2022

Methods For Text Summarization Evaluation

Daniel Deutsch
University of Pennsylvania

Methods For Text Summarization Evaluation

Abstract

The ability to effectively evaluate a learned model is a critical component of machine learning research; without it, progress on tasks cannot be measured and is thus impossible. In the natural language processing task of text summarization, evaluation is incredibly difficult: the notion of the "perfect" summary content is ill-defined, but even if it could be defined, that content can be expressed in many different ways, making it difficult to identify in a summary. The evaluation metrics that researchers propose for text summarization must overcome these challenges in some way. In this thesis, I identify problems with the existing methodologies for evaluating summaries as well as meta-evaluating the quality of an evaluation metric and propose solutions for improving them. I demonstrate that commonly used evaluation metrics fail to properly evaluate the information content of summaries and propose an evaluation metric based on question-answering to address the shortcomings of existing metrics. Then, I argue that the class of metrics which attempt to evaluate the quality of a summary's content without the aid of a human-written reference is inherently biased and limited in its ability to evaluate summaries. Finally, I identify that the methodology for quantifying how well an automatic metric agrees with human judgments of summary quality fails to provide a complete understanding of a metric's performance. To that end, I propose new statistical analysis tools to address the limitations of the standard meta-evaluation procedure and provide a new protocol for meta-evaluating metrics that better evaluates metrics in realistic use cases.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

Dan Roth

Keywords

evaluation, evaluation metrics, question-answering, summarization

Subject Categories

Artificial Intelligence and Robotics

METHODS FOR TEXT SUMMARIZATION EVALUATION

Daniel Deutsch

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2022

Supervisor of Dissertation

Dan Roth, Eduardo D. Glandt Distinguished Professor

Graduate Group Chairperson

Mayur Naik, Professor of Computer and Information Science

Dissertation Committee

Mitchell P. Marcus, RCA Professor of Artificial Intelligence Emeritus

Lyle Ungar, Professor of Computer and Information Science

Chris Callison-Burch, Associate Professor of Computer and Information Science

Mirella Lapata, Professor, Institute for Language, Cognition and Computation, University of Edinburgh

METHODS FOR TEXT SUMMARIZATION EVALUATION

Dedicated to my parents,

who have supported me through over two decades of school,

and to my grandmother,

who instilled in me the value of education

ACKNOWLEDGMENT

On paper, a PhD thesis is an individual effort, a culmination of the research contributions of one person. However, anyone who has done a PhD knows this could not be further from the truth, myself included. There are many people who I must thank for helping me get to this point.

First and foremost is my advisor, Dan Roth. As the dozens of students who came before me know, Dan is a fantastic advisor, both in terms of the technical side of research and his high-level professional advice. There are many occasions that I can recall in which Dan made an argument in one of our meetings, and I only realized he was right three months later. He has provided me with opportunities that very few others in the field could, and for that I am very thankful. However, when I look back on my time working with Dan, I suspect that I will most remember the times that I spent with him in less formal settings. Whether it was during our countless discussions of the latest tennis news, traveling to a Dagstuhl seminar in Germany, riding the cable car to see the Big Buddha in Hong Kong, or getting stranded in the middle of nowhere in the Dominican Republic, chatting with and getting to know Dan was an activity that I always enjoyed and looked forward to.

Then, this thesis would not be the same without my friend and collaborator Rotem Dror. Little did we know that when we met at the Amazon office in Cambridge that we would later go on to have a very successful collaboration together. Our lunchtime conversations helped shape my research agenda; small ideas turned into short papers which turned into long papers that ultimately became a large part of my thesis. Rotem and I have complementary research strengths. Her interest in theory and mine in empirical applications came together to form a great working relationship. I have really enjoyed getting to know her and working with her over the last couple of years.

It cannot go without saying that this entire thesis was completed during the COVID-19 pandemic. The university largely shut down, in-person meetings were cancelled, and students took classes and did research mostly from home. However, for much of that time, Nitish Gupta and I were fortunate enough to be able to work together from the lab on campus, which provided some minor level of normalcy to our lives. We ate countless lunches together in which we discussed our own research

and played tennis in the evenings. Although his name is not listed on any of my publications, Nitish always provided incredibly useful feedback and advice about research that certainly had an impact on my success as a PhD student.

I would like to thank my thesis committee — Mitch Marcus, Lyle Ungar, Chris Callison-Burch, and Mirella Lapata — for their incredibly insightful feedback during my proposal and defense. I count myself incredibly lucky to have such a strong group of researchers be on my thesis committee.

Then, there are many others within CogComp and the NLP community at Penn that I spent time with over the years: Stephen Mayhew, Shyam Uphadyay, Jordan Kodner, Reno Kriz, Jennifer Sheffield, Marianna Apidianaki, João Sedoc, Sihao Chen, Daphne Ippolito, Ben Zhou, Xiaodong Yu, Hangfeng He, Yi Zhang, Liam Dugan, Qing Lyu, Alyssa Hwang, Bryan Li, John Hewitt, Aditya Kashyap, Eleni Miltsakaki, Daniel Khashabi, Daniel Stekol, Anne Cocos, Elior Sulem, Soham Dan, Rebecca Iglesias-Flores, Salvatore Giorgi, Krunal Shah, Xingyu Fu, and Helen Jin. Spending time with you all at conferences, lunch, and TGIF will be a highlight of my time at Penn.

Finally, I would like to thank my family and friends outside of Penn. Although many of you likely never fully understood exactly what I was doing for the last five years, you were supportive of me and kept me sane the entire time.

ABSTRACT

METHODS FOR TEXT SUMMARIZATION EVALUATION

Daniel Deutsch

Dan Roth

The ability to effectively evaluate a learned model is a critical component of machine learning research; without it, progress on tasks cannot be measured and is thus impossible. In the natural language processing task of text summarization, evaluation is incredibly difficult: the notion of the "perfect" summary content is ill-defined, but even if it could be defined, that content can be expressed in many different ways, making it difficult to identify in a summary. The evaluation metrics that researchers propose for text summarization must overcome these challenges in some way. In this thesis, I identify problems with the existing methodologies for evaluating summaries as well as meta-evaluating the quality of an evaluation metric and propose solutions for improving them. I demonstrate that commonly used evaluation metrics fail to properly evaluate the information content of summaries and propose an evaluation metric based on question-answering to address the shortcomings of existing metrics. Then, I argue that the class of metrics which attempt to evaluate the quality of a summary's content without the aid of a human-written reference is inherently biased and limited in its ability to evaluate summaries. Finally, I identify that the methodology for quantifying how well an automatic metric agrees with human judgments of summary quality fails to provide a complete understanding of a metric's performance. To that end, I propose new statistical analysis tools to address the limitations of the standard meta-evaluation procedure and provide a new protocol for meta-evaluating metrics that better evaluates metrics in realistic use cases.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1 : Introduction

Evaluation is a critical component of the development cycle of a machine learning model. Once a model has been trained, measuring how well it performs on the task it was designed to do facilitates both understanding the quality of the model and deciding which models are better than others. Without evaluations that effectively measure model performance, we do not know how well new research proposals work, and thus progress in machine learning is not possible.

In classic natural language processing (NLP) tasks, evaluating the performance of a learned model is relatively straightforward; human annotators provide ground-truth labels for the task at hand, and the predicted outputs from the model can be quickly verified as being right or wrong. For instance, a model's predicted class in k-class classification either matches the ground-truth label or it does not. In structured prediction tasks, such as part-of-speech tagging or constituency parsing, the output can additionally be neither entirely correct nor incorrect, but evaluating the predicted structure still effectively reduces to checking if sub-structures identically match the ground-truth. Although the ground-truth labels for these tasks can sometimes be ambiguous, these cases are the exception rather than the rule; once a ground-truth label is determined, evaluation is straightforward.

However, in recent years, interest in natural language generation tasks has outpaced more classic NLP tasks, and evaluating generated text is far from simple. Consider the task which is the focus of this thesis, text summarization. Although it is ill-defined, from a very high-level, the goal of text summarization is to generate a fluent natural language summary of some input text(s). Because there are practically an infinite number of perfectly acceptable summaries for the same input, it is futile to define a single ground-truth summary and determine if the model's prediction contains the correct salient information based on whether it is the same as the expected summary; in almost every case, the summary would be considered incorrect, which is not very informative.

Ideally, humans would judge the quality of a summary since they would be able to best determine whether the summary contains salient, "summary-worthy" information. However, this is impractical due to the time and cost overhead required to use a human judge every time we need to measure

the quality of a summary. Instead, researchers have proposed various ways of approximating how human judges would rate the quality of a summary without using a human-in-the-loop, known as automatic evaluation metrics. Building such an evaluation metric is rife with challenges, including how to identify and represent salient information and how to determine whether a summary contains that information.

However, because automatic evaluation metrics rely on algorithms or learned models to rate the quality of a summary, their scores are inherently sub-optimal because they do not evaluate summaries the way humans do. As such, it is useful to know how similarly an automatic metric and a human judge would agree on the quality of a summary, that way we understand whether or not the automatic metric is a good substitute for human evaluations. Accurately estimating this similarity (a methodology called metric meta-evaluation) is critically important for understanding how much trust we should put into automatic evaluations of summaries.

In this thesis, I identify various problems in the ways that summaries are evaluated and metrics are meta-evaluated and propose solutions to improve both of these methodologies. An overview of the flaws in each of these types of evaluations is described next.

Evaluating Summaries. The most popular approaches for automatically evaluating summaries do so by assuming access to a ground-truth summary (typically called the reference) and compare the candidate summary to the reference. These metrics implicitly make a (likely incorrect) assumption that the reference summary contains all of the gold-standard salient content from the input, and thus summaries which are more similar to this reference are likely to be higher-quality. Instead of doing an exact match comparison between the two texts, reference-based metrics define ways for extracting and representing the content of the summaries and identifying occurrences of that information in the other summary. Each metric differs on how they solve these problems.

Various metrics, including the de-facto summarization evaluation metric, ROUGE (Lin, 2004), use lexical representations of the summaries and compare two summaries based on their lexical overlap. More recent approaches, such as BERTScore (Zhang et al., 2020), leverage advances in contextual word embeddings (Peters et al., 2018; Devlin et al., 2019) to represent the summaries with

high-dimensional embeddings and calculate a similarity score based on how similar the respective embeddings are. Although these methods appears to have strong correlation to human judgments of summary quality, it is not very clear how well these metrics represent and compare the information content of the summaries. That is, the community is uncertain whether a high ROUGE or BERT-Score value implies the semantic contents of the summaries are similar or it means they are more similar on some less desirably dimension of similarity. This casts doubt on how effective they are as evaluation metrics because they may not evaluate the aspects of summaries which researchers want to measure.

Other approaches to automatic summary evaluation abandon the need for a reference summary altogether and aim to evaluate the content of a summary in a reference-free manner (Louis and Nenkova, 2013; Gao et al., 2020; Vasilyev et al., 2020; Scialom et al., 2019, 2021). Instead of using a human-written reference to identify what input content is salient, they use either a rule-based or learned notion of information salience. While this style of metric is less popular, its advantages are intriguing. Reference summaries can be expensive and difficult to collect, especially in new specialized domains, and thus developing a reliable reference-free evaluation metric would enable quickly and cheaply building summarization systems when references are not available. However, the implications for using a rule-based or learned definition of salience are not well-understood, so it is unclear whether this negatively affects the ability of reference-free metrics to evaluate summaries.

Meta-Evaluating Metrics. Regardless of whether an automatic evaluation metric is reference-based or reference-free, its score is an approximation for how humans would evaluate the quality of a summary. As such, metrics are meta-evaluated by calculating how strongly their scores correlate to those assigned by human judges (Dang and Owczarzak, 2008; Callison-Burch et al., 2006, 2008). These correlations measure the strength of the relationship between the metric and human judgments and allow for arguing that one automatic metric is better than another by comparing their correlations. However, the way in which meta-evaluation is presently done does not provide a complete understanding of the performance of a metric.

