Supplementary Material: Game-Based Video-Context Dialogue

Ramakanth Pasunuru and **Mohit Bansal**
UNC Chapel Hill
{ram, mbansal}@cs.unc.edu

1 Simple Baselines

1.1 Most-Frequent-Response

As a simple non-trained retrieval baseline, we just return (or rank) the responses (in the retrieval ranking list; see supplementary Sec. 2) based on their frequency in the training set.

1.2 Chat-Response-Cosine

We also use another simple non-trained baseline following previous work (Lowe et al., 2015), where we choose/rank the candidate responses based on the cosine similarity between the vector representations of the given chat context and each candidate response (in the retrieval list). We train an LSTM-RNN language model on the Twitch training chat context data. We then use the final hidden state of this pretrained RNN to represent the given chat context/response.

1.3 Nearest Neighbor

We also use a nearest neighbor non-trained baseline similar to Das et al. (2017), where, given the chat context, we find the K-best similar chat contexts in the training set and take their corresponding responses. Next, we again rerank the candidate responses based on the mean similarity score between the candidate response and these K nearest-neighbor responses. Here again, we use a pretrained Twitch-LM RNN for vector representation of chat contexts and responses.

1.4 Logistic Regression and Naive Bayes

Apart from the non-trained baselines, we also present simple trained baselines based on logistic regression and Naive Bayes. Here again, we use the pretrained RNN for chat context and response vector representations.

2 Experimental Setup

2.1 Evaluation

We first evaluate both our discriminative and generative models using retrieval-based recall@k scores, which is a concrete metric for such dialogue generation tasks (Lowe et al., 2015). For our discriminative models, we simply rerank the given responses (in a candidate list of size 10, based on 9 negative examples as described below) in the order of the probability score each response gets from the model. If the positive response is within the top-k list, then the recall@k score is 1, otherwise 0, following previous Ubuntu-dialogue work (Lowe et al., 2015). For the generative models, we follow a similar approach, but the reranking score for a candidate response is based on the log probability score given by the generative models' decoder for that response, following the setup of previous visual-dialog work (Das et al., 2017). In our experiments, we use recall@1, recall@2, and recall@5 scores. For completeness, we also report the phrase-matching metric scores: METEOR (Denkowski and Lavie, 2014) and ROUGE (Lin, 2004) for our generative models. Because dialogue models are hard to evaluate using such phrase-matching metrics (Liu et al., 2016), we also perform human evaluation based comparison for our two strongest generative models (Seq2seq+attention and Seq2seq+attention+BiDAF).

2.2 Negative Samples (Training and Val/Test)

Both our discriminative and generative models need negative samples during training and as well as for the test time (dev/test) retrieval lists for recall@k scores. For training, for every positive triple (video, chat, response) in the training set, we sample 3 random negative triples elsewhere from the training set such that the negative sample

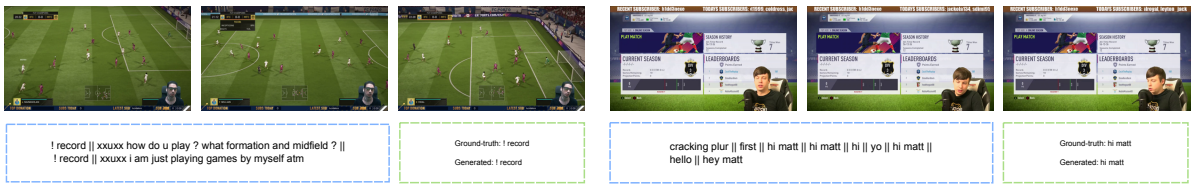

Figure 1: Output generative examples from our BiDAF model.

triples do not come from the video corresponding to the positive triple. We refer Sec. 4.2.1 (of main paper) for details about how we add these negative samples in the training. For validation/test, we sample 9 random negative responses elsewhere from the validation/test set for the given video and chat context, such that they don't come from the video corresponding to the positive response, so as to create a 10-sized retrieval list.

2.3 Training Details

For tokenizing the word-level utterances in the chat context and response, we use the Twitter Tokenizer from NLTK library[1]. All the video clips are down-sampled to 3 fps (frames per second) and also cropped to 244×244 size. Further, we extract the inception-v3 (Szegedy et al., 2016) frame-level features (standard penultimate layer) of 2048 dimension from each video clip and use it as input to our video context encoder. For all of our models, we tune the hyperparameters on the validation set. Our model selection criteria is based on recall@1 for discriminative model and METEOR for the generative model. We set the hidden unit size of 256 dimension for LSTM-RNN. We down-project the inception-v3 frame level features of size 2048 to 256 dimension before feeding it as input to the video context encoder. We use word-level RNN model with 100 dimension word embedding size and a vocabulary size of $27,000$. We initialize the word embeddings with Glove (Pennington et al., 2014) vectors. We unroll the video context LSTM to a maximum of 60 time steps and the chat context to a maximum of 70 time steps. For the response, we unroll the RNN to a maximum of 10 time steps. All of our models use Adam (Kingma and Ba, 2015) optimizer with a learning rate of 0.0001 (unless otherwise specified) and a batch size of 32. Also, we use gradient clipping with maximum clip norm value of 2.0. We set the margin M in our max-margin loss to 0.1 for all models. We use $\lambda = 1.0$ for the weighted loss in the generative model.

[1] http://www.nltk.org

3 Output Examples

Fig. 1 presents additional examples for output responses generated by our BiDAF model.

References

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *CVPR*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826.