Identifying Visual Relationships using Object Tokens and Hand-crafted Features

Kartik Jain*
Department of Computer Science
University of Central Florida
Orlando, FL 32816
kjain@knights.ucf.edu

Abstract

Visual Relationship Detection (VRD) is a computer vision task that involves detection multiple objects in complex scenes and determining the relationship between the objects. While the traditional forms of human-object, and object-object relationships are predominant, large scale datasets such as OpenImagesV5 [Google (2019)] also consist of object-attribute relationships which simply describes the object. To handle both of these cases and approch VRD effectively, we first establish a baseline of our own using objects as visual tokens. Next, we improve upon this baseline by using Recurrent, and Attention models from Language Modelling literature. Next, on top of our best model, we add extra information based on how two objects interact. This is done by using their bounding boxes to compute distance and orientation w.r.t. each other. We find that this simple trick gives us a boost at the VRD task. We report our results on the OpenImagesV5 dataset [Google (2019)] by both Object Detection, and VRD. We also discuss limitations of our current method and suggest future improvements.

1 Introduction

Visual Relationship Detection (VRD) is an extension to the task of Object Detection (OB) which itself is an extension to Image Classification (IC). Moving from IC to VRD increases the scope, and complexity of the problem being tackled significantly. Therefore, as part of a Kaggle challenge, OpenImagesV5 [Google (2019)] included the VRD task with a total price money of $25000 involved. The winners got a chance to present their work at the ICCV 2019 workshop for VRD. Unfortunately for us, as none of the submissions actually uploaded their code (even after the competition ended), we now make our own baseline loosely following the VRD literature and make modifications based on VRD, as well as other Deep Learning literature.

More specifically, our baseline (called BaseModel from here on), is a feature extractor for the two objects involved, and the general scene. We simply apply classification on those features in BaseModel. Consulting work from Jung and Park [Jung and Park (2019)], and Liao et. al. Liao et al. (2017)], we see them approaching VRD through a Natural Language perspective. We take a similar approach in that we consider the object-relationship-object as a sentence that Language Models (LMs) should be effectively able to predict thereby solving our VRD task as a corollary. For this, we employ powerful LMs such as Long Short Term Memory networks [Hochreiter and Schmidhuber (1997)] (called LSTMModel from here on), and Transformer network [Vaswani et al. (2017)] (called TransformerModel from here on).

Finally, using our best model, we simply add information obtained from the two bounding boxes of objects (called BestModel from now on). First, we compute the Euclidean distance between the two

*Mini-project 3 of course CAP6614: Current Topics in Machine Learning.

3rd Conference on Neural Information Processing Systems (NeurIPS 2042.8), Earth-616.

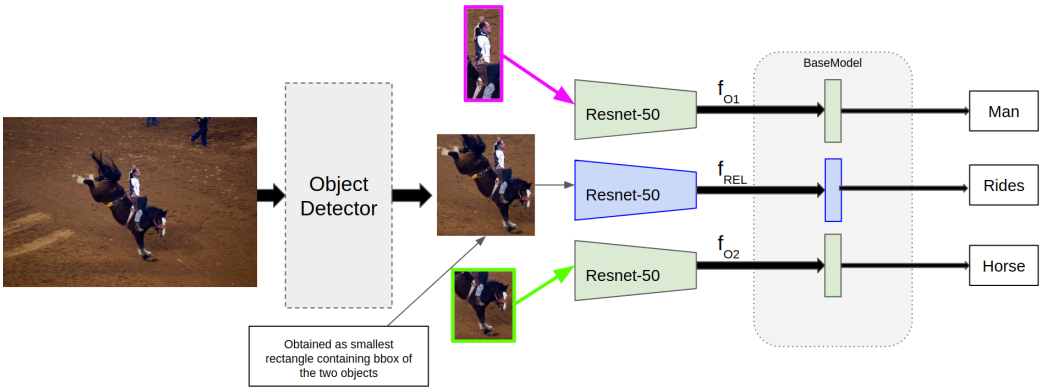

Figure 1: Illustration of base model used for Visual Relationship Detection. The obtained bounding box information of objects are used to find the smallest rectangle that fits both the objects to train the relationship classifier. Same color indicates weight-shared networks. BaseModel simply used object and scene features for object, as well as relationship classification.

