Visual Conversational Agent

Vincent Kang
vkang@andrew.cmu.edu

Serena Wang
serenaw1@andrew.cmu.edu

David Zeng
dzeng@andrew.cmu.edu

Abstract

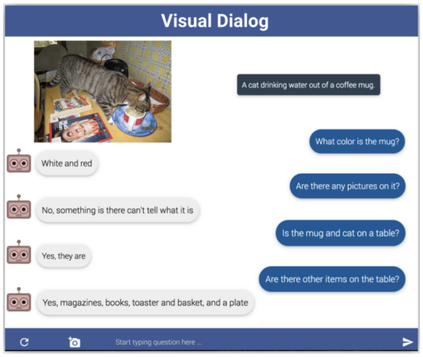
This paper focuses on the problem of Visual Dialog, or creating agents that can answer questions that are part of a larger dialogue with respect to a visual context. One challenge with this dataset that distinguishes it from visual question answering is the need to incorporate information from previous questions to answer the current question. Another challenge is the unclear training signal produced when training with a single ground-truth answer when there are multiple similar answer choices. We propose a novel application of MCB fusion to visual dialog to better capture relationships between the question, image, and history. We also explore incorporating similarity metrics into the training loss function and using soft labels based on similarity to ground truth. Although our new models do not improve performance statistically significantly, we also propose directions of future research for incorporating semantic similarity and advanced fusion techniques into Visual Dialog models.

1 Introduction

1.1 Research Problem

Visual Dialog (VisDial) is a new dataset proposed by Das, et al. [4]. Given an image, dialog history, and a natural-language question about the image, our agent should answer the question correctly in natural language. Specifically, when given an image, a history of question-answer pairs, a follow-up question, and 100 candidate answers, our model should return a ranking of the candidate answers. The main challenge is to properly combine the information presented from different modalities to best rank the candidate answers. A secondary challenge is to create a generative model that can generate accurate and realistic answers.

The goal for artificial intelligence (AI) systems is to be able to handle all kinds of complex human-machine interactions. A Visual Dialog agent is a step towards that goal and can serve as a Turing test for modern AI systems. Furthermore, as AI conversational units, such as Google Assistant and Amazon Alexa, become more complex, they will inevitably have to handle visual elements in conversations. A Visual Dialog agent can also be used to help a visually impaired user understand visual content or to guide robotics operators in situations where the operator is "situationally blind" [4].



A cat drinking water out of a coffee mug.

What color is the mug?

White and red

Are there any pictures on it?

No, something is there can't tell what it is

Is the mug and cat on a table?

Yes, they are

Are there other items on the table?

Yes, magazines, books, toaster and basket, and a plate

Figure 1: An example scenario demonstrating the Visual Dialog task

Visual Dialog is an extension of the visual question answering (VQA) task [1]. Many models for VQA focus on finding the section of the image referred to by the question so that the model can generate an answer based on the relevant image region and the question itself. Since Visual Dialog includes a history of previous questions, models for

Preprint. Work in progress.

Visual Dialog also need to use context from dialog history
to find the relevant section of the image.

In this paper, we present two models to address the challenges of having semantically similar answer options and properly combining images and text modalities. We incorporate similarity metrics as both evaluation and training metrics in our late fusion model. In our other model, we use multimodal compact bilinear pooling to build on top of a baseline model.

2 Related Work

2.1 Visual Dialog

Das, et al., [4] introduced the dataset and the research problem in 2017. The data was collected using a live chat interface between a questioner and answerer. The questioner sees a caption of the image and is required to ask questions about the hidden image to understand the image better. The answerer sees the image and caption and answers questions asked by the questioner. We also use the evaluation protocol introduced in this paper for our baseline model evaluations.

The paper uses a family of encoder-decoder models to model the answerer. The two types of decoders used are generative and discriminative. The encoders used are late fusion, hierarchical recurrent, and memory network. Results indicate that discriminative models perform better than generative models and that naively incorporating history does not improve the model. The memory network encoder performs the best out of all encoders with both the generative and discriminative decoders.

