VISUAL DIALOG

Gauri Pradhan [1] **Rutuja Moharil** [2] **Shruti Sinha** [3]

Abstract

We believe that the next generation of intelligent systems will need to possess the ability to hold a dialog about visual content for a variety of applications. We thus present the task of Visual Dialog with this project, which requires an AI agent to hold a meaningful dialog with humans in natural, conversational language about visual content.

1. Introduction

Visual dialog is a challenging vision-language task, which requires the agent to answer multi-round questions about an image. It typically needs to address two major problems: (1) How to answer visually-grounded questions, which is the core challenge in visual question answering (VQA); (2) How to infer the co-reference between questions and the dialog history. Specifically, it'll work as follows: Given an image and a dialog history associated with it, and a question about the image, the agent has to ground the question in image, the model(s) to be used will infer context from history and answer the question as accurately as possible. A possible example of the end interface with the Visual Dialog- aided AI agent is shown in figure (1). This project

Figure 1. VisDial-aided AI agent's dialoge with another anonymous agent

has potential use in development of visual intelligence systems which aims to enhance user- machine interactions in visual context (an example of practical applications are intelligent AI assistants, aiding visually impaired people in understanding their surroundings).

1.1. Contributions

The Visual Dialog dataset was trained with a Late Fusion encoder and Discriminative decoder. We experimented with different word embeddings for the textual inputs i.e Questions, Answers and Captions.

1) default word2vec nn embedding (Baseline)
2) BERT embedding
3) Glove embedding

Among the three, glove embeddings gave the best performance, while BERT performed the poorest on a training set of 10,000. Using our results we draw comparisons between the embedding techniques and bolster the performance results using evaluation metrics in section 5.5.

2. Background

The Visual Dialog dataset is curated from a conversation between two agents : Agent1 and Agent2. Agent1 is the questioner in this scenario and has to ask questions regarding the image based on the image caption available to it, while the actual image remains hidden. On the other hand, Agent2 can see the image alongwith the caption and is tasked with answering questions posed by the questioner based on the image. This conversation in the form of questions and answers and the caption comprise the dialog history.
Validation of visual-dialog models is harder than general image based networks as the answers or dialogs are more open ended relatively. Here are some of the notations we would be using throughout the report.

- Image $= I$
- Dialog history $H = (C, (Q_1, A_1), ..., (Q_{t-1}, A_{t-1}))$
- Question Q_t
- Candidate answers $A_t = A_t^{(1)}, ..., A_t^{(100)}$

3. Related/Prior Work

Lately, there has been significant interest in the area intersecting vision and language through - Visual ques answering(VQA)[4], visual storytelling, image captioning among others. There has been a lot of work involving chatbots and conversational modeling[3]. We are using Visual

dialog paper by Abhishek Das et.al [1], as our inspiration. Their work enables the machine to answer the question by first comprehending the history of the past dialog and then understanding the image to give the most appropriate answer. Another research done by Geman et al.[2], proposed a visual binary question test, compared to a more free-form language question answering experiment by Das [1].

4. Approach

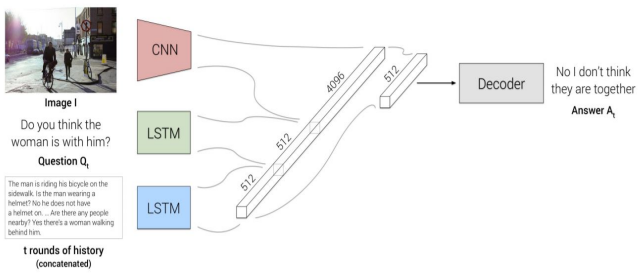


Figure 2. Late Fusion Encoder Architecture

Here we develop neural Visual Dialog models with inputs as the image (I) , ground-truth dialog history (H) and the image Question(Q_t).
Broadly our model consists of two major components :

1. *Encoder network*: Encode individual contributions of the image(VGG16 feature extraction), history and question into a single joint representation in the vector space.
2. *Decoder Network* : This part of the network decodes the latent space representation into an output.

Every dialog for the associated image (in training and validation) in dataset is supported with 100 answer candidates. At any given time t, given input image I, current question Qt, dialog history H = (C,(Q_1, A_1), . . . ,(Q_{t1}, A_{t1}), and 100 candidate answers $A_t = A(1)_t$,. . , $A(100)_t$, the task is to arrive at an answer by ranking candidate answers At. We draw inspiration from [1] for network architectures and plan to incorporate different embeddings for word representations in the encoder architecture.

Embeddings Comparison: Word embeddings are dense vector representations of words in lower dimensional space. Word2Vec is a predictive embedding model that uses the current word to predict the surrounding window of context words[5]. Word2Vec does not take advantage of global context. GloVe embeddings by contrast leverage the same intuition behind the co-occuring matrix used distributional embeddings, but uses neural methods to decompose the co-occurrence matrix into more expressive and dense word vectors.BERT is a dynamic attention based model that incorporate context, handling polysemy and nuance much better. In the context of this report we will only touch the feature extraction side of BERT by just obtaining ELMo-like word embeddings from it.