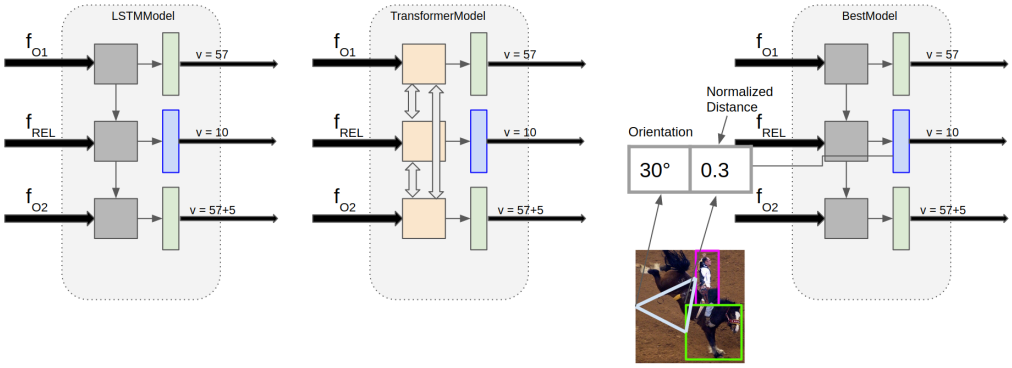

Figure 2: Illustration of the updates made to the Visual Relationship Detector. **(Left)** Instead of three-independent classifiers, LSTM combines to predict "words" at each time-step with varying vocabulary sizes since the "sentence" follows a fixed structure: object-relationship-object/attribute. **(Middle)** A Transformer block combines all three "words" of the sequence to selectively attend to features for "word" prediction. **(Right)** Augments the best model by adding two features to the relationship classifier: distance, and orientation. The distance is computed between the center of the two bounding boxes while the angle orientation is computed from a fixed point. For single-object predictions i.e. is-attribute relations, we repeat the first object twice reducing the distance, and orientation vector to a zero-vector: $[0°, 0.0]$.

boxes. Second, we compute the orientation of those boxes with respect to each other by means of the angle θ subtended by the two vectors from the origin to the box centers. Adding these two as hand-crafted features improved VRD accuracy on the test set by a bit.

The rest of the paper is organized as described next. In Section 2, we define the model in detail covering the relevant works, as well as Kaggle solutions. We talk about our LM inspired updates, and the included hand-crafted features. In Section 3, we discuss training details and experiments conducted on Open Images V5. In Section 4, we discuss the shortcomings of this approach and conclude in Section 5.

2 Method

We outline the major ideas borrowed from relevant VRD, and LM literature in the following subsections in the context of how they fit in our method.

2.1 Object Detector

We use FasterRCNN [Ren et al. (2015)] as our object detection network as it is achieves a balance between fast, and accurate owing to it's region proposal networks. It can handle multi-scale objects, and is therefore a reliable backbone for many VRD models − multiple teams from the top-5 on Kaggle challenge used FasterRCNN as their backbone. The ease of use is also an advantage since an we were able to use weights pre-trained on COCO [Lin et al. (2014)] dataset.

Although, as observed from our initial experiments with FasterRCNN, training for OD on OpenImagesV5 is compute intensive. In order to make the compute feasible, we ended up not using any object detectors since our primary task is VRD and not OD anyway. Instead, we used ground truth bounding box but point our that any sufficiently trained object detector should perform extremely close to our models.

2.2 BaseModel

We only need the bounding box output from the object detector as we use only the objects, and the scene containing those objects for further processing. First, we select the top two object predictions (replication in case of single-object images), and crop the images to pass those objects to a pre-trained ResNet-50 [He et al. (2015)] backbone pre-trained on ImageNet [Deng et al. (2009)]. The weights are shared between those two networks to speed up training.