This work has spurred various works of research for Visual Dialog. Chattopadhyay, et al. [3] , design a cooperative game which extends the Visual Dialog problem in order to benchmark the progress made in human-AI team research. Reinforcement learning techniques have been built on the Visual Dialog dataset [5], [17]. Das, et al. [5], train cooperative Visual Dialog agents that end up developing their own means of a communication in a synthetic ungrounded world, as well as creating more informative dialogs within a supervised dataset. Reinforcement learning can also be used to train agents in goal-oriented dialog [17]. Kottur, et al. [9] use a neural module network architecture for coreference resolution in the Visual Dialog dataset.

2.2 Similarity Metrics

Prior work has focused on using similarity metrics for better evaluation of model that generate answers. ROUGE-G is one such algorithm for computing semantic similarity scores between n-grams [14]. Word Mover's Distance is another metric for the difference in meaning between two documents [10]. Mitchell, et al. [13], also developed a vector-based model for phrase similarity.

Semantic similarity has been shown to be useful for information retrieval [8], but also for image caption retrieval [15]. Similarity metrics are also useful for image caption generation since Fang, et al. [6], achieved better results than all existing models at the time by using a semantic similarity model on candidate captions.

2.3 Improved Fusion

An effective joint representation captures many associations between modalities. However, capturing complex associations between modalities while keeping the performance of the representation learning reasonable may be challenging. Fukui, et al. [7], use multimodal compact bilinear pooling to build a joint representation that can capture interactions between all elements of both the image and text modalities of VQA. Their approach approximates the outer product between the image and text unimodal representation vectors by randomly projecting them to a higher dimensional space and performing an element-wise product in the Fast Fourier Transform space. Ben-younes, et al. [2], developed a different approach for multimodal fusion for VQA called MUTAN, which uses Tucker decomposition of the correlation vector of the image and text representations to represent all bilinear interactions between the two vectors. Liu, et al. [11], developed another fusion method for VQA called Low-rank Multimodal Fusion (LMF). LMF uses a low-rank decomposition of the weight tensor for the first layer that the product of the image and text representation tensors are passed through, and since that decomposition allows computation of the output of that layer without doing the complete tensor product, this fusion method also captures all multimodal interactions with improved performance and fewer parameters.

2.4 Contribution

To our knowledge, this paper is the first to explore better fusion methods and textual semantic similarity in a model for Visual Dialog. Using VQA fusion methods should create a better multimodal representation for our models, and computing scores for candidate answers based on semantic similarity to the ground truth answer should improve the model's ability to differentiate between correct and incorrect answers.

3 Proposed Approaches

Overall, we can frame the task in the following way. Given the input image I, dialog history $H = (C, (Q_1, A_1^{gt}), ..., (Q_{t-1}, A_{t-1}^{gt}))$, question Q_t, and candidate answers $\mathcal{A} = \{A_t^{(1)}, A_t^{(2)}, ..., A_t^{(100)}\}$, output a ranking for the candidate answers using model parameters θ that minimizes the rank of the ground truth answer A_t^{gt}.

$$\underset{\theta}{\operatorname{argmin}} \, R(A_t^{gt} | Q_t, I, H, \mathcal{A}; \theta)$$

3.1 Similarity Metric

We introduce a similarity-metric approach as well as a new loss function.

