KPQA: A Metric for Generative Question Answering Using Keyphrase Weights

Hwanhee Lee[1], Seunghyun Yoon[2], Franck Dernoncourt[2]
Doo Soon Kim[3]*, Trung Bui[2], Joongbo Shin[1] and **Kyomin Jung[1]**

[1]Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]Adobe Research, San Jose, CA, USA, [3]Roku Inc., San Jose, CA, USA
{wanted1007, jbshin, kjung}@snu.ac.kr
{syoon, franck.dernoncourt, bui}@adobe.com, dskim@roku.com

Abstract

In the automatic evaluation of generative question answering (GenQA) systems, it is difficult to assess the correctness of generated answers due to the free-form of the answer. Especially, widely used n-gram similarity metrics often fail to discriminate the incorrect answers since they equally consider all of the tokens. To alleviate this problem, we propose KPQA-metric, a new metric for evaluating the correctness of GenQA. Specifically, our new metric assigns different weights to each token via keyphrase prediction, thereby judging whether a generated answer sentence captures the key meaning of the reference answer. To evaluate our metric, we create high-quality human judgments of correctness on two GenQA datasets. Using our human-evaluation datasets, we show that our proposed metric has a significantly higher correlation with human judgments than existing metrics. Code for KPQA-metric will be available at https://github.com/hwanheelee1993/KPQA.

1 Introduction

Question answering (QA) has received consistent attention from the natural language processing community. Recently, research on QA systems has reached the stage of generating free-form answers, called GenQA, beyond extracting the answer to a given question from the context (Yin et al., 2016; Song et al., 2017; Bauer et al., 2018; Nishida et al., 2019; Bi et al., 2019, 2020). However, as a bottleneck in developing GenQA models, there are no proper automatic metrics to evaluate generated answers (Chen et al., 2019).

In evaluating a GenQA model, it is essential to consider whether a generated response correctly contains vital information to answer the question. There exist several n-gram similarity metrics such

---

*This research was done while the author was affiliated with Adobe Research.

> **Context** : ... , this process, called hypothesis testing, consists of **four steps**. , ...
>
> **Question** : **How many steps** are involved in a hypothesis test?
> **Reference Answer** : **Four steps** are involved in a hypothesis test.
> **Generated Answer** : There are **seven steps** involved in a hypothesis test .
>
> **Human Judgment** : 0.063
>
> **BLEU-1** : 0.778   **BLEU-1-KPQA** : 0.057
> **ROUGE-L** : 0.713   **ROUGE-L-KPQA** : 0.127

Figure 1: An example from MS-MARCO (Bajaj et al., 2016) where widely used n-gram similarity metrics does not align with human judgments of correctness. On the other hand, our *KPQA*-metrics focus on the key information and give low scores to incorrect answers similar to humans.

as BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004), that measure the word overlaps between the generated response and the reference answer; however, these metrics are insufficient to evaluate a GenQA system (Yang et al., 2018a; Chen et al., 2019).

For instance, in the example in Figure 1 from the MS-MARCO (Bajaj et al., 2016), the generated answer receives a high score on BLEU-1 (0.778) and ROUGE-L (0.713) due to the many overlaps of words with those in the reference. However, humans assign a low score of 0.063 on the scale from 0 to 1 due to the mismatch of critical information. As in this example, we find that existing metrics often fail to capture the correctness of the generated answer that considers the key information for the question.

To overcome this shortcoming of the existing metrics, we propose a new metric called KPQA-metric for evaluating GenQA systems. To derive the metric, we first develop Keyphrase Predictor for Question Answering (KPQA). KPQA computes

the importance weight of each word in both the generated answer and the reference answer by considering the question. By integrating the output from the KPQA, we compute the KPQA-metric in two steps: (1) Given a {*question*, *generated answer*, *reference answer*}, we compute importance weights for each question-answer pair {*question*, *generated answer*} and {*question*, *reference answer*} using a KPQA; (2) We then compute a weighted similarity score by integrating the importance weights into existing metrics. Our approach can be easily integrated into most existing metrics, including n-gram similarity metrics and the recently proposed BERTScore (Zhang et al., 2020).

Additionally, we newly create two datasets for assessing automatic evaluation metrics with regard to the correctness in the GenQA domain. We first generate answers using state-of-the-art GenQA models on MS-MARCO and AVSD (Alamri et al., 2019) where the target answers are natural sentences rather than short phrases. We then collect human judgements of correctness over the 1k generated answers for each dataset.

