Ph.D. DISSERTATION

Factual Consistency Evaluation for Conditional Text Generation Systems

조건부 텍스트 생성 시스템에 대한 사실 관계의 일관성 평가

BY

HWANHEE LEE

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Factual Consistency Evaluation for Conditional Text Generation Systems

조건부 텍스트 생성 시스템에 대한 사실 관계의 일관성 평가

BY

HWANHEE LEE

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Factual Consistency Evaluation for Conditional Text Generation Systems

조건부 텍스트 생성 시스템에 대한 사실 관계의 일관성 평가

지도교수 정 교 민
이 논문을 공학박사 학위논문으로 제출함

2022년 5월

서울대학교 대학원

전기 · 정보 공학부

이 환 희

이환희의 공학박사 학위 논문을 인준함

2022년 6월

위 원 장:	심 규 석
부위원장:	정 교 민
위 원:	문 태 섭
위 원:	황 승 원
위 원:	박 진 영

Abstract

Despite the recent advances of conditional text generation systems leveraged from pre-trained language models, factual consistency of the systems are still not sufficient. However, widely used n-gram similarity metrics are vulnerable to evaluate the factual consistency. Hence, in order to develop a factual consistent system, an automatic factuality metric is first necessary. In this dissertation, we propose four metrics that show very higher correlation with human judgments than previous metrics in evaluating factual consistency, for diverse conditional text generation systems. To build such metrics, we utilize *(1) auxiliary tasks* and *(2) data augmentation methods*.

First, we focus on the keywords or keyphrases that are critical for evaluating factual consistency and propose two factual consistency metrics using two different auxiliary tasks. We first integrate the keyphrase weights prediction task to the previous metrics to propose a KPQA (Keyphrase Prediction for Question Answering)-metric for generative QA. Also, we apply question generation and answering to develop a captioning metric QACE (Question Answering for Captioning Evaluation). QACE generates questions on the keywords of the candidate. QACE checks the factual consistency by comparing the answers of these questions for the source image and the caption.

Secondly, different from using auxiliary tasks, we directly train a metric with a data-driven approach to propose two metrics. Specifically, we train a metric to distinguish augmented inconsistent texts with the consistent text. We first modify the original reference captions to generate inconsistent captions using several rule-based methods such as substituting keywords to propose UMIC (Unreferenced Metric for Image Captioning). As a next step, we introduce a MFMA (Mask-and-Fill with Masked-Article)-metric by generating inconsistent summary using the masked source and the masked summary. Finally, as an extension of developing data-driven factual consistency metrics, we also propose a faster post-editing system that can fix the factual errors in the system.

keywords: factual consistency, text generation, evaluation metric

student number: 2017-26066

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently various natural language generation systems such as chatbot, story generation, abstractive summarization, and image captioning have shown great success following the advances in pre-trained language models. Among these systems, the goal of conditional text generation systems is to generate a text for a given specified source such as an image or an article as shown in Figure 1.1. In such systems, it is necessary to generate a text that is factual consistent with the source. In other words, all of the contents in the text must be entailed by the source. However, previous studies [2, 3] have shown that factual consistency of the current conditional text systems are still insufficient and often generate the texts that have at least one factual errors as in *"Candidate Summary 1"* of Figure 1.2. In order to overcome this factual inconsistency problem, an automatic metric that is easily able to evaluate the factual consistency of the current systems is necessary. But as shown in Figure 1.2, widely used n-gram similarity based evaluation metrics such as BLEU [4] or ROUGE [5] are vulnerable to evaluate the factual consistency and often give higher score to the inconsistent text. These n-gram similarity metrics do not consider the importance of each word in evaluating factual consistency and simply decide the quality of each text with the overlap between the human generated reference and the machine generated text. Also, as in *"Candidate Summary 2"* in Figure 1.2, these metrics often give lower score to the factually consistent examples

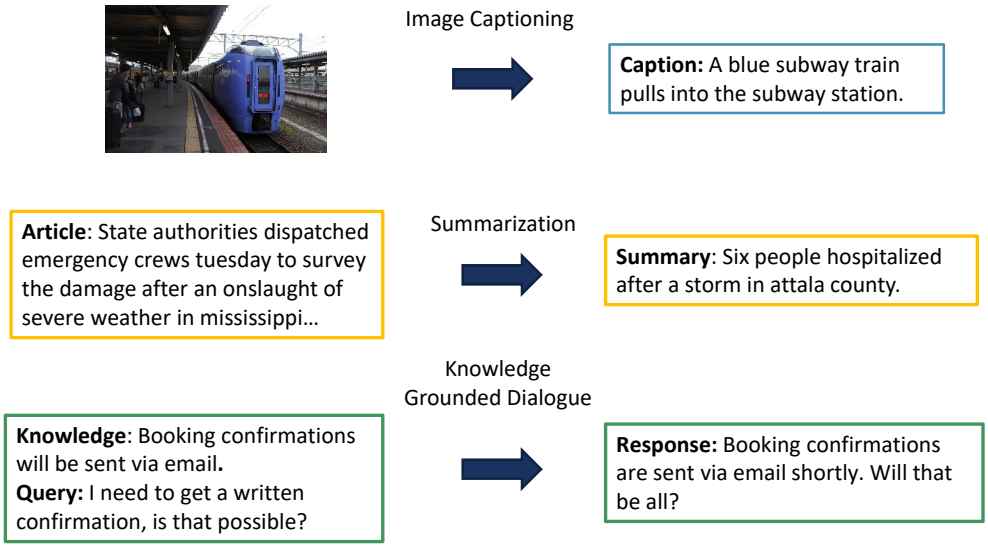


Figure 1.1: Examples of conditional text generation systems.

that are differently expressed with the reference summary. These examples show that judgments between the humans and the n-gram similarity metrics are quite different especially for evaluating the factual consistency.

In this dissertation, we introduce four novel types of factual consistency metrics in several conditional text generation systems to overcome the limitation of the widely used evaluation metrics. The four proposed factual consistency metrics are specified into two categories as follows; *1) using auxiliary tasks* and *2) data augmentation*, based on the types of the approaches to develop each metric.

Part1: Using Auxiliary Tasks for Factual Consistency Evaluation: As shown in the examples in Figure 1.2, there are keywords or keyphrases that are crucial for evaluating the factual consistency. Hence, to evaluate the factual consistency, we first investigate the auxiliary tasks that can focus on keywords or keyphrases to evaluate factual consistency.

Article: Scientists from harvard medical school have discovered a way of turning stem cells into killing machines to fight brain cancer. In experiments on mice, the stem cells were genetically engineered to produce and secrete toxins which kill brain tumours, without killing normal cells or themselves. Researchers said the next stage was to test the procedure in humans. (. . .)

Reference Summary: Scientists in the us have developed a stem cell therapy for brain tumours.

Candidate Summary 1: Scientists in the us have developed a stem cell therapy for *killing normal cells*.

Factual Consistency: *inconsistent*
BLEU-4: 0.758
ROUGE-L: 0.820

Candidate Summary 2: Scientists from harvard have discovered a new therapy for tumours in the brain.

Factual Consistency: *consistent*
BLEU-4: 0.000
ROUGE-L: 0.462

Figure 1.2: An example of factually inconsistent/consistent examples and the wrong evaluation of the widely used metrics in abstractive summarization.

We start from changing the previous widely used metrics by adding importance weights to them. We first utilize the keyphrase weights from an auxiliary task for factual consistency evaluation in generative question answering (GenQA), where the task is to generate a free-form answer for a given passage in the following research.

- **Hwanhee Lee**, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Joongbo Shin, and Kyomin Jung, KPQA: A Metric for Generative Question Answering Using Keyphrase Weights, in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2021, Online.

In the automatic evaluation of GenQA systems, it is difficult to assess the correctness of generated answers due to the free-form of the answer. In this task, 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.

We further study the usage of another auxiliary tasks to focus on the keywords to evaluate factual consistency. Different from KPQA, we develop a metric that is totally different from the n-gram similarity metrics. We adopt Question Generation and Question Answering (QGQA) that are often used for evaluating the factual consistency some conditional text generation systems [6]. We focus on the usage of QGQA in multimodal text generation system, an image captioning in the following research.

- **Hwanhee Lee**, Thomas Scialom, Seunghyun Yoon, Franck Dernoncourt, and Kyomin Jung, QACE: Asking Questions to Evaluate an Image Caption, *In Find-*

ings of the Association for Computational Linguistics: EMNLP 2021 (Findings of EMNLP), November 2021, Punta Cana, Dominican Republic.

In this study, we propose QACE, a new metric based on **Q**uestion **A**nswering for **C**aption **E**valuation. QACE generates questions on the evaluated caption and checks its content by asking the questions on either the reference caption or the source image. We first develop QACE_{Ref} that compares the answers of the evaluated caption to its reference, and report competitive results with the state-of-the-art metrics. To go further, we propose QACE_{Img}, which asks the questions directly on the image, instead of reference. A Visual-QA system is necessary for QACE_{Img}. Unfortunately, the standard VQA models are framed as a classification among only a few thousand categories. Instead, we propose Visual-T5, an abstractive VQA system. The resulting metric, QACE_{Img} is multi-modal, reference-less, and explainable. Our experiments show that QACE_{Img} compares favorably w.r.t. other reference-less metrics.

Part2: Data Augmentation for Factual Consistency Evaluation: The main goal of the factual consistency metric is to distinguish the inconsistent text, where at least one of the contents in the generated text are not consistent with the source, with the consistent text. Inspired by this point, we introduce novel data augmentation techniques to generate synthetic inconsistent samples in two studies included in this part to develop factual consistency metrics.

We first generate inconsistent samples using pre-defined rules such as keyword substitution to develop a metric for image captioning system in the following research.

- **Hwanhee Lee**, Seunghyun Yoon, Franck Dernoncourt, Trung Bui, and Kyomin Jung, UMIC: An Unreferenced Metric for Image Captioning via Contrastive

Learning, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), August 2021, Online.*

Despite the recent advancements in model-based text evaluation metrics such as BERTScore [7], it is still difficult to evaluate the image captions without enough reference captions due to the diversity of the descriptions, and the properties of the multimodal task. In this paper, we introduce a new image captioning metric UMIC, an **U**nreferenced **M**etric for **I**mage **C**aptioning which does not require reference captions to evaluate image captions. Based on Vision-and-Language BERT, we fine-tune the model to discriminate erroneous captions via contrastive learning. Specifically, we apply various approaches such as keyword substitution that can imitate the common error types in the model-generated captions to human written captions to build negative captions. Then, we fine-tune Vision-and-Language BERT to distinguish these negative captions with the original reference captions to develop a captioning metric. Also, we observe critical problems of the previous benchmark dataset (i.e., human annotations) on image captioning metric, such as unbalanced score distribution in the dataset. To solve such problems, we introduce a new collection of human annotations named CapEval1k on the generated captions using recent captioning systems. We validate our proposed evaluation metric UMIC on four datasets, including our new dataset CapEval1k, and show that UMIC has a higher correlation than all previous metrics that require multiple references. Furthermore, our in-depth analysis demonstrates that UMIC properly determines the quality of the caption using both the image and the caption.

As a next step of simple rule-based inconsistent data augmentation method, we study a more advanced inconsistent data augmentation method based on mask-and-fill. We generate inconsistent samples that are more difficult to be distinguished from con-

sistent samples, regarding the relation between the source and the generated text in the following research.

- **Hwanhee Lee**, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung, Masked Summarization to Generate Factually Inconsistent Summaries for Improved Factual Consistency Checking, in *Findings of the Association for Computational Linguistics: NAACL 2022 (Findings of NAACL)*, July 2022, Seattle, WA, USA.

Despite the recent advances in abstractive summarization systems, it is still difficult to determine whether a generated summary is factual consistent with the source text. To this end, the latest approach is to train a factual consistency classifier on factually consistent and inconsistent summaries. Luckily, the former is readily available as reference summaries in existing summarization datasets. However, generating the latter remains a challenge, as they need to be factually inconsistent, yet closely relevant to the source text to be effective. In this paper, we propose to generate factually inconsistent summaries using source texts and reference summaries with key information masked. Experiments on seven benchmark datasets demonstrate that factual consistency classifiers trained on summaries generated using our method generally outperform existing models and show a competitive correlation with human judgments. We also analyze the characteristics of the summaries generated using our method.

Part3: Improving Factual Consistency from Evaluation Metrics: Furthermore, we also study the way to mitigate the factual inconsistency problem itself. As a first step, we propose a post-editing system to correct the factual errors inspired by the approaches to train factual consistency metrics in the following work.

- **Hwanhee Lee**, Cheoneum Park, Seunghyun Yoon, Trung Bui, Franck Dernoncourt, Juae Kim, Kyomin Jung, Factual Error Correction for Abstractive Summaries Using Entity Retrieval, in *arXiv*, May 2022

Despite the recent advancements in abstractive summarization systems leveraged from large-scale datasets and pre-trained language models, the factual correctness of the summary is still insufficient. One line of trials to mitigate this problem is to include a post-editing process that can detect and correct factual errors in the summary. In building such a post-editing system, it is strongly required that 1) the process has a high success rate and interpretability and 2) has a fast running time. Previous approaches focus on regeneration of the summary using the autoregressive models, which lack interpretability and require high computing resources. In this paper, we propose an efficient factual error correction system RFEC based on entities retrieval post-editing process. RFEC first retrieves the evidence sentences from the original document by comparing the sentences with the target summary. This approach greatly reduces the length of text for a system to analyze. Next, RFEC detects the entity-level errors in the summaries by considering the evidence sentences and substitutes the wrong entities with the accurate entities from the evidence sentences. Experimental results show that our proposed error correction system shows more competitive performance than baseline methods in correcting the factual errors with a much faster speed.

This dissertation is organized as shown in Figure 1.3. The next chapter, Chapter 2 provides background on text evaluation metrics and evaluation methods for metrics. In Chapter 3-4, we introduce two independent studies that utilize auxiliary tasks to focus on keywords-aware evaluation, Keyphrase Weight Prediction (Chapter 3) and Question Generation and Question Answering (Chapter 4), to develop factual consistency metric for generative QA and image captioning respectively. In Chapter 5-6, we introduce two works that generate synthetic inconsistent data using rule-based methods (Chapter

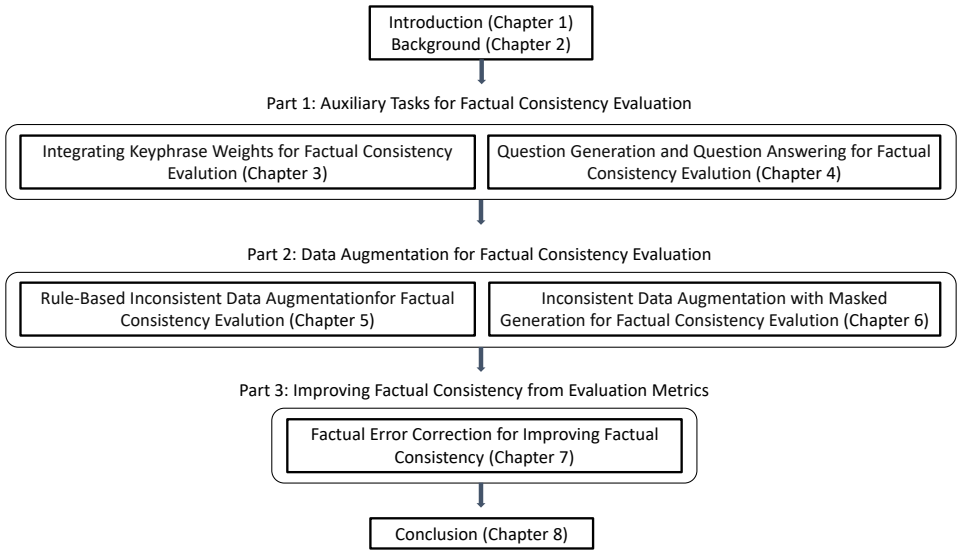


Figure 1.3: Outline of this dissertation.

5) and masked generation (Chapter 6) to train factual consistency metric. We also introduce a method that can improve factual consistency to mitigate the factual errors by post-editing in Chapter 7. Finally, we conclude the dissertation in Chapter 8.

Chapter 2

Background

2.1 Text Evaluation Metrics

We briefly review the current automated text evaluation metrics that have been used to evaluate conditional text generation systems that are specified as shown in Figure 2.1.

2.1.1 *N*-gram Similarity Metrics

N-gram/embedding similarity metrics directly measure the similarity between the reference text and the candidate text. The most widely used evaluation metrics are n-gram similarity metrics such as BLEU or ROUGE. These metrics simply compute the similarity between the reference text with the candidate text by measuring the word overlaps.

BLEU is one of the most popular evaluation metrics for the generated texts based on n-gram precision. BLEU computes a score of the candidate text by counting the number of occurrence in the reference among the n-gram of the candidate. Generally, n varies from 1 to 4, and the geometric mean of the various n are widely used.

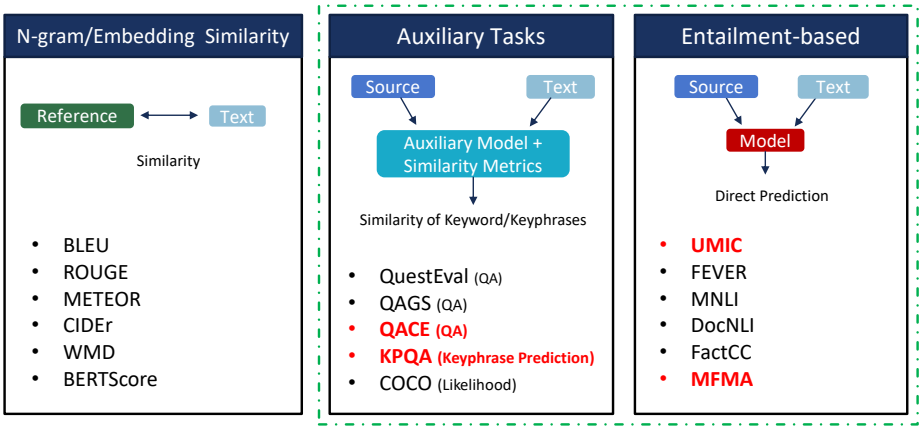


Figure 2.1: Automatic evaluation metrics for conditional text generation systems. The metrics based on auxiliary tasks and entailment are specialized for factual consistency evaluation.

ROUGE is also one of the widely used evaluation metrics used for automatic text generation tasks such as machine translation and summarization. Among various ROUGE variants, ROUGE-L, which is a F-measure based on the longest common subsequence between a candidate and the reference, is frequently used.

METEOR [8] is a F1-score based metric computed by unigram alignments. Different from other metrics, METORS utilizes synonyms, paraphrases, and stemmed words, as well as the general exact word matches.

CIDER [9] is a consensus-based evaluation metric that is specialized for evaluating the generated captions in image captioning task. CIDEr uses Term Frequency-Inverse Document Frequency (TF-IDF) weights to consider the importance of each n-grams for human-like evaluation.

2.1.2 Embedding Similarity Metrics

As an advanced method, embedding based metrics such as BERTScore [10] computes the similarity in the embedding space using the pre-computed embedding for both the reference text and the candidate text. These metrics are relatively easy to compute than the advanced metrics.