3.1.1 Model Description

Rather ranking directly, models on this dataset typically learn a probability function P such that minimizes the following loss function. A ranking is then generated from a ranking of answer probabilities.

$$\mathcal{L}_\theta = \log P(A_t^{gt} | Q_t, I, H; \theta)$$

Here, there is a single answer that is labeled as the ground-truth. Our proposed approach utilizes pretrained GloVe word vectors to generate an answer vector based on an average of word vectors. These answer vectors are compared using cosine-similarity to the ground truth to generate a similarity metric. Then, we change the loss function to be cross-entropy with soft labels:

$$\mathcal{L}_\theta = -\sum_i l(A_t^i) \log P(A_t^i | Q_t, I, H; \theta)$$

where

$$l(A_t^i) = \frac{e^{\lambda \cdot \text{sim}(A_t^{gt}, A_t^i)}}{\sum_j e^{\lambda \cdot \text{sim}(A_t^{gt}, A_t^j)}}$$

3.1.2 Parameter Learning

We train the model using the above described loss function and tune the hyperparamter λ using the validation set.

3.2 MCB

We introduce multilinear compact bilinear pooling (MCB) as a fusion method for Visual Dialog.

3.2.1 Model Description

Existing models for Visual Dialog combine data from multiple modalities by concatenating them. The history-conditioned image-attentive encoder (HCIAE) developed by Lu, et al., [12] concatenate the question embedding and an attended history representation to produce attention weights for the image. This encoder also concatenates the question embedding, attended history representation, and attended image features to create the final embedding of the input data.

In our proposed architecture (shown in Appendix Figure 6), we replace these concatenations in the encoder with MCB. We use MCB to create a multimodal embedding of the question and attended history and use this multimodal embedding to produce image attention weights. We also use MCB to combine the question embedding with the attended image features and concatenate this question-image embedding with the attended history representation for the final embedding given as input to the rest of the model. We keep the final concatenation with the attended history representation since MCB can combine only two modalities at once. Although we also experimented with applying MCB a second time to combine all three modalities, we found that this approach of applying MCB twice in a row did not lead to better performance and may have introduced too much complexity in the data embedding.

3.2.2 Parameter Learning

We use the n-pair loss function developed by Sohn [16].

$$L_D = L_{n\text{-pair}}(\{e_t, a_t^{gt}, \{a_{t,i}^-\}_{i=1}^{N-1}\}, f) = \log(1 + \sum_{i=1}^{N-1} \exp(e_t^T f_t(a_{t,i}^-) - e_t^T f_t(a_t^{gt})))$$

This is a multi-class loss function for the answer embedding f. Our network takes in the embedding e_t of the current question, history, and image generated using MCB and outputs an answer embedding function f that uses an LSTM. This loss function encourages the output answer embedding function to make the similarity of e_t with the embedding ground truth answer $f_t(a_t^{gt})$ greater than the similarity of e_t with the other incorrect candidate answers $f_t(a_{t,i}^-)$.

The gradients of this loss function are backpropagated to the parameters in the MCB layers according to the architecture of the discriminator network.

4 Experimental Setup

Our main research hypotheses are the following:

1. Focusing on semantic similarity, not lexical similarity, should help our model answer questions that have multiple correct answers.
2. Using improved fusion methods like MCB should produce better input representations that should improve model performance.

4.1 Dataset

The v0.9 dataset consists of a training and validation set. For the training set, 82,783 images taken from the COCO 2014 train set. For each image, there one dialog, consisting of 10 rounds of questions. Each image is annotated with a caption. The v0.9 validation set is similar, containing 40,504 images taken from COCO 2014 validation set.

For our experiments, we divide the training set into 80,000 images for training and 2,783 images for validation. We then use the v0.9 validation set as our testing set, reporting results on this set.

4.2 Baseline Model

4.2.1 Similarity Metrics

We use the late fusion model introduced by Das, et al.[4] as our baseline for our similarity metric loss function. This model uses an encoder-decoder architecture.

The discriminative decoder takes in the input encoding $E(Q_t, I, H)$ and an LSTM encoding for each of the answer options and then computes dot products between the input and answer encodings. These dot products are put through a softmax layer to convert them into probabilities for each option.

The late fusion encoder encodes each modality separately before passing the encodings through a concatenation and final linear layer to produce the joint representation. The dialog history H is constructed as the concatenation of each individual round of question/answer with the caption. The Q_t, H are each encoded using an LSTM.