Second, using the smallest rectangle possible to fit the two bounding boxes, we crop the relevant "scene" and discard the irrelevant information. This image is used for relationship prediction, again, by means of a Resnet-50 backbone pre-trained on ImageNet. Next, in this particular model (BaseModel), we simply use the 2048-length vectors from both objects (f_{O1}, f_{O2}) along with a linear classifier (fully-connected layer) for object classification as an auxiliary task to improve relationship classification. As seen from Figure 1, relationship classifier uses f_{REL}, again, with a simple linear classifier to predict one out of the ten possible relations.

2.3 LSTM Model

VRD via object-object or object-attribute relations read like a natural sentence. For example, "man rides horse", "girl plays guitar", "bottle on table", etc. all seem like small captions describing an image. Also, since this "sentence" follows a structure i.e. object-relation-object/attribute, we can even have a dynamic vocab size while predicting the three word tokens. For example, in timestep one, the vocab is restricted to only objects (Therefore, 57 in case of OpenImagesV5), while the next word i.e. relation can take one out of only 10 distinct values.

Since LSTMs [Hochreiter and Schmidhuber (1997)] are popular in Image Captioning [Vinyals et al. (2014)], we chose them as our initial LM model. More specifically, we used a 3-layer, unidirectional LSTM where the output from each timestep go to their respective linear classifier for object/relationship prediction. In addition, we would not expect any shortcomings of LSTMs such as inability to handle long sequences since our sequence is pretty short i.e. `seq_len = 3`.

2.4 Transformer Model

Continuing the object-relation-object/attribute "sentence", we seek to leverage the progress made in LMs over the past few years. Transformer model [Vaswani et al. (2017)] stands out in this regard, as it significantly pushed the state-of-the art at the time by employing self-attention i.e. the ability of a sequence to attend to itself over a long range. In our setup, we used a encoder-decoder architecture that auto-encodes the features $[f_{O1}, f_{REL}, f_{REL}]$ by right-masking in the decoder stage.

More precisely, in our setup, we use a 4-head, 2-layer Transformer network that hopefully learns better representations in order to ease the task of their respective linear classifiers.

2.5 Hand-crafting Features

Although hand-crafting features is frowned upon within the Deep Learning community, we have seen many hybrid systems outperform fully-differentiable or purely-symbolic models. The winners of the VRD challenge too used a pipeline that predicts/models more that what the task-at-hand demands i.e. they used bounding box centroids in their Gradient Boosting Machines (GBMs). We simply append two such features to our f_{REL} feature vector and make it a $2048 + 2$-length vector. We describe them below.

Distance - Euclidean Distance between the centers of the two bounding box provides a lot of information as we can simply identify "is" relationships since there is only one object thereby giving `distance = 0`. Furthermore, relationships such as "inside of" may have a shorter distance than, say, a relation such as "plays" depending on how the scene is structured. We normalize the distance based on the image size to keep it in the same range as others (see Figure 2 for an illustration).

Orientation - In cases where simple distance may not provide enough information, we can seek help of the orientation of said objects. For example, an "on" relationship may almost always have 90° with the horizontal axis since that is how the gravity works :-p. Here, we pick a vector at the origin to compute the angle between the vectors origin-object_1, and origin-object_2. Even though Figure 2 shows orientation in degrees, we use radians to, again, keep the range comparable to other activations.

3 Experiments

We use the 5^{th} release of OpenImages [Google (2019)] which contains millions of object detection annotations but we restrict ourselves to the subset which has visual relationship annotation. There are 390K instances that meet this criteria. Next, to make the compute feasible, we train our models on 25% of the training set and evaluate on a separate set with similar annotations available for testing performance during inference. We run our experiments for 10 epochs each since many of the sub-networks/modules themselves are pre-trained on larger datasets. For the evaluation criteria, we simply look at accuracy obtained on test sets for both object classification, and, more importantly, relationship classification tasks.

Other hyperparameter details are as follows: `lr = 3e-4, optimizer = "adam", momentum = 0.9, weight_decay = 1e-4, batch_size = 32, img_size = 256`. We have implemented the entire project in PyTorch. The code is available in the same submission zip archive where you may have found this report.

Quantitative Results - Observing the top left graph in Figure 3, we can see the Object 1 accuracy for the models increase as the training progresses. We see a generally high object 1 accuracy for the models (exception: TransformerModel; more in discussion).