Word Mover's Distance (WMD) [11] computes the minimum transportation cost between two sets of texts using the embeddings from word2vec [12]. In other words, it computes how much cost is necessary to move the words in one text to another text.

BERTScore is also an embedding similarity based evaluation metric that uses pre-trained representations from BERT [13]. BERTScore first independently computes the contextual embeddings of the given references and the candidate texts with BERT. Then, it computes pairwise cosine similarity scores between two embeddings. When computing the similarity, BERTScore integrates Inverse Document Frequency (IDF) to consider the importance of each token.

2.1.3 Auxiliary Task Based Metrics

For factual consistency focused evaluation, various metrics that utilize auxiliary tasks have introduced. These metrics use auxiliary tasks that can focus on keywords or keyphrases which are crucial for factual consistency evaluation. Among them, question answering based metrics [14, 6] are widely used due to the intuitiveness and the interpretability. However, these metrics often require heavy computational cost.

QA-Based Metrics QuestEval [14] and QAGS [6], which are originally built to evaluate the summaries, use the question generation and answering framework to evaluate the factual consistency of the generated text. These metrics usually compute the factuality score by generating the question for the generated summaries, and then

comparing the answers of them with both summaries and the article. In this dissertation, we introduce a new QA-based metric QACE [15] for image captioning evaluation.

Likelihood-Based Metrics BARTScore [16] directly computes the likelihood of the generated texts using BART [17]. CoCo [18] utilizes the difference of the likelihood for each summary using the original article and the masked source where the key information is masked.

2.1.4 Entailment Based Metrics

To simply evaluate the factual consistency, entailment-based metrics that can directly measure the factual consistency using trained systems are widely used. These metrics usually use deep neural networks to compute the entailment score between the source text and the candidate text, which are relatively faster than using auxiliary tasks.

Related Tasks For factual consistency evaluation, the output of several natural language understanding tasks such as natural language inference [19] or fact checking [20] are also used. These tasks are similar to factual consistency evaluation in detecting the contradiction between the source text and the input text. However, the length of the input text in these tasks are usually shorter than the general output of the conditional text generation systems, and this leads to the difficulty of direct application to factual consistency evaluation.

Data Augmentation Methods Previous studies [21, 3] have shown that the factual errors in the conditional text generation systems are often trivial. For this reason, FactCC [22] and DocNLI [23] imitate the factually inconsistent samples using predefined rules and then directly train factual consistency metrics using the augmented data. We propose UMIC [15] for image captioning based on the rule-based data augmentation.We also present MFMA [24] through the masked generation in this dissertation.

2.2 Evaluating Automated Metrics

An ideal automatic evaluation metric can imitate the evaluation procedure of the humans. Hence, the goal of developing automatic evaluation metric is to develop a metric that is similar to the human judgments. Therefore, to evaluate the quality of the evaluation metrics, correlation with the human judgments are widely used. We briefly review the correlation coefficient in this section.

Pearson Correlation Coefficient Pearson correlation coefficient (r) [25] measures the linear correlation between two species of input data. This coefficient computes the ratio between the covariance of two input variables and then compute the product of their standard deviations. The coefficient value is between -1 and +1, and the higher the value means the higher similarity between the two input data. This Pearson correlation coefficient assesses the linear relationships between the automated metrics and the human judgments when evaluating the metrics.

Spearman Correlation Spearman's rank correlation coefficient(ρ) [26] is a rank based correlation between the two input variables. Similar to Pearson correlation coefficient, it computes how well the relationship between two input variables through a monotonic function. In other words, this correlation coefficient shows the similarity of the rank (i.e. relative position) between two input variables. In evaluating metrics, this coefficient is used to measure the similarity of the rank between the automated metrics and the human judgments.

Chapter 3

Integrating Keyphrase Weights for Factual Consistency Evaluation

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 [27, 28, 29, 30, 31, 32]. However, as a bottleneck in developing GenQA models, there are no proper automatic metrics to evaluate generated answers [33].

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 as BLEU [34] and ROUGE-L [35], that measure the word overlaps between the generated response and the reference answer; however, these metrics are insufficient to evaluate a GenQA system [36, 33].

For instance, in the example in Figure 3.1 from the MS-MARCO [1], 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

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 3.1: An example from MS-MARCO [1] 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.

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 [7].

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 [37] 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.

3.1 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 [1, 38, 39, 40], 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 [41] objective similarly 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 [42, 43, 44] 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 [45, 36, 33] 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.

3.2 Proposed Approach: KPQA-Metric

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.

3.2.1 KPQA

For GenQA, we observe that each word has different levels of importance when assessing a generated answer. As shown in Figure 3.1, 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 impor-

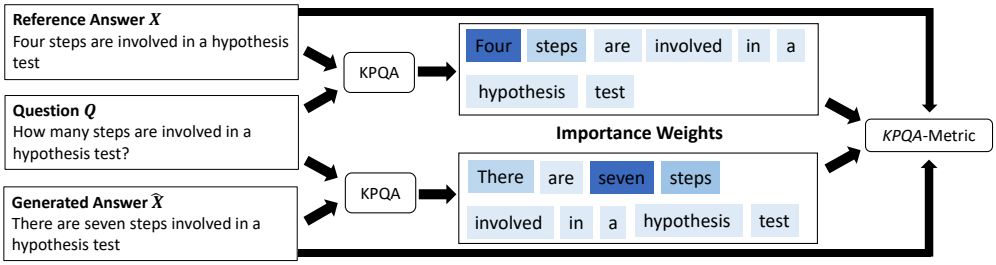


Figure 3.2: Overall flow of KPQA-metric. Importance weights are computed by pre-trained KPQA for each question-answer pair. And then these weights are integrated into existing metrics to compute weighted similarity.

tance of each word when evaluating GenQA systems. As shown in Figure 3.3, KPQA is a BERT-based [13] 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 [46], 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 [47], there are multiple supporting sentences for the single answer span. For these cases, we use SpanBERT [48] 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.

3.2.2 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

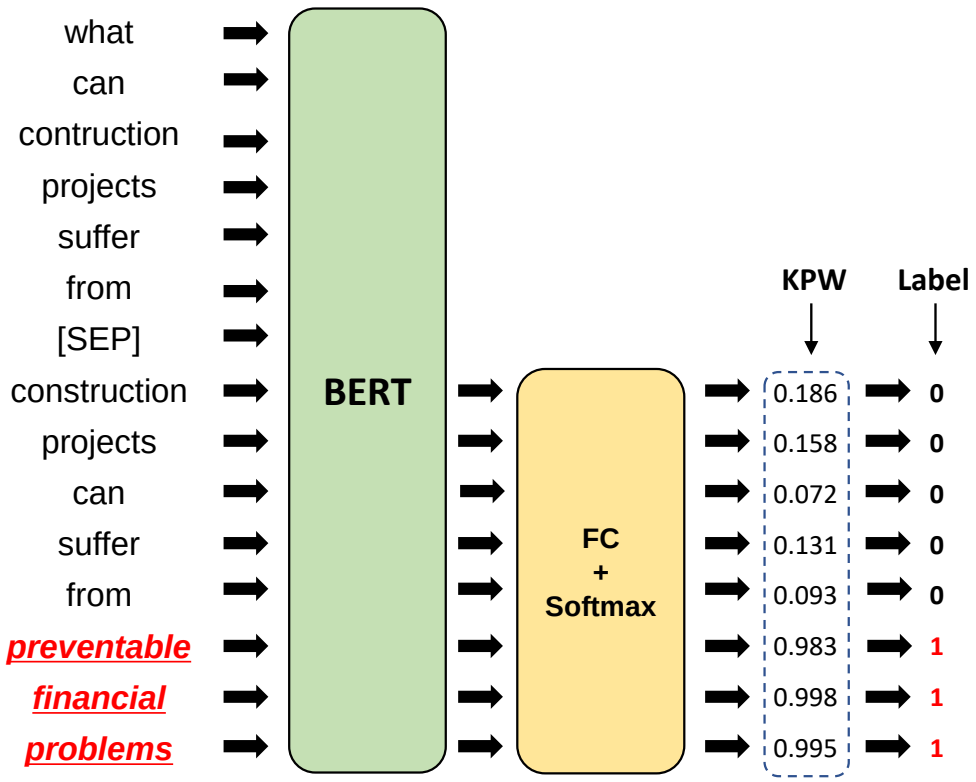


Figure 3.3: Overall architecture and an output example of KPQA. KPQA classifies whether each word in the answer sentences is in the answer span for a given question. We use the output probability KPW as an importance weight to be integrated into KPQA-metric.

that are useful for evaluating the correctness of generated answers in GenQA. The overall flow of our KPQA-metric is described in Figure 3.2. 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 3.3. 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 incorporating 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 ($P^{KPQA}_{Unigram}$) as follows:

$$P^{KPQA}_{Unigram} = \frac{\Sigma^m_{i=1}\Sigma^n_{j=1}\text{KPW}^{(Q,\hat{X})}_i \cdot I(i,j)}{\Sigma^m_{i=1}\text{KPW}^{(Q,\hat{X})}_i}, \tag{3.1}$$

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^{KPQA}_{LCS}), recall(R^{KPQA}_{LCS}) and F1($F1^{KPQA}_{LCS}$), as follows:

$$P^{KPQA}_{LCS} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma^m_{i=1}\text{KPW}^{(Q,\hat{X})}_i}, \tag{3.2}$$

$$R^{KPQA}_{LCS} = \frac{LCS^{KPQA}(X,\hat{X})}{\Sigma^n_{i=1}\text{KPW}^{(Q,X)}_i}, \tag{3.3}$$

$$F^{KPQA}_{LCS} = \frac{(1+\beta^2)R^{KPQA}_{LCS}P^{KPQA}_{LCS}}{R^{KPQA}_{LCS} + \beta^2 P^{KPQA}_{LCS}}, \tag{3.4}$$

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:

$$LCS^{KPQA}(X,\hat{X}) = \Sigma^m_{i=1} I_i \cdot \text{KPW}^{(Q,\hat{X})}_i, \tag{3.5}$$

where I_i is an indicator function which is 1 if each word is in the LCS and 0 otherwise. β is defined in [35].

BERTScore-KPQA Similar to ROUGE-L-KPQA, we compute BERTScore-KPQA using KPW. We first compute contextual embedding $\hat{\mathbf{x}}$ 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:

$$P_{BERT}^{KPQA} = \frac{\Sigma_{i=1}^{m} \text{KPW}_i^{(Q,\hat{X})} \cdot \max_{x_j \in x} \mathbf{x_i}^T \hat{\mathbf{x}}_\mathbf{j}}{\Sigma_{i=1}^{m} \text{KPW}_i^{(Q,\,\hat{X})}} \tag{3.6}$$

$$R_{BERT}^{KPQA} = \frac{\Sigma_{i=1}^{n} \text{KPW}_i^{(Q,X)} \cdot \max_{\hat{x}_j \in \hat{x}} \mathbf{x_i}^T \hat{\mathbf{x}}_\mathbf{j}}{\Sigma_{i=1}^{n} \text{KPW}_i^{(Q,X)}} \tag{3.7}$$

$$F1_{BERT}^{KPQA} = 2 \cdot \frac{P_{BERT}^{KPQA} \cdot R_{BERT}^{KPQA}}{P_{BERT}^{KPQA} + R_{BERT}^{KPQA}} \tag{3.8}$$

$$P_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X, \hat{X})}{\Sigma_{i=1}^{m} \text{KPW}_i^{(Q,\hat{X})}}, \tag{3.9}$$

$$R_{LCS}^{KPQA} = \frac{LCS^{KPQA}(X, \hat{X})}{\Sigma_{i=1}^{n} \text{KPW}_i^{(Q,X)}}, \tag{3.10}$$

$$F_{LCS}^{KPQA} = \frac{(1+\beta^2) R_{LCS}^{KPQA} P_{LCS}^{KPQA}}{R_{LCS}^{KPQA} + \beta^2 P_{LCS}^{KPQA}}, \tag{3.11}$$

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:

$$LCS^{KPQA}(X, \hat{X}) = \Sigma_{i=1}^{m} I_i \cdot \text{KPW}_i^{(Q,\,\hat{X})}, \tag{3.12}$$

where I_i is an indicator function which is 1 if each word is in the LCS and 0 otherwise. β is defined in [35].

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

Table 3.1: Statistics of the generative question answering dataset.

3.3 Experimental Setup and Dataset

3.3.1 Dataset

Generating Answers

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, [33] (2019) released human judgments of correctness for two GenQA datasets, NarrativeQA [39] and SemEval-2018 Task 11 (SemEval) [49]. 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 3.1. 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 3.1. 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 [50] and MHPGM [29]

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 3.2: Performance of the model we trained to generate answers

for MS-MARCO, MTN [51] and AMF [52, 53] for AVSD. We present the performance of each model we trained in Table 3.2.

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.

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 [54], where 1 means completely wrong, and 5 means completely correct.

Instructions to Annotators The full instructions to annotators in MTurk are shown in Figure 3.4. We hire the annotators whose HIT approval rate are higher than 95% and pay $0.02 for each assignment.

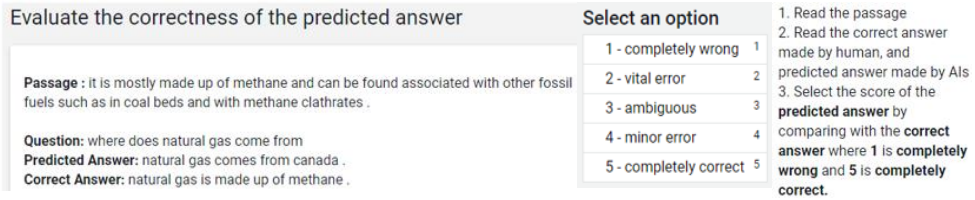


Figure 3.4: Instruction for MTurk workers

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

Table 3.3: Inter annotator agreement measured by Krippendorff's alpha(α) and the average of number of annotators for each dataset.

Filtering Noisy Workers: Some workers did not follow the instructions, producing poor-quality judgments. To solve this problem, we filter noisy ratings using the z-score, as in [55]. 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 3.3. 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 [56, 57], 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 3.3. These coefficient numbers indicate a "substantial" agreement according to one of the general guidelines [58] for kappa-like measures.

3.3.2 Implementation Details

We choose three datasets SQuAD v1.1 [46], HotpotQA [47] 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.

Hyperparameters 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 use max sequence length of 256 for the inputs of KPQA. We use AdamW [59] 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.

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.

Evaluation Methods To compare the performance of various existing metrics and our 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 [33].For all of the correlation coefficients we computed in the paper, we

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 3.4: 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).

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.

3.4 Empirical Results

3.4.1 Comparison with Other Methods

Performance Comparison: We present the correlation scores for the baseline metrics and KPQA-augmented ones for multiple datasets in Table 3.4. 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

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

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 3.6. On the other hand, our KPQA integrated metric assigns weights 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 3.4.

3.4.2 Analysis

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 3.5. This drop shows the effect of the negative domain shift for our KPQA-metric. However, "-KPQA$_{/\text{MARCO}}$" is still much higher than all 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 3.3. 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 3.5. 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.

Correlation Among Question Type: Since MS-MARCO provides the question type information (*PERSON*, *NUMERIC*, *DESCRIPTION*, *LOCATION*, *ENTITY*) for each

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 3.6: 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.

{*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 3.5, 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 [40]. To verify our KPQA-metric on multiple sentence answers, we collect additional 100

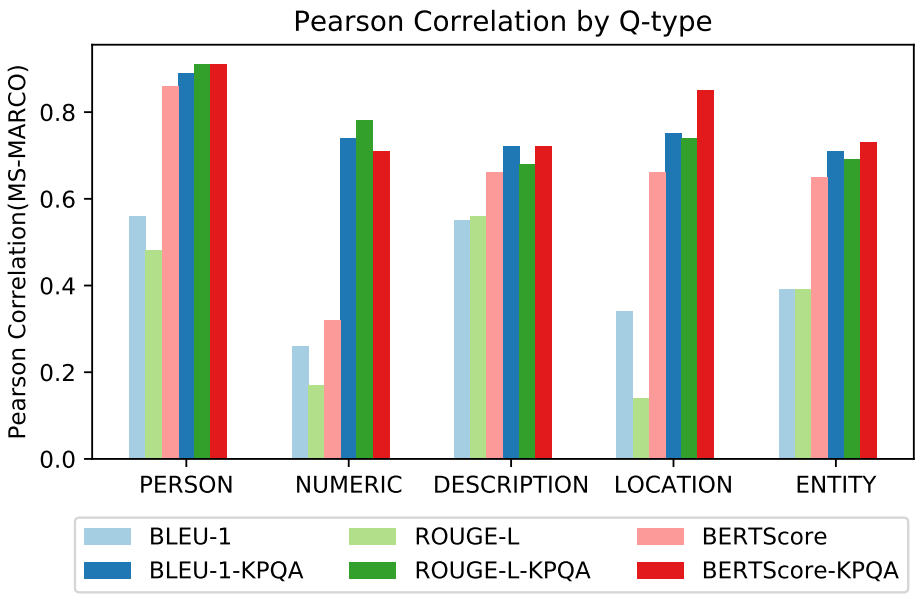


Figure 3.5: Pearson correlation coefficient among question types on MS-MARCO dataset.

human judgments for the generated answer whose answers are multiple sentences in the MS-MARCO like the example in Figure 3.6, and evaluate the various metrics on this dataset. As shown in Table 3.7, 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 3.4. 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 3.5.

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 3.6 and analyze the error cases.

From the analysis, we observe some obvious reasons for the different judgments

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

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 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

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 3.7: Correlation coefficients between various automatic metrics and human judgments of correctness for evaluating multiple sentence answers in MS-MARCO [1].

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

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 3.8 shows the percentage of

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 3.9: 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).

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.

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 3.9 show that our proposed metric outperforms other metrics in both of the GenQA models for each dataset.

3.5 Conclusion

In this study, 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.

Chapter 4

Question Generation and Question Answering for Factual Consistency Evaluation

Image captioning is a task that aims to generate a description containing the main content of a given image. The field of caption generation is prolific [60, 61], and it is, therefore, important to provide reliable evaluation metrics to compare the systems. Most of the prior works still report n-gram similarity metrics such as BLEU [4] or CIDEr [9]. However, these n-gram similarity metrics often fail to capture the semantic errors in the generated captions [62].

To overcome this limitation, we propose QACE, a radically different evaluation framework from n-gram metrics. QACE first generates questions about the candidate caption, and then checks if the answers are consistent w.r.t. either the reference or the source image. We depict QACE in Figure 4.1.

Specifically, we propose two variants of QACE, depending on what content the evaluated caption is compared to: QACE_{Ref} when it is compared to the reference, and QACE_{Img} when it is compared to the source image. QACE_{Img} has the desired feature to be reference-less, i.e., the score can be computed without requiring a gold reference.

In this reference-less setup, a Visual Question Answering (VQA) system is required to answer those questions. However, in the VQA literature [63], the task is usually seen

as a classification task on 3k pre-defined answer choices (e.g., blue, sea, or banana). Therefore, these VQA models are not general QA systems; their usage off-the-shelf for QACE_{Img} would limit the comparison to these very few pre-defined categories, which is not satisfying. To solve this issue, we also propose an abstractive VQA system Visual-T5 as a new module for QACE_{Img} that can generate free-form abstractive answers given a textual question and an image. We conduct a human evaluation of Visual-T5 and show that it is capable of generating accurate abstractive answers. Using Visual-T5, we are now able to compare the answers of the candidate caption directly with the answers of the corresponding image.