4.2.2 MCB

We use the discriminative model with the HCIAE encoder trained with the n-pair loss (HCIAE-D-NP) developed by Lu, et al., [12] as our baseline model for using MCB in Visual Dialog. This model has near state-of-the-art performance for Visual Dialog, so we wanted to build on top of this model to see whether MCB could improve performance even further.

The model uses the HCIAE encoder, which we described in Section 3.2.1 and explain further here. The model first creates an LSTM encoding of the question, an LSTM encoding of the dialog history, and the convolutional features of the image generated by VGG-19. These embeddings are then passed to the HCIAE encoder. The HCIAE encoder creates an attended representation of history conditioned on the question embedding. The encoder also creates an attended representation of the image conditioned on the concatenation of the question embedding and attended history embedding. The encoder then outputs the concatenation of the question embedding, attended history embedding, and the attended image embedding.

The input embedding created by the encoder is used to learn an answer embedding space defined by an embedding function f. The discriminative model then uses the n-pair loss function described in section 3.2.2 to update the learned embedding space to encourage the ground-truth answer to be scored higher than the other candidate answers.

4.3 Experimental Methodology

4.3.1 Evaluation Metrics

We use the following evaluation metrics originally used by Das, et al. [4]. Our discriminative models rank candidate answers for the current question, so we use mean rank, mean reciprocal rank, and recall metrics. We also introduce additional metrics based on semantic similarity.

Mean Rank (Mean), Mean Reciprocal Rank (MRR) The mean rank our model assigns to the ground truth answer (lower is better) and the mean reciprocal of the rank assigned to the ground truth answer (higher is better).

Recall @ k The percent of questions where our model ranks the ground truth answer as one of the top k answers.

Semantic Similarity-Based Metrics We use cosine similarity based on GloVe embedding averages as well as word's mover distance as a performance metric.

Sentence Similarity We also use BLEU and ROUGE_L, which are both non-semantic sentence similarity metrics.

4.3.2 Similarity Metrics

For the late fusion model, we use the same hyperparameters and architecture choice used by Das, et al. [4]. We use a word embedding size of 300 and a LSTM hidden state size of 512. For training, we use the Adam optimizer with learning rate 10^{-3}, decay rate 0.999759, and minimum learning rate $5 \cdot 10^{-5}$. We train with a batch size of 12 for 20 epochs.

Our loss function introduces a new hyperparameter λ that adjusts the sharpness of the labels such that higher λ results in labels closer to the original hard labels. We present results for $\lambda = 20$, chosen by performance on the validation set.

Model	MRR	R@1	R@5	R@10	Mean	Cosine-Sim	WMD	BLEU	ROUGE_L
LF-Baseline	0.5981	45.56	76.73	85.70	5.44	0.78	2.88	0.47	0.49
LF-Sim	0.5813	44.32	73.79	82.66	6.46	0.77	2.96	0.45	0.48
HCIAE-D-NP-Baseline	0.6187	48.09	78.44	87.28	4.89	0.79	2.73	0.51	0.52
HCIAE-D-NP-MCB	0.6149	47.79	77.82	86.65	5.07	0.79	2.75	0.51	0.53

Table 1: Results with similarity metrics. For both ranking-based and similarity-based metrics, our introduced models perform similar or slightly worse on those metrics.

Model	N	MRR	R@1	R@5	R@10	Mean
LF-Baseline-Color	3144	0.4113	26.56	57.57	72.71	8.95
LF-Baseline-Count	2035	0.3248	17.40	48.89	67.08	10.04
LF-Baseline-Yes/No	16988	0.7476	61.08	91.25	95.06	2.90
LF-Baseline-Other	7833	0.4197	26.82	60.15	75.46	8.50
LF-SIM-Color	3144	0.3853	24.65	53.09	67.18	10.47
LF-SIM-Count	2035	0.3049	15.08	48.21	64.86	11.80
LF-SIM-Yes/No	16988	0.7527	61.99	90.45	94.71	2.96
LF-SIM-Other	7833	0.3598	21.50	52.64	67.37	11.03

Table 2: Results categorized by question type on 30,000 question answer pairs. The late fusion model trained with semantic similarity did not achieve better performance than the baseline model across all categories.