Moving to the top right graph, we see that the BaseModel suffers significantly in identifying the second object. We hypothesize that this is mainly due to the added categories in the form of attributes which adds ambiguity to the visual features. We also see that LSTM based models do not suffer in this category and believe that this is due to the context gathered from the previous two timesteps (more in discussion).

Finally, the graph at the bottom shows the Relationship accuracy as the training progresses. As we add updates to the BaseModel, we see a jump in the achievable relationship accuracy with the BestModel partly using hand-crafted features performing the best in this context. Since this is the task at hand, we highlight out this graph as the most important finding from our study.

Qualitative Results - As seen from Figure 4, out of the randomly chosen nine samples, we see five showing correct predictions as marked by the green border. At location (0, 1), the ground truth is "Wine Glass on Table" while the prediction does not focus on the glass at all. Similarly, at (1, 1), the ground truth is "Bottle is Transparent" while the model does not focus on the attribute at all. For examples at (2, 2), and (2, 3), the respective ground truths are "Woman wears Handbag", and "Man holds Microphone". In these cases, while the model produced a structurally, and visually plausible relations, they do not match the instance's ground truth.

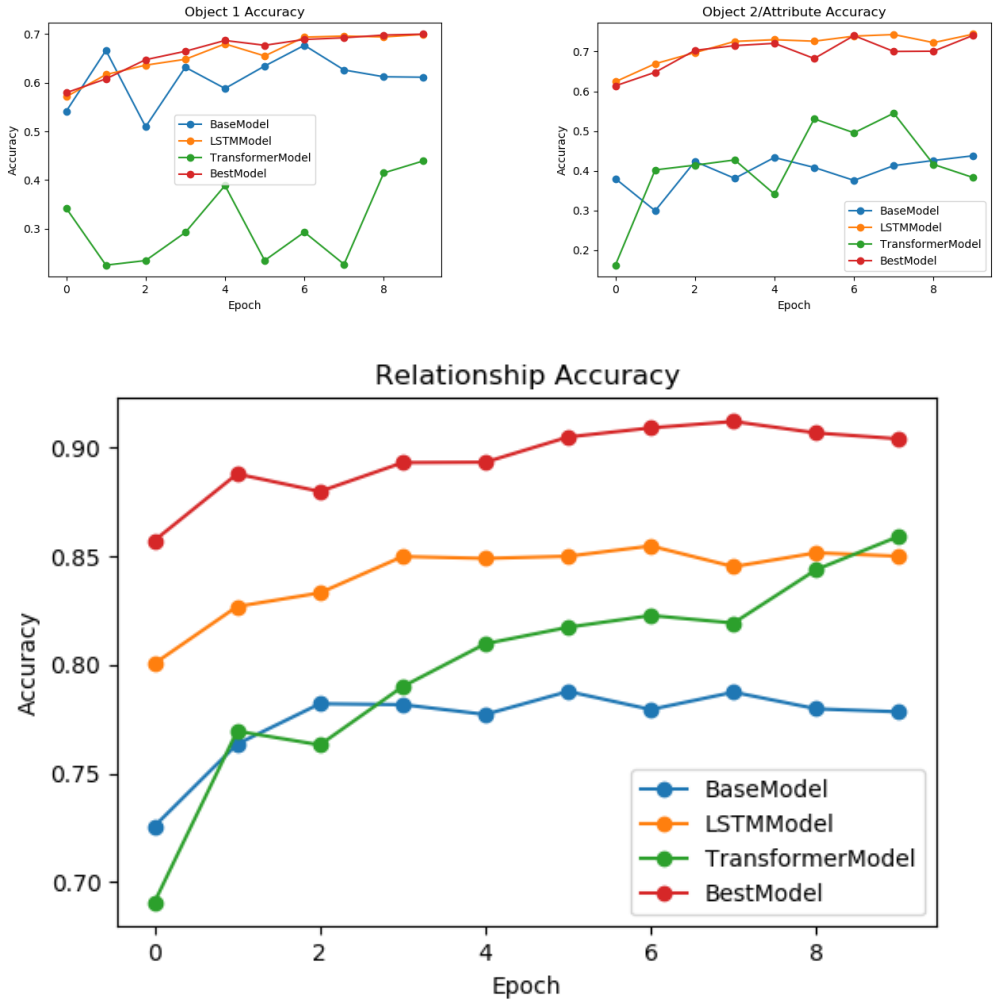

Figure 3: **(Top Left)** Object 1 Accuracy vs. Epochs on the test set for a total of 57 object categories. **(Top Right)** Object 2 Accuracy vs. Epochs on the test set for a total of 57 object + 5 attribute categories. **(Bottom)** Relationship Accuracy vs. Epochs on the test set for a total of 9 relationship + 1 is-attribute categories.