Experimental results show that our proposed QACE_{Ref} and QACE_{Img} show promising results compared to other reference and reference-less metrics on three benchmark datasets: Pascal50s [9], Composite [64] and Flickr8k [65]. Also, as shown in Figure 4.1, QACE has a natural form of interpretability through the visualization of the questions and the answers.

4.1 Related Work

Image Captioning Metrics Similar to other text generation tasks such as machine translation and summarization, n-gram similarity metrics such as BLEU, METEOR [8] and ROUGE [5] are arguably the standard in automatic evaluation. Among them, the most widely used metric is CIDEr [9] which uses TF-IDF based weighted n-gram similarity. SPICE [66] metric is based on scene graph, while more recently, BERTScore [10] compute the similarity of the contextualized embeddings. Different from prior works, we are the first to use Question Generation (QG) and Question Answering (QA) to evaluate the image captions.

Question and Answering for Evaluation [67] proposes a new method to generate informal captions that can answer the visual questions. In our work, we focus on caption evaluation using the QA systems, not on generating the captions. Several QA-based

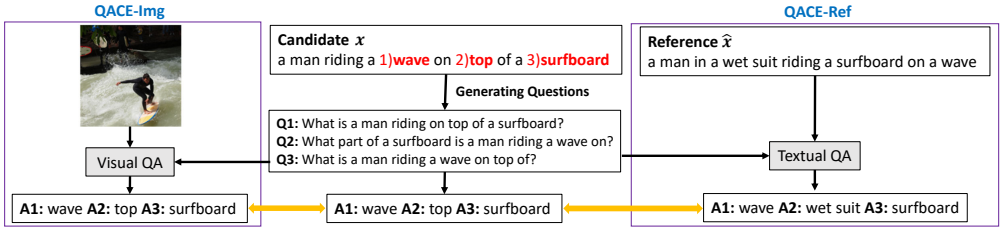


Figure 4.1: The overall flow of QACE. QACE extracts possible answer spans and generates answer-aware questions for a given candidate caption x. The VQA and TQA answer these questions given the image and reference captions, respectively. The correctness of the candidate caption is evaluated by comparing the answers.

evaluation metrics [68, 69] are recently proposed to evaluate abstractive summarization. However, all those prior works are limited to text-to-text evaluation, while our work develops a multi-modal metric.

4.2 Proposed Approach: QACE

We propose QACE, which is a QG- and QA-based framework for evaluating an image caption. As shown in Figure 4.1, QACE first extracts answer candidates (i.e., 1) wave, 2) top, 3) surfboard) from a candidate caption and generates corresponding questions. With these questions, visual-QA (VQA) and textual-QA (TQA) models answers given their context (i.e., image and reference $\hat{x}$). By comparing the answers from each source, we can directly judge the correctness of the candidate caption.

4.2.1 Question Generation

The goal of this component is to generate questions that ask the primary information of the candidate caption. Our QG model is a text-to-text generation model (i.e., T5 [70]), fine-tuned on SQuAD v2 [71] to generate answer-aware questions. Given a caption, we extract possible answer span; in particular, we focus on extracting noun phrases since they mostly contain salient information and can be easily foiled [72]. We argue that

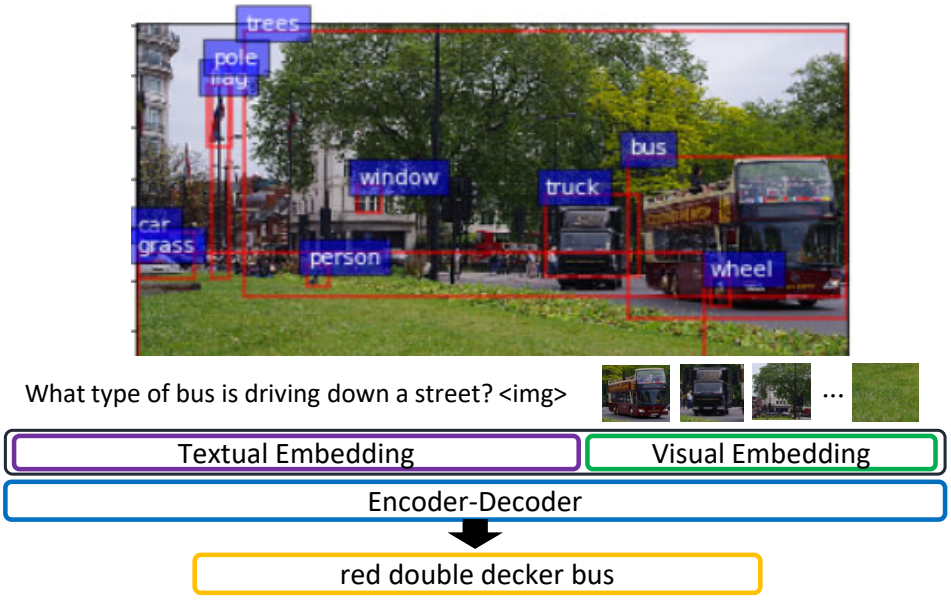


Figure 4.2: The overview of Visual-T5, an abstractive VQA model. We embed questions with additional special separation token and concatenate the visual embeddings to make inputs for T5.

questions generated on this salient information should be answered similarly from the image or the captions if they share the same information.

4.2.2 Question Answering

For QACE_{Ref}, we use a TQA model. We train T5 to answer the generated questions (see 4.2.1) with the reference captions as context. Conversely, QACE_{Img} requires a VQA model. We propose a new architecture, Visual-T5, that can generate abstractive answers given an image and a question, as opposed to the standard multiple-choice VQA.

4.2.3 Abstractive Visual Question Answering

When no reference captions are available, one of the most important parts of QACE is the VQA model that can produce correct answers. To move beyond VQA as a classification task, we are the first, to the best of our knowledge, to develop an abstractive VQA model that can generate free-form answers. Specifically, we enable multimodal encoding for T5, inspired by the previous works on adapting pre-trained language models for multimodal tasks [73]. We illustrate our proposed Visual-T5 in Figure 4.2. Based on default T5 architecture, Visual-T5 has an additional visual embedding layer that encodes regional features of the image from Faster RCNN [74]. This linear layer maps detection features to 768 dimensions, same as the dimension of textual embedding. This 768d features are therefore considered as a standard token in Visual-T5, which can encode an image and a question together.

We provide the training details including the additional output examples of our proposed abstractive VQA model, Visual-T5 in this section.

Visual Embedding We extract the regional features for each object using Faster RCNN [74]. We fixed the number of boxes to 36 and each regional feature consists of dimension 2048 and 6 additional dimensions consists of the location and the size of each box. We concatenate this additional dimensions to make dimension of 2054 for each regional feature. And single linear layer maps these 2054d features to 768d to be considered as a token in T5.

Answer Examples We provide more examples of our abstractive VQA models in Figure 4.3. We observe that many predicted answers are correct, but expressed in a different form as in the first and the second example. Also, model outputs unanswerable to the questions that are unanswerable for a given image like the third example.

Answerability We make unanswerable visual questions by randomly sampling the questions from the different images to the given image. We mixed 20% of these

Question: What does a child sleep in a bed with?

Prediction: stuffed animals
Ground-Truth : stuffed toys

Question: What day are people out on their snow boards?

Prediction: sunny day
Ground-Truth : clear blue day

Question: What vegetable is a small child holding?

Prediction: unanswerable
Ground-Truth : unanswerable

Figure 4.3: Various output examples on the evaluation set of abstractive VQA model, Visual-T5.

unanswerable questions similar to the third example in Figure 4.3 to train VQA model.

Human Evaluation We hire the workers whose locations in one of the US, UK, CA, NZ, AU to guarantee the fluency in English. We restrict the workers whose HIT approval rates are higher than 95%, and minimum hits are over 500. We pay workers more than USD $10 in an hour through several preliminary experiments on the compensation. We provide the full instructions and the interface in Figure 4.4. We compute the annotator agreement using Krippendorff's α [56]. We observe that Krippendorff's α is 0.56 that

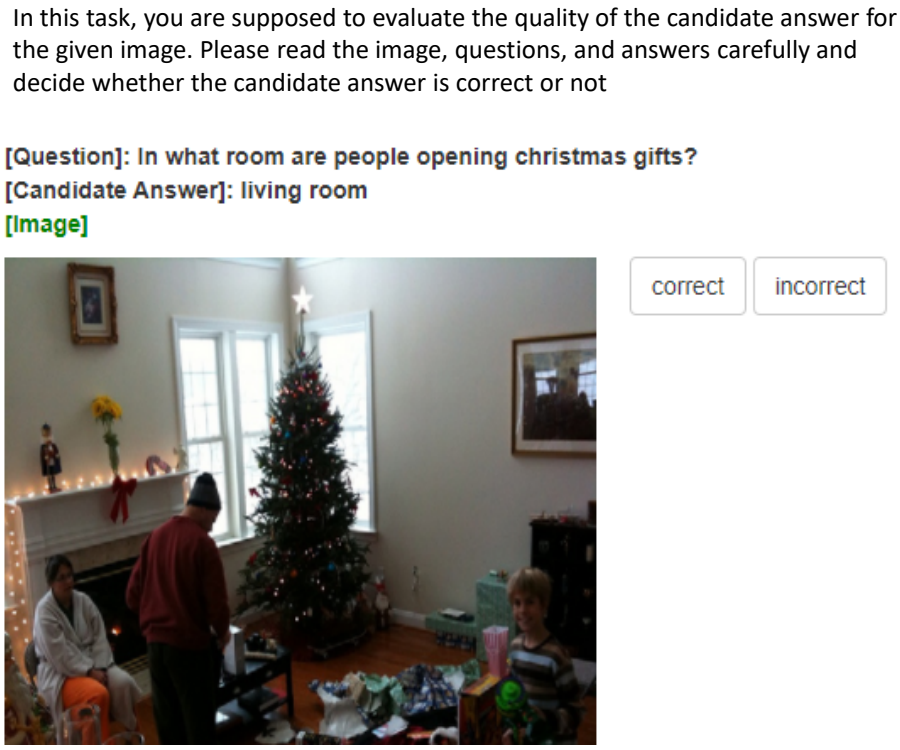


Figure 4.4: Full instructions and interface to workers for evaluating the answers of VQA model.

indicates a “moderate“ agreement according to one of the referenced guidelines [58] for kappa-like measures.

4.2.4 QACE Metric

For a given candidate caption x, We use QG to generate questions $Q = (q_1, ..., q_M)$ for all of M noun phrases of x. Then, we compare the answers for each question in Q on x with the answers on the reference source. We introduce two QACE variants, QACE_{Ref} for which the reference caption is compared, and QACE_{Img} for which the source image is compared. Using QG and QA, we compute QACE_{Ref} and QACE_{Img} as follows:

$$QACE = \frac{\Sigma_{i=1}^{M} f(QA(q_i, x), QA(q_i, ctx))}{M}, \tag{4.1}$$

where ctx corresponds to the image for QACE_{Img} and the gold reference for QACE_{Ref}, $f(A_1, A_2)$ is the function that measures the similarity between two answers A_1 and A_2. The standard metric in QA is the F1, as introduced by [46]. However, two abstractive answers can be similar but written in two different ways, limiting the effectiveness of a naive F1. Hence, in addition to the F1, we propose to use the BERTScore. Finally, we also complete the similarity metrics using the answerability of the questions for function f, in order to measure whether the question is answerable. The answerability corresponds to $1 - P_{unanswerable}$, where $P_{unanswerable}$ is the probability attributed by the model to the token unanswerable.[1] To consider all the different aspects, we use the average of three values computed using each function as the default value of QACE.

4.3 Experimental Setup and Dataset

4.3.1 Dataset

Synthetic Data Generation for VQA

As discussed in 4.2.3, relying on a VQA dataset such as VQA v2 [75] limits possible answers to a small size of pre-defined categories. To train a general and abstractive VQA model, we create synthetic abstractive VQA datasets. We generate Questions/Answers pairs using the captions in the training set of MS-COCO [76]. Specifically, we extract noun phrases from a reference caption and generate an answer-aware question using our QG model. To increase the validity of these synthetic questions, we apply the round trip consistency [77], filtering out the questions for which the QA model predicts a different answer than the extracted noun phrase. We convert these synthetic QA dataset to create $\{$*question, answer, image*$\}$ triples by concatenating the corresponding images to these captions.

[1] SQuAD v2 contains unanswerable questions, for which we associate the token unanswerable as the correct answer during training. Therefore, our QA model associates this token with the probability that the question is not answerable.

In addition, we randomly add 20% of unanswerable questions[2] to the synthetic training set, so that the model learns to judge the answerability of a given question. Through this, if a candidate caption contains any hallucinating content that is not included in the image, questions about it can be marked as unanswerable by our VQA model, as shown in the second example of Figure 4.5. This synthetic dataset enables the training of the abstractive VQA model. We report the performance of the model through a human evaluation in Section 4.4.2.

Benchmark Dataset

We evaluate our proposed metric on three benchmark datasets (i.e. human annotations), *PASCAL-50S*, *Composite* and *Flickr8k*.

PASCAL-50S provides 4k caption triplet $<A, B, C>$, where "A" is composed of 50 reference captions(A) and two candidate captions(B, C) for the given image. There are human judgments as to which "B" or "C" is more appropriate caption for a given image compared to "A".

Composite is composed of 11,985 human judgments scores range from 1 to 5 depending on the relevance between each candidate caption-image pair with 5 reference captions.

Flickr8k provides three human-expert judgments for 5,822 candidate caption-image pairs. The scores are from 1 to 4, depending on the relevance of each caption-image pair.

4.3.2 Implementation Details

For all of the results on reference based metrics we reported in the paper, we compute the average of each metric score with each reference for all of the references on each dataset.

[2]We consider an image and a question that are not paired to be unanswerable, and do negative sampling.

	Ref?	Pascal50s	Composite	Flickr8k
BLEU-4	✓	65.2	45.7	28.6
ROUGE-L	✓	67.7	47.7	30.0
METEOR	✓	**80.5**	46.6	40.3
CIDEr	✓	77.8	47.4	41.9
SPICE	✓	76.1	48.6	**45.7**
BERTScore	✓	72.0	45.6	30.5
QACE-Ref (ours)	✓	75.1	49.3	40.5
F1	✓	57.5	**55.1**	9.2
BERTScore	✓	76.4	46.0	30.9
Answerability	✓	71.6	47.3	39.0
-Perplexity	✗	46.8	1.7*	10.1
VIFIDEL	✗	69.0	13.1	**33.6**
QACE-Img (ours)	✗	**70.0**	**19.1**	29.1
F1	✗	62.0	12.5	27.3
BERTScore	✗	65.9	12.8	27.1
Answerability	✗	**74.5**	15.7	27.8

Table 4.1: First column represents the accuracy of matches between human judgments in PASCAL50s. Columns 2 to 3 show the Kendall Correlation between human judgments and various metrics. All p-values in the results are < 0.05 except for *.

4.4 Empirical Results

4.4.1 Comparison with Other Methods

We present the experimental results for all three datasets in Table 4.1. For the reference-aware metrics, QACE_{Ref} shows best results on Composite and comparable to the best

metrics for Pascal50s and Flickr8k, indicating the relevance of a QA based metric to evaluate image captioning.

For the reference-less metrics, all the correlations are lower this time, showing the difficulty of evaluating the captions without reference. Nonetheless, among these metrics, QACE_{Img} shows the best results for Pascal50s and Composite and comparable results in Flickr8k. For Flickr8k, we found that more than half of the human judgments of the candidate captions are less than 0.2 as 0 to 1 scale. In other words, most of the captions in this dataset are totally not related to the image. For this reason, most of the generated questions are unanswerable for an image and we explain that this leads to relatively lower performance of QACE_{Img} in Flickr8k compared to other metrics.

Furthermore, We investigate the independent contribution of each answer similarity function, f, in computing QACE and present the results in Table 4.1 (note that default QACE-Img uses the mean of F1, BERTScore and answerability). The table reveals that each similarity function has a different aspect, and averaging three results suggests the best performance for two of three datasets.

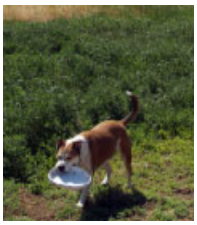
4.4.2 Analysis

VQA Model Performance

Visual-T5 is one of the main components of QACE_{Img}. Since it can generate free-form answers, its automatic evaluation is challenging. We therefore conduct a human evaluation on 200 examples randomly sampled from the test set. We hire three annotators to judge whether the generated answer is correct or not given the image. On the majority vote from three annotators, VQA model correctly answers for the average 69% of the examples. Among these 69% correct answers, half of them were written differently from the original answer, showing that our model can generate abstractive answers.

Candidate: a man^{A1} is standing on a sunny beach^{A2} **(Human: 1.0)**
Reference: a man walks down the beach near the ocean
Q1: What is standing on a sunny beach?
Q2: What is a man standing on?

Ref	**A1**:man **A2:**beach	$QAEC_{Ref}$: 0.88
Img	**A1:**man **A2**:sand	$QAEC_{Img}$: 0.79



Candidate: a cow^{A1} is standing in a field^{A2} of grass^{A3} **(Human: 0.2)**
Reference: a dog with a frisbee standing in the grass
Q1: What animal is standing in a field of grass?
Q2: What is a cow standing in?
Q3: What type of field is a cow standing in?

Ref	**A1:**dog **A2:**grass **A3**:grass	$QAEC_{Ref}$: 0.60
Img	**A1:**dog **A2**:unanswerable **A3:**grassy field	$QAEC_{Img}$: 0.47

Figure 4.5: Case study on QACE metric. Human judgments are normalized to between 0 and 1.

Case study

Different from the previous metrics, QACE can be easily interpreted through the visualization of the generated questions and the following answers as shown in Figure 4.5. In the first example, we observe that the second question is answered differently by the VQA model (*sand* VS *beach*). Despite, the answer itself being correct - it is true that the man is standing on the sand - it results in a lower score for QACE_{Img} compared to QACE_{Ref}. This emphasizes the importance to use other similarity metrics than the F1 when comparing two answers (see Section 4.2.4). For instance, BERTScore should be able to consider closer *sand* and *beach* than *sand* and a random word.

The second example is very illustrative: for the first question, both TQA and VQA answer *dog*, hence detecting an error in the candidate caption that talks about a *cow*. The second question refers to the *cow*, which makes it ambiguous. The VQA model considers it as *unanswerable*, while the TQA model correctly answers *grass*. Following this study, we expect that QACE_{Img} can be improved through a finer answer comparison method in future work.

4.5 Conclusion

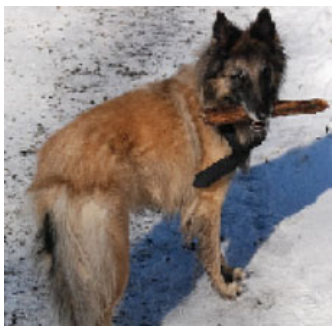
In this paper, we propose QACE, a captioning metric that directly compares each content in the candidate caption with either the source image or a gold reference caption by asking questions. To enable asking questions directly on the source image, we introduce Visual T5, an abstractive VQA model to generate free-form visual answers, for which we report strong results based on a human evaluation. Our proposed metric can be applied in both reference and reference-less settings. It holds high explainability and compares favorably to the state-of-the-art in terms of correlations with human judgments.