4.3.3 MCB

To train the HCIAE-D-NP model modified to use MCB, we use the same hyperparameters used by Lu, et al. [12]. The only architecture modifications we made to the baseline HCIAE-D-NP model were the additions of MCB to the HCIAE encoder. The rest of the network architecture remains unchanged.

All LSTMs in the network have a single layer and 512 hidden units. For training, we use the Adam optimizer with base learning rate 4×10^{-4}, decay rate 0.8 for the first moment of the gradient, and decay rate 0.999 for the second moment of the gradient. We train with batch size 100 for 20 epochs.

5 Results and Discussion

In this section, we discuss the performance of our late fusion model trained with similarity metrics and the HCIAE-D-NP model modified with MCB relative to their respective baseline models.

5.1 Similarity Metrics

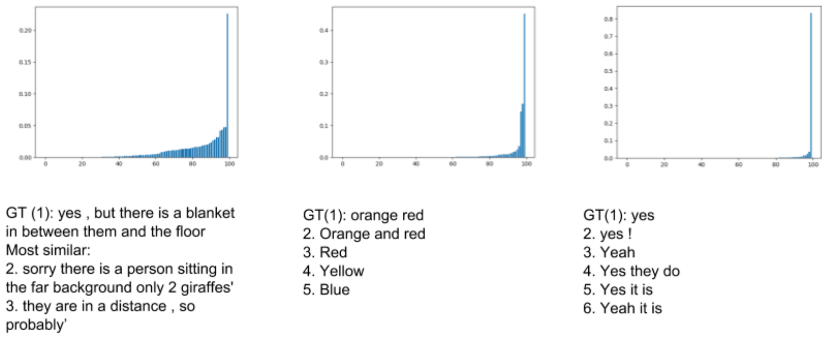
5.1.1 Performance of Similarity Metric Model

Overall, we see similar or slightly worse performance on the original evaluation metrics between the baseline late fusion model and late fusion with semantic similarity as shown in Table 1. Furthermore, the similarity model performs slightly better on Yes/No questions and performs worse on all the other types of questions as shown in Table 2. We also see similar or slightly worse performance on the similarity-based metrics. The latter is unexpected, and we discuss some potential issues with similarity metrics that might explain the results.

5.1.2 Analysis of similarity metrics

We analyze the distribution of cosine similarity scores in our dataset, which can be seen in Appendix Figure 4. Given that each question has around 100 candidate answers, on average, we expect a few candidate answers besides the ground truth to have similarity score above 0.9.

We identify several issues with the similarity metric used by the model to generate soft labels. We provide a few examples of soft label values for $\lambda = 20$, shown in Figure 2. We present two cases where the soft labeling does not properly capture semantic similarity and one where it does. In the first example in Figure 2, while the soft labeling suggests many similar answers, the most similar answers

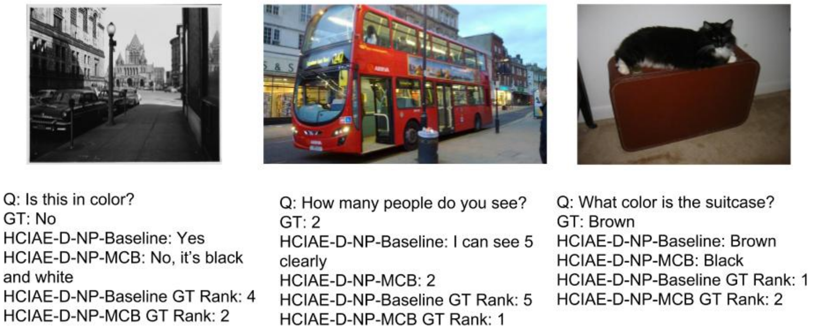


GT (1): yes , but there is a blanket in between them and the floor
Most similar:
2. sorry there is a person sitting in the far background only 2 giraffes'
3. they are in a distance , so probably'