Specifically, little is known about how precise the estimates of these correlations are, leaving doubt

about how much the metrics' scores actually correspond to summary quality. Then, it is not clear whether observed differences in correlations are due to metric improvements or random chance, so there are no clear recommendations to practitioners for what metrics should be used. Further, the standard definitions of the correlations that are most frequently used evaluate metrics do so in an unrealistic — and likely far too easy — setting, so the performance of metrics how they are used in practice by researchers is not well-understood.

1.1. Thesis Statement

In this thesis, I argue that the way evaluation is done in summarization is flawed: standard automatic reference-based evaluation metrics do not evaluate summaries in a way that aligns with the community's research goals; automatic reference-free metrics are an inherently flawed and limited class of evaluation metric; and evaluation metrics themselves are not properly meta-evaluated. Together, these problems negatively impact the community's ability to evaluate summaries, thereby limiting progress on building automatic summarization systems.

In the chapters that follow, I discuss in detail both the explanations of these problems as well as a proposal of various solutions. I demonstrate how question-answer (QA) pairs are an easy-to-use representation framework for the semantic content of summaries and show that they can be leveraged to create a high-quality evaluation metric as well as a better summarization model. Then, I explain that reference-free metrics should be used as diagnostic tools for analyzing model behavior and not as bellwethers for progress on summarization. Finally, I propose new statistical techniques based on resampling, which improve the rigor of metric meta-evaluation, and provide a new protocol for how metric meta-evaluation should be done to better align with how evaluation metrics are actually used in practice.

1.2. Thesis Outline

The rest of this document is organized as follows:

- Chapter 2 provides an overview of text summarization evaluation, including the different approaches for evaluating summaries as well as the methodology for meta-evaluating metrics. It also contains a brief discussion of how a related field of NLP, machine translation, addresses

the same problems that are discussed in this thesis.

- Chapter 3 explores the fundamental question of what type of similarity between reference and candidate summaries should be measured by reference-based evaluation metrics. We argue that these metrics should ideally evaluate summaries by measuring how much information they have in common with the references, and we perform an analysis that demonstrates that two key metrics, ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020), largely fail to evaluate summaries this way.

- To address the shortcomings of existing evaluation metrics, in Chapter 4 we explore using question-answering (QA) to evaluate summaries through a proposal of a reference-based evaluation metric called QAEval. Our metric represents the information in a reference summary with QA pairs, then calculates what proportion of that information is contained in the candidate summary by automatically answering the QA pairs against it. We demonstrate improved performance over previous work in many evaluation settings, argue that QAEval's upper-bound performance surpasses all other metrics where it currently lags behind, and show evidence that it better evaluates a summary's information than ROUGE and BERTScore.

- In Chapter 5, we explore the idea of reference-free metrics, which aim to evaluate summaries without references, for instance, by predicting a set of salient questions for evaluation from the input document rather than generating them from the reference summary. However, we argue that reference-free metrics are inherently flawed and should not be used as primary evaluation methods because they can be directly optimized during inference to obtain the "perfect" output under the metric, which is very unlikely to be the best possible output according to humans.

- In Chapter 6, we briefly depart from evaluation and explore how QA-based representations can be used to improve summary generation by using them to identify salient document content. First, salient document spans are identified through a question-generation and answering procedure, similar to that used by QAEval. Then, the spans are used to incorporate an

inductive bias into a two-stage summarization model, which first predicts salient spans in the document, then conditionally generates a summary. We demonstrate how this procedure leads to better end-to-end summaries in addition to improving the controllability of the summary's content.

- Then, the focus of the thesis switches to the methodology used for meta-evaluation metrics. Chapter 7 identifies that the community is very uncertain how precisely metrics correlate to human judgments of summary quality as well as which metrics truly correlate more strongly than others. To that end, we propose a set of statistical methods for estimating confidence intervals for metric correlations and running hypothesis tests to argue one metric has a higher correlation than another. We demonstrate how these methods provide new insights into the behavior of summarization evaluation metrics.

- Chapter 8 re-examines the standard definition of the system-level correlation that is most frequently used to meta-evaluate metrics and points out there are at least two ways in which there is a disconnect between how meta-evaluation is performed and how metrics are used in practice. We propose fixes to the meta-evaluation methodology and demonstrate that, although the improvements allow us to more precisely estimate the metrics' correlations, they also reveal that the standard evaluation procedure largely over-estimates the quality of metrics as they are used in practice. This latter result casts significant doubt over how reliably the most commonly used evaluation metrics replicate human judgments of summarization system quality.

- Finally, Chapter 9 summarizes the contributions of the thesis and discusses possible directions for future research.

CHAPTER 2 : Background

2.1. Introduction

Text summarization is an ill-defined task. From a high-level, the goal is to write a natural language summary of some input text in which the summary expresses the salient information from the input. However, it is often difficult to describe exactly what content is salient because reasonable people may disagree about what constitutes a high-quality summary of the same input.

Despite the task's ambiguity, the need to evaluate the generated summaries is still clear. As such, researchers have proposed various metrics for evaluating summaries, both with a human in-the-loop and automatically with an algorithm or learned model. Because the automatic metrics are generally an approximation of how a human judge would evaluate a summary, the metrics themselves can be evaluated to quantify how similar their scores are to those judged by humans through a process known as metric meta-evaluation.

This Chapter provides the necessary background on both of these types of evaluation — evaluating summaries and meta-evaluating metrics — for the research presented in this thesis as well as a brief discussion of how evaluation is performed in another text generation task, machine translation, in order to better contextualize the research presented in this thesis.

2.2. Evaluating Summaries

Accurately evaluating the quality of a summary is critical — and often overlooked — problem within text summarization. If reliable evaluation procedures do not exist, then it is impossible to determine whether research claims that a newly proposed model generates better summaries are true, and thus progress on text summarization is not possible.

Different Aspects of a Summary. There are many different aspects of summaries which can be evaluated. Various dimensions and their definitions include the following:

- **Linguistic quality**: Is the summary grammatically correct?
- **Coherence**: Is the content of the summary presented in a well-organized manner?

- **Faithfulness**: Is all of the information in the summary present in the input document(s)?
- **Content quality**: Does the summary contain the salient information from the input document(s)?

This list is by no means exhaustive nor are the aspects and their definitions standardized within the community (Howcroft et al., 2020). Although they are related and correlated to each other, evaluating each aspect involves its own set of challenges and requires different approaches and solutions.

We argue that determining what content should or should not be included in the summary is the defining challenge for the task of summarization, and the vast majority of summarization evaluation research has been focused on the content quality dimension, likely for this reason. As such, the evaluation work in this thesis is also focused on evaluating content quality. When we mention the "quality" of a summary throughout this thesis, we refer to extent to which the summary contains the salient information from the input document(s). The scope of the subsequent discussion about summary evaluation in this Chapter is also limited to evaluating summary content.

Evaluation Challenges. Evaluating the content of a summary is a very difficult task. First, due to the ill-defined nature of the summarization task, it is infeasible to explicitly define what salient content from the document should be included in the summary. If two expert annotators were asked to write summaries of the same input document(s), it is almost certainly the case that they will not both write summaries which include exactly the same content. They both will mention common information, but they will also include content that the other did not. However, both summaries could be considered equally good. This stands in contrast to more traditional tasks, such as part-of-speech tagging, where there is one correct tag per token (ignoring some ambiguous cases, which are the exception rather than the rule). The output is either right or wrong.

Even if we did know exactly what content should be in the summary, the same information can be expressed in different ways, and identifying phrases which do share the same meaning is non-trivial. For example, expert annotators have marked the underlined phrases as expressing the same

information:[1]

- The European Airbus A380 flew its maiden test flight from France 10 years after design development started.
- The first A380 arrived in January 2005, taking its maiden flight April 27.
- The largest passenger airline ever built, the Airbus 380 (A380), took off on its maiden four-hour flight on April 27, 2005 in France.
- The A380, carrying between 555 and 840 passengers, was unveiled in January 2005 and test flown in April.

Automatically identifying these phrases are semantically equivalent requires understanding the syntax of the sentences, identifying that "flew/flown" are synonyms of "taking" and "took off," and detecting that "European Airbus A380" and "A380" refer to the same entity. It is easy to imagine more complicated examples which might require a higher level of reasoning to match semantically equivalent phrases correctly.

Evaluation Metrics. Researchers have proposed various different metrics for evaluating the content of summaries, each with their own methods for attempting to solve the above problems. Evaluation metrics can be described and categorized by various different dimensions, such as whether they are intrinsic or extrinsic evaluations, whether they require a reference summary, etc. However, arguably the biggest distinction among metrics is whether the metrics are manual or automatic. Manual evaluation metrics require a human-in-the-loop to score a new summary, whereas automatic metrics can estimate the quality of a new summary with no additional human input. The two types of metrics are discussed in more detail in the following two sections, respectively.

2.2.1. Manual Evaluations of Summaries

High-quality manual evaluations of summaries are largely considered to be the preferred way to evaluate summaries because the human judges are able to perform tasks that automatic methods

[1]This example was taken from the Pyramid Method annotations (Nenkova and Passonneau, 2004; Nenkova et al., 2007) on the TAC 2008 dataset (Dang and Owczarzak, 2008), both discussed later in this Chapter.

cannot do as reliably. For instance, it is theoretically possible to train a human judge to score summaries according to a specific set of criteria. Performing the same evaluation through automatic means is likely incredibly difficult for any practical application of summarization, and thus manual evaluations are considered to be more trustworthy than automatic metrics.

Methods for manually evaluating summaries can be categorized as either direct assessments, in which the candidate summary is evaluated based on its own merit as a summary, or reference-based, in which the candidate content is compared to a gold-standard summary under the assumption that the reference contains the "perfect" summary content. Both methods have their advantages and disadvantages.

Direct assessments provide methods for human judges to score summaries using any set of criteria that is provided to them. This allows for soliciting judgments for the exact aspects of summaries that you want to evaluate, which is a rather useful property of an evaluation metric. However, this requires clearly describing the scoring rubric and training judges to adhere to it so the scores of different summaries are comparable, which is likely quite difficult to do. Further, accurately evaluating the content of a summary requires a good understanding of the input documents' contents. In a multi-document scenario — or even in single-document settings — this can be quite cognitively taxing and a difficult task for judges to perform.

In contrast, the reference-based methods evaluate a summary by determining how similar it is to a "perfect" summary, which is typically written by a human annotator. This task is likely simpler than direct assessments because the amount of text which the judges must read and understand is significantly smaller. However, the downside is that the content evaluation is limited to what is contained in the reference summary. It is almost certainly true that the reference summary does not contain all of the document content that would be deemed as acceptably salient by human judges. Therefore, if the candidate summary contained information that is not present in the reference, you cannot directly conclude that the information is bad or unimportant, but the candidate summary will be penalized for it anyway.

Below, we discuss the different ways researchers have used direct assessments and reference-based evaluations for summarization.

Direct Assessment. Direct assessment evaluations of summaries aim to score the inherent quality of a summary based on a set of pre-defined criteria. This criteria can be any description, but it often is based around whether the summary contains the salient information from the input or whether the summary successfully achieved the task it was generated to perform (e.g., provide a specific information need to a user).

Some of the first attempts to define a description of a high-quality summary for direct assessment came during the Document Understanding Conferences (DUC) and Text Analysis Conferences (TAC), which ran a series of shared tasks on summarization. Across different iterations of DUC and TAC, the organizers developed and refined the definition of a high-quality summary that was given to expert judges, who were instructed to use it to assign each summary a score. For instance, in DUC 2005 the summarization task which was evaluated was to write a multi-document summary that answers an information need at a pre-specified level of granularity. The instructions provided to the annotators for a metric called "responsiveness" was the following:[2]

> Responsiveness should be measured primarily in terms of the amount of information in the summary that actually helps to satisfy the information need expressed in the topic statement, at the level of granularity requested in the user profile. The linguistic quality of the summary should play a role in your assessment only insofar as it interferes with the expression of information and reduces the amount of information that is conveyed.