In experiments on the human-evaluation datasets, we show that our KPQA-metrics have significantly higher correlations with human judgments than the previous metrics. For example, BERTScore-KPQA, one of our KPQA-integrated metrics, obtains Pearson correlation coefficients of 0.673 on MS-MARCO whereas the original BERTScore obtains 0.463. Further analyses demonstrate that our KPQA-metrics are robust to the question type and domain shift. Overall, our main contributions can be summarized as follows:

- We propose KPQA metric, an importance weighting based evaluation metric for GenQA.
- We collect high-quality human judgments of correctness for the model generated answers on MS-MARCO and AVSD, where those two GenQA datasets aim to generate sentence-level answers. We show that our proposed metric has a dramatically higher correlation with human judgments than the previous metrics for these datasets.
- We verify the robustness of our metric in various aspects such as question type and domain effect.
- We release the human-annotated benchmark dataset and pre-trained models to compute the KPQA-metric to the research community[1].

---

[1]https://github.com/hwanheelee1993/KPQA

2 Preliminaries: Automated Text Evaluation Metrics

We briefly review the current automated text evaluation metrics that have been used to evaluate GenQA systems.

BLEU is a popular evaluation metric for generated text based on n-gram precision. BLEU scores a candidate by counting the number present in the reference among the n-gram of the candidate. In general, n varies from 1 to 4, and the scores for varying n are aggregated with a geometric mean.

ROUGE is a set of evaluation metrics used for automatic text generation such as summarization and machine translation. Typically, most studies use ROUGE-L, which is a F-measure based on the longest common subsequence between a candidate and the reference.

METEOR (Banerjee and Lavie, 2005) is an F1 score of a set of unigram alignments. METEOR has a unique property that it considers stemmed words, synonyms, and paraphrases, as well as the standard exact word matches.

CIDER (Vedantam et al., 2015) is a consensus-based evaluation metric that is designed for a high correlation with human judgment in the image captioning problem. CIDEr uses Term Frequency-Inverse Document Frequency (TF-IDF) weights for human-like evaluation.

BERTScore is a recently proposed text evaluation metric that use pre-trained representations from BERT (Devlin et al., 2019). BERTScore first computes the contextual embeddings for given references and candidates independently with BERT, and then computes pairwise cosine similarity scores. When computing similarity, BERTScore adopts Inverse Document Frequency (IDF) to apply importance weighting.

3 Proposed Metric for Evaluating GenQA

To build a better metric for GenQA, we first propose KPQA. By considering the question, the KPQA assigns different weights to each token in the answer sentence such that salient tokens receive a high value. We then integrate the KPQA into existing metrics to make them evaluate correctness as well.

3.1 KPQA

For GenQA, we observe that each word has different levels of importance when assessing a gen-
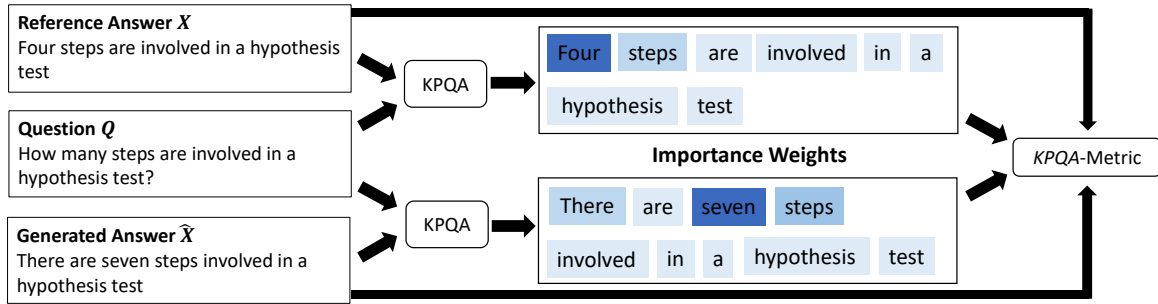
Figure 2: Overall flow of KPQA-metric. Importance weights are computed by pre-trained KPQA for each question-answer pair. And then these weights are integrated into existing metrics to compute weighted similarity.
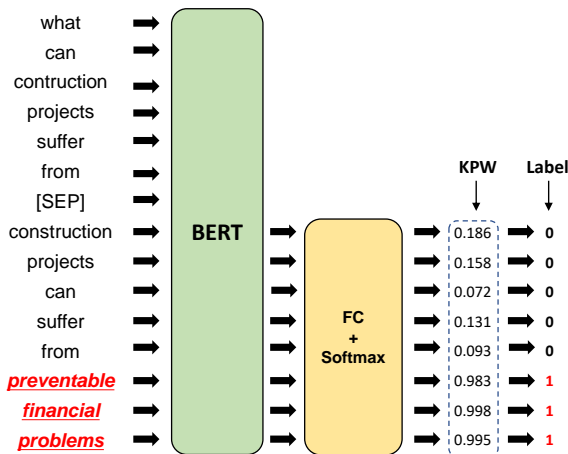