GT(1): orange red
2. Orange and red
3. Red
4. Yellow
5. Blue

GT(1): yes
2. yes !
3. Yeah
4. Yes they do
5. Yes it is
6. Yeah it is

Figure 2: Success and failure cases in soft labeling. The soft label of each of the 100 candidate answers (including the ground truth) is plotted in sorted order from lower to highest.



Q: Is this in color?
GT: No
HCIAE-D-NP-Baseline: Yes
HCIAE-D-NP-MCB: No, it's black and white
HCIAE-D-NP-Baseline GT Rank: 4
HCIAE-D-NP-MCB GT Rank: 2

Q: How many people do you see?
GT: 2
HCIAE-D-NP-Baseline: I can see 5 clearly
HCIAE-D-NP-MCB: 2
HCIAE-D-NP-Baseline GT Rank: 5
HCIAE-D-NP-MCB GT Rank: 1

Q: What color is the suitcase?
GT: Brown
HCIAE-D-NP-Baseline: Brown
HCIAE-D-NP-MCB: Black
HCIAE-D-NP-Baseline GT Rank: 1
HCIAE-D-NP-MCB GT Rank: 2

Figure 3: Examples of success and failure cases for HCIAE-D-NP model modified with MCB

to the ground truth actually have very little semantic similarity. This is likely due to our approach of taking the average of word embeddings, which will tend to result in long answers all averaging to a similar embedding. The second example is one that does capture semantic similarity, where colors close to orange red have soft labels of > 0.1 and remaining colors are given relatively small soft labels. The final example is one where there are many similar answers that should be reflected in the soft labels but are not. This is most likely an artifact of the underlying word embedding used (GloVe embeddings), where despite being semantically similar, the word embeddings are not considered similar enough due to insufficiently high co-occurence.

5.2 MCB

5.3 Analysis of performance of model modified with MCB

The HCIAE-D-NP model modified to use MCB did not achieve noticeably better results than the baseline HCIAE-D-NP model. In each category of questions (counting, color, yes/no), we observe similar performance metrics for both models (Table 3). For most of the metrics, the model modified with MCB has slightly worse performance than the baseline model, but only by very small decimal differences. The similarity metric performance of the model with MCB is also almost exactly the same as the performance of the original HCIAE-D-NP model on similarity metrics (Table 1). Any difference in the performance metrics is quite small, so adding MCB to the model does not seem to improve or degrade performance. This implies that replacing concatenation with an MCB layer in the network architecture is not sufficient to improve performance for the Visual Dialog task.

Model	N	MRR	R@1	R@5	R@10	Mean
HCIAE-D-NP-Baseline-Color	15943	0.4466	29.82	61.82	76.84	7.67
HCIAE-D-NP-Baseline-Count	10589	0.3311	17.17	50.98	68.95	9.52
HCIAE-D-NP-Baseline-Yes/No	89879	0.7416	61.60	89.22	93.90	3.09
HCIAE-D-NP-Baseline-Round 1	15000	0.6303	51.31	76.86	85.56	5.16
HCIAE-D-NP-Baseline-Round 4	15000	0.5977	45.39	77.32	86.41	5.09
HCIAE-D-NP-Baseline-Round 7	15000	0.6234	48.29	78.99	87.95	4.74
HCIAE-D-NP-Baseline-Round 10	15000	0.6577	52.10	81.89	89.67	4.27
HCIAE-D-NP-MCB-Color	15943	0.4269	28.15	58.93	73.82	8.32
HCIAE-D-NP-MCB-Count	10589	0.3268	16.76	50.62	68.60	9.66
HCIAE-D-NP-MCB-Yes/No	89879	0.7431	61.97	89.21	93.81	3.13
HCIAE-D-NP-MCB-Round 1	15000	0.6306	51.61	76.49	85.37	5.24
HCIAE-D-NP-MCB-Round 4	15000	0.5946	45.33	76.27	85.77	5.27
HCIAE-D-NP-MCB-Round 7	15000	0.6200	48.05	78.59	87.26	4.91
HCIAE-D-NP-MCB-Round 10	15000	0.6511	51.41	81.29	88.89	4.49