5. Experimental Results

5.1. Dataset

Our project is evaluated on the recently released real-world dataset VisDial v1.0. The training set of VisDial contains 123K images from Coco dataset with 10 rounds of dialog for each image, totalling about 1.2M dialog pairs. The validation sets were collected from Flickr, with 2K COCO-like images. An interesting category of answers emerge like 'I think so', 'I can't tell', or 'I can't see' which depict uncertainty or lack of information.
This is a direct consequence of the fact that one of the agent is not shown the image and hence not all questions are answerable with certainty. We can expect answers to be free-form and longer.
This allows a more human-like answer approach which is desirable in conversational AI.

5.2. Preprocessing

We followed the process introduced by [1]. Firstly, we lowercased all letters in sentences, converted digits to words and removed contractions. After that, we used Python NLTK for tokenizing, followed by padding captions, questions and answers. A dictionary of words that appear at least 5 times in the training set is constructed, giving a vocabulary of around 11k.
The final dataset created (to feed into the network) comprises of following fields: image id, the ground truth answer, dialog (history, caption, question), lengths(history, question), answer options and image features.

5.3. Network architecture

We have employed Late fusion encoder model and the architecture is described below.
Encoder model :

1. An embedding layer for the questions and history (separately) is initialized. Embedding dimension of 300 is used in all cases. The question embedding layer and history embedding layer are passed separately, through a custom function to obtain right padded sequence to ensure consistency in sequence lengths.
2. We use one LSTM to encode the 10 rounds of dialog history ($H_0, ...H_9$) and another LSTM to encode the question.
   An input layer of dimension 300 and hidden layer of

dimension 512 is used in both the LSTMs. The image features from VGG16 are projected to a size of 512. The projected features are passed through a softmax layer followed by a tanh non-linearilty to obtain the encoder output.

The term "fusion" of the encoder is due to the concatenation or fusion of fully connected Q, H and I .

Decoder model :
We employ a discriminator based decoder model. Given an encoder output and candidate option sequences, the decoder predicts a score for each option sequence.
The decoder computes dot product similarity between the encoder output and LSTM output of each of the answer options which is fed into a softmax layer which serves as a score.
During training, we maximize the log-likelihood of the correct option, while at test/evaluation time, options are simply ranked based on the probability scores.

5.4. Training

We used softmax cross-entropy loss to train the model and Adam optimizer with a learning rate of 0.001. Dropout was also applied with rate of 0.5. We trained our model for 10 epochs on 16 GB NVIDIA Tesla K80 GPU.

5.5. Evaluation Metric

The scores generated by decoder are further converted to an efficient ranking using the following evaluation techniques:

Retrieval metrics / evaluation using sparse annotations: We have mean reciprocal rank (MRR), recall (R@1, 5, 10), and mean rank as described in the Visual Dialog paper[(1)].
Specifically, at test/validation time, the system is given an image I, the 'ground-truth' dialog history (including the image caption) $C, (Q1, A1), ..., (Qt1, At1)$, the question Qt, and a list of $N = 100$ candidate answers, and asked to return a sorting of the candidate answers. The model is evaluated on retrieval metrics
(1) rank of human response (lower is better)
(2) recall@k, i.e. existence of the human response in top-k ranked responses, and
(3) mean reciprocal rank (MRR) of the human response (higher is better).

Evaluation using dense annotations- As some of the candidate options may be semantically identical (e.g. 'yeah' and 'yes'), we used answers from five human annotators indicate whether each of the 100 candidate answers is correct for each val and test phase instance. For evaluation, we report the normalized discounted cumulative gain (NDCG) over the top K ranked options, where K is the number of answers marked as correct by at least one annotator. It is widely used for measuring ranking quality. For this computation, we consider the relevance of an answer to be the fraction of annotators that marked it as correct. NDCG requires to rank relevant candidates in higher places, rather than just to select the ground-truth answer.
We use only the non-zero relevance scores for NDCG calculation in our evaluation. NDCG is invariant to the order of options with identical relevance and to the order of options outside of the top K which makes the metric consistent across all queries.

5.6. Results

Model Metric	**Word2Vec LFE**	**Glove LFE**	**BERT LFE**
MRR	0.45568	0.48992	0.38018
R@1	0.32136	0.35038	0.37369
R@5	0.59801	0.63992	0.46041
R@10	0.71114	0.75339	0.51138
Mean	10.4809	8.71996	21.5723
NDCG	0.42436	0.44719	0.40475