Chapter 5

Rule-Based Inconsistent Data Augmentation for Factual Consistency Evaluation

Image captioning [78] aims to generate a short description that explains the main content in the given image with a natural language. While there have been many advances for image captioning systems [60, 79, 61, 80] and captioning datasets [81, 82], few studies [9, 66, 83, 84, 85] have focused on assessing the quality of the generated captions. Especially, most of the evaluation metrics only use reference captions to evaluate the caption although the main context is an image. However, as shown in the examples Figure 5.1, since there are many possible reference captions for a single image, a candidate caption can receive completely different CIDEr [9] scores depending on the type of reference [86]. Also, like the candidate caption in second example, candidate caption may get a high score depending on the reference caption even though the fact *"kicking"* is completely wrong.

Due to difficulties of evaluating an image caption caused by diverse nature of image captions, reference-based metrics usually use multiple human written reference captions which are difficult to obtain. To overcome this limitation, we propose UMIC, an **U**nreference **M**etric for **I**mage **C**aptioning, which is not dependent on the reference captions and only use an image-caption pair to evaluate a caption. We develop UMIC



Reference 1: A dog standing in the snow with a stick in its mouth.
Reference 2: A little dog holding sticks in its mouth.

Candidate: A dog standing on the snow with a dog.

CIDEr with Reference 1: 3.166
CIDEr with Reference 2: 0.281

Human Judgments : 1.875 out of 5

Reference 1: There is a man running on a field with a soccer ball.
Reference 2: The soccer player is bringing back the ball into play.

Candidate : A young man **kicking** a soccer ball on a field.

CIDEr with Reference 1 : 2.869
CIDEr with Reference 2 : 0.782

Human Judgments : 3 out of 5

Figure 5.1: An example where the metric score for a given candidate caption varies significantly depending on the reference type.

upon UNITER [87] which is a state-of-the-arts pre-trained representation for vision-and-language tasks. Since UNITER is pre-trained to predict the alignment for large amounts of image-text pairs, we consider that UNITER can be a strong baseline for developing an reference-less metric. We fine-tune UNITER via contrastive learning, where the model is trained to compare and discriminate the ground-truth captions and diverse synthetic negative samples. We carefully prepare the negative samples that can represent most of the undesirable cases in captioning, such as *grammatically incorrect*, *irrelevant to the image*, or *relevant but have wrong keyword*.

When evaluating the metric's performance, it is required to compare the correlations between human judgments and the metric's evaluation score for given datasets. We choose three standard benchmark datasets (i.e., Composite [64], Flickr8k [65],

PASCAL-50s [9]) and further analyze the quality of the dataset. Interestingly, we found that there exist critical issues in the benchmark datasets, such as poor-label or polarized-label. To perform a rigorous evaluation as well as stimulate the research in this area, we collect new 1,000 human judgments for the model-generated caption. Finally, we evaluate our proposed metric on four benchmark datasets, including our new dataset. Experimental results show that our proposed reference-less metric is highly correlated with human judgments than all of the previous metrics that use reference captions. Overall, our main contributions can be summarized as follows:

- We propose an image captioning metric UMIC that does not utilize reference captions through contrastive learning.
- To evaluate the quality of the proposed metric, we introduce a new benchmark dataset CapEval1k composed of high-quality human judgments for 1k captions generated from four recent captioning models.
- We demonstrate that our proposed metric has a higher correlation with human judgments than the previous metrics in four benchmark datasets including CapEval1k.
- We verify the effectiveness of our metric in various aspects such as case study and an observation on attention map.

5.1 Related Work

Image Captioning Metrics Following other text generation tasks such as dialogue systems and machine translation, n-gram similarity metrics such as BLEU [4], ROUGE [5] and METEOR [8] are widely used to evaluate an image caption. Especially, CIDEr [9], which weights each n-gram using TF-IDF, is widely used. SPICE [66] is a captioning metric based on scene graph. BERTScore [10], which computes the similarity of the contextualized embeddings, are also used. BERT-TBR [86] focuses on the variance in multiple hypothesis and ViLBERTScore (VBTScore) [84] utilizes ViLBERT [88] to improve BERTScore.

Different from these metrics, VIFIDEL [85] computes the word mover distance [11] between the object labels in the image and the candidate captions, and it does not require reference captions. Similar to VIFIDEL, our proposed UMIC does not utilize the reference captions. However, UMIC directly uses image features and evaluates a caption in various perspectives compared to VIFIDEL.

Quality Estimation Quality Estimation (QE) is a task that estimates the quality of the generated text without using the human references and this task is same as developing an unreferenced metric. QE is widely established in machine translation (MT) tasks [89, 90, 91]. Recently, [92] introduces a large scale human ratings on image-caption pairs for training QE models in image captioning tasks. Our work also trains caption QE model, (i.e. unreferenced captioning metric) but we do not use human ratings to train the metric. Instead, we create diverse synthetic negative samples and train the metric with these samples via ranking loss.

5.2 Proposed Approach: UMIC

We propose UMIC, an unreferenced metric for image captioning using UNITER. We construct negative captions using the reference captions through the pre-defined rules. Then, we fine-tune UNITER to distinguish the reference captions and these synthetic negative captions to develop UMIC.

5.2.1 Modeling

Since UNITER is pre-trained to predict the alignment of large amounts of image-text pairs, we use the output of the layer that predicts this alignment as the baseline of UMIC to be fine-tuned. Specifically, we compute the score of a caption $\mathrm{S}(I, X)$ for given image $I = (i_1, ..., i_N)$ and $X = (x_1, ..., x_T)$ as follows.

We first compute the contextual embedding for I and X using UNITER to get the

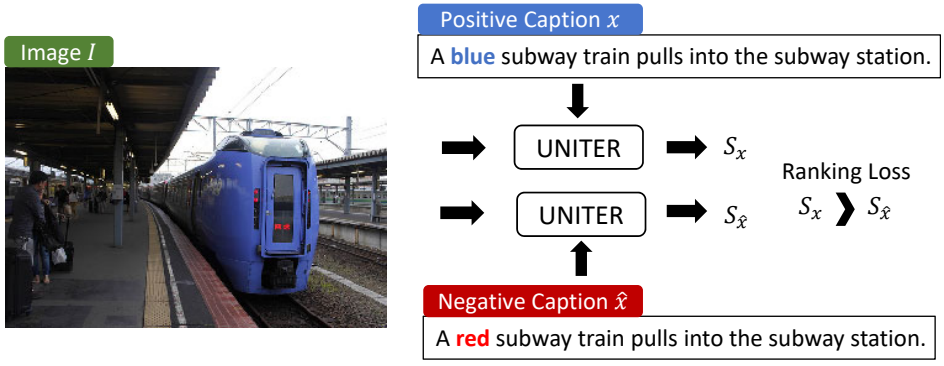


Figure 5.2: Overall training procedure of UMIC. Given an image I, a positive caption x and a negative caption $\hat{x}$, we compute the score of each image-caption pair S_x and $S_{\hat{x}}$ using UNITER respectively. Then, we fine-tune UNITER using raking loss that S_x is higher than $S_{\hat{x}}$.

joint representation of image and text as follows.

$$i_{[CLS]}, i_1, ..., i_N, x_1, ..., x_T = \text{UNITER}(I, X), \tag{5.1}$$

where $i_{[CLS]}$ is a joint representation of the input image and input caption. Then we feed it into a single fully-connected layer to get a score as follows.

$$\text{S}(I, X) = sigmoid(W i_{[CLS]} + b), \tag{5.2}$$

where W and b are trainable parameters.

5.2.2 Negative Samples

To model negative captions, we observe the captions' common error types in the model-generated captions. Specifically, we pick 100 bad captions in the order of whose human judgments are low in Composite and Flickr8k, respectively. Then, we categorize the main errors into three types:*relevant but have wrong keywords, totally irrelevant to the image, grammatically incorrect.* To model most imperfect captions including these frequent type errors, we prepare negative captions as follows.

Target Image

Similar Image

Original: a woman hugging a girl who is holding a suitcase

Substitiution: a boy hugging a girl who is holding a suitcase

Random(Hard Negative): a very small cute child by a suitcase

Repetition & Removal: a woman hugging a girl is holding a suitcase suitcase

Figure 5.3: An example of the generated negative captions for the left image to train UMIC. Hard negative caption is one of the reference captions for the right image which is similar to the left image.

Substituting Keywords To mimic the captions that are relevant but have wrong keywords, as in the example of Figure 5.2, we randomly substitute 30% of the words in the reference captions and use them as negative samples like Figure 5.3. The motivation we choose 30% is that the average length of the generated caption is about 10 words and the number of keywords is usually around three. We only substitute *verb*, *adjective*, and *noun*, which are likely to be keywords since they are usually visual words. Also, we substitute them with the words with the same POS-Tags using the pre-defined dictionaries for the captions in the training set to conserve the sentence structure.

Random Captions We randomly sample captions from other images and use them as negative samples to generate totally irrelevant captions for the given image. Also,

similar to the image-text retrieval task, we use hard-negative captions, which are difficult to be discerned, with a probability of 50%. Specifically, we utilize the captions of the images similar to the given images using the pre-trained image retrieval model. We get negative captions that are the captions of the similar image sets computed by image-text retrieval model VSE++ [93] as in [94]. Then, we sample the captions in the reference captions of the Top-3 similar image sets like the example in Figure 5.3.

Repetition & Removal We find that some of the captions have repeated words or have incomplete sentences. Hence, we randomly repeat or remove some words in the reference captions with a probability of 30% in the captions to generate these kinds of captions. Specifically, we choose to repeat or remove with a probability of 50% for the sampled word.

Word Order Permutation We further generate negative samples by randomly changing the word order of the reference captions, so that the model sees the overall structure of the sentence, not just the specific visual words.

5.2.3 Contrastive Learning

Using the negative captions generated by the above rules, we fine-tune UNITER via contrastive loss for positive caption X and negative caption $\hat{X}$ as follows.

$$Loss = max(0, M - (\mathrm{S}(I, X) - \mathrm{S}(I, \hat{X}))), \tag{5.3}$$

where M is the margin for the ranking loss, which is a hyperparameter. We make each batch composed of one positive caption and four negative captions that are made by each negative sample generation technique.

5.3 Experimental Setup and Dataset

5.3.1 Dataset

We briefly explain the previous benchmark datasets for captioning metrics and analyze the problems for two of these datasets, Flickr8k and Composite. Also, we introduce a new benchmark dataset to alleviate the addressed problems.

Commonly Used Datasets

Composite consists of 11,985 human judgments for each candidate caption generated from three models and image pair. This dataset's human judgments range from 1 to 5, depending on the relevance between candidate caption and image.

Flickr8k provides three expert annotations for each image and candidate caption on 5,822 images. The score ranges from 1 to 4, depending on how well the caption and image match. All of the captions in this dataset are reference captions or captions from other images.

PASCAL50s contains 1,000 images from UIUC PASCAL Sentence Dataset with 50 reference captions for each image. Different from other datasets, this dataset provides 4,000 caption triplet $<A, B, C>$ composed of 50 reference captions(A) and two candidate captions(B, C) for the given image. There are human annotated answers to which is more similar to "A", "B" or "C".

Problems in Flickr8k and Composite

We investigate the human judgments in Flickr8k and Composite, and visualize the distributions of judgment scores for two datasets, Flickr8k and Composite in Figure 5.4, and find several problems.

For the Flickr8k, most of the scores are less than 0.2 since the candidate captions

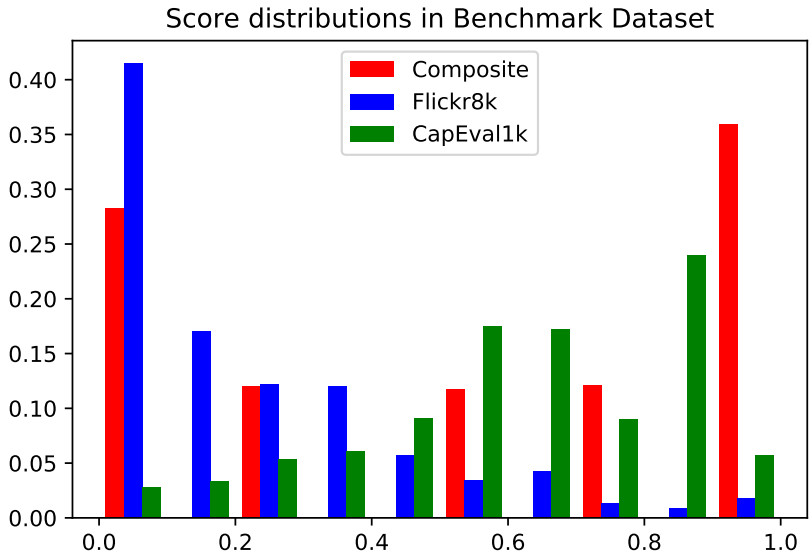


Figure 5.4: Score distributions of human judgments in Composite, Flickr8k and our proposed CapEval1k dataset. All scores were normalized from 0 to 1.

were sampled by an image retrieval system from a reference caption pool, not model-generated captions. Therefore, most captions are not related to images and differ significantly from the model-generated captions. We argue that this naive configuration is not enough to distinguish the performance of the metric precisely.

For the Composite, most of the scores are placed near 0 or 1. We explain this because only a single annotator annotates each sample's score resulting in biased output. We also manually investigated the captions and found that the captions are coarsely generated. Note that the captions for this dataset were generated by the old model [95, 64]. For these reasons, we conclude that additional benchmark dataset is necessary to evaluate the captioning metrics.

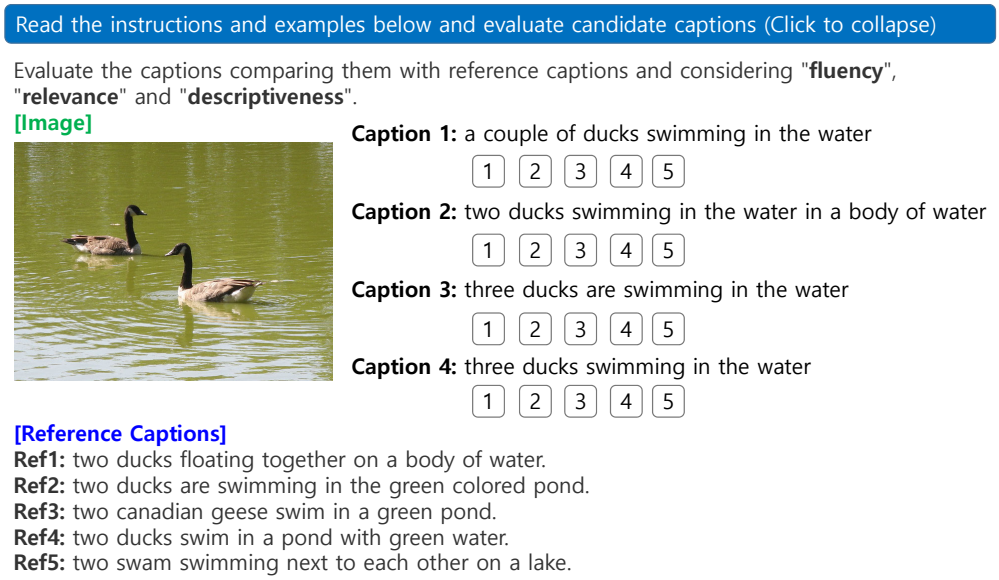


Figure 5.5: Annotation interface and short instructions for captioning evaluation task.

CapEval1k Dataset

To alleviate the addressed issues in Flickr8k and Composite, we introduce a new dataset CapEval1k, which is composed of human judgments for the model-generated captions from four recently proposed models: Att2in [79], Transformer [96], BUTD [61] and AoANet [80]. Different from Flickr8k and Composite, we ask each annotator to evaluate the captions by considering three dimensions: *fluency*, *relevance*, *descriptiveness*. We hire 5 workers who are fluent in English for each assignment from Amazon Mechanical Turk and use the average score.

Instructions to Annotators The interface and instructions to annotators in MTurk are shown in Figure 5.5 and Figure 5.6. We request the worker to evaluate four captions at once in a single assignment so that the worker can consider the difference among the captions.

[Overview]
In this task, you are supposed to evaluate the quality of the caption for the given image.
Please read the image and the captions carefully and assign the score for each caption considering three criterias.

[Instructions]
1. Read the candidate captions, reference captions and see the given image.
2. Evaluate the four candidate captions considering three criterias(refer to the negative examples below) and comparing them to the reference captions
- Note that reference captions are not always perfect.

Criterias & Common negative examples in the captions
Please consider 3 things comprehensively and rate the overall score for the capture.
(1) Fluency
Whether the caption is fluent, natural and grammatically correct
Ex) Grammatically correct but strange
a plate of food and food
(2) Relevance
Whether the sentence correctly describes the visual content and be closely relevant to the image.
Ex) Relevant/Minor Mistake: relevant but tiny parts are wrong
a plate of fruits and a crepe on a grey dish
(3) Descriptiveness
Whether the sentence is a precise, informative caption that describes important details of the image.
Ex) Too General Capton
a plate of fruits

Figure 5.6: Full instructions for the captioning evaluation task. We provide an image and five reference captions to the workers and request them to evaluate four captions.

Inter-annotator Agreement We compute the annotator agreement using Krippendorff's α [56]. We observe that Krippendorff's α is 0.37 that indicates a "fair" agreement according to one of the general guidelines [58] for kappa-like measures.

Worker Pool & Pay We hire the annotators whose locations in one of the US, UK, CA, NZ, AU. We restrict the workers whose HIT approval rates are higher than 96%,

and minimum hits are over 5000. We pay workers more than USD $10 in an hour through several preliminary experiments on the compensation.

Score Distribution Since our CapEval1k dataset is composed of annotations via recently proposed models, the overall scores are relatively higher than other datasets as shown in Figure 5.4. Compared to other datasets, CapEval1k contains the annotators' comprehensive judgment across multiple dimensions in evaluating the quality of the generated captions, so we can see that the score distribution score is not concentrated in a particular area.

5.3.2 Implementation Details

Hyperparameters We use the pre-trained UNITER-base with 12 layers in the official code provided by the authors [87][1]. We use the COCO dataset [81] to fine-tune UNITER through ranking loss. We use the train and validation split of COCO dataset in [87]. The number of the training set is 414k, and the validation set is 25k. We set the batch size of 320, learning rate of 2e-6, and fine-tune UNITER for a maximum of 4k steps. We select the model that shows the minimum loss in the validation set. We set margin M as 0.2 in the ranking loss. We repeat training 5 times for each best-performing model.

Computing Infrastructure We use AMD Ryzen Threadripper 2950X (3.50 GHz) with GeForce GTX 2080 Ti for the experiments. The software environments are Python 3.6.8 and PyTorch 1.1.0.

Average runtime for each approach Each epoch of our training UMIC on average takes 20 minutes using a single GPU. For evaluation, it takes a minute.