Figure 3: Overall architecture and an output example of KPQA. KPQA classifies whether each word in the answer sentences is in the answer span for a given question. We use the output probability KPW as an importance weight to be integrated into KPQA-metric.

erated answer. As shown in Figure 1, there exist keywords or keyphrases that are considered significant when evaluating the correctness of the answer. Additionally, some words, such as function words are mostly irrelevant to the correctness of the answer. Inspired by this observation, we introduce KPQA, which can predict the importance of each word when evaluating GenQA systems. As shown in Figure 3, KPQA is a BERT-based (Devlin et al., 2019) classifier that predicts salient tokens in the answer sentences depending on the question. We regard it as a multi-class classification task where each token is a single class. To train KPQA, we first prepare extractive QA datasets such as SQuAD (Rajpurkar et al., 2016), which consist of {*passage*, *question*, *answer-span*}. We transform these datasets into pairs of {*answer-sentences*, *question*, *answer-span*}. We extract the answer-sentences that contain answer-span in the passage

since these sentences are short summaries for the given question. Specifically, for a single-hop QA dataset such as SQuAD, we pick a single sentence that includes answer-span as the answer sentence. For the answers in a multi-hop QA dataset such as HotpotQA (Yang et al., 2018b), there are multiple supporting sentences for the single answer span. For these cases, we use SpanBERT (Joshi et al., 2020) to resolve the coreferences in the paragraphs and extract all of the supporting sentences to compose answer sentences. The {*question*, [SEP], *answer-sentences*} is then fed into the KPQA to classify the answer-span, which is a set of salient tokens, in the given answer-sentences considering the question.

3.2 KPQA Metric

Since KPQA's training process allows KPQA to find essential words in the answer sentences to a given question, we use a pre-trained KPQA to get the importance weights that are useful for evaluating the correctness of generated answers in GenQA. The overall flow of our KPQA-metric is described in Figure 2. We describe how we combine these weights with existing metrics to derive the KPQA-metric.

We first compute the importance weights for a given question $Q = (q_1, ..., q_l)$, reference answer $X = (x_1, ..., x_n)$ and generated answer $\hat{X} = (\hat{x}_1, ..., \hat{x}_m)$ using pre-trained KPQA. We provide each pair {*question*, *generated answer*} and {*question*, *reference answer*} to pre-trained KPQA and get the output of the softmax layer. We define these parts as KeyPhrase Weight (KPW) as shown in Figure 3. We note that $\text{KPW}^{(Q,\hat{X})} = (w_1, ..., w_m)$ is an importance weight of generated answer $\hat{X}$ for a given question Q. These weights reflect the importance of each token for evaluating the correctness.

We then compute KPQA-metric by incorporat-

ing the KPW into several existing metrics modifying the precision and recall to compute the weighted similarity.

BLEU-1-KPQA: We derive BLEU-1-KPQA, which is an weighted precision of unigram ($P_{Unigram}^{KPQA}$) as follows:

$$P_{Unigram}^{KPQA} = \frac{\Sigma_{i=1}^{m}\Sigma_{j=1}^{n}\text{KPW}_i^{(Q,\hat{X})} \cdot I(i,j)}{\Sigma_{i=1}^{m}\text{KPW}_i^{(Q,\hat{X})}}, \quad (1)$$

where $I(i,j)$ is an indicator function assigned the value of 1 if token x_i is the same as $\hat{x}_j$ and 0 otherwise.

ROUGE-L-KPQA: We also derive ROUGE-L-KPQA, which is a modified version of ROUGE-L using KPW to compute weighted precision(P_{LCS}^{KPQA}), recall(R_{LCS}^{KPQA}) and F1($F1_{LCS}^{KPQA}$), as follows:

$$P_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma_{i=1}^{m}\text{KPW}_i^{(Q,\hat{X})}}, \quad (2)$$

$$R_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma_{i=1}^{n}\text{KPW}_i^{(Q,X)}}, \quad (3)$$

$$F_{LCS}^{KPQA} = \frac{(1+\beta^2)R_{LCS}^{KPQA}P_{LCS}^{KPQA}}{R_{LCS}^{KPQA} + \beta^2 P_{LCS}^{KPQA}}, \quad (4)$$

where LCS is the Longest Common Subsequence between a generated answer and a reference answer. The $LCS^{KPQA}(X,\hat{X})$ is defined as follows:

$$LCS^{KPQA}(X,\hat{X}) = \Sigma_{i=1}^{m}I_i \cdot \text{KPW}_i^{(Q,\hat{X})}, \quad (5)$$

where I_i is an indicator function which is 1 if each word is in the LCS and 0 otherwise. β is defined in (Lin, 2004).

