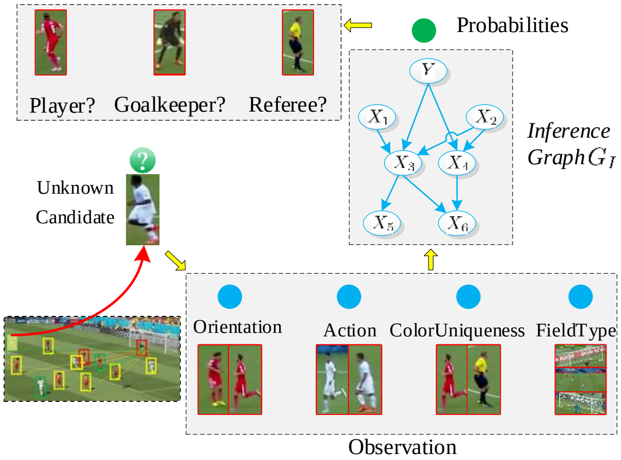


Figure 5: The pipeline of inference graph used for inferring the role of a person object.

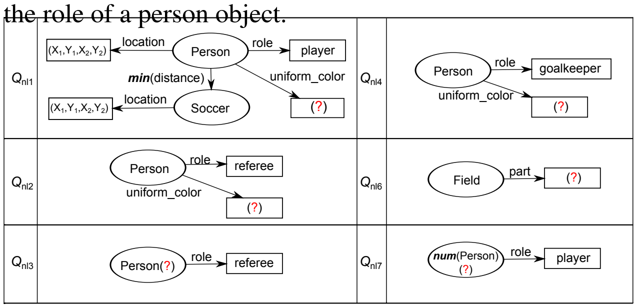


Figure 6: Query graphs

After inference, one can either use a complete EAG to answer queries, or directly apply inference graph to find answers to certain queries (see Section **??** for an example).

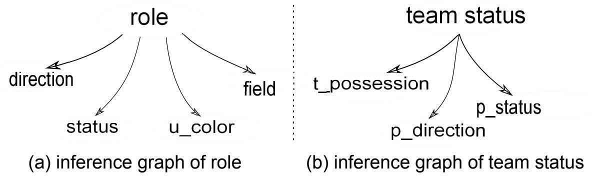
6. Experimental Studies

In this section, we conducted two sets of experiments to evaluate (1) the performance of our visual processing module, (2) the accuracy of our inference module, and (3) the overall performance of our approach.

Experimental Setting.

DataSet. We used two datasets: (1) Soccer dataset that we annotated; and (2) X dataset from []. We extracted images with subjects of golf and tennis (report Statistics about the dataset). We split Soccer (resp. X) data into two parts:*I* (one third) and *II* (two thirds), and used *II* as training data, and *I* as testing data.

Queries. We used two sets of questions: (1) the set of ques-



(a) inference graph of role

(b) inference graph of team status

Figure 7: Inference Graphs

tions given in Table 2 for Soccer dataset; and (2) another set of questions listed in Table **??** for X dataset.

Id	Question	Difficulty
Q_{nl_1}	Who is holding the soccer?	Easy
Q_{nl_2}	What is the uniform color of the referee?	Easy
Q_{nl_3}	Is there any referee in the image?	Easy
Q_{nl_4}	Which team does the goalkeeper belong to?	Medium
Q_{nl_5}	Who is the defending team?	Medium
Q_{nl_6}	Which part of the field are the players being now?	Hard
Q_{nl_7}	How many players are there in the image?	Hard
Q_{nl_8}		
Q_{nl_9}		
$Q_{nl_1 0}$		

Table 2: A set of questions

6.1. Performance of Visual Processing

Peixi, please report your results with details here.

6.2. Effectiveness of Inference

Accuracy of Role. Only report results with Reinforcement learning

Accuracy of Team-Status. Only report results with Reinforcement learning

Accuracy of Kick-Off. SHOW INFERENCE GRAPH AND RESULT TABLE!

Accuracy of Penalty Kick. SHOW INFERENCE GRAPH AND RESULT TABLE!

Accuracy of Corner Kick. SHOW INFERENCE GRAPH AND RESULT TABLE!

Accuracy of Attacking Free Kick. SHOW INFERENCE GRAPH AND RESULT TABLE!

Accuracy of Balls. Over new dataset. SHOW INFERENCE GRAPH AND RESULT TABLE!

6.3. Overall Performance

We compared the following state-of-the-art methods: X1, X2 and X3 with ours.

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. 2

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4

[3] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub) graph isomorphism algorithm for matching large graphs. *TPAMI*, 26(10):1367–1372, 2004. 2

[4] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 379–387, 2016. 4

[5] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997. 5

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in neural information processing systems*, pages 2296–2304, 2015. 1

[8] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3, 2017. 2

[9] D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 5

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016. 4

[11] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2015. 1

[12] K. Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001. 4

[13] F. Petitjean, W. Buntine, G. I. Webb, and N. Zaidi. Accurate parameter estimation for bayesian network classifiers using hierarchical dirichlet processes. *Machine Learning*, 107(8-10):1303–1331, 2018. 5

[14] J. Pound, A. K. Hudek, I. F. Ilyas, and G. E. Weddell. Interpreting keyword queries over web knowledge bases. In *CIKM*, pages 305–314, 2012. 2

[15] M. Ren, R. Kiros, and R. Zemel. Image question answering: A visual semantic embedding model and a new dataset. *Proc. Advances in Neural Inf. Process. Syst*, 1(2):5, 2015. 1

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural*

Information Processing Systems - Volume 1, NIPS'15, pages 91–99, Cambridge, MA, USA, 2015. MIT Press. 4

[17] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. *arXiv preprint*, 2017. 2

[18] Y. Tian and J. M. Patel. TALE: A tool for approximate large graph matching. In *ICDE*, pages 963–972, 2008. 2

[19] J. R. Ullmann. An algorithm for subgraph isomorphism. *JACM*, 23(1):31–42, 1976. 2

[20] A. Wagner, D. T. Tran, G. Ladwig, A. Harth, and R. Studer. Top-k linked data query processing. In *ESWC*, pages 56–71, 2012. 2

[21] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 4

[22] P. Xiong, H. Zhan, X. Wang, B. Sinha, and Y. Wu. Visual query answering by entity-attribute graph matching and reasoning. In *CVPR*, 2019. 2, 3, 4

[23] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016. 1

[24] W. Yih, M. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1321–1331, 2015. 2

[25] W. Zheng, H. Cheng, L. Zou, J. X. Yu, and K. Zhao. Natural language question/answering: Let users talk with the knowledge graph. In *CIKM*, pages 217–226, 2017. 2

[26] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017. 1

[27] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016. 1

[28] L. Zou, L. Chen, and M. T. Özsu. Distancejoin: Pattern match query in a large graph database. *PVLDB*, 2(1):886–897, 2009. 2