Because the judgments in DUC were done by expert annotators who read all of the input documents and understood the task description, the collected responsiveness scores are considered to be high-quality. However, only one judge scored each summary, so inter-annotator agreement was not calculated. Nevertheless, the responsiveness scores (or later "overall responsiveness," which included an evaluation of the summary's linguistic quality) are used as ground-truth scores to meta-evaluate automatic metrics.

[2] `https://duc.nist.gov/duc2005/responsiveness.assessment.instructions`

Although DUC used expert judges in its evaluations, direct assessments which are performed today are largely done using crowd workers and platforms like Mechanical Turk. There is no standard description of a high-quality summary that is used (Howcroft et al., 2020) nor a standard annotation interface. Evaluating summaries with crowd workers is incredibly difficult. The annotations frequently have very poor inter-annotator agreements (Fabbri et al., 2021), potentially due to the cognitive difficulty of task and lack of annotator training. Tasks given to crowd workers are typically rather small and well-defined. In contrast, directly assessing summary quality requires fully understanding all of the input text, and providing consistent judgments likely requires training the annotators to teach them exactly how the summaries should be scored. Together, these challenges make crowd-based evaluations of summaries incredibly difficult.

Reference-Based. Reference-based manual evaluations of summaries are largely done via the Pyramid Method (Nenkova and Passonneau, 2004; Nenkova et al., 2007) or the Lightweight Pyramid Method (Shapira et al., 2019). Each of these methods is described below.

The Pyramid Method is the gold-standard for comparing the content of a candidate summary to a reference summary. It leverages expert annotators to extract phrases which express atomic units of meaning from the reference summaries and identify occurrences of the same unit of meaning in the candidate summaries. The intuition is that more matches found in the candidate summary indicates that it shares more meaning with the references and is likely a good summary.

These units of meaning, know as Summarization Content Units (SCUs), are not precisely defined in terms of what they should represent. Instead, the authors describe a procedure for identifying them. Given two texts, expert annotators are instructed to first find similar sentences and then break down the sentences into subparts which share the same meaning. The annotators define an abstract meaning for the SCU which is common across all of its occurrences, even though that same meaning may be expressed differently in each case. In practice, we find that the information represented by each SCU is roughly equivalent to what is expressed by the predicate-argument relations in the text.

The annotation procedure to calculate a score for a summary proceeds as follows. First, the expert annotators construct what is known as the pyramid, a collection of weighted SCUs extracted from

the reference summaries. For each of the k reference summaries for the same input document(s), an expert annotator identifies all of the SCUs in each of the references, marking occurrences of the same SCU across the references. Let $M_i \in \mathcal{M}_{\text{pyr}}$ be a specific SCU in the set of all identified SCUs, $\mathcal{M}_{\text{pyr}}$. M_i is assigned a weight $w_i \in \{1, \ldots, k\}$ equal to the number of reference summaries that M_i appears in. The set of SCUs and their weights define the pyramid. Typically, there is a small number of SCUs with a large weight (forming the top of the pyramid) and a larger number of SCUs with a small weight (the bottom of the pyramid). Although the Pyramid Method was designed to handle multiple references, it can still be applied in principle to datasets with only one reference.

Then, the pyramid is used to score a candidate summary. For each SCU in the pyramid, an expert judge determines whether the candidate summary contains an occurrence of the SCU. Let $\mathcal{M}_{\text{sum}}$ be the set of SCUs in the candidate summary. Then, the unnormalized score of the summary is equal to the sum of the weights the SCUs in $\mathcal{M}_{\text{sum}}$:

$$D = \sum_{M_i \in \mathcal{M}_{\text{sum}}} w_i \tag{2.1}$$

Then, let O_n be equal to the highest possible weight of a summary with n SCUs. That is, it contains all SCUs of weight k, then $k-1$, etc., until it has n SCUs. The Pyramid Score is calculated by normalizing D by O_n where n is equal to the average number of SCUs in the reference k reference summaries.[3]

Calculating the Pyramid Score is labor intensive and expensive because it requires expert annotators who have been trained to identify and match SCUs. As such, researchers have proposed methods for relaxing the assumption that the annotators are experts and try to perform the Pyramid Method with crowdworkers. This approach is known as the Lightweight Pyramid (Shapira et al., 2019).

The Lightweight Pyramid makes several changes to the Pyramid Method to decrease the required

[3]This is technically known as the modified Pyramid Score because D is normalized by the weight of an optimal summary with the number of SCUs equal to the average number of SCUs in the references. The original Pyramid Score instead normalizes by the weight of an optimal summary that contains the same number of SCUs as are in the candidate summary. The modified score can be calculated by only identifying which SCUs in $\mathcal{M}_{\text{pyr}}$ appear in the summary; the original score requires exhaustively annotating all of the SCUs in the summary itself, which is much more time consuming and expensive.

level of human annotation. First, the SCUs are defined to be small important facts that are copy-and-pasted from the text of the references; they do not define an abstract meaning for the SCUs.

Then, they do not exhaustively annotate all of the SCUs in the references to form a Pyramid. Instead, each crowd worker is instructed to identify up to 8 SCUs per reference, which is repeated using 2 crowd workers, resulting in 16 SCUs per reference. The SCUs are not merged within a single reference, so there may be repeated information.

To construct the Pyramid, they gather all of the SCUs annotated across the k reference summaries and define the pyramid to be a randomly selected subset of them. Again, no merging of SCUs is done; the assumption is that if an SCU occurs across multiple references, it is more likely to be represented multiple times in the random sample.

Finally, 5 crowd workers judge whether each of the SCUs in the pyramid are present in the candidate summaries, taking the majority vote to decide whether or not it is truly present. The Lightweight Pyramid score is equal to the percent of matched SCUs out of the total number in the pyramid.

Because the Lightweight Pyramid does not exhaustively annotate the full pyramids and does not require expert judges to compute, it is a faster and cheaper evaluation metric than the standard Pyramid Method. Although it is less likely that the Lightweight annotations would perfectly match those done by an expert, Shapira et al. (2019) show the Lightweight score correlates well with the expert-based Pyramid Score.

2.2.2. Automatic Evaluations of Summaries

Although manual evaluation metrics are considered to be the preferred style of evaluation, they are very slow, often quite expensive to calculate, and do not scale to a large number of summaries. This makes them difficult to use, especially during model development when researchers require quick feedback about model quality.

Instead, researchers turn to automatic evaluation metrics, which are relatively fast and cheap to calculate and can easily scale to hundreds of thousands of summaries. Automatic metrics require no human-in-the-loop to score a newly generated summary. Many of them do require human input

Metric	Salient Content Representation	Salient Content Extractor	Content Identification Method	Content Matcher
Pyramid Score (Nenkova and Passonneau, 2004)	SCUs	Experts	Semantic Matching	Experts
Lightweight Pyramids (Shapira et al., 2019)	SCUs	Crowd Workers	Semantic Matching	Crowd Workers
ROUGE (Lin, 2004)	n-grams	Automatic	Lexical Matching	Automatic
Basic Elements (Hovy et al., 2006)	Dependency Relations	Automatic	Lexical Matching	Automatic
BEwT-E (Tratz and Hovy, 2008)	Dependency Relations	Automatic	Lexical Matching	Automatic
AutoSummENG (Giannakopoulos et al., 2008)	Character n-gram Graphs	Automatic	Graph Similarity	Automatic
MeMoG (Giannakopoulos and Karkaletsis, 2010)	Character n-gram Graphs	Automatic	Graph Similarity	Automatic
NPowER (Giannakopoulos and Karkaletsis, 2013)	Character n-gram Graphs	Automatic	Graph Similarity	Automatic
PEAK (Yang et al., 2016)	OpenIE Tuples	Automatic	Lexical Matching	Automatic
PyrEval (Gao et al., 2019)	SCUs	Automatic	Embedding Similarity	Automatic
S^3 (Peyrard, 2019)	Textual Features	Automatic	Learned Model	Automatic
BERTScore (Zhang et al., 2020)	Word Embeddings	Automatic	Cosine Similarity	Automatic
MoverScore (Zhao et al., 2019)	Word Embeddings	Automatic	Movers Distance	Automatic
APES (Eyal et al., 2019)	Fill-in-the-Blank Q's	Automatic	Reading Comprehension Model	Automatic
QAEval (Chapter 4; Deutsch et al., 2021a)	Wh-Questions	Automatic	Question-Answering Model	Automatic

Table 1: An overview of the differences between reference-based metrics. Every metric assumes the reference summary contains the salient information, creates some representation of that information, then identifies whether or not that content exists in the candidate summary.

in the form of an example reference summary, however once that reference is written, no additional human intervention is required. The key trade-off between manual and automatic evaluations is that automatic evaluations often fail to directly measure the exact aspects of summaries that we are interested in evaluation, but instead approximate it with some substitute score.

There are many different ways to characterize automatic evaluation metrics, however like manual evaluations, arguably the most important distinction is whether they require a reference summary or not. Reference-based evaluations rely on reference summaries to identify what content from the input document is salient and then evaluate a candidate summary based on how similar it is to the reference, effectively reducing the evaluation task to a text comparison problem. In contrast, reference-free evaluations directly or indirectly define a model of document content salience and use that to evaluate the content of the candidate text.

The majority of the discussion which follows is focused on reference-based metrics, which are more popular than their reference-free counterparts. We include detailed explanations about ROUGE and BERTScore since they are used frequently throughout this thesis and provide high-level overviews of other reference-based metrics (see Table 1 for a table which compares them). Finally, we include brief descriptions of various approaches to reference-free evaluation.

ROUGE. Recall-Oriented Understudy for Gisting Evaluation, or ROUGE, is a reference-based metric which is largely considered to be the de-facto automatic evaluation method for summarization (Lin, 2004). One potential reason for its popularity is that it was an official evaluation metric for the Document Understanding Conference during the 2000s, which largely standardized various tasks in summarization, including evaluation. It is still used in nearly every summarization paper to this day.

There are several different variants of ROUGE, however, at their core, they each measure the similarity of the candidate and reference summaries based on the number of tokens they have in common via lexical matching.

For the sake of this discussion, let X be the candidate summary with m tokens and Y be the reference summary with n tokens. Each variant of ROUGE is described below.

The most common variant of ROUGE, called ROUGE-N, counts the number of n-grams of order N that X and Y have in common, effectively using a bag of n-grams to represent the summaries' contents. This value is then normalized calculate recall and precision:

$$\text{ROUGE-N}_{\text{Recall}} = \frac{|\text{N-GRAMS}(X) \cap \text{N-GRAMS}(Y)|}{|\text{N-GRAMS}(Y)|} \tag{2.2}$$

$$\text{ROUGE-N}_{\text{Precision}} = \frac{|\text{N-GRAMS}(X) \cap \text{N-GRAMS}(Y)|}{|\text{N-GRAMS}(X)|} \tag{2.3}$$

Here, N-GRAMS$(\cdot)$ is used as a function that maps from the summary tokens to multiset of all of the n-grams of order N.[4]

ROUGE-S represents the summaries as skip-bigrams, which are pairs of tokens in the summary

[4] A multiset is a generalization of a set that allows for multiple members of the same value. The intersection of two multisets A and B produces a multiset that has n occurrences of member x where n is the minimum number of x times appears in A or B.

with an arbitrary number of tokens in between. The calculation is analogous to ROUGE-N:

$$\text{ROUGE-S}_{\text{Recall}} = \frac{|\text{SKIP-BIGRAMS}(X) \cap \text{SKIP-BIGRAMS}(Y)|}{|\text{SKIP-BIGRAMS}(Y)|} \tag{2.4}$$

$$\text{ROUGE-S}_{\text{Precision}} = \frac{|\text{SKIP-BIGRAMS}(X) \cap \text{SKIP-BIGRAMS}(R)|}{|\text{SKIP-BIGRAMS}(X)|} \tag{2.5}$$

ROUGE-SU extends ROUGE-S by also including all unigrams in the summary representations and including them in the multisets. In practice, both ROUGE-S variants limit the length of the gap to decrease the number of spurious matches. This value has been standardized to a gap length of 4, so the metrics are referred to as ROUGE-S4 and ROUGE-SU4.