BERTScore-KPQA Similar to ROUGE-L-KPQA, we compute BERTScore-KPQA using KPW. We first compute contextual embedding $\hat{x}$ for generated answer $\hat{X}$ and $\mathbf{x}$ for reference X using the BERT model. Then, we compute weighted precision(P_{BERT}^{KPQA}), recall(R_{BERT}^{KPQA}) and F1($F1_{BERT}^{KPQA}$) with contextual embedding and KPW of each token as follows:

$$P_{BERT}^{KPQA} = \frac{\Sigma_{i=1}^{m}\text{KPW}_i^{(Q,\hat{X})} \cdot \max_{x_j \in x}\mathbf{x_i}^T\hat{\mathbf{x}_j}}{\Sigma_{i=1}^{m}\text{KPW}_i^{(Q,\hat{X})}} \quad (6)$$

$$R_{BERT}^{KPQA} = \frac{\Sigma_{i=1}^{n}\text{KPW}_i^{(Q,X)} \cdot \max_{\hat{x}_j \in \hat{x}}\mathbf{x_i}^T\hat{\mathbf{x}_j}}{\Sigma_{i=1}^{n}\text{KPW}_i^{(Q,X)}} \quad (7)$$

$$F1_{BERT}^{KPQA} = 2 \cdot \frac{P_{BERT}^{KPQA} \cdot R_{BERT}^{KPQA}}{P_{BERT}^{KPQA} + R_{BERT}^{KPQA}} \quad (8)$$

$$P_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma_{i=1}^{m}\text{KPW}_i^{(Q,\hat{X})}}, \quad (9)$$

$$R_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma_{i=1}^{n}\text{KPW}_i^{(Q,X)}}, \quad (10)$$

$$F_{LCS}^{KPQA} = \frac{(1+\beta^2)R_{LCS}^{KPQA}P_{LCS}^{KPQA}}{R_{LCS}^{KPQA} + \beta^2 P_{LCS}^{KPQA}}, \quad (11)$$

where LCS is the Longest Common Subsequence between a generated answer and a reference answer. The $LCS^{KPQA}(X,\hat{X})$ is defined as follows:

$$LCS^{KPQA}(X,\hat{X}) = \Sigma_{i=1}^{m}I_i \cdot \text{KPW}_i^{(Q,\hat{X})}, \quad (12)$$

where I_i is an indicator function which is 1 if each word is in the LCS and 0 otherwise. β is defined in (Lin, 2004). Similar to ROUGE-L-KPQA, we also derive BLEU-1-KPQA and BERTScore-KPQA by intergating KPW and provide the formulas in Appendix.

4 Collecting Human Judgments

4.1 Generating Answers

Dataset	Answer Length (avg.)	# Samples
MS MARCO	16.6	183k
AVSD	9.4	118k
Narrative QA	4.7	47k
SemEval	2.5	14k

Table 1: Statistics of the generative question answering dataset.

GenQA Datasets: To evaluate GenQA metrics, it is necessary to measure the correlation between human judgments and automated text evaluation metrics for evaluating the model generated answers. Recently, Chen et al. (2019) released human judgments of correctness for two GenQA datasets, NarrativeQA (Kočiský et al., 2018) and SemEval-2018

2108

Task 11 (SemEval) (Ostermann et al., 2018). However, we find that the average lengths of the answer sentence are 4.7 and 2.5 for NarrativeQA and SemEval, respectively, as shown in Table 1. These short answers are often short phrases and cannot be representative of GenQA, because the answers could be long and may deliver complex meaning. We argue that evaluating long and abstractive answers is more challenging and suitable for studying the metrics for general form of GenQA. To fill this gap, we collect the human judgments of correctness for model generated answers on two other GenQA datasets, MS-MARCO and AVSD, which have longer answers than NarrativeQA and SemEval as shown in Table 1. For the MS-MARCO, we use the Natural Language Generation (NLG) subset, which has more abstractive and longer answers than the Q&A subset.