Correlation Coefficient We compute Kendall-C for Flickr8k [65], since we could produce the similar results for most of the previous papers. And we compute Kendall-B

[1] https://github.com/ChenRocks/UNITER

Metric	Flickr8k	Composite	CapEval1k	PASCAL50s
BLEU-1	0.274	0.406	0.233	74.3
BLEU-4	0.286	0.439	0.238	73.4
ROUGE-L	0.300	0.417	0.220	74.9
METEOR	0.403	0.466	0.288	78.5
CIDEr	0.419	0.473	0.307	76.1
SPICE	0.457	0.486	0.279	73.6
BERTScore	0.396	0.456	0.273	79.5
BERT-TBR	0.467	0.439	0.257	**80.1**
VBTScore	**0.525**	**0.514**	**0.352**	79.6
VIFIDEL	0.336	0.191	0.143	70.0
UMIC	**0.468**	**0.561**	**0.328**	**85.1**
UMIC$_{\text{C}}$	0.431	0.554	0.299	84.7

Table 5.1: Columns 1 to 3 represent Kendall Correlation between human judgments and various metrics on Flickr8k, Composite and CapEval1k. All p-values in the results are < 0.01. The last column shows the accuracy of matches between human judgments in PASCAL50s.

for Composite [64] and CapEval1k. For Composite, we use five references and some of the candidate captions in this dataset are exact same with one of the references.

5.4 Empirical Results

5.4.1 Comparison with Other Methods

We compute caption-level Kendall's correlation coefficient with human judgments for the Composite, Flickr8k, and our proposed CapEval1k. For the PASCAL50s, we compute the number of matches between human judgments for each candidate caption pair. For all of the reference based metrics, we use five reference captions and then get average score among the five references except for BERTScore where we use maximum.

We present the experimental results for all four datasets in Table 5.1. We show that although UMIC does not utilize any reference captions, UMIC outperforms the

baseline metrics except for VBTScore in all of the datasets that depend on multiple references. We also report the strong unreferenced baseline UMIC$_{\text{-C}}$, which is directly using the pre-trained weights from UNITER without contrastive learning. Interestingly, UMIC$_{\text{-C}}$ shows a higher performance than most of the metrics. This high performance shows that pre-trained image-text matching layer of UNITER already has a good representation for evaluating image captions. Especially for Composite, both UMIC and UMIC$_{\text{-C}}$ significantly outperform baseline metrics. We explain this in the polarized distribution of human judgments as we explained in Section 5.3.1. In other words, the relevance of most image-caption pairs in this dataset is too obvious so that UNITER can easily distinguish them. However, while UMIC shows higher performance on all datasets, UMIC$_{\text{-C}}$ shows relatively low performance on Flickr8k and CapEval1k. And this demonstrates the effectiveness and generalization ability of our contrastive learning objective to develop UMIC.

Also, we can observe that the performance of each metric is relatively low and the rank of each metric changes in our proposed CapEval1k dataset. We explain that this is because the captions in CapEval1k are relatively difficult to be evaluated since the score distribution is not biased as explained in Section 5.3.1.

5.4.2 Analysis

Case Study

We visualize one sample each showing the strengths and weaknesses of UMIC in Figure 5.7. In the above example, the candidate caption is partially relevant to the image, but the single word “three” in the caption is totally incorrect since there are only “two” giraffes in the image. And this leads to a low human judgment of 0.2. Nevertheless, unlike our UMIC, widely used metrics and UMIC$_{\text{-C}}$ give this caption a high score due to the many words overlaps or missing the keywords. The bottom example shows one of the error cases and the limitations of our proposed method. Since the detection model in UMIC could not recognize the important object like the “baseball bat”, UMIC outputs

References
- two giraffe standing next to each other in a field.
- two giraffes are climbing a hill with mountains in the background.

Candidate
- **three** giraffes standing in a field of grass

BLEU1: 0.324	**ROUGE-L**: 0.320	**METEOR**: 0.173	**CIDER**: 0.866
SPICE: 0.289	**UMIC**: 0.352	**UMIC**$_{/-C}$: 0.770	**Human**: 0.200

References
- a person breadking a bottle with a baseball bat
- a boy in yellow shirt swinging a baseball bat

Candidate
- a man swinging a **baseball bat** at a ball

BLEU1: 0.360	**ROUGE-L**: 0.354	**METEOR**: 0.176	**CIDER**: 1.205
SPICE: 0.192	**UMIC**: 0.094	**UMIC**$_{/-C}$: 0.062	**Human**: 0.450

Figure 5.7: Case study for the various metrics on candidate captions in CapEval1k Dataset. Human judgments are normarlized from 0 to 1.

very low score.

Interpreting the Output

To interpret the output of UMIC, we visualize the attention maps to interpret the output of UMIC as shown in Figure 5.8. To compute attention weights, we first find the head that is specialized in inter-modaility attention between the image and the caption. Following [87], we find a *reversed block* that can show the cross-modality relations between a caption and an image by computing the ratio of the attention weights in cross-modality alignment for each head. We choose the third head that has the highest attention ratio in cross-modal alignment for interpreting the output of the metric. Then,

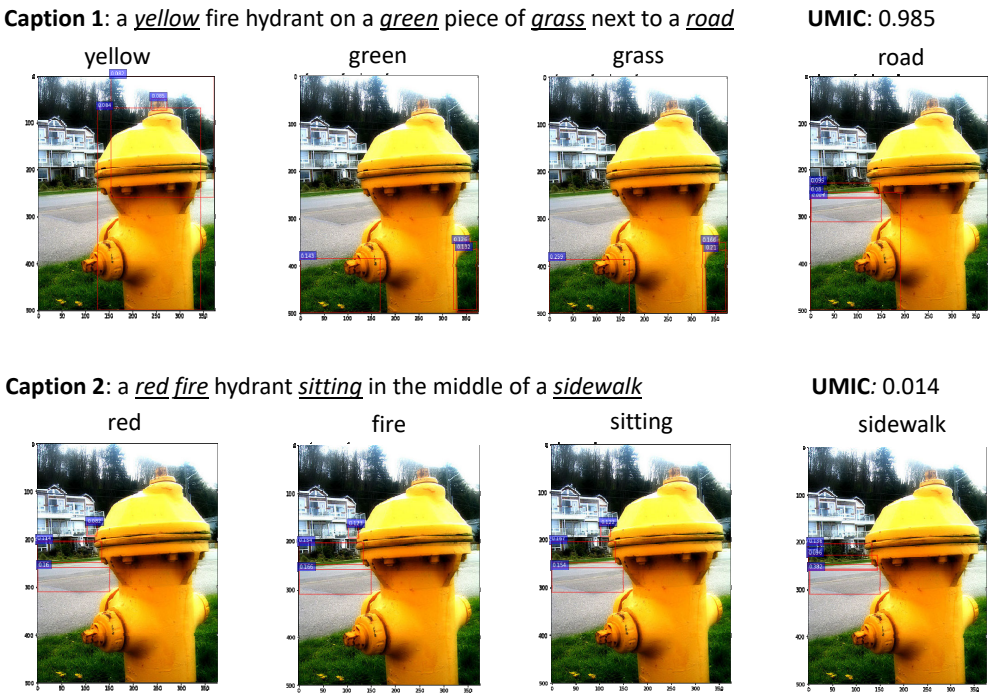


Figure 5.8: Text-to-image attention map visualization of our metric on two different captions for a same image. We represent the top-3 regions in the images according to the attention weights with specific words in the caption.

we pick top-3 regions among detection features in the image that have higher attention weights to the specific words as shown in Figure 5.8. For the first example that gets higher UMIC score, we observe that attention weights on each word are assigned to desired image regions. For example, for the *"yellow"*, yellow parts in the fire hydrant are aligned to the word. However, for the wrong captions in the second example, the aligned regions are usually not relevant to the words, and each word point the same regions in the image. We explain that *red* fire hydrant does not exist in the image and the model cannot find the proper region in the image for this case.

5.5 Conclusion

In this paper, we propose UMIC, an unreferened metric that does not require any reference captions for image captioning task through contrastive learning in one of the vision-and-language pre-trained models UNITER. To train a metric, we propose several methods such as keyword substitution and retrieving hard negative captions to prepare negative captions that imitate the wrong captions in the captioning systems. Also, we analyze the problems widely benchmark datasets for image captioning metrics and introduce a new benchmark dataset CapEval1k that relieve the issues such as polarity in previous datasets. Experimental results on four benchmark datasets, including our proposed CapEval1k, show that UMIC outperforms previous metrics. We also validate the properties of UMIC through an observation on the attention map. In future work, we will study a way to integrate our metric to the training step in image captioning to improve the quality of the generated caption following [97].

Chapter 6

Inconsistent Data Augmentation with Masked Generation for Factual Consistency Evaluation

As textual content available on- and offline explodes, automated text summarization is becoming increasingly crucial [98]; with the advances in neural text generation methods, abstractive summarization systems that generate paraphrases are quickly replacing extractive ones that simply select essential sentences from the source text [99]. While abstractive summaries can be more coherent and informative (given the same length) than their extractive counterparts, they frequently contain information inconsistent with the source text. This is a critical issue, as it directly affects the reliability of the generated summaries. [2, 100, 101].

Unfortunately, existing approaches to identify such factual inconsistency without constructing new resources have not been satisfactory. Directly measured similarity between the summary and its source text—using popular n-gram similarity metrics such as ROUGE [35] and BLEU [34]—exhibits low correlation with human judgments for factual consistency. Also, leveraging related tasks—such as natural language inference (NLI) [19] and fact verification [20]—is not ideal. This is because these tasks aim to identify relations between two sentences, whereas factual consistency checking involves a multi-sentence summary and an even longer source text [102, 103].

Article: Guus Hiddink, the Russia and Chelsea coach, has had much to smile about in his 22-year managerial career. ,. . . , Enjoying success around the world – at different levels with different players in different cultures – has made Guus Hiddink one of the most admired bosses around. ,. . . , Hiddink's resume includes stints in other high-pressure jobs such as Fenerbahce, Valencia and Real Madrid. ,. . . , But the straight-speaking Dutchman is loyal to the project he has in charge of the Russian national side and insists he will leave Chelsea at the end of the season regardless.

Reference Summary: Born in 1946, Hiddink has become one of the best managers in the world . Dutchman has enjoyed huge success at club and international level. He's currently coach of Russia and is in charge of Chelsea until end of the season.

Mask-and-fill Summary Without Article:
Born in 1946, *Dutchman* has become one of *the most respected politicians* in the world. Dutchman is enjoyed *success at the Olympics and World Cup*. He's currently the *President of Russia* and is in charge of *the country* until the end of the season.

Mask-and-fill Summary With Masked Article:
Born in 1946, *Hiddink* has become one of *the most admired managers* in the world. Dutchman has enjoyed *successful spells* at *Chelsea and Real Madrid*. He's currently *manager of Russia* and is in charge of *the country* until the end of the season.

Figure 6.1: An example of generated negative summary using masked article. Spans that are highlighted are masked when generating the negative summary. Note that red spans are factually inconsistent with the given article and blue spans are factually consistent.

A remaining solution is to train a factual consistency classifier with a dataset specifically constructed for this purpose. Note that ***positive summaries*** are readily available. That is, the reference summaries from existing text summarization datasets can be assumed to be factually consistent with the respective source texts. Thus, the main challenge is in generating effective ***negative summaries***, i.e., summaries that are

factually inconsistent with the source text. Recent works generate negative summaries by simply replacing keywords in the reference summaries or sentences extracted from the source texts [22, 23]. This, however, results in negative summaries that significantly diverge from the source texts and positive summaries, which is not ideal for training factual consistency classifiers. For instance, Figure 6.1 shows that *coach* in the reference summary is changed to *President of Russia*, which is an inconsistency that is too obvious.

In this study, we propose a novel method, Masked-and-Fill with Masked Article (MFMA), where parts of the source text and reference summary is masked and later inferred to generate a plausible but factually inconsistent summary. Experiments on seven benchmark datasets demonstrate that factual consistency classifiers trained on negative summaries generated with our method mostly outperform existing models and show competitive correlation with human judgment. We also analyze the characteristics of the negative summaries generated. Our main contributions are as follows:

- We propose a novel negative summary generation method for training factual consistency classifiers for abstractive summaries.
- We show the efficacy of our method on seven benchmark datasets using classification performance and correlation with human judgement.
- We analyze the characteristics, such as affinity and diversity, of the negative summaries generated using our method.

6.1 Related Work

Factual Inconsistency in Summarization Systems

Previous works [101, 100, 2] have studied the factual inconsistency in abstractive summarization systems. Especially, [2] demonstrates that 30% of the model generated summaries have at least one factual errors and this obstacle the practical usage. [101] specifies these factual errors in the abstractive summarization system into two types:

intrinsic errors and *extrinsic errors*. Intrinsic errors occurs using the contents present in the source article like ”Switzerland” and ”England” in the negative summary example in Figure 6.2. On the other hand, extrinsic extrinsic errors are the errors generated by ignoring the source article when generating summaries. ”in the second half” in Figure 6.2, which is not included in the source article, is an example of extrinsic errors.

In this work, we propose a system for detecting these various factual errors that are necessary for developing summarization system. We propose a unified method for intentionally modeling both types of errors to build a dataset for training this system.

Measuring Factual Consistency

As a better way to evaluate the factual consistency, recent works such as QAGS [6] and QuestEval [14] adopt question generation and question answering framework to evaluate the factual consistency. Both methods firstly generate questions using entities or noun phrases in the candidate summary and then compare the answers of these questions between the source and the summary. Although these methods do not require any reference summaries, they have higher correlation with human judgments than previous metrics in consistency checking. Also, the generated questions and their answers are easily interpretable. But due to their complicated structure, computational complexity of these methods is relatively heavy and the errors in each component can be cascaded. Following the idea that all of the contents in the summaries should be entailed by source document, models from the related tasks such as Natural Language Inference(NLI) [19, 104, 103] are also used to verify factual consistency of the summaries. This approaches are simpler and more intuitive than QA-based metrics. But the data pairs in these datasets are usually composed of single sentences, and this makes it difficult to be directly used for factual consistency checking in summarization where the task requires multi-sentence level reasoning. For this reason, two recent studies FactCC [22] and DocNLI [23] have studied ways to make synthetic datasets for training factual consistency checking model. Both works create synthetic negative

summaries using the pre-defined rules such as entity substitution or mask-and-fill. In this study, we propose a more general negative summary generation method additionally using the masked source.

CoCo [18] compares the likelihood of the generated summaries using the original source and the masked source to estimate the counterfactual samples. Different from CoCo, our work directly augments the negative summaries and train the classifier using them.

6.2 Proposed Approach: MFMA and MSM

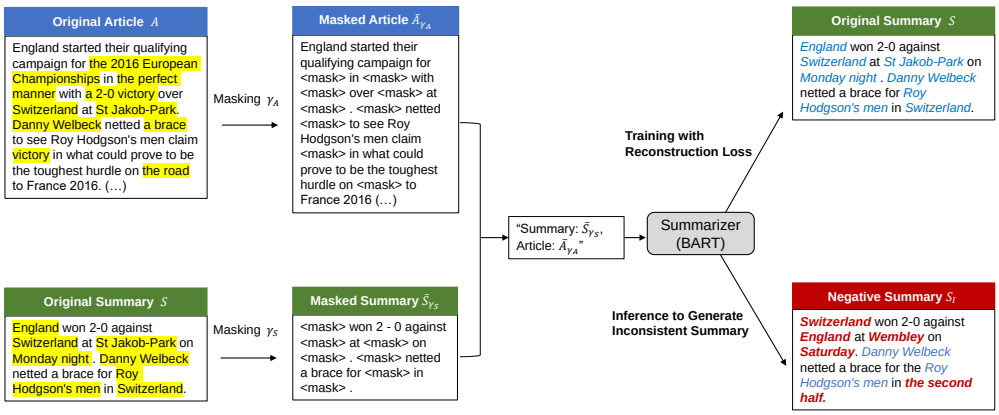


Figure 6.2: Overall flow of our proposed negative summary generation method Mask-and-Fill-with-Masked Article.

For a given article A and a summary S, we aim to develop a factual consistency checking system that can evaluate whether S is factual consistent with A. In other words, the system is required to discriminate a factual consistent summary S_C with the factual inconsistent summary S_I that consists of at least one factual error. We consider this problem as a classification task between S_C and S_I. However, large-scale human-annotated training datasets for this task have not been constructed yet, especially for the inconsistent summaries S_I.

In this study, we focus on effective augmentation methods of the inconsistent summaries. In order for that, there are two crucial conditions: 1) guarantee of inconsistency; the generated summaries should be indeed inconsistent with the source article, 2) relevance to the source article; the generated summaries should include contents related to the article. These two factors are in trade-off relations, which means that when the generated summaries are strongly inconsistent they might not be related to the article and vice versa. Therefore appropriate negative summary augmentation is required to improve the factual consistency classifier.

To generate confusing and hard negative summaries, we propose a summary generation using a masked article and masked reference summary where some salient information is hidden. By doing so, we let the summarizer model infer hidden information through the masked article to generate plausible negative summaries. Note that, previous works such as FactCC and DocNLI generate negative summaries S_I by changing positive summaries S_C through entity replacements or mask-and-fill methods without referring to the source article. We observe that previous methods can easily guarantee negativeness, but they often generate summaries that are very irrelevant to the source article or unnatural as shown in Figure 6.1.

6.2.1 Mask-and-Fill with Masked Article

To model inconsistent summaries but related to the article, we propose a method, **M**ask-and-**F**ill with **M**asked **A**rticle (MFMA), which generates negative summaries with masked articles and masked reference summaries, as shown in Figure 6.2.

Specifically, we assumed *noun phrases* and *entities* in the articles are salient information, and mask them with the ratio of γ_A, resulting in masked article $\overline{A}_{\gamma_A}$. Similarly, we also mask the salient spans in the positive summary, i.e., reference summary, with the ratio of γ_S to form a masked summary $\overline{S}_{\gamma_S}$. Then, we concatenate $\overline{A}_{\gamma_A}$ and $\overline{S}_{\gamma_S}$ by prepending prefix token for each input text (i.e., "Summary: $\overline{S}_{\gamma_S}$, Article: $\overline{A}_{\gamma_A}$") as shown in Figure 6.2. Next, we train a summarizer based on an encoder-decoder model,

BART [17], to reconstruct the original summary S with the following loss:

$$\mathcal{L} = \sum_{t} -\log P(S_t|S_{<t}, [\overline{S}_{\gamma_S}; \overline{A}_{\gamma_A}]). \tag{6.1}$$

After training, we generate negative summaries of unseen and masked article-summary pairs through inference. Obviously, if the mask ratio is high enough, the model is hard to correctly fill the masked contents from the erased article and reference summary. However, we assume the trained reconstruction model is able to fill the masks with plausible contents by inferring the related contents with the masked article.

6.2.2 Masked Summarization

As a variant of MFMA, we also study another negative summary generation model, **M**asked **S**u**M**marization(MSM). The model aims to generate summaries using masked articles $\overline{A}_{\gamma_A}$ but without masked reference summaries as follows:

$$\mathcal{L} = \sum_{t} -\log P(S_t|S_{<t}, \overline{A}_{\gamma_A}). \tag{6.2}$$

The MSM model is trained to generate the entire summaries without the information guidance of masked reference summaries, so MSM has merits in generating more diverse summaries than MFMA.

6.2.3 Training Factual Consistency Checking Model

Finally, for the factual consistency checking model, we train a binary classifier of consistent summaries and inconsistent generated summaries. The pair of summary and the corresponding article are concatenated and then fed into the classification model as an input. We fine-tuned the pre-trained ELECTRA [105] by adding a classifier head with binary cross-entropy loss.