The final variant of ROUGE is known as ROUGE-L. ROUGE-L is calculated based on the length of the longest common subsequence between the two summaries. A subsequence of length k is a list of k increasing indices that correspond to tokens in a summary. A common subsequence between two summaries is a pair of subsequences $(i_1, \ldots, i_k)$ and $(j_1, \ldots, j_k)$ such that $x_{i_\ell} = y_{j_\ell}$ for $\ell = 1, \ldots, k$ where x_i and y_j are tokens in X and Y. The longest such common subsequence can be efficiently computed with a dynamic program. The precision and recall for ROUGE-L are defined as:

$$\text{ROUGE-L}_{\text{Recall}} = \frac{\text{LCS}(X, Y)}{n} \tag{2.6}$$

$$\text{ROUGE-L}_{\text{Precision}} = \frac{\text{LCS}(X, Y)}{m} \tag{2.7}$$

where $\text{LCS}(\cdot)$ is a function that calculates the length of the longest common sequence between two summaries. A weighted extension of ROUGE-L, called ROUGE-W, which gives a higher score to subsequences that contain more adjacent tokens.

Depending on the particular evaluation setting, either the recall, precision, or corresponding F_1 value of the ROUGE variants are reported as approximations of summary quality.

BERTScore. Since the introduction of large-scale language models ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), contextual word embeddings have made a significant impact on the

field of NLP. Due to their clear successes and advantages over lexical representations and context-independent word embeddings, there has been interest among researchers in understanding the ways in which these models can be used to develop better evaluation metrics. BERTScore (Zhang et al., 2020) is among them.

BERTScore is a reference-based metric which calculates a similarity score between the reference and candidate summaries by inducing a weighted token-level alignment between the two summaries' tokens. Each token is represented by its BERT embedding. Then, all of the tokens in one summary are aligned to some token in the other summary based on which token's BERT embedding has the highest cosine similarity. The corresponding weight of the token alignment is equal to that cosine similarity, and the final score of the metric is a normalized sum of all of the alignment weights.

Specifically, let $Y = y_1, \ldots, y_n$ be the reference summary that is being used to evaluate candidate summary $X = x_1, \ldots, x_m$. First, the embedding for each token in the summaries is computed via BERT (or, more generally, any other method of embedding tokens). Then, the cosine similarity between each pair of BERT embeddings in the reference and candidate summaries is computed. Let matrix $B \in [-1, 1]^{m \times n}$ be the pairwise cosine similarities of each token X and Y. Entry B_{ij} is the similarity of the embeddings for x_i and y_j.

Next, two separate alignments are created between the summary tokens, one to calculate recall and one to calculate precision. The recall alignment maps every reference token to its most-similar candidate token. There are no restrictions on the alignment other than every reference token is aligned, so it is possible for no candidate tokens to be included in the alignment or for two reference tokens to be aligned to the same candidate token. The sum of the aligned similarities is then normalized by the number of reference tokens to calculate recall. This is defined mathematically in Equation 2.9. The same procedure is followed to calculate precision (Equation 2.8), but the roles of the reference

and candidate summaries are reversed.

$$\text{BERTScore}_{\text{Recall}} = \frac{1}{n}\sum_{j=1}^{n}\max_{i} B_{ij} \tag{2.8}$$

$$\text{BERTScore}_{\text{Precision}} = \frac{1}{m}\sum_{i=1}^{m}\max_{j} B_{ij} \tag{2.9}$$

The recall score can be interpreted as the amount of content in the reference which is also contained in the candidate. Similarly, the precision can be understood as the amount of content in the candidate that is in the reference. The recall, precision, or corresponding F_1 score can be used as the final evaluation score. In our own experiments, we observe that recall seems to correlate the best with human judgment.

Other Reference-Based Metrics. ROUGE and BERTScore can be viewed as representative metrics that use lexical and embedding-based representations of the summaries. However, other related metrics have also been proposed and used to evaluate summaries. Some notable metrics are briefly mentioned here, but not discussed in detail.

Meteor is another lexical metric which explicitly creates an alignment between the two summaries' tokens with a lookup table of synonyms and hyperparameters that encourage adjacent tokens in one summary to be aligned to nearby tokens in the other (Denkowski and Lavie, 2014).

Similar to ROUGE, Basic Elements (Hovy et al., 2006) and its extension BEwT-E (Tratz and Hovy, 2008) perform lexical matching between the two summaries, but they do so between "basic elements," which are sets of semantic units that are derived from the syntax of the summaries' sentences.

A series of metrics represent the summaries with character n-gram graphs and calculate a similarity score between the two summaries' graphs. Such metrics include AutoSummENG (Giannakopoulos et al., 2008), MeMoG (Giannakopoulos and Karkaletsis, 2010), and NPowER (Giannakopoulos and Karkaletsis, 2013).

MoverScore (Zhao et al., 2019) is a generalization of BERTScore in which the embedding-based

alignment is influenced by a cost matrix that penalizes alignments that aligns tokens that are far apart in the semantic space.

Some metrics, such as PEAK (Yang et al., 2016) and PyrEval (Gao et al., 2019) attempt to automate the Pyramid Method. PEAK represents SCUs by OpenIE tuples which are automatically extracted from and matched across the two summaries. PyrEval identifies SCUs by decomposing sentences into clauses, then matches SCUs based on the clauses' embeddings.

S^3 (Peyrard et al., 2017) is a learned evaluation metric that calculates features from the two summaries and predicts a score using a trained model. The features capture a variety of different types of similarity between the two texts, including different ROUGE scores, TF-IDF vector similarities, etc. The scoring model is trained on a dataset of collected human judgments of summary quality.

Finally, APES (Eyal et al., 2019) evaluates summaries using a learned reading comprehension model. It removes the named entities from the reference summaries and tries to predict the entity that was removed using the candidate summary. This metric is discussed more in §4.9.

Calculating Human-Level Performance. It is common practice in NLP to establish human performance on a task in order to contextualize the results from learned models. In the case of reference-based metrics, this amounts to evaluating the quality of the reference summaries. However, reference-based metrics use references to evaluate summaries, so evaluating a reference against itself would artificially inflate its score because it would essentially be a "perfect" summary. Therefore, care needs to be taken to ensure the evaluation is fair, which is done through a process called jackknifing.

Assume for the sake of this discussion that we have N reference summaries for the same input document. Each reference summary should be evaluated against the remaining $N-1$ summaries using the reference-based metric; the summary itself should not be used as a reference during evaluation. Thus, establishing a human baseline for reference-based metrics requires at least $N = 2$ references.

Then, to fairly compare the resulting human score to a learned system's score, the system's summary must be evaluated N different times, once against each possible subset of $N-1$ references, and then

averaged. If the reference-based metric handles multiple reference summaries by independently scoring the candidate against each one and then taking the average, the score does not change. However, some metrics may aggregate over multiple summaries a different way, for example by calculating the maximum score over the references, and the resulting score will change.

If jackknifing is not used and the reference is scored against itself, the score will be artificially inflated and not comparable to models' scores. If both model and human summaries are used to meta-evaluate a metric (discussed more in §2.3) and jackknifing is not performed, the metric's performance will be incorrectly estimated. This is a mistake which we have observed in the literature, and it invalidates the results of the meta-evaluation.

Reference-Free Metrics. In contrast to reference-based metrics, the class of reference-free metrics either implicitly or explicitly model what content in the input documents is salient instead of assuming the salient content is provided in a reference summary. Although this style of metric is less popular, there have been various approaches proposed in the literature.

SIMetrix is a toolkit which calculates various similarity scores between the summary and input documents using different distance measures described in Louis and Nenkova (2013). For example, the authors propose to calculate the Jensen Shannon divergence between the two texts using unigram frequency statistics and the cosine similarity between the texts' TF-IDF vector representations.

Other metrics, such as SUPERT (Gao et al., 2020), create psuedo-references from the input documents that are used to evaluate other generated summaries. Specifically, SUPERT identifies salient sentences that form the pseudo-references by finding sentences that have embeddings which are most similar to the documents' embeddings. This approach is similar to LexRank (Erkan and Radev, 2004), a well-known graph-based approach for ranking document sentences by their relative importances.

The BLANC (Vasilyev et al., 2020) metric leverages large-scale language models and scores the quality of a generated summary based on how well it improves the performance of a masked language model which operates on the input document.

Finally, two related metrics, SummaQA (Scialom et al., 2019) and QuestEval (Scialom et al., 2021) leverage QA to evaluate summaries, similarly to our proposal in Chapter 4. However, the main difference is that these metrics compare the generated summary to the input document via questions generated from one text and answered using the other. QuestEval also contains a learned question salience model that was trained to predict how likely a question is answered in the reference summary.

We discuss reference-free evaluation metrics at length in Chapter 5 and argue that they should not be used to measure progress on the task of summarization because they can be directly optimized during inference. This means it is possible to define algorithms which find the (near) optimal summaries under the metrics and thus the "perfect" output is already known, limiting their ability to effectively evaluate summaries.

2.3. Meta-Evaluating Metrics

Automatic evaluation metrics are often used as substitutes for manual evaluations of summary quality. Although they are much cheaper and faster to calculate than manual annotations, they typically do not evaluate the exact properties of summaries that we want to measure, which can only be done using human judges. As such, they are used as approximations of manual evaluations. Measuring the similarity between automatic metrics and human judgments is done via metric meta-evaluation (Dang and Owczarzak, 2008; Callison-Burch et al., 2006, 2008, 2010; Przybocki et al., 2008; Bojar et al., 2016).

Accurately meta-evaluating metrics is critically important for the development of summarization systems. Automatic metrics are the primary method that researchers use for evaluation during system development. Their scores are used to quickly determine whether a new model generates higher-quality summaries, and as a result, they end up influencing whether or not researchers pursue a specific idea. Therefore, it is important to know how likely an improvement in an automatic metric corresponds to an actual improvement in summary quality as judged by humans. If automatic metrics correlate poorly with human judgments—or we do not know how well they correlate at all—then it is not clear what conclusions we can reach about the quality of a summary based on

the value of an automatic metric.

2.3.1. Data Collection

Meta-evaluating metrics requires collecting a set of summaries that will be evaluated by human judges to establish a ground-truth quality score. These summaries are typically collected as follows.

First, a set of M input documents (or clusters, in a multi-document setting) is selected. The documents could be the entire test split of a dataset if it is small enough in size (Dang and Owczarzak, 2008, 2009) or subset of the test split sampled uniformly at random (Fabbri et al., 2021) or a stratified sample based on some difficulty heuristic (Bhandari et al., 2020).

Then, a set of N summarization systems is identified. This set could be all of the systems submitted to a summarization shared task (Dang and Owczarzak, 2008, 2009) or a collection of recently proposed models (Bhandari et al., 2020; Fabbri et al., 2021). Each of the N systems is used to generate a summary for all M inputs, resulting in $M \times N$ summaries.

Once the summaries have been collected, human annotators judge the quality of each of the summaries, establishing the ground-truth score that the automatic metrics will be evaluated against. The quality score that the annotators assign should be reflective of the aspect of the summary which is going to be evaluated through automatic metrics, such as the linguistic quality or faithfulness (see §2.2.1 for more details). This human annotation procedure is denoted by $\mathcal{Z}$. The result of this judgment procedure is a matrix of ground-truth scores, called Z, in which z_j^i is the human-judged score of the summary produced by the ith summarization system on the jth input document.

Then, the automatic metric, denoted $\mathcal{X}$, is used to calculate a score for all $M \times N$ summaries, resulting in matrix X in which x_j^i is the automatic metric score of the summary produced by the ith summarization system on the jth input document.

Once matrices Z and X have been collected, they can be used to estimate the similarity of the human-judgment procedure $\mathcal{Z}$ and metric $\mathcal{X}$ through different correlation calculations, discussed in §2.3.2. Next, we describe several datasets which are commonly used to meta-evaluate summarization metrics. Dataset statistics are included in Fig. 2.

