A Data Collection

A.1 Datasets

We collect human judgments of correctness for two GenQA datasets, MS-MARCO (Bajaj et al., 2016) and AVSD (Alamri et al., 2019). We describe the properties of each dataset in this section.

MS-MARCO MS-MARCO is a large-scale english machine reading comprehension dataset that provides ten candidate passages for each question. The model should consider the relevance of the passages for the given question and answer the question. One of the main features of this dataset is that it contains free-form answers that are abstractive. MS-MARCO provides two tasks, Natural Language Generation (NLG) task and Q&A task. For the NLG task, the model should generate an abstractive summary of the passages for given questions, which is a well-formed answer rather than an answer span in the passage. Although the Q&A task also provides some abstractive answers, most of the answers are short and do not contain the context or rationale of the question. Hence, we use the NLG subset of MS-MARCO dataset as a GenQA dataset to study the metrics for GenQA. Also, we use the training set of Q&A subset to train and evaluate KPQA, since most of the samples in this subset has exact answer spans in the passage like SQuAD.

Audio Visual Scene-aware Dialog (AVSD) To study more general metrics for GenQA, we also use a multimodal GenQA dataset for our work. Audio Visual Scene-aware Dialog (AVSD) is a multimodal dialogue dataset composed of QA pair about Charades videos. Although the name of the dataset contains dialog, all of the dialog pairs are composed of questions answering about a video. The task of this dataset is to generate an answer for a question about a given video, audio, and the history of previous turns in the dialog. In other words, this task is to generate a free-form answer for a given multimodal context, which can be considered as GenQA.

A.2 Instructions to Annotators

The full instructions to annotators in MTurk are shown in Figure 1. We hire the annotators whose HIT approval rate are higher than 95% and pay $0.03 for each assignment.

Dataset	Model	BLEU-1	ROUGE-L
MS-MARCO	UniLM	60.2	63.1
	MHPGM	43.7	53.9
AVSD	MTN	67.3	52.6
	AMF	62.6	48.7

Table 1: Performance of the model we trained to generate answers on develepment set of each dataset

A.3 Models

To investigate the performance of automatic metrics, we gather pairs of a sentence, {generated answer, *reference answer*}. Collecting high-quality answer candidates for a given context and question is an essential step; thus, we choose two models for each dataset from the latest research in the literature. We train two models UniLM (Dong et al., 2019) and MHPGM (Bauer et al., 2018) for MS-MARCO dataset. For AVSD dataset, we train two models MTN (Le et al., 2019) and AMF (Alamri et al., 2018). We present the performance of each model we trained in Table 1. We briefly describe the models and the training details to generate the answer for two datasets.

UniLM UniLM, which stands for unified language model pre-training, is a powerful seq2seq model based on pre-trained representations from BERT (Devlin et al., 2019). UniLM is a pre-trained transformer network that can be easily fine-tuned for NLU and NLG. UniLM achieves higher performance for various NLG tasks, such as abstractive summarization and question generation. We fine-tune UniLM for GenQA similar to the way fine-tuning UniLM to NLG, where source sequences are each question and paragraphs, the target sequence is an answer. We add [SEP] tokens between the question and each paragraph. Then, we fine-tune UniLM for 3 epochs with this setting using the public code[1].

MHPGM MHPGM, which stands for multi-hop pointer generator networks, uses multi-hop reasoning QA model that can integrate commonsense information. This model uses pointer-generator decoder to generate the answer. We train the model for three epochs with batch size 24 using the public code[2].

[1] https://github.com/microsoft/unilm

[2] https://github.com/yicheng-w/CommonSenseMultiHopQA

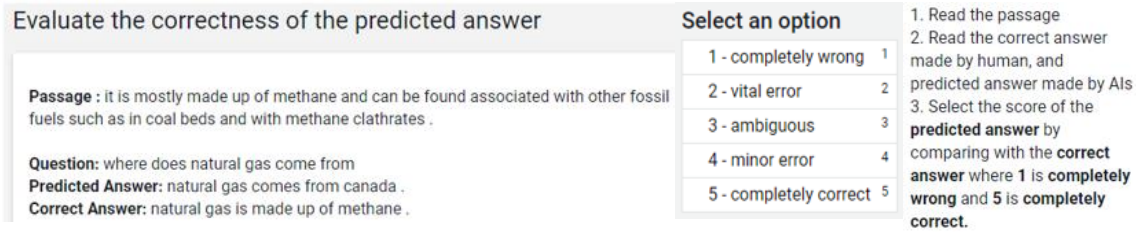

Figure 1: Instruction for MTurk workers

MTN MTN (Le et al., 2019), which is a multimodal transformer encoder-decoder framework, is a state-of-the-art model for AVSD. MTN employs multimodal attention blocks to fuse multiple modalities such as text, video, and audio. We train 10 epochs with batch size 256 and generate the answers for the testset released in the DSTC7 workshop (Alamri et al., 2018) using the publicaly available code[3].