6.3 Experimental Setup and Dataset

6.3.1 Dataset

For evaluating the performance of factual consistency checking system, it is necessary to compare the human judgments of the consistency for the summary with the system. And these human judgment exist in two forms, binary level(*consistent*, *inconsistent*) or numerical levels such as likert scale. In general, in the case of binary level data, performance is measured through accuracy with human judgments. For the case of numerical levels, correlation with human judgments is measured. In addition to using the results for the existing benchmark dataset in this way, we also report the accuracy by casting these numerical level datasets to the binary level dataset since we develop classifier based system. We report the results on the following datasets.

FC-Test [22] release a human-annotated factual consistency for the model generated summaries for CNN/DM Dataset in binary-level to test the performance of FactCC. There are 513 instances in this dataset.

XSumHall [101] study the types of hallucination in the generated summaries and collect the annotation on the errors in the 2K model generated summary for BBC XSum dataset [106]. We use the datasets as binary level benchmark for XSum dataset as in [22].

SummEval [107] collect the likert scale human judgments for the 1600 summaries generated from sixteen abstactive summarizer on CNN/DM testset. This dataset provides human judgments scores in terms of *"coherence" , "consistency", "fluency", and "relevance"* by three expert annotators in likert scale. We only use *"consistency"* score of three annotators, for evaluating our proposed metric. For casting this score to binary level, we let the cases where at least one annotators give less than 5 points for *"consistency"* as *inconsistent*, otherwise *consistent*.

QAGS-CNN/DM & XSum [6] release a human judgments for factual consistency on the model generated summaries for 235 summaries on CNN/DM testset and 239 summaries on XSum testset. Each summary is annotated by three annotators. We also cast the dataset to binary level by assigning *inconsistent* if at least one annotators give *inconsistent* label, otherwise *consistent*.

FRANK-CNN/DM & XSum [3] releases a benchmark dataset FRANK for summarization factual metrics which consists of 2246 summaries on the model generated summaries for 1250 summaries in CNN/DM and 996 summaries XSum. Three annotators evaluated factual consistency of the generated summaries in this dataset. We also convert this dataset to binary level as same as QAGS-CNN/DM and QAGS-XSum.

6.3.2 Implementation Details

Negative Summary Generation We randomly split the training set of CNN/DM dataset [108] in half and use half for training negative summarizer and the other half for generating negative summary after training. We use *spaCy* for finding entities and noun phrases in both summaries and articles. We train *bart-base*[1] for five epochs to train MFMA, and use *bart-base* model without fine-tuning for MF. We use *t5-small* [70][2] for MSM, which shows better results than *bart-base* for this task. We attach the further details in Appendix.

Training Classifier We train *google/electra-base-discriminator*[3] for five epochs with learning rate 2e-5, batch size of 96 using adam optimizer [109] with the dataset we generate using MF, MFMA and MSM. For DocNLI and FactCC, we get the original training dataset that each author release, and we train a model with the same setting as our method except for the training datasets for a fair comparison. We choose model

[1] https://huggingface.co/facebook/bart-base

[2] https://huggingface.co/t5-small

[3] https://huggingface.co/google/electra-base-discriminator

using the balanced accuracy on validation set of FactCC [22] which consists of 1k human annotated summaries.

Hyperparameters We train five epochs for MFMA and MSM using *bart-base* for MFMA and *t5-small* for MSM respectively. We train the model with batch size of 48, max input sequence size of 1024, and max target sequence size of 140. We conduct experiment with various article masking γ_A ratio-summary masking ratio γ_S combinations, at 0.2 intervals from (0.2, 0.2) to (1.0, 1.0). For the case of training classifier, we train *google/electra-base-discriminator* for five epochs with learning rate 2e-5 and batch size of 96. We choose the best parameters using the validation set provided by the [22]. The best mask ratio combination is γ_A = 0.6 and γ_S = 0.8.

6.4 Empirical Results

6.4.1 Comparison with Other Methods

Dataset	FactCC-Test		SummEval		QAGS-CNN/DM		FRANK-CNN/DM		Average	
Metric	F1	BA	F1	BA	F1	BA	F1	BA	F1	BA
Baselines										
FactCC	71.0	71.3	65.1	68.2	69.3	69.6	64.1	63.9	67.4	68.2
DocNLI	67.2	71.0	**71.5**	**71.3**	62.4	66.2	66.0	66.0	66.8	68.6
MNLI	55.0	56.0	51.7	51.7	48.6	53.4	50.4	53.3	51.4	53.6
FEVER	57.9	56.2	52.6	53.6	39.4	53.3	49.8	55.6	49.9	54.7
MF	59.9	64.1	68.2	67.5	47.6	56.9	62.4	62.7	59.5	62.8
Ours										
MFMA	**79.7**	**84.5**	71.3	69.6	**70.5**	**72.3**	**69.5**	69.2	**72.8**	**73.9**
MSM	70.6	72.7	66.8	68.2	67.6	68.7	69.6	**69.3**	68.6	69.7

Table 6.1: Macro F1-score(F1) and class-balanced accuracy(BA) of the human annotated factual consistency for the benchmark datasets based on CNN/DM.

Dataset	XSumHall		QAGS-XSum		FRANK-XSum		Average	
Metric	**F1**	**BA**	**F1**	**BA**	**F1**	**BA**	**F1**	**BA**
Baselines								
FactCC	52.1	**61.8**	63.6	63.7	50.7	58.0	55.5	61.2
DocNLI	55.1	**56.4**	65.3	66.0	**60.3**	**63.4**	60.2	**61.9**
MNLI	33.3	52.1	45.2	51.1	28.8	50.6	35.8	51.3
FEVER	53.1	55.5	62.2	63.7	54.9	63.5	56.7	60.9
MF	53.6	53.3	54.6	54.9	55.7	55.3	54.6	54.5
Ours								
MFMA	**55.5**	56.0	**66.6**	**67.0**	59.6	59.6	**60.6**	60.9
MSM	52.6	53.9	50.8	55.5	50.8	51.3	51.4	53.6

Table 6.2: Macro F1-score(F1) and class-balanced accuracy(BA) of the human annotated factual consistency for the benchmark datasets based on XSum.

Classification Accuracy Due to the imbalance in each dataset, we report the macro-f1 and the class balanced accuracy in Table 6.1 and Table 6.2. We observe that one of our proposed methods MFMA outperforms baseline entailment metrics in four of seven benchmark datasets. MFMA shows better performances than other methods in especially for CNN/DM benchmarks, and shows similar performance to other baseline in XSum datasets. We explain that this is because we only use training set of CNN/DM to construct training set. On the other hand, DocNLI additionally uses the datasets from related tasks such as ANLI [110] and SQuAD [46] except for synthetic negative summaries. Another proposed method MSM also shows competitive performance for CNN/DM benchmarks, but relatively lower performance in XSum based benchmark datasets. We explain the performance gap between MSM and MFMA is due to the properties that directly generates summaries, resulting in many noisy samples that are relatively easy to be distinguished.

Dataset	SummEval		QAGS-CNN/DM		QAGS-XSum		FRANK-CNN/DM		FRANK-XSum	
Metric	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Baselines										
ROUGE-L	0.16	0.14	0.29	0.24	0.13	0.13	0.16	0.13	0.16	0.13
BLEU-4	0.11	0.12	0.18	0.23	0.03	0.03	0.16	0.17	0.11	0.14
METEOR	0.18	0.16	0.26	0.25	0.11	0.12	0.29	0.28	0.18	0.16
BERTScore	0.16	0.14	0.37	0.36	0.11	0.13	0.33	0.30	0.19	0.17
QuestEval	0.35	0.30	0.42	0.36	0.20	0.20	0.46	0.41	0.19	0.18
CoCo	0.42	0.36	**0.67**	0.57	0.20	0.18	0.50	0.45	0.14	0.12
FactCC	0.38	0.36	0.45	0.48	0.30	0.30	0.32	0.36	0.09	0.08
DocNLI	0.51	**0.41**	0.60	0.59	0.36	0.35	0.49	**0.49**	**0.25**	**0.21**
MNLI	0.11	0.13	0.19	0.22	0.08	0.10	0.15	0.16	0.02	0.03
FEVER	0.33	0.32	0.40	0.34	**0.38**	**0.41**	0.38	0.43	0.20	0.19
MF	0.44	0.35	0.43	0.30	0.10	0.10	0.40	0.39	0.10	0.13
Ours										
MFMA	**0.52**	0.38	0.62	**0.65**	0.37	0.38	**0.52**	0.45	0.16	0.17
MSM	0.43	0.36	0.50	0.48	0.20	0.22	0.51	0.48	0.05	0.09

Table 6.3: Summary level Pearson Correlation(r) and Spearman's Correlation(ρ) between various automatic metrics and human judgments of factual consistency for the model generated summaries. Note that we use the confidence of consistency label for entailment based metrics.

Correlation with Human Judgments To compare with general metrics that are not classification level, we also report the correlation with human judgments for five datasets in Table 6.3. We demonstrate that our proposed method has higher pearson correlation coefficient with human judgments in three of five benchmark datasets and competitive with best results results in spearman correlation coefficient. Especially, entailment based methods, which are relatively easy to compute, including our proposed methods show better results than QA-based QuestEval or likelihood based CoCo. Also, reference based method such as ROUGE-L show very lower performance than other methods that does not require any references.

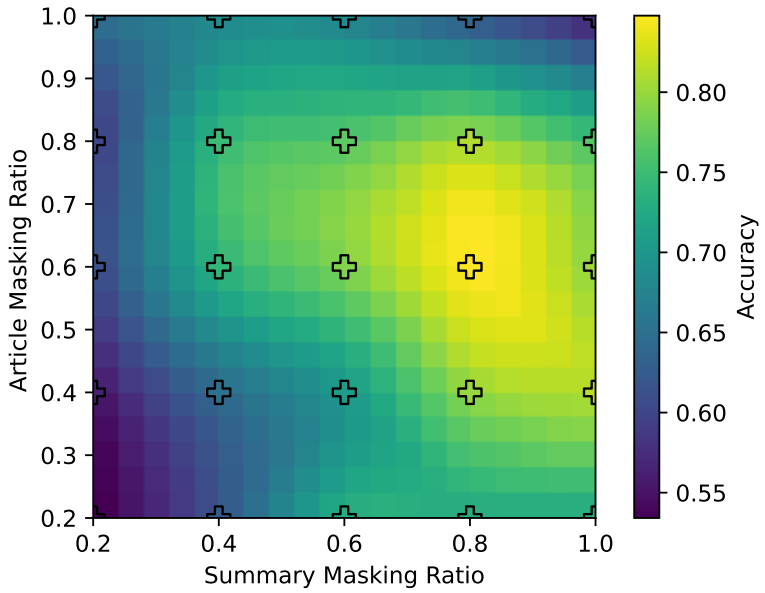


Figure 6.3: Validation Performance among Masked Ratio for Mask-and-Fill with Masked Article. We experiment with each of the five combinations of article mask ratio and summary mask ratio, and then plot the interpolated results.

6.4.2 Analysis

Performance among Masked Ratio We analyze the effects of the mask ratio for both source article and summary in our proposed method MFMA and present results using the validation set in Figure 6.3. Through this experiment, we investigate the tradeoff in adjusting both article masking ratio and summary masking ratio. As shown in Figure 6.3, we find that too high masking ratio decreases performance by sacrificing affinity. On the other hand, if the masking ratio is insufficient, the generated negative sample is often not really negative and this leads to lower performance. Also we can infer that there is an optimal masking ratio combination.

Generated Samples among Masking Ratio We visualize the generated negative summaries through our proposed method MFMA and MSM using CNN/DM in Figure 6.4. We also visualize the example through MF, which simply fills in the mask without the article. We observe that if the article masking ratio γ_A is too low, the generated summaries become almost similar to the original summary since there are enough information to fill the mask. However, if the γ_A is too high, the generated examples are too far from the article, resulting in too negative summary similar to filling the mask without article.

Dataset	Avg-CNN/DM	Avg-XSum
NP/Ent	**73.9**	**60.9**
Token	58.6	53.9
Sentence	53.5	53.4

Table 6.4: Balanced accuracy of the human annotated factual consistency among masking unit. NP/Ent denotes *noun phrases* and *entities*.

Performance among Masking Unit We basically perform masking operation in the *noun phrases* and *entities* units for both summary and article. In order to see the effect of the masking unit, we also conduct an experiment on word level masking and sentence level masking, and present the classification level results in Table 6.4. We observe that *noun phrases* level masking shows the best results following the work [111] where many errors in summarization system are related to *noun phrases* and *entities*.

Distance from Original Reference Summary Using the results on various combinations of article masking ratio and summary masking ratio for MFMA as presented in Figure 6.3, we also investigate the relation between the average distance from the reference summary on each mask ratio combination and the performance. We compute BERTScore between original reference summary and the negative summary generated

using the reference summary to get the distance. Interestingly, as shown in Figure 6.5, we observe the distribution in which performance is maximized within the appropriate distance around 0.8 as the two-dimensional distribution with an R^2 of 0.74. This result shows how far the synthetic negative summaries must be from the reference summaries to help training the factual consistency checking model.

Diversity among Masked Ratio Our proposed method can generate various samples depending on the location of the mask for the same summary-article pair with the fixed mask ratio. Hence, we analyze the diversity of the generated negative summaries among the combinations of mask ratio for MFMA and present the result using validation set in Figure 6.6. We define the diversity of each mask ratio combination as the negation of pairwise similarity score for each sample following [112]. We sample four negative summaries using the given article for each method and then compute the pairwise similarity scores for all of the combinations. We also use BERTScore as a similarity measure. Similar to the distance, we observe that diversity has also similar to a two-dimensional form with an R^2 of 0.7, in which the accuracy is maximized at an appropriate point.

Case Study To understand the pros and cons of our proposed factual consistency checking system, we conduct a case study and illustrate the representative examples in Figure 6.7. We observe that our proposed method is good at judging the facts themselves in the summary like the first example, but still not perfect in examples that require high-level reasoning like the second example.

Article: Tropical Storm Andrea formed in the Gulf of Mexico on Wednesday, marking the first storm of the 2013 Atlantic hurricane season. The storm was about 270 miles west-southwest of Tampa, Florida, and moving north at 6 mph, said the National Hurricane Center. It had maximum sustained winds of 40 mph. ,..., Thursday afternoon and move over southeastern Georgia and eastern South Carolina Thursday night and Friday. Little change in strength is expected over the next 48 hours.

Reference Summary: Storm moving north at faster rate (6 mph) A storm warning is issued for parts of the Florida coast . Little change in strength is forecast for the next 48 hours . Andrea the first storm of the 2013 Atlantic hurricane season .

MF: Storm moving north at 6 mph (11 kph) A storm warning is issued for parts of the Carolinas. Little change in strength is forecast for the Carolines. Andrea the first storm of the season.

MFMA $\gamma_A = 0.4$**:** Andea moving north at faster rate (6 mph) A storm warning is issued for parts of the Florida coast. Little change in strength is forecast for the next 48 hours. Andea the first storm of the 2013 Atlantic hurricane season.

MFMA $\gamma_A = 0.6$**:** Storm moving north at faster rate (6 mph) Tropical storm watch is issued for west coast of Florida. Little change in strength is forecast for next 48 hours. Storm the first storm of the 2013 Atlantic hurricane season.

MFMA $\gamma_A = 0.8$**:** Andrea moving north at 12 mph (6 mph) A tropical storm warning is issued for parts of the East Coast. A significant increase in strength is forecast for Thursday. Andrea the first storm of the Atlantic hurricane season.

MFMA $\gamma_A = 1.0$**:** Storm moving north at 6 mph (10 kph) A tropical storm watch is issued for the west coast of Florida. A hurricane watch in effect is forecast for the East Coast. The center of the storm forms.

MSM $\gamma_A = 0.4$**:** The storm is about 270 miles west-southwest of Fort Myers, Florida. NEW: The storm is in effect for the west coast of Florida. The storm is the first of the 2013 Atlantic hurricane season.

Figure 6.4: Generated negative summaries among through various masking ratio in CNN/DM dataset. For MFMA and MF, we fix the summary masking $\gamma_S = 0.6$:

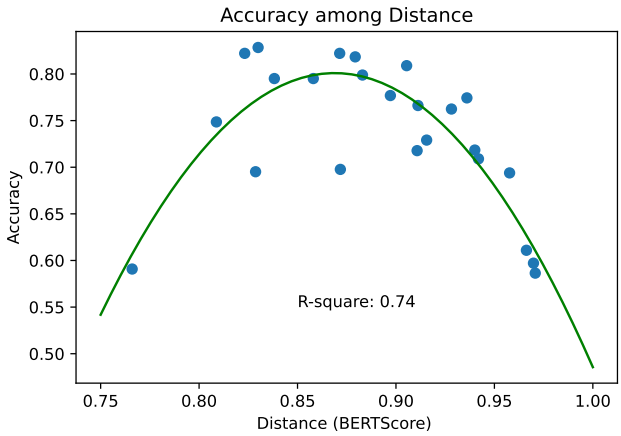


Figure 6.5: Validation Set Performance among BERTScore between the original reference summaries and the negative summaries we generate using the various combinations of article and summary masking ratios.

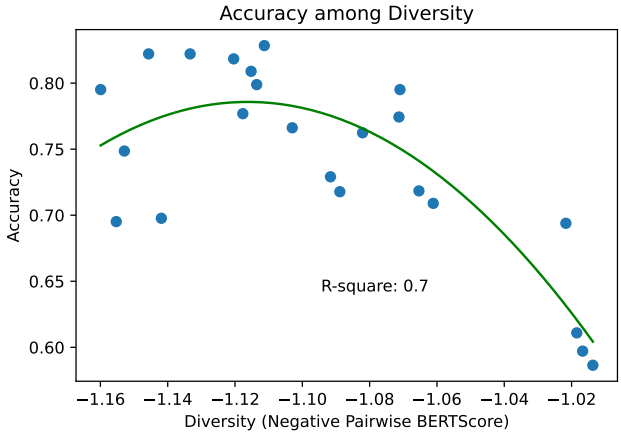


Figure 6.6: Validation Set Performance among diversity among various combinations of article masking ratio and summary masking ratio. Diversity is computed as negative of the pairwise BERTScore between four negative samples generated by each masking ratio.

Article: Nkaissery told reporters the university will be able to confirm Saturday if everyone has been accounted for. Thursday's attack by al-Shabaab militants killed 147 people, three security officers and two university security personnel. The attack left 104 people injured, including 19 who are in critical condition, Nkaissery said.,...,

Candidate Summary: 147 people, including 142 students, are in critical condition.

Ground Truth: *INCONSISTENT*
MFMA: *INCONSISTENT*
MSM: *INCONSISTENT*
DocNLI: *INCONSISTENT*
FactCC: *CONSISTENT*

Article: Media playback is not supported on this device United remain 15 points clear at the top of the table with eight games left after a 1-0 win at Sunderland. "We are not concerned with what we have left behind us, we are only focusing on what is in front of us," said Ferguson. ",...,

Candidate Summary: Manchester United manager Sir Alex Ferguson says he is not concerned about his side's unbeaten start to the season as they attempt to win the Premier League title.