GenQA Models: For each of the two datasets, we first generate answers for questions on validation sets using two trained GenQA models: UniLM (Dong et al., 2019) and MHPGM (Bauer et al., 2018) for MS-MARCO, MTN (Le et al., 2019) and AMF (Alamri et al., 2018; Hori et al., 2017) for AVSD. Details on these QA models are in Appendix. After training, we select 1k samples for each dataset in the validation set. Specifically, we first randomly pick the 500 questions in the validation set of each dataset and collect the corresponding model generated answers for each model so that we have two generated answers for each sample. Therefore, we collect a total of 1k samples, two different answers for 500 questions for each dataset. Also, we discard samples if one of two GenQA models exactly generates the ground-truth answer since human evaluation is useless during the sampling.

4.2 Collecting Human Judgments of Answer Correctness

We hire workers from the Amazon Mechanical Turk (MTurk) to rate the correctness of the generated answers from the models we trained. We assign ten workers for each sample to get reliable data. We ask the workers to annotate correctness using a 5-point Likert scale (Likert, 1932), where 1 means completely wrong, and 5 means completely correct. We provide the full instruction in Appendix.

Filtering Noisy Workers: Some workers did not follow the instructions, producing poor-quality judgments. To solve this problem, we filter noisy

Dataset	α	# Annotators (avg.)
MS MARCO	0.817	7.08
AVSD	0.725	6.88

Table 2: Inter annotator agreement measured by Krippendorff's alpha(α) and the average of number of annotators for each dataset.

ratings using the z-score, as in (Jung and Lease, 2011). We first compute the z-score among the ten responses for each sample. Then, we consider the responses whose z-score is higher than 1 to be noise and remove up to five of them in the order of the z-score. The average number of annotators after filtering is shown in Table 2. We use the average score of the annotators for each sample as a ground-truth evaluation score to assess the quality of the evaluation metric.

Inter-Annotator Agreement: The final dataset is further validated with Krippendorff's alpha (Krippendorff, 1970, 2011), a statistical measure of inter-rater agreement for multiple annotators. We observe that Krippendorff's α is higher than 0.6 for both datasets and models after filtering, as shown in Table 2. These coefficient numbers indicate a "substantial" agreement according to one of the general guidelines (Landis and Koch, 1977) for kappa-like measures.

5 Experiments

5.1 Implementation Details

We choose three datasets SQuAD v1.1 (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018b) and MS-MARCO Q&A subset to train KPQA. We combine the training set of the three datasets and use a 9:1 split to construct the training and development set of KPQA. For HotpotQA, we exclude *yes/no* type questions where the answers are not in the passage.

For model parameters, we choose *bert-base-uncased* variants for the BERT model and use one fully-connected layer with softmax layer after it. We train 5 epochs and choose the model that shows the minimum evaluation loss. We provide more details in Appendix.

5.2 Results

Evaluation Methods for Metrics: To compare the performance of various existing metrics and our

Dataset	MS-MARCO		AVSD		NarrativeQA		SemEval	
Metric	r	ρ	r	ρ	r	ρ	r	ρ
BLEU-1	0.349	0.329	0.580	0.562	0.634	0.643	0.359	0.452
BLEU-4	0.193	0.244	0.499	0.532	0.258	0.570	-0.035	0.439
ROUGE-L	0.309	0.301	0.585	0.566	0.707	0.708	0.566	0.580
METEOR	0.423	0.413	0.578	0.617	0.735	0.755	0.543	0.645
CIDEr	0.275	0.278	0.567	0.600	0.648	0.710	0.429	0.595
BERTScore	0.463	0.456	0.658	0.650	**0.785**	0.767	0.630	0.602
BLEU-1-KPQA	0.675	0.634	0.719	0.695	0.716	0.699	0.362	0.462
ROUGE-L-KPQA	**0.698**	0.642	0.712	0.702	0.774	0.750	**0.742**	**0.687**
BERTScore-KPQA	0.673	**0.655**	**0.729**	**0.712**	0.782	**0.770**	0.741	0.676

Table 3: Pearson Correlation(r) and Spearman's Correlation(ρ) between various automatic metrics and human judgments of correctness. All of the results are statistically significant (p-value < 0.01).

metric, we use the Pearson coefficient and Spearman coefficient. We compute these correlation coefficients with human judgments of correctness. We test using MS-MARCO, AVSD, from which we collected human judgments, and NarrativeQA and SemEval from (Chen et al., 2019).

