OSCAR: Occluding Spatials, Category, And Region under discussion

Raymond Liu
Princeton University
rl27@princeton.edu

Indu Panigrahi
Princeton University
indup@princeton.edu

Abstract

Visual dialog is an important task that lies at the intersection of natural language processing and computer vision. It has far-reaching implications in the real world, so it is important to understand what information VD models depend on. In this paper, we explore one model for visual dialog called Question-Category-Spatial with a Region under Discussion. In addition to reproducing the results of the original work, we introduce our own ablations on the input embedding in order to understand what information is most useful for the model. Finally, we contribute a more efficient input embedding for the posed questions.

1 Introduction

Visual dialog (VD) is a joint task between natural language processing and computer vision that involves a machine engaging in dialog about a visual input. This task has many applications both in real-world deployment and within research. In applications used by the general public, it can help visually-impaired people approach visual challenges by answering questions about their surroundings. More generally, VD can form the base for smart assistants that answer questions with visual inputs. In research, VD can be used to evaluate whether or not a model really understands a visual input. Given the multitude of potential applications of VD, it is worth investigating what information VD models depend on as doing so can help improve the reliability of the system.

In this paper, we reproduce the Question-Category-Spatial (QCS) model with a Region under Discussion (RuD), a model that performs VD. In addition to performing the feature ablations from the original work, we introduce our own ablations on the input embedding to understand the effect of each component of the embedding on the model's performance. Furthermore, we experiment with different ways of producing the question embedding and find that using a GRU is less computationally expensive than the originally-used LSTM while yielding a similar performance.

2 Related Work

2.1 Visual Dialog

Visual dialog is a conversation comprised of a series of questions and answers that form a dialog history (i.e., each question and answer could depend on the past questions and answers in the conversation) (Das et al., 2017). The popular task of Visual Question Answering is essentially one iteration of VD.

2.2 QCS Model

The QCS model takes an input embedding composed of the target category of the object of interest, an LSTM embedding of the posed question, and spatial information about the object with respect to the entire image. The embedding is fed into a Multilayer Perceptron (MLP), and the model outputs an answer (Yes, No, or N/A).

The target category is encoded as a dense category embedding that is obtained from a one-hot class vector (Fig. 1). The embedding of the question is generated by an LSTM. The spatial information of the object (annotated as part of the COCO dataset (Lin et al., 2014)) with respect to the image consists of eight coordinates (de Vries et al., 2017). These three parts are concatenated together to form the input embedding.

2.3 QCS + RuD

QCS+RuD model is identical to QCS except it appends to the input embedding the spatial information of the object with respect to a *Region under Discussion* (RuD) (Fig. 2). The RuD uses dialog

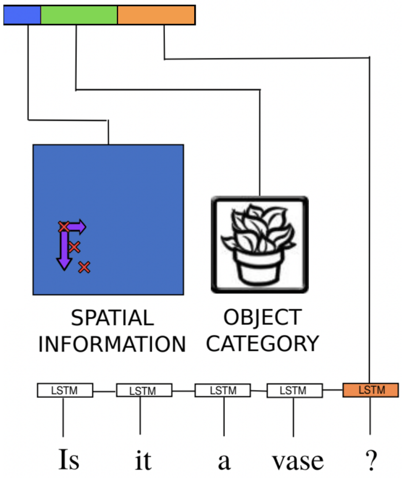

Figure 1: QCS Input Embedding.

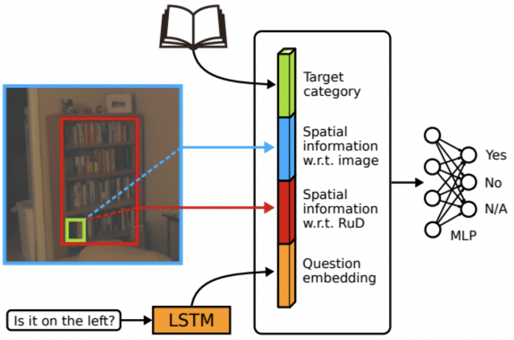

Figure 2: Schematic of the QCS+RuD model including the input embedding.

history to narrow down the region of the image that is relevant to the conversation. This addition is meant to help improve the accuracy of the Oracle's answers.

2.4 Interpretability of Embeddings

Understanding what information a model depends on is useful for diagnosing shortcomings in the model and improving its reliability. Since the input embedding to QCS+RuD is comprised of four distinct parts, we seek to determine the effect of each part on the model.

There exist a few methods for examining embeddings. One such method is the word intrusion test which can be applied to topic models to determine if the topics that the model predicts are reasonable (Chang et al., 2009). However, this method involves several rounds where a human chooses from a set of words from a model's generated collection of words under a topic. As a result, this method is labor-intensive.

Another set of methods forms more interpretable embeddings by projecting dense embeddings to sparser embeddings with higher dimensions (Arora et al., 2016; Templeton, 2021). However, these methods do not improve upon the existing dense embeddings.