Dataset	Number of Clusters	Docs per Cluster	Avg. Doc Length (Tokens)	Number of References	Avg. Reference Length (Tokens)	Number of Systems	Avg. Candidate Length (Tokens)
TAC 2008 (Dang and Owczarzak, 2008)	48	10	512	4	96	58	96
TAC 2009 (Dang and Owczarzak, 2009)	44	10	552	4	98	55	95
REALSumm (Bhandari et al., 2020)	100	1	740	1	51	25	72
SummEval (Fabbri et al., 2021)	100	1	359	1	44	16	63

Table 2: Basic statistics of the datasets commonly used to meta-evaluate summarization metrics.

TAC 2008 & 2009. During the 2000s, the Document Understanding Conference (DUC) and the Text Analysis Conference (TAC) ran highly influential shared tasks on summary generation and summarization evaluation. These conferences collected a large number of high-quality summarization datasets and expert judgments of various aspects of summary quality, largely standardizing task definitions and the evaluation and meta-evaluation methodologies for summarization.

Two datasets which emerged as the benchmarks for meta-evaluating metrics are TAC 2008 (Dang and Owczarzak, 2008) and 2009 (Dang and Owczarzak, 2009). During these years, the shared tasks included topic-focused multi-document summarization, in which the goal is to write a 100 word summary of a cluster of documents with respect to a specific information need expressed by a topic statement. An example information need is the following:

> *Describe developments in the production and launch of the Airbus A380.*

There were 48 and 44 input clusters with 10 input documents per cluster in 2008 and 2009, respectively. The documents were manually selected to be about a coherent topic from the news domain. Each document cluster also contains 4 expert-written reference summaries.

There were 58 and 55 systems which were submitted in 2008 and 2009, respectively, which had their summaries evaluated by expert annotators across different dimensions of summary quality, including an *overall responsiveness* score, which evaluates both the content and readability of the summaries, as well as the Pyramid Method.

REALSumm & SummEval. The CNN/DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016) is a large-scale, generic single-document summarization task that is largely considered a benchmark summarization dataset for current summarization research. In contrast to the datasets

popularized by DUC and TAC, the reference summaries are about one document instead of a cluster of documents, are notably shorter at ≈ 50 tokens compared to ≈ 100 for TAC, and tend to be far more extractive (Grusky et al., 2018).

Further, the types of systems which are trained and evaluated on the CNN/DailyMail dataset are rather different from those evaluated during DUC and TAC. Recent systems are trained on orders of magnitude more data than systems from the 2000s. They are often abstractive models instead of extractive models, and they produce summaries which are higher quality in the sense that they are more similar to the reference summaries (Peyrard, 2019). As such, it is not clear to what extent conclusions made from meta-evaluating metrics on the TAC datasets will still hold for datasets and models which are popular today.

To that end, researchers collected new datasets for meta-evaluating metrics on the CNN/DailyMail dataset with recently published summarization systems. REALSumm (Bhandari et al., 2020) is a collection summaries from 25 systems on 100 input documents that were evaluated using the Lightweight Pyramid (see §2.2.1). The input documents were selected to capture both easy- and hard-to-summarize documents. Documents were grouped into difficulty buckets based on the average reference-based metric scores of the 25 systems, and a stratified sample was performed using those buckets.

The SummEval (Fabbri et al., 2021) dataset contains expert judgments of summary quality for 16 systems on 100 input documents. The input documents were selected by a uniform sample of the CNN/DailyMail dataset. Each of the 1600 summaries were annotated for four different dimensions of summary quality:

1. **Coherence**: The collective quality of how well the summary is structured and organized
2. **Consistency**: The extent to which the summary contains information which is factually supported by the input document
3. **Fluency**: The grammaticality quality of the sentences

4. **Relevance**: How well the summary selects important content from the source document.

Each summary was was assigned a score on a Likert scale from 1-5 by three different expert judges for each of the four dimensions.

2.3.2. Correlation Levels

Within summarization, there are two standard correlations that are used to estimate the correlation between metric $\mathcal{X}$ and human judgments $\mathcal{Z}$, called the system- and summary-level correlations. Both are functions of the matrices X and Z, described next.

System-Level Correlation. The system-level correlation quantifies the extent to which an automatic metric $\mathcal{X}$ and human judgments $\mathcal{Z}$ agree on the quality of a *system*. It is defined as the correlation between the metric's score and the human judged score for each of the N systems.

Each system is assigned a score by the metric and human judgments by aggregating over the M individual summary scores for that system. Typically the scores are averaged, as in Equations 2.10 and 2.11, although other aggregation functions can be used.

$$\bar{x}_i = \frac{1}{M} \sum_{j=1}^{M} x_i^j \tag{2.10}$$

$$\bar{z}_i = \frac{1}{M} \sum_{j=1}^{M} z_i^j \tag{2.11}$$

Then, the system-level correlation is calculated as the correlation between the metric and human system scores:

$$r_{\text{SYS}}(X, Z) = \textsc{Corr}\left(\{(\bar{x}_1, \bar{z}_1), \ldots, (\bar{x}_N, \bar{z}_N)\}\right) \tag{2.12}$$

in which the $\textsc{Corr}(\cdot)$ function calculates the correlation between the paired observations, discussed more in §2.3.3.

Geometrically, the system-level correlation is equivalent to calculating a per-row score for X and Z by aggregating over the columns, then taking a correlation between the per-row scores (see Fig. 1).

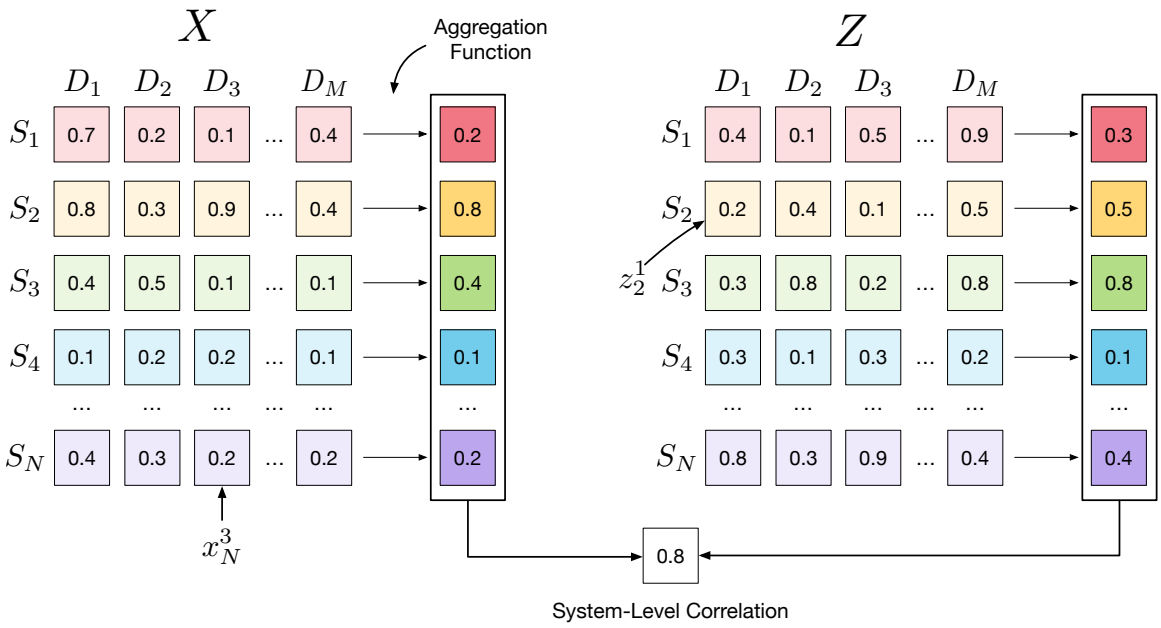

Figure 1: The system-level correlation calculates the correlation between the metric and ground scores for each system, typically calculated by aggregating over the scores per-input.

Summary-Level Correlation. In contrast to the system-level correlation, the summary-level correlation measures how similarly a metric and human judgments score individual *summaries* for the same input document. It is defined as the average correlation between the N summaries per-input, averaged over all M inputs:

$$r^j = \textsc{Corr}\left(\left\{\left(x_1^j, z_1^j\right), \ldots, \left(x_N^j, z_N^j\right)\right\}\right) \tag{2.13}$$

$$r_{\textsc{sum}}(X, Z) = \frac{1}{M}\sum_j^M r^j \tag{2.14}$$

Geometrically, this is equivalent to taking average correlation between the columns of X and Z (see Fig. 2.

Other Similarity Functions. There are other ways that X and Z could be used to calculate a correlation. For example, an average per-row correlation could be calculated, which would measure how similarly the metric and human judgment score summaries per-system, or by calculating the correlation between the two matrices after they have been flattened into vectors, which would measure how similarly they score individual summaries (not necessarily grouped by input document).

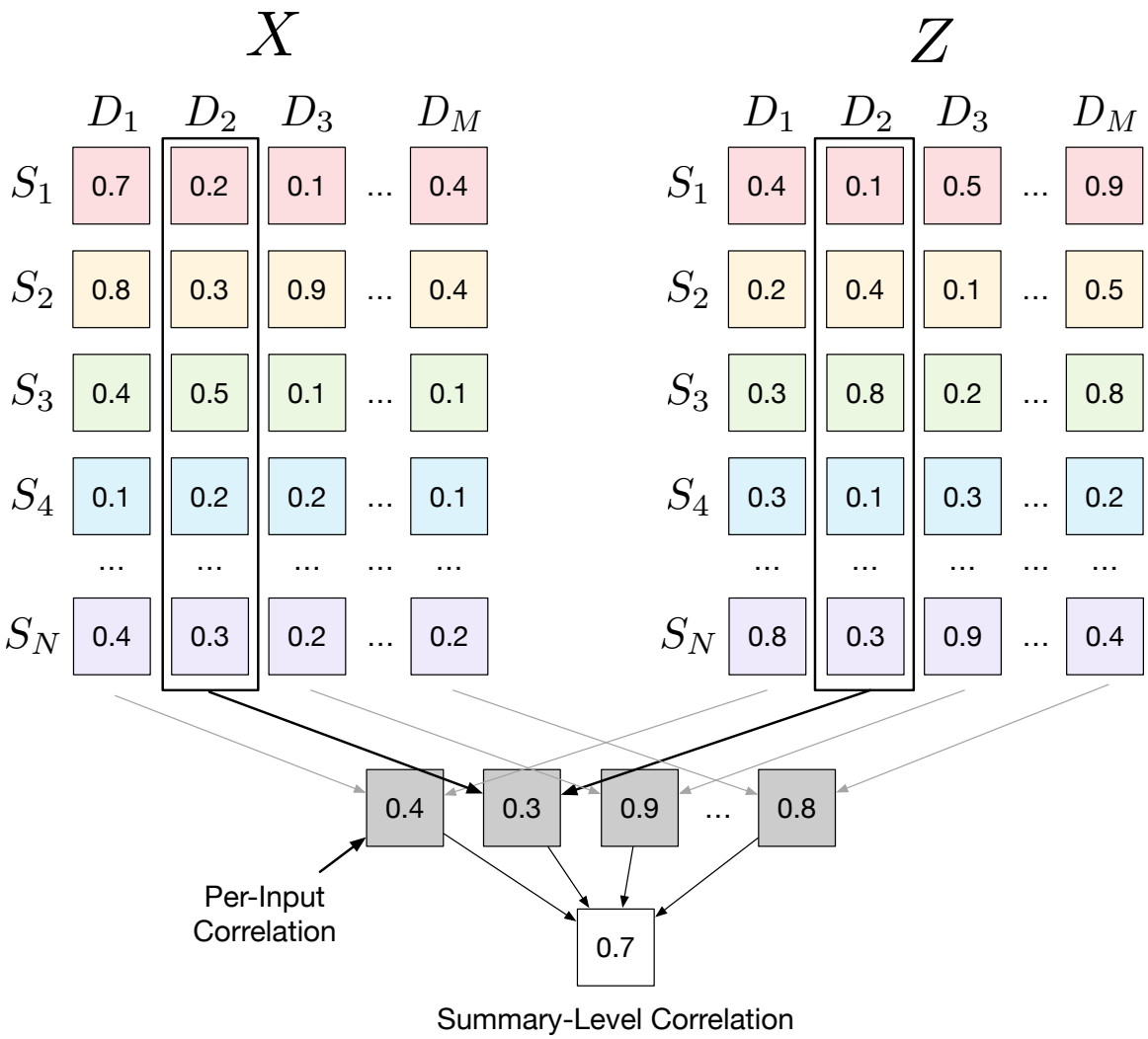

Figure 2: The summary-level correlation calculates the correlation between the metric and ground scores per input, then averages the correlations to calculate the final score.