Ground Truth: *CONSISTENT*
MFMA: *INCONSISTENT*
MSM: *INCONSISTENT*
DocNLI: *INCONSISTENT*
FactCC: *CONSISTENT*

Figure 6.7: Case study on entailment based models. First example comes from and FactCC-Test and second example comes from XSumHall.

6.5 Conclusion

In this study, we proposed an effective generation method of factually inconsistent summaries, called MFMA. In this method, some proportion of the source text and corresponding reference summaries is hidden, then a summarization model generates plausible but factually inconsistent summaries by inferring the masked contents. Experiments on seven benchmark datasets demonstrate that factual consistency classifiers trained using our method generally outperform existing models and show competitive correlation with human judgement.

Chapter 7

Factual Error Correction for Improving Factual Consistency

Text summarization is a task that aims to generate a short version of the text that contains the important information for the given source article. With the advances of neural text summarization systems, abstractive summarization systems [99] that generate novel sentences rather than extracting the snippets in the source are widely used [113]. However, factual inconsistency between the original text and the summary is frequently observed in the abstractive summarization system [2, 100, 101] as shown in the system summary of Figure 7.2. As in the example of Figure 7.2, many of these errors in the summaries occur at the entry-level such as person name and number. But these types of errors are sometimes trivial and can often be easily solved through simple modification like changing the wrong entities, as shown in Figure 7.2. For this reason, previous works [114, 115] have introduced post-editing systems to alleviate these factual errors in the summary. But all of those works adopt the seq2seq model, which requires a similar cost to the original abstractive summarization systems, as a post-editing. Therefore, using such systems based on seq2seq doubles the inference time for performing post-editing, resulting in significant inefficiency. In addition, seq2seq based post-editing model can be affected by the model's own bias to the input summary.

Article: *Singer-songwriter David Crosby hit a jogger with his car Sunday evening, a spokesman said.* The accident happened in Santa Ynez, California, near where Crosby lives. Crosby was driving at approximately 50 mph when he struck the jogger, according to California Highway Patrol Spokesman Don Clotworthy. The posted speed limit was 55. The jogger suffered multiple fractures, and was airlifted to a hospital in Santa Barbara, Clotworthy said.,...

System Summary with Factual Error: ***Don Clotworthy*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

After Correction: ***David Crosby*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

Figure 7.1: An example of generated summary with factual errors and the correct summary after minor modification.

To overcome this issue and develop efficient factual corrector for summarization systems, we propose a totally different approach, RFEC(**R**etrieval-based **F**actual **E**rror **C**orrector) that efficiently corrects the factual errors with much faster running time compared to seq2seq model. RFEC first retrieves the evidence sentences for the given summary for correcting and detecting errors. By doing so, we shorten the input length of the model to obtain computational efficiency. Then, RFEC examines all of the entities whether each entity has a factual error. If any entities have a factual error, RFEC substitutes these wrong entities with the correct entity by choosing them among the entities in the source article. Through these steps, we do not create a whole sentence as in the seq2seq model, but decide whether to fix and correct it through the retrieval, resulting in higher computational efficiency. Experiments on both synthetic and real-world benchmark datasets demonstrate that our model shows competitive performance

with the baseline model with much faster running time. Also, as shown in Figure 7.3, RFEC has a natural form of interpretability through the visualization of the erroneous score and the scores of each candidate entity for correcting the wrong entities.

7.1 Related Work

With the advancement of pre-training language models such as BERT [13] and BART [17], abstractive summarization systems have adopted these models to use the rich information inherent in parameters. While these models improved the performance, the generated summaries are still often factually inconsistent with the source article. [3].

To solve the factual inconsistency in abstractive summarization systems, FASUM [115] adopted graph attention network [116] for generating the correction summary. [117] studied contrast candidate generation and selection by ranking approach as a model-agnostic post-processing technique to correct the extrinsic hallucinations.

Another line of mitigating the factual errors is to develop a post editing system to fix the errors. [114] presented a post-editing corrector module using a BART-based auto-regressive model. To train such system, the study generated corrupted summary by substituting the key information such as an entity or a number to construct a training dataset. [118] also develop a seq2seq based error correction system in the claim of FEVER dataset [119] by correcting the words after masking some words. Different from seq2seq based previous works, we develop a faster retrieval based factual error correction system that does not generate the whole summary, only corrects the entity-level errors by substituting them with one of the entities in the article.

Article: *Singer-songwriter David Crosby hit a jogger with his car Sunday evening, a spokesman said.* The accident happened in Santa Ynez, California, near where Crosby lives. Crosby was driving at approximately 50 mph when he struck the jogger, according to California Highway Patrol Spokesman Don Clotworthy. The posted speed limit was 55. The jogger suffered multiple fractures, and was airlifted to a hospital in Santa Barbara, Clotworthy said.,...

System Summary with Factual Error: ***Don Clotworthy*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

After Correction: ***David Crosby*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

Figure 7.2: An example of generated summary with factual errors and the correct summary after minor modification.

7.2 Proposed Approach: RFEC

7.2.1 Problem Formulation

For a given summary S and an article A, we aim to develop a factual error correction system that can fix the possible factual errors in S. Since most of the factual errors appear in entity-level, we develop a system that is specialized in correcting entity-level errors. Specifically, we define this problem as two steps, entity-level error detection and entity-level error correction as shown in Figure 7.3. For given ns entities $E_S = \{es_1, es_2, ..., es_{ns}\}$ in a summary S, we first classify whether each entity is factually consistent with the article A. If any entity e_{S_i} is factually inconsistent, the system substitutes it with one of the na entities in the article $E_A = \{ea_1, ea_2, ..., ea_{na}\}$.

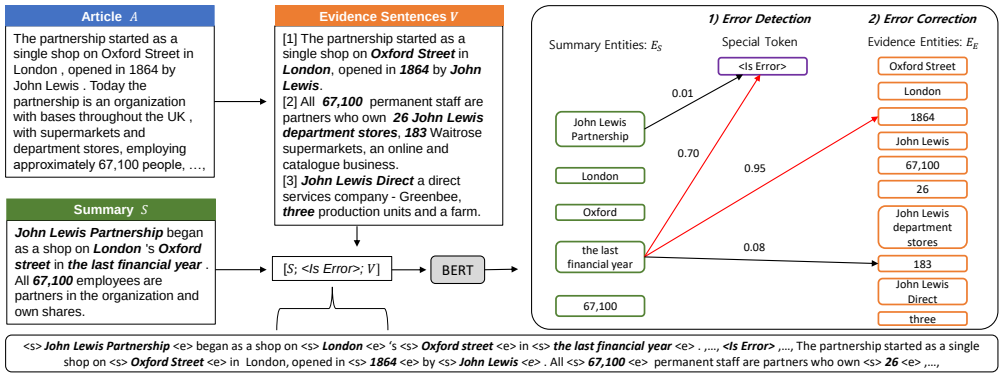


Figure 7.3: Overall flow of our proposed retrieval-based factual error correction system. Given a summary S and an article A, we first retrieve evidence sentences V. Using S and V, we compute BERT embeddings for entities in summary E_S and evidence sentence V. Note that $\langle$ *Is Error* $\rangle$ is a special token for classifying whether each entity is an error. If the erroneous score computed using $\langle$ *Is Error* $\rangle$ token is above threshold, we regard those entity as an error and substitute it with one of the entities in the evidence sentences that obtains highest score.

7.2.2 Training Dataset Construction

To train a factual error correction system, we need a triple composed of an input summary S_1 that may have factual errors, an article A and a target summary S_2 that is a modified version of S_1 without factual errors. However, it is difficult to obtain S_1 that has the errors with the position annotated and the right ground truth correction of such errors. Hence, to train a system, we construct a synthetic dataset by modifying the reference summaries following previous works [114, 115, 22]. We corrupt reference summaries in CNN/DM dataset [108] by randomly changing one of the entities with the same type of other entities in the dataset to make a corrupted summary. Finally, we construct a triple (S_1, A, S_2). Meanwhile, in the real world dataset, a significant number of summaries are factually consistent, so we only make errors for 50% of the summaries and set $S_1 = S_2$ for the rest of the summaries in the dataset. Through this procedure,

we construct the synthetic training dataset where the number of each train/validation split is 133331/6306, respectively.

7.2.3 Evidence Sentence Retrieval

Generally, a summary does not treat all of the contents in the article but only contains some important parts of the article. Hence, in most cases, checking for errors within the summary and correcting them does not require the entire article, and using the part related to the summary is sufficient, as shown in Figure 7.3. Inspired by this observation, we extract some of the sentences in the article according to the similarity with the summary to increase the efficiency of the system by shortening the input length. We use ROUGE-L [35] score as a similarity measure to extract top-2 evidence sentences for each sentence in summary. Then, we remove the duplicates and sort them according to the order in which they appear in the article, and combine them to form $V = \{V_1, V_2, ..., V_M\}$, a set of evidence sentences for detecting and correcting errors in the summary S.

7.2.4 Entity Retrieval Based Factual Error Correction

Computing Embedding Using summary S and the evidence sentences V, we first extract entities E_S and E_V respectively using SpaCy[1] named entity recognition model. And we insert special tokens $\langle$ s $\rangle$ and $\langle$ e $\rangle$, before and after each extracted entity. Then we also insert an additional token $\langle$ *Is Error* $\rangle$, which is later used for checking the factual consistency between S and V and concatenate them to make an input for the BERT [13]. Using BERT, we obtain the contextualized embedding of each entity in S and V as follows:

$$H=[h_1,h_2,...,h_l]=BERT([S;<IsError>;V]), \tag{7.1}$$

where l is maximum sequence length of the input.

[1] https://spacy.io/api/entityrecognizer

And we get the embedding of start token ¡s¿ for each entity as the entity embeddings $HE_V = \{h_{ev_1}, h_{ev_2}, ..., h_{ev_{nv}}\}$ and $HE_S = \{h_{es_1}, h_{es_2}, ..., h_{es_{ns}}\}$ for V and S respectively. We also get h_{err}, an embedding of *¡Is Error¿*.

Error Detection Using the computed embeddings, we compute the erroneous score for all of the entities, in summary, using the embedding of *¡Is Error¿* token h_{err} as follows.

$$\hat{s}_{err_i}=P(Err|es_i)=sigmoid(h_{es_i}^{\intercal} W_{det} h_{err}+b_{det}) \tag{7.2}$$

,where $i = 1, 2, 3, ..., ns$. The W_{det} and b_{det} are model parameters.

Error Correction For the entities that are factual errors, we compute the correction score between the entities and all of the entities in the evidence sentences similar to error detection as follows.

$$\hat{s}_{cor_{ij}}=P(Cor|es_i,ev_j)=sigmoid(h_{es_i}^{\intercal} W_{cor} h_{ev_j}+b_{cor}) \tag{7.3}$$

,where $i = 1, 2, 3, ..., ns_{err}$, $j = 1, 2, 3, ..., nv$. ns_{err} is the number of errors in the summary. The W_{cor} and b_{cor} are model parameters.

Training Objective We train the model using binary cross entropy loss for both detection and correction as follows.

$$L_{det}=-\frac{\sum_{i=1}^{ns}(s_{err_i}\log(\hat{s}_{err_i})-(1-s_{err_i})\log(1-\hat{s}_{err_i}))}{ns} \tag{7.4}$$

$$L_{cor}=-\frac{\sum_{i=1}^{ns}\sum_{i=1}^{nv}(s_{cor_{ij}}\log(\hat{s}_{cor_{ij}})-(1-s_{cor_{ij}})\log(1-\hat{s}_{cor_{ij}}))}{ns\cdot nv} \tag{7.5}$$

$$L=L_{det}+L_{cor} \tag{7.6}$$

,where $s_{err_i} \in \{0, 1\}$ and $s_{cor_{ij}} \in \{0, 1\}$, which are the ground truth labels for detection and correction.

Inference For the inference stage, we do not have the label as to whether each entity is an error. Therefore, we calculate the two results sequentially, error detection and error correction, using the same BERT embeddings. For each entity, if an erroneous score is above thr_{det}, then we let that entity be an error as shown in Figure 7.3. And then, we search the candidate of correction among the evidence entities HE_V, and substitute it with the entity that gets the maximum score as in Figure 7.3. We conduct correction only when the maximum score is higher than thr_{cor} to prevent unnatural correction caused by failure to find the appropriate entity within the candidate.

7.3 Experimental Setup and Dataset

7.3.1 Dataset

For our experiments, we evaluate our proposed factual error correction method on both synthetic dataset and real-world dataset, based on CNN/DM. We briefly describe the details of two benchmark datasets below. Using the same method in Section 7.2.2, we make a separate 3,000 test tests. As same as the training dataset, the corrupted summaries, and the reference summaries are mixed at the same ratio in this testset. For this synthetic testset, we know the ground truth correction for each summary. Hence, we measure the success rate of correction through whether the post-editing model's correction is the same as the ground truth correction. In addition to this synthetic data, we also use the FactCC-Test set [22] that has labels on the 503 system-generated summaries whether they are factually consistent or not. Among them, 62 summaries are inconsistent, and 441 summaries are consistent. Different from the synthetic testset, FactCC-Test Dataset does not provide the ground truth correction for the inconsistent summaries. Hence, we manually check the results of all of the systems as in the example of Figure 7.4.

7.3.2 Implementation Details

Hyperparameters For our experiments, we use *bert-base-cased*[2] for RFEC. We train the model for five epochs using Adam Optimizer [109] with a learning rate of 3e-5. For baseline seq2seq model, we use *bart-base*[3] following the previous work [114] and train the model using the same dataset we used for training RFEC with same epochs for fair comparison.We set both thr_{det} and thr_{cor} for 0.5 using the validation set. For maximum sequence length, we set 1024 for BART, 256 for BART without evidence selection, 256 for RFEC, and 512 for RFEC without evidence sentence selection.

7.4 Empirical Results

Computing Infrastructure All of the experiments are done using NVIDIA RTX A5000 24G with Python 3.8.8 and PyTorch 1.10.1. We measure the running time, including the preprocessing time of each method using a single A5000 GPU and Intel(R) Xeon(R) Silver 4210R CPU (2.40 GHz).

7.4.1 Comparison with Other Methods

Synthetic Dataset We present the results for the 3k synthetic testset in Table 7.1. We observe that the performance of BART is slightly better than RFEC, but our proposed retrieval-based model has a much faster running time. We also observe that accuracy for all of the models is very high for the synthetic dataset since the type of the errors is relatively trivial. Also, we find that using only evidence sentences performs slightly lower than using the whole article sentences but have advantages in computing speed for both systems. Especially for RFEC, it does not take much time to calculate the model output, but it costs relatively much time on preprocessing, especially for named entity recognition. And reducing the input length through the sentence selection also

[2]https://huggingface.co/bert-base-cased

[3]https://huggingface.co/facebook/bart-base

reduces the preprocessing time, resulting in faster running time, as shown in Table 7.1. For computing the throughput, we make the best effort to set the maximum batch size for each setting using a same environment for a fair comparison.

Method	Sample/min	Accuracy
Seq2seq - BART	933	90.93
- sentence selection	629	92.20
RFEC	4024	91.06
- sentence selection	1810	91.15

Table 7.1: Factual error correction results on test split of synthetic Test Dataset with the average running time.

Method	Inconsistent(62)		Consistent(441)	
	Changed	Edited	Changed	Edited
Seq2seq - BART	8	15	2	14
- sentence selection	9	23	7	78
RFEC	7	9	2	23
- sentence selection	6	8	3	31

Table 7.2: Factual error correction results on FactCC-Testset. Each column represents how many corrections each system has performed for the sample of each label, and how many labels have changed from the correction.

FactCC-Test Dataset We present the results for the FactCC-Test Dataset in Table 7.2. Compared to the results in the synthetic dataset, both seq2seq and RFEC do not correct many errors, only 9 and 7 for the best settings in both systems among 62 errors. However, as in the synthetic dataset, our proposed method shows almost the same results with eight times less running time compared to the seq2seq method. Also, we can observe that using the correction model also creates a significant number of new errors especially

for the seq2seq model without sentence selection.

7.4.2 Analysis

We present the representative success and failure cases of our proposed retrieval-based factual error correction system with the top-3 retrieved entities for the errors in Figure 7.4. For the first example, RFEC successfully corrects the error *Valerie Braham* by substituting it with *Philippe Braham* that gets a higher correction score among the entities in the evidence sentences. Also, as the object to be corrected is a person's name, we can observe that other correction candidates are also names. On the other hand, for the second example, although RFEC detects the error *Raymond*, but do not find the correction candidates whose correction score is above thr_{cor}. For this example, *Raymond* should be changed to *the front bench*, but the named entity recognition model fails to capture it and leads to missing it from the correction candidate.

7.5 Conclusion

In this study, we proposed an efficient factual error correction system RFEC based on two retrieval steps. RFEC first retrieves evidence sentences based on textual similarities between the summary and the article for detecting and correcting factual errors. Then, if there is an entity that is a cause of factual errors, RFEC substitutes it with one of the entities in the evidence sentences as a retrieval-based approach. Experiments on two benchmark datasets demonstrate that our proposed method shows competitive results compared to strong baseline seq2seq with a much faster inference speed.

Example 1) - Success

Evidence Sentences: Her husband, Philippe Braham, was one of 17 people killed in January's terror attacks in Paris. One month after the terror attacks in Paris, a gunman attacked a synagogue in Copenhagen, Denmark, killing Dan Uzan, who was working as a security guard for a bat mitzvah party.

Input Summary: Valerie Braham was one of 17 people killed in January 's terror attacks in Paris

Corrected Summary: *Philippe Braham* was one of 17 people killed in January's terror attacks in Paris.

Top3 Correction Candidates for Valerie Braham:

Philippe Braham, Dan Uzan, bat mitzvah

Example 2) - Failure

Evidence Sentences: Sawyer Sweeten grew up before the eyes of millions as a child star on the endearing family sitcom " Everybody Loves Raymond." Sweeten , best known for his role Geoffrey Barone , was visiting family in Texas, entertainment industry magazine Hollywood Reporter reported, where he is believed to have shot himself on the front porch.

Input Summary: He is believed to have shot himself on Raymond

Corrected Summary: He is believed to have shot himself on Raymond.

Top3 Correction Candidates for Raymond:

Everybody Loves Raymond, Geoffrey Barone, Sawyer Sweeten

Figure 7.4: Case study on our proposed factual error correction system. The entities in the evidence sentences are highlighted. The color on each entity in each input summary represents the erroneous score, and the darker the color, the higher the erroneous score.

Chapter 8

Conclusion

In this dissertation, we propose four novel factual consistency metrics that has higher correlation with human judgments than previous methods for various conditional text generation systems based on two approaches; *(1) using auxiliary tasks*, *(2) data augmentation methods* as shown in Table 8.

First, we utilize auxiliary tasks to focus on keywords that are salient for evaluating factual consistency and propose two metrics, KPQA and QACE. We develop a KPQA-metric specialized for generative QA by integrating keyphrase weights to existing metrics such as BLEU or BERTScore using a pre-trained auxiliary task. We use the soft labels of the pre-trained auxiliary model designed to get the keyphrase weights. Experimental results show that keyphrase weights are helpful in evaluating the correctness. Furthermore, we propose a QACE metric for image captioning task based

Name	Method	Reference	Computation	Interpretability
KPQA	Auxiliary Tasks	O	Slow	High
QACE		X		
UMIC	Data Augmentation	X	Fast	Low
MFMA		X		

Table 8.1: Properties of the proposed factual consistency metrics.

on question generation and question answering system. Different from KPQA-metric, QACE do not require human generated reference. QACE generates questions for the given captions and then directly evaluate the factual consistency of the captions by comparing the answers of the questions for the source and the candidate caption.