2.5 GuessWhat?!

To evaluate the QCS model, de Vries et al. developed the GuessWhat?! dataset (de Vries et al., 2017). The GuessWhat?! dataset consists of a game where a "Questioner" asks an "Oracle" a series of questions in order to identify an object in an image. The questioner and oracle are two tasks, and the QCS model plays the role of the oracle. We use this dataset to train and evaluate our models.

3 Reproducing QCS+RuD

We reproduce the baseline results from the original QCS+RuD work (Mazuecos et al., 2021) as well as their original ablations (Table 1). We obtained similar results as the original paper for both the baseline model and the ablations.

Mazuecos et al. perform three feature ablations by alternately removing supercategory matching, second-token matching, and negative history. Some noun phrases (NPs) use supercategories. These are nouns that cover several COCO categories. For example, "food" covers "apple", "orange", "cucumber", etc. These supercategories are used in the semantic history for each game. The semantic history is the list of positive and negative relations to the supercategories from previous rounds of dialog (e.g. "vehicle" being positive and "car" being negative means that the object is a vehicle but not a car) (Mazuecos et al., 2021). Some category questions use two-token NPs (i.e. "red car") in which case only the second token refers to the category (i.e. "car" in "red car"). The QCS+RuD model only matches to the second token if the answer is "yes". Lastly, negative history removes all objects from negated supercategories from the set of possible objects to be included in the RuD.

4 Ablations on Input Embeddings

This section details our novel ablations on the QCS+RuD model.

As the input embedding consists of several parts, we alternately ablate these parts to examine the effect of each on the model. We do not ablate the question embedding as the task is to engage in dialog by answering the questions.

Type	QCS	QCS+RuD	-super	-2nd	-neg
object	0.908	**0.911**	0.909	0.912	0.909
spatial	0.677	**0.687**	0.695	0.694	0.698
color	0.624	**0.629**	0.633	0.625	0.630
action	0.647	**0.662**	0.662	0.658	0.667
size	**0.633**	0.628	0.629	0.637	0.620
texture	**0.724**	0.706	0.714	0.710	0.716
shape	0.668	**0.698**	0.688	0.678	0.678
GW	0.781	**0.787**	0.790	0.789	0.790

Table 1: Reproduced paper results. We obtain similar results as the original paper.

We use the term "–image" to refer to ablating the spatial information of the object with respect to the image. Similarly, "–target" refers to ablating the target category embedding, and "–RuD" refers to ablating the Region under Discussion. We do not include an explicit ablation on the RuD alone because that is the equivalent of the QCS model itself.

The test set is divided into questions with and without history information. We test each model on the overall test set, as well as on the two subsets of questions with and without history.

Without history, RuD is defined to be the whole image, so the RuD spatial embedding is the same as the spatial embedding for the entire image. So –image without history is equivalent to QCS.

We train each model for 16 epochs, with a learning rate of 0.0001, and a batch size of 1024, using an NVIDIA Tesla P100 with 16GB GPU memory.

4.1 Results

1. The RuD captures a significant amount of image information.

The performance of QCS+RuD and –image are similar. Furthermore, –target–RuD performs worse than –target. These trends imply that the model can eventually deduce the target object from the RuD (Table 2). This suggests that the RuD alone still captures enough information about the image for the model to successfully engage in dialog.

2. Removing the target category is generally more detrimental to the model's performance than removing the other parts of the input embedding.

All models that include the target category in the input embedding usually perform significantly better than models that ablate the target category. Compared to -image and QCS (which is equivalent to -RuD), -target demonstrates a significant drop on most types of questions, except size and spatial. This trend seems reasonable because when we remove the target category, the model does not necessarily know what it is looking for until it accumulates enough dialog history.

3. Sometimes, the ablations remove too much information.

Removing information can be detrimental to performance. For example, –image–RuD, -target-RuD, and –image–target perform much worse than QCS+RuD.

4. Accessing the dialog history helps when the model does not know the target category.

The reduction in performance from QCS+RuD w/o history to –target w/o history is larger than that from QCS+RuD w/ history to –target w/ history. This observation suggests that removing history negatively impacts the model that does not have the target category. This trend is reasonable because if the model does not know what type of object it is looking for, it needs some supporting information. In other words, the model without the target category benefits from accessing the dialog history (Table 4).

5 Modifying the Question Embedding

Next, we perform a set of novel experiments in which we evaluate different ways of embedding the posed question.

Specifically, we alternately replace the original LSTM model with a GRU (Paszke et al., 2019), a transformer encoder (Paszke et al., 2019), and a naïve embedding. We then compare the performance corresponding to each embedding. We implement a GRU because GRUs have fewer parameters and consequently take less time to train (Chung et al., 2014). Next, we try a transformer since they generally perform better than LSTMs in practice. Finally, we try a naïve method that simply averages the word embeddings in a question to produce the question embedding. Additionally, we considered using a bag-of-words model; however, we decided that the model would be too large given the vocabulary size. We also tried using a pre-trained BERT model but found that training was unreasonably slow.