While the system-level correlation has been used since at least TAC 2009 (Dang and Owczarzak, 2009), to the best of my knowledge, the definition of the summary-level correlation above was standardized in Peyrard et al. (2017) and subsequently used by later works. Variants of the summary-level correlation have been used before, such as reporting the percent of the M summary-level correlations which are significant (Louis and Nenkova, 2013).

Metrics have also been evaluated with pairwise accuracy (also called "discriminative power"), including in TAC (Dang and Owczarzak, 2009; Owczarzak et al., 2012). Pairwise accuracy is defined

as the proportion of pairs of systems which are ranked the same according to the metric and human judgments subject to the difference between the human scores being statistically significant. This is equivalent to calculating Kendall's τ at the system-level with the added requirement of statistical significance.

Discussion. The system-level correlation is arguably a more important quantity than the summary-level correlation or other ways to calculate the similarity between X and Z. Evaluation metrics are most frequently used to reach conclusions about the qualities of systems, for instance, by arguing that one system generates better summaries than another because it has a higher metric score. The system-level correlation most directly evaluates metrics for this use case. In contrast, metrics are rarely used to compare individual summaries of the same input document, which is evaluated by the summary-level correlation.

In general, the actual values of summary-level correlations tend to be much lower than the system-level correlations. It is far more difficult to accurately evaluate an individual summary, both by humans and automatic metrics, than it is to accurately estimate the quality of a system, which is an aggregate over dozens of summaries. Therefore, the scores for individual summaries have higher variance than those for systems, resulting in less agreement between human judgments and metrics.

2.3.3. Correlation Coefficients

There are three correlation coefficients that are typically used in the system- and summary-level correlations calculations. All three coefficients measure the strength of the relationship between the metric and human-judged scores for systems or summaries, but they make different assumptions about what type of relationship they assume exists. As such, their values should be interpreted accordingly.

For the sake of this discussion, we will assume that we are calculating the correlation between two variables, A and B, using a sample of paired observations $\{(a_1, b_1), \ldots, (a_k, b_k)\}$.

Pearson's Correlation Coefficient. Pearson's correlation coefficient measures the extent to which a linear relationship exists between A and B. The sample correlation is defined as:

$$r = \frac{\sum_{i=1}^{k}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{k}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{k}(b_i - \bar{b})^2}} \tag{2.15}$$

where $\bar{a}$ and $\bar{b}$ are the average a_i and b_i values.

The value the Pearson correlation ranges from -1 to $+1$. A value of $r = 1$ means the paired observations have a perfect linear relationship, and $r = -1$ means they have a perfect negative linear relationship. Importantly, a low value of $|r|$ does not imply that no relationship exists between the paired observations, only that the strength of the *linear* relationship is weak.

Spearman's Rank Correlation Coefficient. Closely related to Pearson's correlation coefficient, Spearman's rank correlation coefficient measures the extent to which a monotonic relationship exists between the paired observations. It is calculated as the Pearson correlation between the ranks of the observations. Specifically, let r_{a_i} be the ordinal rank of the ith observation when the pairs are sorted by their A scores and likewise for r_{b_i}. Then, Spearman's correlation is defined as the Pearson correlation of $\{(r_{a_1}, r_{b_1}), \ldots, (r_{a_k}, r_{b_k})\}$.

By converting the observations to their corresponding ranks, Spearman relaxes Pearson's assumption that the observations have a linear relationship and measures the strength of a monotonic relationship between the two sets of scores. Thus, the value of Spearman's correlation coefficient has the same interpretation as Pearson's r except the relationship is monontonic instead of linear.

Kendall's Rank Correlation Coefficient. Finally, Kendall's rank correlation coefficient measures how similarly A and B rank the observations irrespective of any specific parametric relationship between the scores. For every possible pair $\{(i, j) : \forall i, j = 1, \ldots, N \text{ and } i < j\}$, Kendall's correlation coefficient, denoted τ, counts how often A and B agree on whether observation i is ranked higher or lower than observation j. A pair (i, j) is called concordant if A and B agree on the ranking. That is, $a_i > a_j$ and $b_i > b_j$ or $a_i < a_j$ and $b_i < b_j$. If A and B disagree on the ranking, the pair is called discordant.

Then, Kendall's τ is defined as

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) \cdot (P + Q + U)}} \tag{2.16}$$

where P and Q are the number of concordant and discordant pairs, respectively, and T and U are the number of ties only in A or B, respectively. Similarly to Pearon's and Spearman's coefficients, $\tau \in [-1, 1]$. A value of $\tau = 1$ means the observations are ranked identically by A and B, whereas $\tau = -1$ means the observations are ranked in the exact opposite order by A and B.

2.4. Evaluation in Machine Translation

The evaluation challenges discussed in the previous sections of this chapter largely apply to every text generation task; because there is no single-ground truth, evaluation metrics which are more sophisticated than exact match need to developed, and those metrics themselves must be meta-evaluated against human judgments of text quality. In order to contextualize the work in this thesis within the broader field of natural language processing, we briefly discuss the evaluation methodology for, arguably, the most well-studied text generation task, machine translation (MT).

Task Definition. At the task-level, MT is more well-defined than summarization. Although there are nuances and variations, the goal of MT is to generate a translation for some input source text that preserves the meaning of the source text, in contrast to summarization in which some of the meaning of the input text is lost by design. The ill-definition of summarization comes from the fact that what information to include in the summary is ambiguous, whereas a translation should contain the same information as the input. However, like summarization, MT is not completely defined as various aspects could impact the perceived translation quality, such as who is the audience and what is the purpose for generating the translation, which are not always clear.

MT systems typically operate at the sentence level, translating one sentence at a time, whereas summarization models most often process at least one input document, which is several paragraphs long, and outputs several sentences as the summary. This has implications for the difficulty of soliciting human evaluations of their outputs and what challenges automatic metrics for these tasks need to solve.

Workshop on Machine Translation. The evaluation methodology for MT is largely standardized by the Workshop on Machine Translation (WMT), which is an annual conference that has been running since 2006. Similarly to DUC and TAC for summarization, WMT runs various shared tasks each year related to developing and benchmarking translation systems as well as automatic evaluation metrics. Over the years, they have explored different methods of manually evaluating translations and established the meta-evaluation procedures for automatic metrics. Their methodologies are discussed below.

Manual Evaluation. Like summarization, the gold standard evaluation methodology for MT uses human annotators to judge the quality of a translation. Historically, WMT had judges assign a ranking to a set of translations output by various systems (Callison-Burch et al., 2010). While this provided relative rankings, it requires a large number of judgments, and converting the relative rankings into an absolute ranking of systems is not straightforward (Bojar et al., 2011; Lopez, 2012; Sakaguchi et al., 2014). In more recent years WMT has pivoted toward direct assessments of translation quality in order to balance the cost and reliability of the solicited judgments (Akhbardeh et al., 2021). Although the length of text being evaluated in MT is often much shorter than for summarization, human evaluation is still very challenging and subject to low inter-annotator agreements (Bojar et al., 2016).

Automatic Evaluation. In order to approximate human judgments, MT researchers have proposed various automatic evaluation metrics that are most typically reference-based. Many of the MT metrics which have been proposed or widely used are similar to those in summarization. For instance, the de-facto metric, BLEU (Papineni et al., 2002), compares the reference and candidate translations based on lexical overlap similarly to ROUGE. BERTScore (discussed in §2.2.2) was actually developed with MT in mind.

One key difference between MT and summarization is the success of metrics that have been trained to predict human judgments of translation quality, such as BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020) in MT. Over the years, WMT has collected tens of thousands of sentence-level quality judgments of candidate translations, and these metrics used trained models to predict

the candidate translation quality given the source text and reference translation. Although such an approach would likely be successful for summarization (and has been explored before; Peyrard et al., 2017), the data necessary to directly build a high-quality metric likely does not exist as the scale required. In total, the number of summaries which have been judged by reliable experts is orders of magnitude smaller than the equivalent number of translations. Further, due to the fact that summaries are frequently longer than a single sentence, the number of judged summaries required for training a high-quality evaluation metric is likely larger than for MT, making a learned metric even more challenging.

The evaluation metric which we propose in Chapter 4 leverages question-answering to compare the semantic content of two summaries. In theory, the same evaluation methodology could be directly applied to evaluate MT as it can effectively be treated as a black-box method for comparing two texts, and previous work has explored this direction for manual evaluations (Callison-Burch, 2009). However in practice, it is unknown whether such an approach may be immediately successful; the question answering model which we employ is trained on SQuAD 2.0 (Rajpurkar et al., 2018), which contains instances in which questions need to be answered against paragraphs of text. It is unclear whether a model trained on SQuAD 2.0 would generalize well to single sentence contexts, which is the most typical case for summarization. We suspect that the most direct application of our proposed metric would be in evaluating paragraph- or document-level MT, which will look more similar to the text that the question answering model was trained on.

Metric Meta-Evaluation. MT uses a similar metric meta-evaluation methodology to that of summarization (Callison-Burch et al., 2006, 2008, 2010; Przybocki et al., 2008). The quality of a metric is quantified by calculating the correlation between its scores and human judgments on a large set of summaries.

Both MT and summarization study how similar a metric and humans score systems via system-level correlations, but the two tasks differ in how they compute an input-level correlation. Summarization calculates an average per-document correlation via the summary-level correlation, whereas MT reports a "segment-level" correlation, which directly computes the correlation between the met-

ric and human scores for all of the translations without any sort of grouping. Neither method is necessarily better than the other; they simply have different interpretations. The summary-level correlation quantifies how similarly the scores are for summaries of the same document, whereas the segment-level does so for a translation for any source text.

While statistically testing differences between metric correlations in summarization is largely not performed, it is standard in MT to use Williams' test (Williams, 1959). In Chapter 7, we propose methods for calculating confidence intervals for correlations and statistically testing their differences within the context of summarization. We include a detailed discussion of the performance of our proposed methods and Williams' test for MT and summarization §7.5. Our methods can be directly applied to MT, and they were, in fact, adopted during the most recent iteration of the WMT metrics shared task (Freitag et al., 2021).

Finally, in Chapter 8, we point out that the way that the system-level correlation is calculated for summarization does not necessarily align with how the metrics are used in practice. The changes we propose to address this discrepancy could also be directly applied to MT, and it is certainly an analysis worth doing. A related study for MT was performed by Mathur et al. (2020a) who point out that outlier systems disproportionally affect the metrics' correlations and are thus likely not good estimates of the true correlations. Our proposed change to the definition of system-level correlations which calculates the correlation on pairs of systems which have similar automatic metric scores reaches a similar conclusion.

In summary, the tasks of MT and summarization are very related despite seeming very different at the surface. Both fields of NLP face similar challenges for evaluation and meta-evaluation, and advancements and insights from one task are likely to be useful and hold true for the other.

CHAPTER 3 : Understanding & Interpreting Evaluation Metrics

As discussed in Chapter 2, reference-based metrics are the primary method used by researchers to automatically evaluate summaries. They implicitly assume all of the salient document content is contained in the reference summary and calculate a score for a candidate summary based on its similarity to the reference. Therefore, reference-based evaluation is reduced the problem of deciding how similar two texts are.

However, there are many ways in which two texts can be similar. For instance, they could discuss different aspects of the same entities or concepts and be topically similar, or they could further express similar information about those common topics (see Fig. 3 for an example). It is unclear what dimension of similarity is captured by existing evaluation metrics.