AMF AMF is an Attentional Multimodal Fusion based model (Hori et al., 2017) introduced as a baseline system for DSTC7 AVSD workshop (Alamri et al., 2018), It is composed of RNN and multimodal attention architecture. This model encode the multimodal inputs with LSTM (Gers et al., 2000) and fuse the information with modality-dependent attention mechanism. We train this model with 15 epochs with batch size 64 using the public code[4].

B Further Experiments

B.1 Correlation by Models

The dataset we collect has human judgments on a generated answer from two models for each dataset; thus we can observe how the performance of each metric depends on the type of GenQA model. The experimental results in Table 2 show that our proposed metric outperforms other metrics in both of the GenQA models for each dataset.

C Experimental Details

In this section, we describe experimental details that are not mentioned in the previous sections including some items in the reproducibility checklist.

C.1 Reproducibility Checklist

Source Code We provide the source code for both training KPQA and computing KPQA metric as a supplementary material. We will publicly release the full source with the pre-trained model to easily compute KPQA-metric.

Computing Infrastructure We use Intel(R) Core(TM) i7-6850K CPU (3.60 GHz) with GeForce GTX 1080 Ti for the experiments. The software environments are Python 3.6 and PyTorch 1.3.1.

Average runtime for each approach Each epoch of our training KPQA on average takes 150 minutes using the single GPU. For evaluation, it takes 5 minutes.

Number of Model Parameters The number of parameters in KPQA model is about 109.4M.

Hyperparameters We use max sequence length of 256 for the inputs of KPQA. We use AdamW (Loshchilov and Hutter, 2018) optimizer with learning rate 2e-5, and mini-batch size of 16 for all of the experiments. We use *bert-base-uncased* with additional one fully-connected layer of 768 units and tanh activation function. And then we add a softmax layer after it. We train KPQA for 5 epochs and choose the model that shows the minimum evaluation loss over the development set. We repeat training 5 times for each best-performing model.

C.2 Significant Test

For all of the correlation coefficients we computed in the paper, we use a t-test using a null hypothesis that is an absence of association to report p-value, which is the standard way to test the correlation coefficient.

C.3 KPQA Performance

We present the performance of KPQA on keyphrase prediction for evaluation data in Table 3.

[3]https://github.com/henryhungle/MTN

[4]https://github.com/dialogtekgeek/AudioVisualSceneAwareDialog

Dataset	MS-MARCO				AVSD			
Model	**UniLM**		**MHPGM**		**MTN**		**AMF**	
Metric	r	ρ	r	ρ	r	ρ	r	ρ
BLEU-1	0.369	0.337	0.331	0.312	0.497	0.516	0.655	0.580
BLEU-4	0.173	0.224	0.227	0.26	0.441	0.492	0.579	0.553
ROUGE-L	0.317	0.289	0.305	0.307	0.510	0.528	0.648	0.575
METEOR	0.431	0.408	0.425	0.422	0.521	0.596	0.633	0.608
CIDEr	0.261	0.256	0.292	0.289	0.509	0.559	0.627	0.602
BERTScore	0.469	0.445	0.466	0.472	0.592	0.615	0.701	0.645
BLEU-1-KPQA	0.729	**0.678**	0.612	0.573	0.687	0.681	0.736	0.673
ROUGE-L-KPQA	**0.732**	0.667	**0.667**	0.624	0.681	0.682	0.731	**0.700**
BERTScore-KPQA	0.696	0.659	0.659	**0.655**	**0.712**	**0.703**	**0.738**	0.695

Table 2: Pearson Correlation(r) and Spearman's Correlation(ρ) between various automatic metrics and human judgments of correctness for MS-MARCO dataset and AVSD dataset. We generate the answers and collect human judgments for two models on each dataset. All of the results are statistically significant (p-value < 0.01).

Dataset	F1
SQuAD	55.81
MS-MARCO Q&A	59.26
HotpotQA	69.28

Table 3: Performance of our keyphrase predictor in development set of each dataset.

C.4 BERTScore

For computing BERTScore we use *bert-large-uncased-whole-word-masking-finetuned-squad* variant from (Wolf et al., 2019)[5] which is a BERT model fine-tuned on QA dataset SQuAD. We observe that computing BERTScore through this BERT model shows slightly higher correlation with human judgments than the BERT model without fine tuning. We use the first layer of it after the word embedding layer to compute the embedding. We experiment among different layers and found that the first hidden layer yielded the best result. We compute all of the BERTScore including original BERTScore and BERTScore variants using this BERT model.

References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, et al. 2019. Audio visual scene-aware dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7558–7567.

Huda Alamri, Chiori Hori, Tim K Marks, Dhruv Batra, and Devi Parikh. 2018. Audio visual scene-aware dialog (avsd) track for natural language generation in dstc7. In *DSTC7 at AAAI2019 Workshop*, volume 2.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471.

Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4202.

Hung Le, Doyen Sahoo, Nancy Chen, and Steven Hoi. 2019. Multimodal transformer networks for end-to-

[5] https://github.com/huggingface/transformers

end video-grounded dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5612–5623.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.