Performance Comparison: We present the correlation scores for the baseline metrics and KPQA-augmented ones for multiple datasets in Table 3. The correlations between human judgment and most of the existing metrics such as BLEU or ROUGE-L are very low, and this shows that those widely used metrics are not adequate to GenQA. Moreover, the performance of existing metrics is especially low for the MS-MARCO, which has longer and more abstractive answers than the other three datasets.

We observe a significantly higher correlation score for our proposed KPQA-metric compared to existing metrics especially for MS-MARCO and AVSD where the answers are full-sentences rather than short phrases. For the NarrativeQA, where existing metrics also have higher correlations, the gap in performance between KPQA-metric and existing metrics is low. We explain this is because the answers in NarrativeQA are often a single word or short phrases that are already keyphrases.

Comparison with IDF: The next best metric after our proposed metric is the original BERTScore, which uses contextual embeddings and adopts IDF based importance weighting. Since IDF is dependent on the word-frequency among the documents, it can assign a lower weight to some important words to evaluate correctness if they frequently occur in the corpus as shown in Table 5. On the other hand, our KPQA integrated metric assigns weights

Dataset	MS-MARCO	
Metric	r	ρ
BLEU-1-KPQA	0.675	0.634
ROUGE-L-KPQA	**0.698**	0.642
BERTScore-KPQA	0.673	**0.655**
BLEU-1-KPQA$_{\text{MARCO}}$	0.573	0.529
ROUGE-L-KPQA$_{\text{MARCO}}$	0.598	0.564
BERTScore-KPQA$_{\text{MARCO}}$	0.602	0.595
BLEU-1-KP	0.629	0.589
ROUGE-L-KP	0.671	0.640
BERTScore-KP	0.657	0.649

Table 4: Ablation studies for our proposed metrics on domain effect and using the question context.

to words in the answer sentence using the context of the question. This approach provides dynamic weights for each word that leads to a better correlation with human evaluation as shown in Table 3.

5.3 Ablation Study

Domain Effect: Our KPQA metric computes importance weights using a supervised model; thus our proposed method may suffer from a domain shift problem. Although our metric is evaluated on out-of-domain datasets except MS-MARCO, we further examine the effect of the domain difference by changing the trainset of KPQA. Since we train KPQA with the combination of SQuAD, HotpotQA and MS-MARCO Q&A, the original KPQA works as in-domain for MS-MARCO. To measure the negative domain effect, we exclude the MS-MARCO Q&A in the training set of KPQA and measure the performance of KPQA-metric on MS-MARCO. We annotate it "-KPQA$_{\text{MARCO}}$" and report the results in Table 4. This drop shows the effect of the negative domain shift for our KPQA-metric. However, "-KPQA$_{\text{MARCO}}$" is still much higher than all
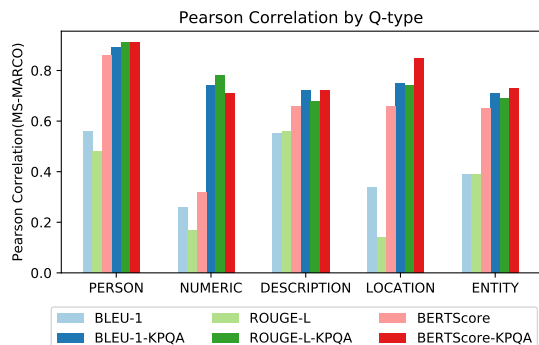
Figure 4: Pearson correlation coefficient among question types on MS-MARCO dataset.

Question : How to cook sausage peppers onions ?

Reference Answer : To cook sausage peppers onions first place the sausage in a large skillet over medium heat, and brown on all sides after that remove from skillet, and slice meelt butter in the skillet, stir in the yellow onion, red onion, and garlic, and cook 2 to 3 minutes and then mix in red bell pepper and green bell pepper season with basil, and oregano in last stir in white wine.

Generated Answer : To cook sausage peppers onions , preheat the oven to 350 degrees fahrenheit . Place the onions in the oven and cook for 20 minutes

Figure 5: An example from MS-MARCO where the answers are composed of multiple sentences.

of the previous metrics.

Using the Question Context: Our KPQA uses the question as an additional context to predict the keyphrases in the sentence, as shown in Figure 3. To examine the power of utilizing the question information for the keyphrase predictor, we remove the question part from the dataset and train the keyphrase prediction model. With the newly trained model, we compute the importance weights for words in the target sentence and apply them to BLEU-1, ROUGE-L, and BERTScore. We call this metric as "-KP" and report the results in Table 4. We observe that "-KPQA" metric is better than "-KP" metric for all of the three variants. These results show that training keyphrase predictor to find the short answer candidate in the sentence is effective for capturing the key information in the generated answer, but it is more effective when the question information is integrated.