In this Chapter, we explore the fundamental question of what type of similarity between reference and candidate summaries should be measured by reference-based evaluation metrics. We argue that, while measures of topic similarity are certainly useful, comparing the two summaries based on the information they express through predicate-argument structures is a more valuable dimension of similarity to measure for summarization. Then, we analyze two key evaluation metrics, ROUGE and BERTScore, to understand the extent to which they can be interpreted as measuring information similarity. We find that these metrics largely cannot be viewed as evaluating the information in a summary, but rather as measures of whether the candidate and reference summaries discuss the same topics. This result highlights the shortcomings of these metrics and motivates the need for evaluation metrics that better evaluate the information in a summary.

This Chapter is based on work previously described in Deutsch and Roth (2021).

3.1. Introduction

When human judges are asked to evaluate the content of a summary, either through a direct assessment or by comparing it to a reference, it is likely the case that they are evaluating the information expressed by the summary. While other aspects certainly contribute to the overall quality, such as whether the summary discusses salient topics, determining whether the summary contains the right

Figure 3: Both candidate summaries are similar to the reference, but along different dimensions: Candidate 1 contains some of the same information, whereas candidate 2's information is different, but it at least discusses the correct topic. The goal of this Chapter is to understand if summarization evaluation metrics' scores should be interpreted as measures of information overlap or, less desirably, topic similarity.

facts is a more meaningful way to measure its quality.

The "information" of a summary is not a fully well-defined concept. However, our definition will be that a summary's information is what can be expressed through predicate-argument relations in the text.

Our analysis of the extent to which ROUGE and BERTScore can be viewed as measures of information quality relies on casting the two metrics into a unified framework in which the similarity of two summaries is calculated based on an alignment between the summaries' tokens (§3.3). This alignment-based view of the metrics enables performing two different analyses of how well they measure the information overlap between the candidate and reference summaries.

The first analysis demonstrates that only a small proportion of the metrics' token alignments are between phrases which contain identical information according to expert annotators (§3.4). The second reveals that token alignments which represent common information are vastly outnumbered by those which represent the summaries discussing the same topic (§3.5). Overall, both analyses support the conclusion that ROUGE and BERTScore largely do not measure information overlap.

Additionally, we briefly explore whether or not nine other evaluation metrics successfully measure information quality (§3.6). By demonstrating that nearly all of the metrics correlate much more strongly to ROUGE than to the summaries' Pyramid Scores, we argue the metrics are likely to

measure information overlap no better than ROUGE does.

While the summarization community has been aware, informally, of the shortcomings of the current evaluation metrics, this study provides experimental evidence beyond correlations to support these intuitions. The contributions of this Chapter include (1) analyses which reveal that ROUGE and BERTScore largely do not measure the information overlap between two summaries and (2) evidence that many other evaluation metrics likely suffer from the problem.

3.2. Motivation: Understanding Evaluation Metrics

As discussed in Section 2.2.2, reference-based evaluation metrics assume that human-written reference summaries have gold-standard content and score a candidate summary based on its similarity to the reference. An ideal evaluation metric that measures the content quality of a summary should score the quality of its information, or what is expressed by its predicate-argument relations. For reference-based metrics, this means that the comparison between the two summaries should measure how much information they have in common.

Metrics such as ROUGE and BERTScore calculate the similarity of two summaries either by how much lexical overlap they have or how similar the summaries' contextual word embeddings are (discussed in more detail in §2.2.2). Although we understand how their scores are calculated, it is not clear how the scores should be interpreted: Are they representative of how much information the two summaries have in common, or do they describe how similar the summaries are on some other less desirable dimension, such as whether they discuss the same topics? The goal of this Chapter is to answer this question.

Knowing the answer is critically important. The goal of summarization is to produce summaries which contain the "correct" information (among other desiderata). Automatic metrics are the most frequent method that researchers use to argue that one summarization model generates better summaries than another. If our evaluation metrics are not aligned with our research goals — or if we do not understand what they measure at all — then we do not know whether we are making progress as a community.

3.3. A Common Framework

The focus of our analyses will be primarily on two evaluation metrics, ROUGE and BERTScore. Although on the surface these two metrics appear to compare two summaries very differently, here we demonstrate how they can both be viewed as calculating a score based on a weighted alignment between the summaries' tokens. This common framework enables us to reason about how to interpret their scores.

Let $X = x_1, \dots x_m$ and $Y = y_1, \dots y_n$ be the tokens of the candidate and reference summaries. ROUGE-1 counts the number of unigrams that are in common between the two summaries:[5]

$$M = \sum_{u \in \text{unigrams}(X)} \min(c_X(u), c_Y(u)) \tag{3.1}$$

where $c_T(u)$ counts the number of times u appears in the summary T and the summand is over unique unigrams. Then, precision and recall are calculated by dividing M by m and n, respectively. When multiple references are available, the precision and recall scores are micro-averaged.

A weighted alignment A is a set of token alignments (i, j, w) that map token x_i to y_j with weight $w \in (0, 1.0]$. The weight of an alignment, denoted $W(A)$, is the sum of the weights of the individual token alignments. ROUGE can be viewed as creating an alignment by pairing $\min(c_X(u), c_S(u))$ occurrences of unigram u in X and Y with weight 1.0 for all unigrams. It additionally imposes a constraint that each token can be aligned to at most one other token. Since a unigram may appear multiple times in a summary, the alignment may not be unique, however, its weight will equal M.

BERTScore calculates a similarity score between two pieces of text based on the pairwise cosine similarities of their tokens' BERT embeddings. Let B_{ij} be the similarity score between the embeddings for x_i and y_j. To calculate recall, BERTScore first aligns every reference token to its most-similar candidate summary token (Eq. 3.2). Then, the sum of the corresponding similarities is

[5] Our analysis focuses on the unigram variant of ROUGE, called ROUGE-1. We refer to it, where clear, as ROUGE for simplicity.

normalized by the number of reference tokens to get the recall score (Eq. 3.3).

$$A_R = \{(i, j, B_{ij}) : \forall j, i = \arg\max_k B_{k,j}\} \tag{3.2}$$

$$\text{BERTScore}_{\text{Recall}} = W(A_R)/n \tag{3.3}$$

A similar procedure is followed to calculate precision, but instead, every candidate summary token is aligned to its most-similar reference token, and the sum of the similarities is normalized by the number of summary tokens. When multiple references are available, the precision and recall scores are defined to be the maximum respective values across references. Because the alignment maps between tokens at specific positions within the summaries, BERTScore's alignment is unique, unlike for ROUGE.

By formulating ROUGE and BERTScore in a framework based on token alignments, we can reason about their behaviors by examining the tokens they align in two different analyses, as described next.

3.4. SCU-Based Analysis

The first analysis compares the two metrics' token alignments to annotations derived from the Pyramid Method (Nenkova and Passonneau, 2004).

As discussed in §2.2.1, the Pyramid Method is a technique to manually evaluate the content of a candidate summary by comparing it to a set of reference summaries. The method uses a domain-expert annotator to exhaustively identify atomic units of meaning in the summaries, known as summary content units (SCUs), and mark their occurrences in the reference and candidate summaries. Two phrases marked with the same SCU are considered to express the same information.[6] Since the Pyramid Method annotation is exhaustive, we can assume that any two phrases in the reference and candidate summaries that are not marked with the same SCU do not have the same meaning.

These annotated phrases can be used to reason about ROUGE and BERTScore: If a large proportion

[6]While there is no explicit definition of exactly what type of meaning an SCU represents, we empirically find that it aligns well with our own definition of information, equivalent to what is represented by the predicate-argument structures of the text.

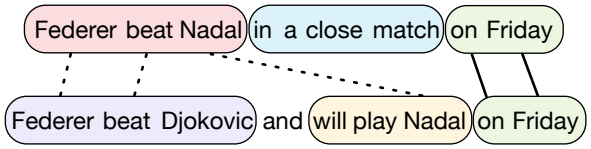

Figure 4: An example token alignment created by ROUGE. Each color represents a summary content unit (SCU) that marks informational content. Only $2/5$ of the token alignments (the solid edges) can be explained by matches between phrases that express the same information (the green phrases).

of their token alignments is between phrases that express the same information, then their scores can potentially be interpreted as representing the summaries' information overlap. Otherwise, it is evidence that they do not compare summaries based on information.

For this analysis, we use the summaries and Pyramid annotations from the TAC 2008 and 2009 English multi-document summarization datasets (Dang and Owczarzak, 2008, 2009). These datasets are discussed in more detail in §2.3, but the relevant information about them is included here. TAC 2008 has 48 document clusters and 58 system summaries, and TAC 2009 has 44 clusters and summaries from 55 systems. All clusters have around 10 documents each and 4 reference summaries, and every summary has been been annotated with SCUs.

For each of the system summaries, we calculate the proportion of the total alignment weight that can be explained by matches between identical SCUs, as defined in Eqs. 3.4 and 3.5:

$$A_{\text{SCU}} = \{(i, j, w) : \ (i, j, w) \in A, \ \text{SCU}(i) \cap \text{SCU}(j) \neq \emptyset\} \tag{3.4}$$

$$\text{Prop}_{\text{SCU}} = \frac{W(A_{\text{SCU}})}{W(A)} \tag{3.5}$$

where $\text{SCU}(i)$ returns the set of SCUs that are annotated for the token at index i. Fig. 4 has an example of this calculation. Since ROUGE does not use a unique alignment, we choose the alignment which maximizes Eq. 3.5, thus calculating an upper-bound.

The distribution of the proportion of ROUGE and BERTScore explained by SCU matches is presented in Figure 5. We find that, on average, only 25% and 15% of these metrics scores comes from

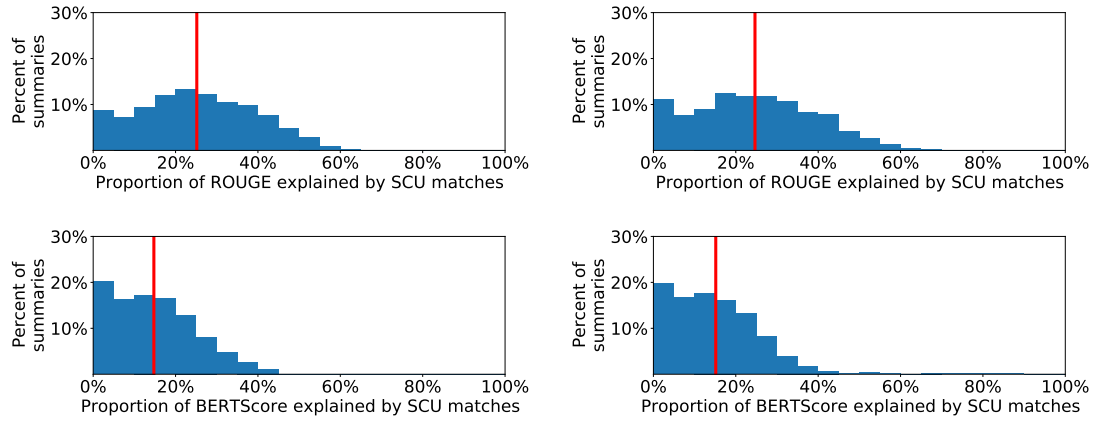

Figure 5: The distribution of the proportion of ROUGE (top row) and BERTScore (bottom row) on TAC 2008 (left column) and TAC 2009 (right column) that can be explained by tokens matches that are labeled with the same SCU (Eq. 3.5). The averages, around 25% and 15% on both datasets (in red), indicate that only a small amount of their scores is between phrases that express the same information.

matches between tokens marked with the same SCUs on both datasets. Since only a relatively small fraction of the overall metric scores comes from phrases with the same information, this suggests that ROUGE and BERTScore's values cannot be interpreted as a measure of information overlap.

In the next Section, we perform a second analysis which supports this conclusion and also better describes how the metrics' scores should be interpreted instead.

3.5. Category-Based Analysis

The second analysis of ROUGE and BERTScore focuses on grouping token alignments into categories (§3.5.1), then using those categories to reason about how much of the metrics' scores is explained by information or topic matches (§3.5.2).

3.5.1. Token Alignment Categorization

We define a category to be a function C that selects the subset of summary token indices for which that category applies. For example, a "noun" category would select only the token indices that correspond to nouns. $C(S)$ denotes the application of a category to summary S.