Table 3: Results categorized by question type to compare the baseline HCIAE-D-NP model and the HCIAE-D-NP model modified with MCB over 150,000 question-answer pairs. The model with MCB did not achieve better performance than the baseline model across most categories of questions.

5.3.1 Qualitative analysis

When going through validation instances, we observe that the HCIAE-D-NP model modified with MCB does sometimes result in better answers than the baseline HCIAE-D-NP model. There are cases where the model with MCB ranks the ground-truth answer as the most likely answer and the baseline model does not (e.g. the second example in Figure 3). There are even cases where the model with MCB ranks topmost an answer that seems better than the ground-truth answer and the baseline model outputs the wrong answer (e.g. the first example in Figure 3). However, in numerous other cases, the model with MCB ranks topmost an incorrect answer, while the original HCIAE-D-NP model ranks topmost the correct ground-truth answer (e.g. the last example in Figure 3).

6 Conclusion and Future Directions

We explored applying similarity metrics and a VQA fusion method to the Visual Dialog dataset. We trained an existing Visual Dialog model using a loss function based on a soft labeling that depends on semantic similarity between a candidate answer and the ground-truth answer. Similarity metrics were also used to evaluate the performance of existing models and our current models. Our second main contribution is applying the VQA fusion method MCB to the Visual Dialog task. Although neither of our models achieved statistically significantly improved results over baseline models, based on observed errors, we propose several ideas to extend our current work.

For soft labels to work, the labels themselves must first be reasonably accurate. Rather taking an average of a pretrained word embedding for each answer, it is necessary to finetune an existing metric or create a new similarity metric that better measures semantic similarity between answers relative to a given question. In addition, training our similarity metric with the model itself might result in a similarity function that better captures meaningful differences in the context of answering visual dialog question. Similarly, fusion methods should be developed to also incorporate dialog history in a better way. The key difference between VQA and Visual Dialog is the addition of previous question-answer pairs, and existing VQA fusion methods may not effectively combine the three modalities of the dialog history, image, and current question.

References

[1] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra. Vqa: Visual question answering. *International Journal of Computer Vision*, 123(1):4–31, 2017.

[2] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 3, 2017.

[3] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh. Evaluating visual conversational agents via cooperative human-AI games. In *Human Computation and Crowdsourcing*, 2017.

[4] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *International Conference on Computer Vision*, 2017.

[6] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[7] 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. In *Empirical Methods in Natural Language Processing*, 2016.

[8] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. Petrakis, and E. Milios. Information retrieval by semantic similarity. *International journal on semantic Web and information systems (IJSWIS)*, 2(3):55–73, 2006.

[9] S. Kottur, J. M. Moura, D. Parikh, D. Batra, and M. Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–169, 2018.

[10] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[11] Z. Liu, Y. Shen, V. Bharadhwaj Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[12] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *Advances in Neural Information Processing Systems*, pages 314–324, 2017.

[13] J. Mitchell and M. Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

[14] E. ShafieiBavani, M. Ebrahimi, R. Wong, and F. Chen. A graph-theoretic summary evaluation for ROUGE. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 762–767, 2018.

[15] A. F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 174–180. ACM, 1996.

[16] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.

[17] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courvile, and O. Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2765–2771. AAAI Press, 2017.