4 Discussion

As mentioned previously, we used the cropped objects to identify themselves, and to aid our relation classification. This makes it crucial that, in practice and while deploying, we need a near perfect object detector which may be difficult to achieve on complex datasets with a lot of variation. Access to GT bounding box is our model's biggest weakness.

Next, as for the LM intution itself, we could employ language priors more heavily in that we could restrict the vocabulary size of `object_2` to just five if the previous timestep outputted is-relationship. We could also consider WordNet [Miller (1995)] as hierarchy of words such that the model extract features for a broader category, and then find it's closest child in the vocabulary. We leave all of this as future work.

For the experiments, we see that the TransformerModel seems to perform poorly on the object classification task but at the same time also seems more unstable than other models i.e. non-monotonic behavior. It may just be the case that the Transformer might have outperformed other models given enough training but we leave this for future exploration.

Figure 4: Few qualitative results shown from different relationship categories in a 3x3 grid on the trained BestModel. Images with green border indicate complete/near-complete correct prediction while images with red border indicate incorrect prediction and/or structure. Images with yellow bounding box indicate a structurally, and visually correct relationship but one which does not match the ground truth relationship-triplet (all of these are based on visual assessment, and are not cherry picked).

On the other hand, the LSTMModel seems to perform consistently well on all three accuracies. It is interesting to see a huge jump in `object_2` accuracy when LSTM's are used which is most likely because of the gathered context from previous timesteps in the object-relationship-object structure. It is because of this we chose LSTMs as the backbone for BestModel. Finally, we saw that even two simple hand-crafted features gave us more than 5% improvement in accuracy suggesting that there is much room for improvement within the realm of smartly hand-crafting features.

After visual introspection of the nine instances included, and 100 instances not included in this report, we find that our BestModel, while predicts the is-relationship quite accurately (possibly because the relationships are class-imbalanced in favor of is-relationship), it has a hard time assessing the correct attribute. For example, "Bottle is Transparent", and "Bottle is (made of) Plastic" are not easily distinguished even with visual features. We see promise in including an attribute prediction network tailored to such cases but are not aware of any robust methods at the moment and therefore leave this to future work.

5 Conclusion

Using powerful Language Models, we were able to convert a three-independent-classifications task to a problem similar to image captioning with temporal dependencies. Next, using simple hand-crafted features which made sense for the Visual Relationship Detection task, we were again able to improve the relationship prediction of our model. While we understand investing in hand-crafting features may not help in the long run of advancing research, it may be suitable for challenge-based problems such as these, where each % improvement counts. We also mention requirement of accurate bounding boxes as one of the main limitations to our method. Finally, we report results on OpenImages dataset

and suggest potential future work. We also had fun while writing the entire code for this project from scratch, and enjoyed learning throughout this course. Thank you :-)

References

Google, Open Images V5 - Visual Relationship Detection. *ICCV* **2019**, *v5*.

Jung, J.; Park, J. Visual Relationship Detection with Language prior and Softmax. *CoRR* **2019**, *abs/1904.07798*.

Liao, W.; Lin, S.; Rosenhahn, B.; Yang, M. Y. Natural Language Guided Visual Relationship Detection. *CoRR* **2017**, *abs/1711.06032*.

Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; Polosukhin, I. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 5998–6008.

Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Cambridge, MA, USA, 2015; pp 91–99.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; Dollár, P. Microsoft COCO: Common Objects in Context. 2014.

He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *CoRR* **2015**, *abs/1512.03385*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. CVPR09. 2009.

Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and Tell: A Neural Image Caption Generator. *CoRR* **2014**, *abs/1411.4555*.

Miller, G. A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41.