Secondly, we tackle the problem of developing a factuality metric as a data-driven approach and propose two novel metrics, UMIC and MFMA. We solve the factual consistency evaluation in image captioning task with data augmentation method by generating inconsistent captions with pre-defined rules to edit the human written caption. We train a metric through contrastive learning to distinguish fake inconsistent captions with the human written caption. This novel training method achieves higher correlation with human judgments, resulting in state-of-the-arts in evaluating image caption. We propose a more general way of creating inconsistent examples through masked generation in evaluating factual consistency of the abstractive summaries. We generate inconsistent examples using masked articles to generate hard negative examples. Experimental results show that proper masking ratio can generate inconsistent samples that are helpful for develping factual consistency metrics, resulting in state-of-the-arts in abstractive summarization evaluation. Finally, we investigate the way to mitigate the factual inconsistency itself inspired by the approaches in data-driven factual consistency metrics. We develop an entity-level faster post-editing system RFEC to correct the factual errors in the abstractive summarization system.

Extensive experiments have demonstrated that all of our proposed metrics outperform previous metrics in evaluating factual consistency for target text generation systems. Also, we confirm that the methods use auxiliary tasks have natural form of interpretability due to the properties of each auxiliary task, but have higher computational cost. On the other hand, data-driven approaches generally show higher performance and faster computation speed, while sacrificing the interpretability.

In the future work, we will study a method to optimize the conditional text generation systems using a factual consistency metrics. Since most of the evaluation metrics

are not differentiable, we will start from the reinforcement learning based approach. Specifically, we will use proximal policy optimization algorithms [120] to directly optimize the conditional text generation systems using factual consistency metrics. We will also study new decoding methods that can utilize factual consistency metrics in the inference stage.

Bibliography

[1] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al., "Ms marco: A human generated machine reading comprehension dataset," arXiv preprint arXiv:1611.09268, 2016.

[2] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the original: Fact aware neural abstractive summarization," in thirty-second AAAI conference on artificial intelligence, 2018.

[3] A. Pagnoni, V. Balachandran, and Y. Tsvetkov, "Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4812–4829, 2021.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318, 2002.

[5] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Text summarization branches out, pp. 74–81, 2004.

[6] A. Wang, K. Cho, and M. Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in Proceedings of the 58th Annual Meeting

of the Association for Computational Linguistics, (Online), pp. 5008–5020, Association for Computational Linguistics, July 2020.

[7] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[8] S. Banerjee and A. Lavie, "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, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, 2005.

[9] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp. 4566–4575, IEEE Computer Society, 2015.

[10] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in International Conference on Learning Representations, 2019.

[11] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in International conference on machine learning, pp. 957–966, 2015.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, 2019.

[14] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, J. Staiano, A. Wang, and P. Gallinari, "QuestEval: Summarization asks for fact-based evaluation," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, (Online and Punta Cana, Dominican Republic), pp. 6594–6604, Association for Computational Linguistics, Nov. 2021.

[15] H. Lee, S. Yoon, F. Dernoncourt, T. Bui, and K. Jung, "Umic: An unreferenced metric for image captioning via contrastive learning," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 220–226, 2021.

[16] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," Advances in Neural Information Processing Systems, vol. 34, 2021.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880, 2020.

[18] Y. Xie, F. Sun, Y. Deng, Y. Li, and B. Ding, "Factual consistency evaluation for text summarization via counterfactual estimation," in Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 100–110, 2021.

[19] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 632–642, 2015.

[20] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The fact extraction and VERification (FEVER) shared task," in Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), (Brussels, Belgium), pp. 1–9, Association for Computational Linguistics, Nov. 2018.

[21] F. Nan, R. Nallapati, Z. Wang, C. dos Santos, H. Zhu, D. Zhang, K. McKeown, and B. Xiang, "Entity-level factual consistency of abstractive text summarization," in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 2727–2733, 2021.

[22] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), (Online), pp. 9332–9346, Association for Computational Linguistics, Nov. 2020.

[23] W. Yin, D. Radev, and C. Xiong, "DocNLI: A large-scale dataset for document-level natural language inference," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, (Online), pp. 4913–4922, Association for Computational Linguistics, Aug. 2021.

[24] H. Lee, K. M. Yoo, J. Park, H. Lee, and K. Jung, "Masked summarization to generate factually inconsistent summaries for improved factual consistency checking," in Findings of the Association for Computational Linguistics: NAACL 2022, july 2022.

[25] Wikipedia contributors, "Pearson correlation coefficient — Wikipedia, the free encyclopedia," 2022. [Online; accessed 6-June-2022].

[26] Wikipedia contributors, "Spearman's rank correlation coefficient — Wikipedia, the free encyclopedia," 2022. [Online; accessed 6-June-2022].

[27] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li, "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 (S. Kambhampati, ed.), pp. 2972–2978, IJCAI/AAAI Press, 2016.

[28] L. Song, Z. Wang, and W. Hamza, "A unified query-based generative model for question generation and question answering," arXiv preprint arXiv:1709.01058, 2017.

[29] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, (Brussels, Belgium), pp. 4220–4230, Association for Computational Linguistics, 2018.

[30] K. Nishida, I. Saito, K. Nishida, K. Shinoda, A. Otsuka, H. Asano, and J. Tomita, "Multi-style generative reading comprehension," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 2273–2284, Association for Computational Linguistics, 2019.

[31] B. Bi, C. Wu, M. Yan, W. Wang, J. Xia, and C. Li, "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), (Hong Kong, China), pp. 2521–2530, Association for Computational Linguistics, 2019.

[32] B. Bi, C. Wu, M. Yan, W. Wang, J. Xia, and C. Li, "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 Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 7424–7431, AAAI Press, 2020.

[33] A. Chen, G. Stanovsky, S. Singh, and M. Gardner, "Evaluating question answering evaluation," in Proceedings of the 2nd Workshop on Machine Reading for Question Answering, (Hong Kong, China), pp. 119–124, Association for Computational Linguistics, 2019.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, 2002.

[35] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Text Summarization Branches Out, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, 2004.

[36] A. Yang, K. Liu, J. Liu, Y. Lyu, and S. Li, "Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task," in Proceedings of the Workshop on Machine Reading for Question Answering, (Melbourne, Australia), pp. 98–104, Association for Computational Linguistics, 2018.

[37] H. Alamri, V. Cartillier, A. Das, J. Wang, A. Cherian, I. Essa, D. Batra, T. K. Marks, C. Hori, P. Anderson, S. Lee, and D. Parikh, "Audio visual scene-aware dialog," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 7558–7567, Computer Vision Foundation / IEEE, 2019.

[38] W. He, K. Liu, J. Liu, Y. Lyu, S. Zhao, X. Xiao, Y. Liu, Y. Wang, H. Wu, Q. She, X. Liu, T. Wu, and H. Wang, "DuReader: a Chinese machine reading comprehen-

sion dataset from real-world applications," in Proceedings of the Workshop on Machine Reading for Question Answering, (Melbourne, Australia), pp. 37–46, Association for Computational Linguistics, 2018.

[39] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, "The NarrativeQA reading comprehension challenge," Transactions of the Association for Computational Linguistics, vol. 6, pp. 317–328, 2018.

[40] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: Long form question answering," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 3558–3567, Association for Computational Linguistics, 2019.

[41] I. Sutskever, O. Vinyals, and Q. V. Le, "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 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, 2014.

[42] C.-W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "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, (Austin, Texas), pp. 2122–2132, Association for Computational Linguistics, 2016.

[43] P. Nema and M. M. Khapra, "Towards a better metric for evaluating question generation systems," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, (Brussels, Belgium), pp. 3950–3959, Association for Computational Linguistics, 2018.

[44] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "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), (Hong Kong, China), pp. 540–551, Association for Computational Linguistics, 2019.

[45] G. Marton and A. Radul, "Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements," in Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, (New York City, USA), pp. 375–382, Association for Computational Linguistics, 2006.

[46] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, 2016.

[47] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2369–2380, 2018.

[48] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," Transactions of the Association for Computational Linguistics, vol. 8, pp. 64–77, 2020.

[49] S. Ostermann, M. Roth, A. Modi, S. Thater, and M. Pinkal, "SemEval-2018 task 11: Machine comprehension using commonsense knowledge," in Proceedings of The 12th International Workshop on Semantic Evaluation, (New Orleans, Louisiana), pp. 747–757, Association for Computational Linguistics, 2018.

[50] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon, "Unified language model pre-training for natural language un-

derstanding 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 (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds.), pp. 13042–13054, 2019.

[51] H. Le, D. Sahoo, N. Chen, and S. Hoi, "Multimodal transformer networks for end-to-end video-grounded dialogue systems," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 5612–5623, Association for Computational Linguistics, 2019.

[52] H. Alamri, C. Hori, T. K. Marks, D. Batra, and D. Parikh, "Audio visual scene-aware dialog (avsd) track for natural language generation in dstc7," in DSTC7 at AAAI2019 Workshop, vol. 2, 2018.

[53] C. Hori, T. Hori, T. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp. 4203–4212, IEEE Computer Society, 2017.

[54] R. Likert, "A technique for the measurement of attitudes.," Archives of psychology, 1932.

[55] H. J. Jung and M. Lease, "Improving consensus accuracy via z-score and weighted voting," in Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[56] K. Krippendorff, "Estimating the reliability, systematic error and random error of interval data," Educational and Psychological Measurement, vol. 30, no. 1, pp. 61–70, 1970.

[57] K. Krippendorff, "Computing krippendorff's alpha-reliability," Computing, vol. 1, pp. 25–2011.

[58] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," biometrics, pp. 159–174, 1977.

[59] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in International Conference on Learning Representations, 2018.

[60] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3156–3164, 2015.

[61] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6077–6086, 2018.

[62] J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser, "Why we need new evaluation metrics for NLG," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (Copenhagen, Denmark), pp. 2241–2252, Association for Computational Linguistics, Sept. 2017.

[63] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in Proceedings of the IEEE international conference on computer vision, pp. 2425–2433, 2015.

[64] S. Aditya, Y. Yang, C. Baral, C. Fermuller, and Y. Aloimonos, "From images to sentences through scene description graphs using commonsense reasoning and knowledge," arXiv preprint arXiv:1511.03292, 2015.

[65] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," Journal of Artificial Intelligence Research, vol. 47, pp. 853–899, 2013.

[66] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in European Conference on Computer Vision, pp. 382–398, Springer, 2016.

[67] A. Fisch, K. Lee, M.-W. Chang, J. H. Clark, and R. Barzilay, "Capwap: Captioning with a purpose," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8755–8768, 2020.

[68] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano, "Answers unite! unsupervised metrics for reinforced summarization models," 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), pp. 3237–3247, 2019.

[69] A. Wang, K. Cho, and M. Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5008–5020, 2020.

[70] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[71] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 784–789, 2018.

[72] R. Shekhar, S. Pezzelle, Y. Klimovich, A. Herbelot, M. Nabi, E. Sangineto, and R. Bernardi, "Foil it! find one mismatch between image and language caption," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 255–265, 2017.

[73] T. Scialom, P. Bordes, P.-A. Dray, J. Staiano, and P. Gallinari, "What bert sees: Cross-modal transfer for visual question generation," in Proceedings of the 13th International Conference on Natural Language Generation, pp. 327–337, 2020.

[74] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, pp. 91–99, 2015.

[75] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6904–6913, 2017.

[76] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision, pp. 740–755, Springer, 2014.

[77] C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins, "Synthetic qa corpora generation with roundtrip consistency," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6168–6173, 2019.

[78] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," ACM Computing Surveys (CsUR), vol. 51, no. 6, pp. 1–36, 2019.

[79] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7008–7024, 2017.

[80] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4634–4643, 2019.

[81] 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, pp. 1473–1482, 2015.

[82] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2556–2565, 2018.

[83] Y. Cui, G. Yang, A. Veit, X. Huang, and S. Belongie, "Learning to evaluate image captioning," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5804–5812, 2018.

[84] H. Lee, S. Yoon, F. Dernoncourt, D. S. Kim, T. Bui, and K. Jung, "Vilbertscore: Evaluating image caption using vision-and-language bert," in Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, pp. 34–39, 2020.

[85] P. S. Madhyastha, J. Wang, and L. Specia, "Vifidel: Evaluating the visual fidelity of image descriptions," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6539–6550, 2019.

[86] Y. Yi, H. Deng, and J. Hu, "Improving image captioning evaluation by considering inter references variance," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 985–994, 2020.

[87] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in European Conference on Computer Vision, pp. 104–120, Springer, 2020.

[88] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in Advances in Neural Information Processing Systems, pp. 13–23, 2019.

[89] L. Specia, K. Shah, J. G. De Souza, and T. Cohn, "Quest-a translation quality estimation framework," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 79–84, 2013.

[90] A. F. Martins, M. Junczys-Dowmunt, F. N. Kepler, R. Astudillo, C. Hokamp, and R. Grundkiewicz, "Pushing the limits of translation quality estimation," Transactions of the Association for Computational Linguistics, vol. 5, pp. 205–218, 2017.

[91] L. Specia, F. Blain, V. Logacheva, R. Astudillo, and A. F. Martins, "Findings of the wmt 2018 shared task on quality estimation," in Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pp. 689–709, 2018.

[92] T. Levinboim, A. V. Thapliyal, P. Sharma, and R. Soricut, "Quality estimation for image captions based on large-scale human evaluations," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Online), pp. 3157–3166, Association for Computational Linguistics, June 2021.

[93] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," 2018.

[94] J. Wang, W. Xu, Q. Wang, and A. B. Chan, "Compare and reweight: Distinctive image captioning using similar images sets," in ECCV, 2020.

[95] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3128–3137, 2015.

[96] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, pp. 5998–6008, 2017.

[97] J. Cho, S. Yoon, A. Kale, F. Dernoncourt, T. Bui, and M. Bansal, "Fine-grained image captioning with clip reward," in Findings of NAACL, 2022.

[98] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," Expert Systems with Applications, p. 113679, 2020.

[99] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[100] Z. Zhao, S. B. Cohen, and B. Webber, "Reducing quantity hallucinations in abstractive summarization," in Findings of the Association for Computational Linguistics: EMNLP 2020, (Online), pp. 2237–2249, Association for Computational Linguistics, Nov. 2020.

[101] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1906–1919, 2020.

[102] R. Bora-Kathariya and Y. Haribhakta, "Natural language inference as an evaluation measure for abstractive summarization," in 2018 4th International Conference for Convergence in Technology (I2CT), pp. 1–4, IEEE, 2018.

[103] T. Falke, L. F. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych, "Ranking generated summaries by correctness: An interesting but challenging application for natural language inference," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2214–2220, 2019.

[104] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1112–1122, 2018.

[105] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in International Conference on Learning Representations, 2019.

[106] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, (Brussels, Belgium), pp. 1797–1807, Association for Computational Linguistics, Oct.-Nov. 2018.

[107] A. R. Fabbri, W. Kryscinski, B. McCann, C. Xiong, R. Socher, and D. Radev, "Summeval: Re-evaluating summarization evaluation," Transactions of the Association for Computational Linguistics, vol. 9, pp. 391–409, 2021.

[108] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, (Berlin, Germany), pp. 280–290, Association for Computational Linguistics, Aug. 2016.

[109] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in ICLR (Poster), 2015.

[110] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, "Adversarial nli: A new benchmark for natural language understanding," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4885–4901, 2020.

[111] T. Goyal and G. Durrett, "Annotating and modeling fine-grained factuality in summarization," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1449–1462, 2021.

[112] G. Tevet and J. Berant, "Evaluating the evaluation of diversity in natural language generation," in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, (Online), pp. 326–346, Association for Computational Linguistics, Apr. 2021.

[113] H. Lin and V. Ng, "Abstractive summarization: a survey of the state of the art," in Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, pp. 9815–9822, 2019.

[114] M. Cao, Y. Dong, J. Wu, and J. C. K. Cheung, "Factual error correction for abstractive summarization models," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6251–6258, 2020.

[115] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, and M. Jiang, "Enhancing factual consistency of abstractive summarization," in Proceedings

of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 718–733, 2021.

[116] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in International Conference on Learning Representations, 2018.

[117] S. Chen, F. Zhang, K. Sone, and D. Roth, "Improving faithfulness in abstractive summarization with contrast candidate generation and selection," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Online), pp. 5935–5941, Association for Computational Linguistics, June 2021.

[118] J. Thorne and A. Vlachos, "Evidence-based factual error correction," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3298–3309, 2021.

[119] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 809–819, 2018.

[120] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

초 록

최근의 사전학습 언어모델의 활용을 통한 조건부 텍스트 생성 시스템들의 발전에도 불구하고, 시스템들의 사실 관계의 일관성은 여전히 충분하지 않은 편이다. 그러나 널리 사용되는 n-그램 기반 유사성 평가 기법은 사실 일관성 평가에 매우 취약하다. 따라서, 사실 일관된 텍스트 생성 시스템을 개발하기 위해서는 먼저 시스템의 사실 관계를 제대로 평가할 수 있는 자동 평가 기법이 필요하다. 본 논문에서는 다양한 조건부 텍스트 생성 시스템에 대해, 이전 평가 기법보다 사실 관계 일관성 평가에서 인간의 판단과 매우 높은 상관관계를 보여주는 4가지 평가 기법을 제안한다. 이 기법들은 *(1) 보조 태스크 활용* 및 *(2) 데이터 증강 기법* 등을 활용한다.

첫째로, 우리는 중요한 핵심 단어또는 핵심 구문에 초점을 맞춘 두 가지 다른 보조 태스크를 활용하여 두 가지 사실 관계의 일관성 평가 기법을 제안한다. 우리는 먼저 핵심 구문의 가중치 예측 태스크를 이전 평가 기법에 결합하여 주관식 질의응답을 위한 평가 기법을 제안한다. 또한, 우리는 질의 생성 및 응답을 활용하여 키워드에 대한 질의를 생성하고, 이미지와 캡션에 대한 질문의 답을 비교하여 사실 일관성을 확인하는 QACE를 제안한다.

둘째로, 우리는 보조 태스크 활용과 달리, 데이터 기반 방식의 학습을 통해 두 가지의 평가 기법을 제안한다. 구체적으로, 우리는 증강된 일관성 없는 텍스트를 일관성 있는 텍스트와 구분하도록 훈련한다. 먼저 규칙 기반 변형을 통한 불일치 캡션 생성으로 이미지 캡션 평가 지표 UMIC을 제안한다. 다음 단계로, 마스킹된 소스와 마스킹된 요약을 사용하여 일관성이 없는 요약을 생성하는 MFMA를 통해 평가 지표를 개발한다. 마지막으로, 데이터 기반 사실 일관성 평가 기법 개발의 확장으로 시스템의 사실 관계 오류를 수정할 수 있는 빠른 사후 교정 시스템을 제안한다.

주요어: 사실 관계의 일관성, 텍스트 생성, 평가 기법
학번: 2017-26066