7 Appendix

7.1 Similarity Score Figures

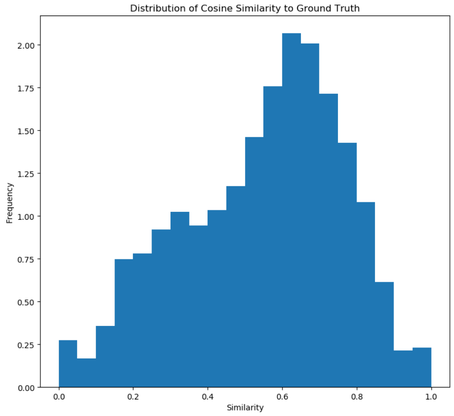
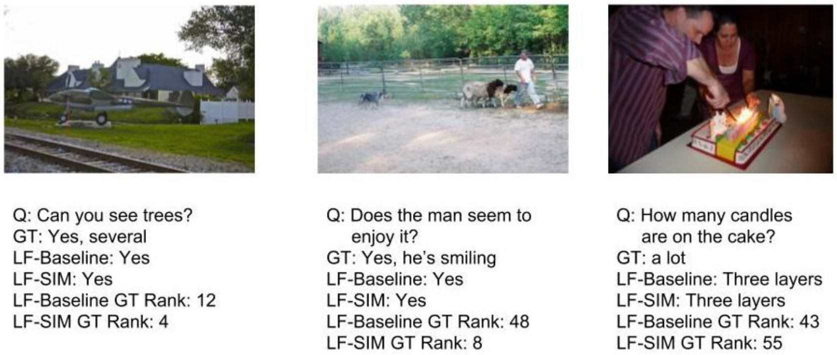


Figure 4: Similarity score distribution



Q: Can you see trees?
GT: Yes, several
LF-Baseline: Yes
LF-SIM: Yes
LF-Baseline GT Rank: 12
LF-SIM GT Rank: 4

Q: Does the man seem to enjoy it?
GT: Yes, he's smiling
LF-Baseline: Yes
LF-SIM: Yes
LF-Baseline GT Rank: 48
LF-SIM GT Rank: 8

Q: How many candles are on the cake?
GT: a lot
LF-Baseline: Three layers
LF-SIM: Three layers
LF-Baseline GT Rank: 43
LF-SIM GT Rank: 55

Figure 5: Qualitative examples for late fusion vs. late fusion with semantic similarity models

7.2 Fusion Figures

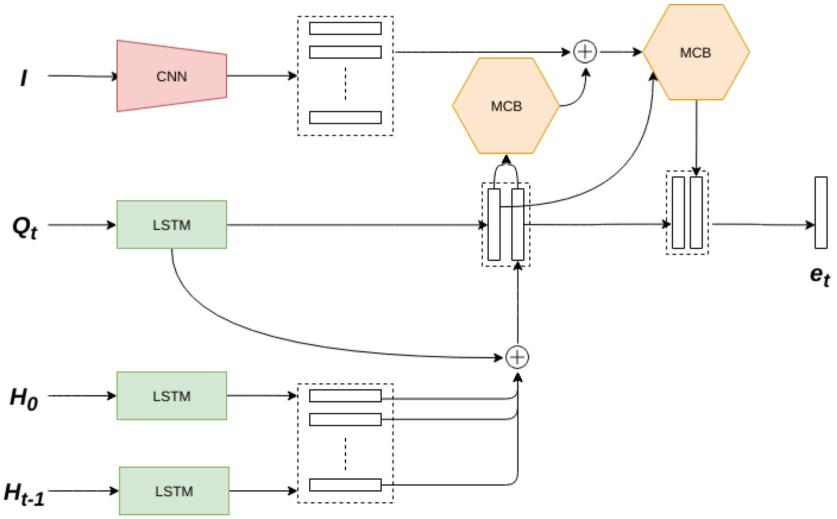


Figure 6: Model architecture for HCIAE-D-NP-MCB