5.4 Analysis

Correlation Among Question Type: Since MS-MARCO provides the question type information (*PERSON*, *NUMERIC*, *DESCRIPTION*, *LOCATION*, *ENTITY*) for each {*question*, *answer*} pair, we evaluate the various metrics by the question type. We split the dataset into these five question types and measure the performance of various metrics with Pearson correlation coefficients. As shown in Figure 4, our KPQA-metric variants outperform their original version in all of the question types. KPQA-metric is especially effective for the *NUMERIC* question type, whose answer sentence often has shorter keyphrase such as a number. For *ENTITY* and *PERSON* question types, the gap between KPQA-integrated metric and original metric

is lower for BERTScore. We speculate that this is because the original BERTScore uses IDF-based importance weighting, unlike other metrics.

Multiple Sentence Answers: Most of the answers in MS-MARCO and AVSD consist of single sentences, but the answers for GenQA can be multiple sentences like (Fan et al., 2019). To verify our KPQA-metric on multiple sentence answers, we collect additional 100 human judgments for the generated answer whose answers are multiple sentences in the MS-MARCO like the example in Figure 5, and evaluate the various metrics on this dataset. As shown in Table 6, our KPQA integrated metric shows still higher correlations than other metrics. We observe that the gap between KPQA integrated metrics and existing metrics is relatively lower than that of Table 3. We speculate this is because many of the multiple sentence answers are *DESCRIPTION* type answers whose keyphrases are sometimes vague, similar to the results in Figure 4.

Error Analysis: We pick 100 error cases from MS-MARCO in the order of a large difference in ranks among 1k samples between human judgments and BERTScore-KPQA. The importance weights have no ground-truth data; thus we manually visualize the weights as shown in Table 5 and analyze the error cases.

From the analysis, we observe some obvious reasons for the different judgments between humans and BERTScore-KPQA. We first classify error cases by the question types and observe that 51 cases belong to *NUMERIC*, and 31 cases belong to *DESCRIPTION*. We further analyze the *NUMERIC* question type and find that many parts of the errors

2111

Context	... , it can take 5-20 hours of walking to lose 1 pound ... , ...
Question	How long do i need to walk in order to loose a pound ?

Reference	Walk for 5 to 20 hours to lose 1 pound .
IDF	Walk for 5 to 20 hours to lose 1 pound .
KPW	Walk for 5 to 20 hours to lose 1 pound .

Human Judgment: 0.94,　**BERTScore**: 0.72,　**BERTScore-KPQA**: 0.93

UniLM	You need to walk for 5 to 20 hours in order to loose a pound .
IDF	You need to walk for 5 to 20 hours in order to loose a pound .
KPW	You need to walk for 5 to 20 hours in order to loose a pound .

Table 5: An example of the scores given by humans, BERTScore and BERTScore-KPQA for the samples from MS-MARCO dataset. BERTScore uses IDF and BERTScore-KPQA uses KPW as importance weights to compute score. Heat map shows IDF and KPW, which are normalized between 0 and 1.

Dataset	MS-MARCO	
Metric	r	ρ
BLEU-1	0.363	0.364
ROUGE-L	0.584	0.607
BERTScore	0.712	0.728
BLEU-1-KPQA	0.529	0.540
ROUGE-L-KPQA	0.642	0.648
BERTScore-KPQA	**0.774**	**0.786**

Table 6: Correlation coefficients between various automatic metrics and human judgments of correctness for evaluating multiple sentence answers in MS-MARCO (Bajaj et al., 2016).

Metrics	MS-MARCO	AVSD
BLEU-1	63.44	72.02
ROUGE-L	61.29	70.98
BERTScore	67.74	78.24
BLEU-1-KPQA	74.19	**81.35**
ROUGE-KPQA	**76.34**	77.20
BERTScore-KPQA	**76.34**	**81.35**

Table 7: The percentage of matches at which human judgment and various metrics on ranking two models' output.

are due to higher weights on units such as "million" or "years." There exist a total of ten error cases for this type, and we believe that there is room for improvement with regard to these errors through post-processing. In the case of the *DESCRIPTION* question type, 17 out of 31 cases are due to inappropriate importance weights. We speculate this result is because the keyphrases for the answers to questions belonging to the *DESCRIPTION* type are sometimes vague; thus, the entire answer needs to be considered when it is evaluated.