Table 1. Metric Evaluation

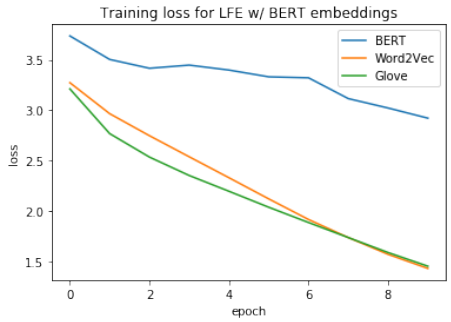


Figure 3. Loss vs Epoch plot

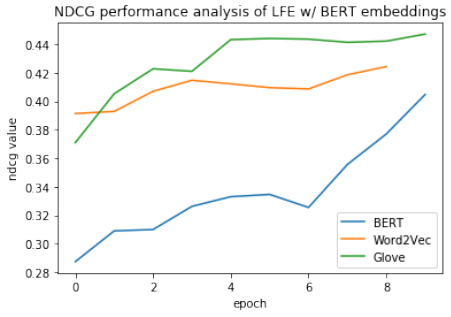


Figure 4. Performance Analysis: NDCG vs Epoch plot

Referencing figure (3), we can see that loss decreased almost linearly with epochs.

With every new epoch, the NDCG for the train set increased, with the 10th epoch achieving 42.4. This shows that our model has a decent performance for the given subset of 10k of the original dataset of size 123k. This performance is comparable to the baseline 51.2 seen from the paper of [1]. We can evidently observe from these plots that performance using pre-trained Glove embeddings is higher and BERT performance is lower than the baseline.

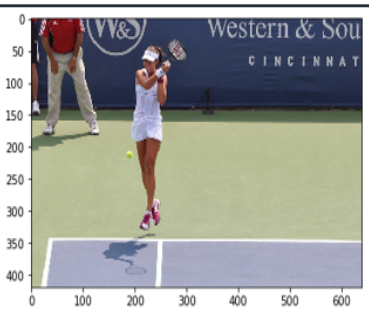
Glove results:
As mentioned in section 5.1, there exists many answer options like "I can't tell","I can't really tell", "can't tell". These are more likely to have the same numeric representation for "can't" ,"tell" regardless of where the words occurred in the answer and hence options where these words occur would have higher scores.

BERT results:
The reason for a poor performance could be two-fold.

1. We employed a custom tokenizer as opposed to a BERT tokenizer and used a pre-trained BERT embedding.

2. As it is a context based encoding method, it might be safe to assume that due to the nature of higher number of negative connotations in answer options, the model cannot generate good scores. For eg answer options like "No", "no birds", "no living things", "I can't see anyone", "I don't know", "Can't tell " have similar latent meaning which leads to inaccurate scoring and ranking.

Figure (5) is a sample image from the validation dataset selected at random on the GloVe embedding model. We chose to display top 5 answer options for 3 dialog rounds. For question 1 the model evaluates constrasting answer options as top 5 ranks. For question 2 the model evaluates options with colors and the correct answer in the top 5 ranks. For question 3 the model is able to identify very similar answers and actually performs reasonably well for this question with options like "no people in sight" ,"can barely see a player".



Caption: a woman playing tennis with the ball in front of a sponsored grandstand

Question_1: do you see a lot of people
Answer_1: just 3
Top 5 Answer options
['no there is no people' 'yes' 'yes a wall' '3' 'no just the 2 of them']

Question_2: what color is her tennis racket
Answer_2: black
Top 5 Answer options
['red, blue and black' 'black' 'i am unable to tell what color it is' 'yes' 'it's blue and yellow']

Question_3: do you see another player
Answer_3: no
Top 5 Answer options
['no people in sight' 'can barely see a player' 'no' 'can barely see a player at third just legs' '1 person behind a pole']

Figure 5. Sample Validation input

6. Discussion

Glove models leverage similarity metrics for word vector calculation and is context independent. Thus Glove performs very well on word analogy tasks[7] which is evident from our analysis. On the other hand BERT is heavily context dependent and learns relationships between sentences, which is why it might work better for a general VQA task (where subject has seen the image and asks more specific queries regarding details) [6] than Visual Dialog which has more uncertain and free form answer format.

In our opinion a possible future option would be to train on the entire dataset for a more crystallised analysis of NDCG. Moving forward, experimenting with potential causal relations embodied in QA alongside different image feature extraction networks (MaskRCNN ,ResNet) and even different neural network models (Attention networks) could be explored.

In this project, we presented a performance comparison on different set of embeddings on Visual Dialog. The overall findings and the intricacies entailed in the dataset indicate the potential of human-like AI conversational systems in commercial and mainstream applications in the coming years.

References

[1] *Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra*. Visual dialog. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, 2017

[2] *D. Geman, S. Geman, N. Hallonquist, and L. Younes*. A Visual Turing Test for Computer Vision Systems. In PNAS, 2014.

[3] *A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, et al*. Smart Reply: Automated Response Suggestion for Email. In KDD, 2016.

[4] *Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick and Devi Parikh* 2015. VQA: Visual Question Answering,

[5] *pytorch.org/tutorials/beginner/nlp/word-embeddings-tutorial.html*

[6] *https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html*

[7] *https://nlp.stanford.edu/projects/glove/*