LSTM and GRU take word embeddings of size 300 and return vectors of size 512. The naïve method returns vectors of size 300.

Type	QCS+RuD	-image	-target	-image-RuD	-image-target	-target-RuD
object	0.911	0.908	0.746	0.903	0.724	0.736
spatial	0.687	0.684	0.688	0.585	0.678	0.670
color	0.629	0.630	0.601	0.601	0.599	0.574
action	0.662	0.659	0.620	0.613	0.617	0.601
size	0.628	0.651	0.662	0.557	0.637	0.638
texture	0.706	0.733	0.634	0.713	0.625	0.617
shape	0.698	0.688	0.638	0.635	0.601	0.638
total	0.787	0.785	0.699	0.741	0.685	0.684

Table 2: Performance on all questions. The leftmost column indicates the question types. “total” is the general performance across all types.

Type	QCS+RuD	-image	-target	-image-RuD	-image-target	-target-RuD
object	0.895	0.893	0.740	0.884	0.715	0.728
spatial	0.684	0.681	0.691	0.579	0.679	0.667
color	0.620	0.620	0.602	0.588	0.601	0.574
action	0.645	0.639	0.614	0.589	0.617	0.595
size	0.618	0.636	0.659	0.551	0.635	0.641
texture	0.701	0.721	0.641	0.701	0.626	0.620
shape	0.650	0.675	0.613	0.588	0.563	0.625
total	0.742	0.740	0.691	0.682	0.678	0.672

Table 3: Performance on questions with history.

Type	QCS+RuD	-image	-target	-image-RuD	-image-target	-target-RuD
object	0.935	0.930	0.753	0.931	0.737	0.749
spatial	0.697	0.693	0.679	0.606	0.676	0.678
color	0.656	0.660	0.596	0.640	0.590	0.573
action	0.719	0.724	0.639	0.692	0.619	0.623
size	0.657	0.694	0.671	0.572	0.643	0.629
texture	0.714	0.751	0.623	0.731	0.623	0.612
shape	0.752	0.702	0.667	0.688	0.645	0.653
total	0.857	0.856	0.711	0.833	0.695	0.703

Table 4: Performance on questions without history.

Type	QCS+RuD	GRU	Transformer	Naïve
object	**0.911**	**0.911**	0.744	0.815
spatial	0.687	**0.696**	0.596	0.607
color	**0.629**	0.626	0.554	0.580
action	0.662	**0.663**	0.574	0.601
size	0.628	**0.631**	0.553	0.584
texture	0.706	**0.717**	0.593	0.613
shape	**0.698**	0.691	0.611	0.598
total	0.787	**0.790**	0.635	0.713

Table 5: Extended results (all questions).

LSTM	GRU	Transformer	Naïve
2100	950	2660	69

Table 6: Approximate times in seconds to complete one training epoch for each question embedding model. LSTM is the original baseline model.

We train all non-transformer models for 16 epochs, with a learning rate of 0.0001, and a batch size of 1024 as we did for our ablations on the input embeddings. As before, all models are trained using an NVIDIA Tesla P100 with 16GB GPU memory.

Due to insufficient GPU memory, we simplify the transformer encoder model to have dimension 256, one head, and two layers. We also reduce the batch size to 512. Lastly, we modify the word embeddings to have size 256 to be compatible with the transformer.

5.1 Results

We find that the GRU trains faster (Table 6) and yields similar accuracy to the LSTM (Table 5). The transformer encoder takes a long time to train despite our simplifications and yields a worse accuracy than the LSTM. Finally, the naïve implementation performs worse than the LSTM but performs better than the transformer encoder. Thus, our results suggest that using a GRU may be more efficient than using an LSTM.

6 Conclusion and Future Work

In this paper, we reproduce the QCS+RuD model and feature ablations from "Region under Discussion for visual dialog" (Mazuecos et al., 2021). We then contribute novel ablations on the input embeddings. Lastly, we improve the question embedding by replacing the LSTM with a GRU.

A potential improvement is to reduce the model's reliance on image annotations by developing a way to leverage visual features detected by computer vision models. The current spatial information relies on human annotations from the COCO dataset (Lin et al., 2014) to construct the RuD because the QCS paper found that visual features worsened the performance. However, as annotations are not necessarily available for all datasets, concurrently computing visual features would be useful.

Acknowledgments

We would like to thank our adviser, Professor Karthik Narasimhan, for guiding this project as part of the COS 484 Natural Language Processing course. We would also like to thank Mauricio Mazuecos, a co-author of the original QCS+RuD paper, for code clarifications.
Code for this project is available at:
https://github.com/ind1010/Region-under-discussion-for-visual-dialog

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Linear algebraic structure of word senses, with applications to polysemy.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312.

Mauricio Mazuecos, Franco M. Luque, Jorge Sánchez, Hernán Maina, Thomas Vadora, and Luciana Benotti. 2021. Region under Discussion for visual dialog. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4745–4759, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA.

Adly Templeton. 2021. Word equations: Inherently interpretable sparse word embeddings through sparse coding. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 177–191, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.