Rank-Pair: One practical usage of the text evaluation metric is ranking outputs of multiple models. Using the collected human judgments of correctness for the same 500 {*question*, *reference answer*} pairs for two models on MS-MARCO and AVSD, we can compare the output of each models through the human-annotated score. To see the alignment of ranking ability among the various metrics with that of human judges, we conduct a "win-lose match" experiment, counting the number of times that a metric ranks the output of two models as the same as human judges. To prepare test samples, we chose only those whose gap between human judgment scores on the two models is greater than 2. Finally, we obtain 93 and 193 samples for MS-MARCO and AVSD, respectively. Considering that the range of scores is 1-5, this approach ensures that each output of the models has a clear quality difference. Table 7 shows the percentage of rank-pair matches for each metric with human judgments of correctness on two datasets. Our KPQA-metric shows more matches than previous metrics in all of the datasets; thus, it is more useful for comparing the generated answers from different models.

6 Related Work

One important next step for current QA systems is to generate answers in natural language for a given question and context. Following this interest, several generative (abstractive) QA datasets (Bajaj et al., 2016; He et al., 2018; Kočiský et al., 2018; Fan et al., 2019), where the answer is not necessarily in the passage, have recently been released. Since the task is to generate natural language for the given question, the QA system is often trained with seq2seq (Sutskever et al., 2014) objective simi-

larly to other natural generation tasks such as neural machine translation. Hence, researchers often use n-gram based similarity metrics such as BLEU to evaluate the GenQA systems, following other natural language generation tasks.

However, most of these n-gram metrics including BLEU were originally developed to evaluate machine translation and previous works (Liu et al., 2016; Nema and Khapra, 2018; Kryscinski et al., 2019) have shown that these metrics have poor correlations with human judgments in other language generation tasks such as dialogue systems. As with other text generation systems, for GenQA, it is difficult to assess the performance through n-gram metrics. Especially, n-gram similarity metrics can give a high score to a generated answer that is incorrect but shares many unnecessary words with the reference answer. Previous works (Marton and Radul, 2006; Yang et al., 2018a; Chen et al., 2019) have pointed out the difficulty of similar problems and studied automated metrics for evaluating QA systems. Inspired by these works, we focus on studying and developing evaluation metrics for GenQA datasets that have more abstractive and diverse answers. We analyze the problem of using existing n-gram similarity metrics across multiple GenQA datasets and propose alternative metrics for GenQA.

7 Conclusion

In this paper, we create high-quality human judgments on two GenQA datasets, MS-MARCO and AVSD, and show that previous evaluation metrics are poorly correlated with human judgments in terms of the correctness of an answer. We propose KPQA-metric, which uses the pre-trained model that can predict the importance weights of words in answers to a given question to be integrated with existing metrics. Our approach has a dramatically higher correlation with human judgments than existing metrics, showing that our model-based importance weighting is critical to measure the correctness of a generated answer in GenQA.

Ethical Considerations

Our paper and dataset follow ethical standards. We compensate the annotators with competitive pay. Furthermore, we follow all ethical procedures for data collection, where we use public datasets to train the models.

References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K. Marks, Chiori Hori, Peter Anderson, Stefan Lee, and Devi Parikh. 2019. Audio visual scene-aware dialog. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7558–7567. Computer Vision Foundation / IEEE.

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*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2521–2530, Hong Kong, China. Association for Computational Linguistics.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2020. Generating well-formed answers by machine reading with stochastic selector networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Sym-*

posium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7424–7431. AAAI Press.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

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 *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4203–4212. IEEE Computer Society.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Hyun Joon Jung and Matthew Lease. 2011. Improving consensus accuracy via z-score and weighted voting. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Klaus Krippendorff. 1970. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Hung Le, Doyen Sahoo, Nancy Chen, and Steven Hoi. 2019. Multimodal transformer networks for end-to-end video-grounded dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5612–5623, Florence, Italy. Association for Computational Linguistics.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike 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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Gregory Marton and Alexey Radul. 2006. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 375–382, New York City, USA. Association for Computational Linguistics.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style generative reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2273–2284, Florence, Italy. Association for Computational Linguistics.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, New Orleans, Louisiana. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

An Yang, Kai Liu, Jing Liu, Yajuan Lyu, and Sujian Li. 2018a. Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 98–104, Melbourne, Australia. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018b. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2972–2978. IJCAI/AAAI Press.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.