Each category is used to filter an alignment A used by ROUGE or BERTScore to a category-specific

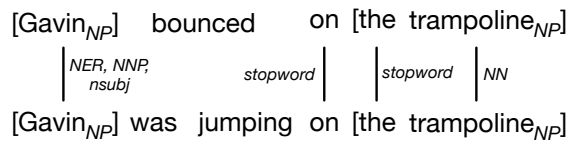

Figure 6: Every token alignment used by ROUGE or BERTScore is assigned to one or more interpretable categories (defined in §3.5). This allows us to calculate, for this example, that matches between named-entities contribute $1/4$ to the overall score, stopwords $2/4$, and noun phrases $3/4$ (assuming alignment weights of 1.0).

alignment between tokens which belong to that category only, denoted A_C:

$$A_C = \{(i, j, w) : \ (i, j, w) \in A, \ i \in C(R), \ j \in C(S)\} \tag{3.6}$$

For the "noun" category, A_C would be the subset of token alignments between nouns in R and S.

Then, the *contribution* of C is defined as the ratio between A_C and A:

$$\text{Contribution}_C = \frac{W(A_C)}{W(A)} \tag{3.7}$$

The contribution of C can be interpreted as the proportion of ROUGE or BERTScore that can be explained by matches between tokens in category C (see Fig. 6 for an example).

Higher-Order Categories. Although our analysis only uses unigram alignments, it is desirable to reason about groups of tokens. This would enable calculating how much of the metrics' scores can be explained by matches between (subject, verb, object) tuples, for instance.

We extend the definition of a category to select a set of *tuples* of indices. Then A_C selects only the token alignments in A that are included in an aligned tuple selected by C. Two tuples $(i_1, \ldots, i_k)$ and $(j_1, \ldots, j_k)$ are said to be aligned if indices i_ℓ and j_ℓ are aligned for $\ell = 1, \ldots, k$. Fig. 7 has an example tuple-based matching.

3.5.2. Category-Based Analysis

Next, we define a set of categories in which each category represents either information or topic matches, then reason about how much of the metrics' scores can be explained by information or

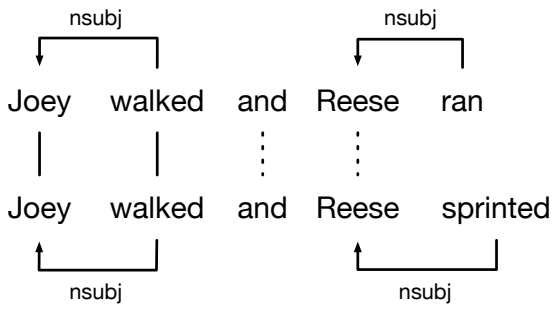

Figure 7: The VB+NSUBJ category selects tuples of verbs and their corresponding NSUBJ dependents in the dependency tree. In this example, $2/4$ of the alignment (the solid lines) can be explained by matches between such tuples. The dashed lines cannot: The "and" alignment is not part of any tuple; Since "ran" and "sprinted" are not aligned, their corresponding tuples are not considered to be aligned, so the "Reese" match does not count toward the total.

topic similarities based on the corresponding category contributions.

We define the following categories:

1. **Stopwords**: One category to select matches between stopwords, denoted STOPWORDS.

2. **Parts-of-Speech**: Six categories, one for selecting alignments between each type of the following part-of-speech tags: common nouns (NN), proper nouns (NNP), verbs (VB), adjectives (ADJ), adverbs (ADV), and numerals (NUM).

3. **Named-Entity**: One category for all named-entities, denoted NER. This category only selects alignments between tokens if they are the same type of named-entity (person, location, or organization).

4. **NP Chunks**: One category to select matches between tokens that are part of noun phrases, denoted NP-CHUNKS.

5. **Dependency**: Three categories that select matches between tokens with the same dependency tree arc label for ROOT, NSUBJ, and DOBJ labels.

6. **Dependency Tuples**: Three categories that match higher-order tuples based on the dependency tree. Each category selects a tuple containing a verb and either its subject child

	TAC'08		CNN/DM	
Category	**ROUGE**	**BERTScore**	**ROUGE**	**BERTScore**
NP-CHUNKS	58.7	46.1	53.6	43.0
STOPWORDS	54.6	32.4	48.4	28.7
NN	17.9	13.7	31.8	24.9
NNP	14.9	11.3	0.3	0.2
NER	13.5	8.5	0.1	0.1
VB	9.0	9.3	14.1	10.6
ADJ	4.1	2.6	6.2	4.0
NSUBJ	3.9	2.2	6.3	4.1
DOBJ	2.0	1.4	2.8	1.7
NUM	1.5	1.7	2.5	1.9
VB+DOBJ	1.3	0.4	3.4	1.0
ROOT	1.1	1.5	3.3	2.5
VB+NSUBJ	1.0	0.5	3.8	2.4
ADV	0.6	0.4	1.6	0.8
VB+NSUBJ+DOBJ	0.3	0.1	1.5	0.5

Table 3: The contributions (Eq. 3.7) of every category to ROUGE and BERTScore on TAC 2008 and CNN/DailyMail indicate the metrics are largely matching nouns and stopwords rather than tuples which express information (e.g., VB+NSUBJ+DOBJ). The contributions do not sum to 100% because more than one category can explain the same token alignment. The NNP and NER for CNN/DailyMail are significantly lower because the candidate summaries were all lower-cased.

(NSUBJ), object child (DOBJ), or both. These categories are denoted VB+NSUBJ, VB+DOBJ, and VB+NSUBJ+DOBJ. They are representative of information expressed as predicate-argument relations (e.g., {subject, verb, object} tuples).

We consider 2 through 5 to be keywords that represent the topics discussed in the summaries, whereas 6 describes tuples which express the summaries' information as predicate-argument relations.

The contributions of each category on the TAC 2008 summaries as well as the summaries produced by baseline (See et al., 2017) and state-of-the-art (Liu and Lapata, 2019) abstractive models on the CNN/DailyMail dataset (Nallapati et al., 2016) are presented in Table 3. All text processing, including POS/NER tagging and parsing, are done with spaCy (Honnibal et al., 2020).

The results across datasets and evaluation metrics largely follow the same trend: Noun- and stopword-

	TAC'08		CNN/DM	
Content Type	**ROUGE**	**BERTScore**	**ROUGE**	**BERTScore**
Topic	70.6	57.9	75.0	59.2
Information	2.2	0.9	6.7	3.2
Stopwords	54.6	32.4	48.4	28.7

Table 4: The contributions of different categories of token matches when grouped by whether they represent topics, information, or stopwords. Clearly, the information categories explain only a small proportion of the overall metrics scores on TAC'08 and CNN/DailyMail.

based matches explain the vast majority of the token alignments used by both ROUGE and BERTScore, whereas the dependency tuple categories explain very little of the overall scores.[7] For instance, on TAC 2008, noun phrase and stopword matches contribute 58.7% and 54.6% to ROUGE, whereas the dependency tuple with the largest contribution, VB+DOBJ only contributes 1.3%.

When the specific categories are grouped by content type in Table 4, it becomes even more apparent that topic and stopwords matches explain most of ROUGE and BERTScore.[8] We find that topic, stopword, and information matches explain 70.6%, 54.6%, and 2.2% of ROUGE on TAC'08.

The low contribution of information-based categories toward each metric is further evidence that neither metric strongly captures the information overlap between summaries, supporting the results found in §3.4. Rather, ROUGE and BERTScore are better measures of how much the two summaries discuss the same topics.

3.6. Other Evaluation Metrics

The analyses thus far have exploited the structure of ROUGE and BERTScore to reason about the extent to which they measure information overlap between two summaries. Although it is desirable to ask the same question about other evaluation metrics, the metrics may not directly fit into this analysis framework or it would require significant effort to repeat this analysis for each one. Instead, we indirectly reason about how much information overlap other metrics measure through their correlations to ROUGE and the Pyramid Score.

[7] Although there are versions of ROUGE that remove stopwords, including them is significantly more common, and therefore we analyze the more popular ROUGE variant.

[8] The numbers in Table 4 numbers cannot be directly read off Table 3 nor do they sum to 100% because multiple categories can explain the same token alignment.

Metric	ROUGE-1	Pyr. Score	Δ
ROUGE-1	1.00	0.59	-
Pyramid Score	0.59	1.00	-
AutoSummENG	0.83	0.61	0.22
BERTScore	0.74	0.59	0.15
BEwT-E	0.81	0.62	0.19
MeMoG	0.68	0.52	0.16
METEOR	0.91	0.63	0.28
MoverScore	0.79	0.61	0.18
NPowER	0.81	0.60	0.21
PyrEval	0.47	0.35	0.12
ROUGE-2	0.79	0.58	0.21
S^3	0.92	0.63	0.29

Table 5: The summary-level Pearson correlations of various metrics to ROUGE-1 and the Pyramid Score (Δ is the difference between them). All of the other metrics correlate more strongly to ROUGE-1 than the Pyramid Score (by $\approx$0.2) and correlate to the Pyramid Score approximately as much as ROUGE-1 does ($\approx$0.6). Together, these results suggest the other metrics measure information overlap as poorly as ROUGE-1.

First, we assume that the Pyramid Score is the gold-standard for measuring the information overlap between summaries. This is a relatively safe assumption because the Pyramid Method is annotated by domain experts, and a candidate's Pyramid Score is based solely on how much information it has in common with a reference. There is no credit given to a candidate for discussing the right topics but with the incorrect information.

Then, the correlations of the other metrics to both ROUGE and the Pyramid Score are calculated and compared. If the correlation to ROUGE is much higher than the correlation to the Pyramid Score, then it is more likely that the metric suffers from the same issues that ROUGE does than it is to directly measure information overlap.

Table 5 contains the summary-level correlations of various other evaluation metrics to ROUGE and the Pyramid Score. The other metrics are: AutoSummENG (Giannakopoulos et al., 2008), BEwT-E (Tratz and Hovy, 2008), MeMoG (Giannakopoulos and Karkaletsis, 2010), METEOR (Denkowski and Lavie, 2014), MoverScore (Zhao et al., 2019), NPowER (Giannakopoulos and Karkaletsis, 2013), PyrEval (Gao et al., 2019), ROUGE-2, and S^3 (Peyrard, 2019). See §2.2.2 for

Metric	Summ-Level		Sys-Level	
	r	ρ	r	ρ
ROUGE	0.49	0.48	0.80	0.80
NP-CHUNKS	0.45	0.44	0.79	0.80

Table 6: The Pearson r and Spearman ρ correlations of ROUGE and calculating ROUGE with only NP chunks are very close, demonstrating that a purely topic based comparison (NP chunks) is a very high baseline for content quality correlations on TAC'08.

details on these metrics. These metrics exhibit a variety of different comparison techniques, from n-gram graph comparisons to contextual word-embedding comparisons, and other alignment based approaches.

Notably, most of the metrics' Pearson correlations to ROUGE are much higher than to the Pyramid Score by around 0.2 points, suggesting these metrics do not measure information overlap well. Further, their correlations to the Pyramid Score are roughly the same as ROUGE's, around 0.6. This means that these metrics correlate to a direct measure of information overlap as well as one would expect a metric which measures information overlap at the level of ROUGE to correlate. Although the results of this experiment are not direct evidence that many of the other evaluation metrics do a poor job at measuring information overlap, they do strongly suggest it.

3.7. Discussion

Responsiveness Correlations. Many of the automatic metrics analyzed in this work have demonstrated very high system-level correlations to ground-truth summary responsiveness judgments (Pearson's $r > 0.8$; Dang and Owczarzak, 2008, 2009), so the results that indicate they do not measure information overlap are somewhat contradictory. Since the metrics appear to compare summaries based on the topics they discuss, it is likely that only comparing summary topics is a very strong baseline for these benchmark datasets.

Indeed, we find in Table 6 that calculating ROUGE with only NP chunks (which represents little-to-no information under our definition of information) achieves nearly the same correlations as ROUGE on TAC'08. It is clear that this is not a good evaluation metric, but it does demonstrate that the baseline for this task is quite high.

Limitations. There are some limitations to our analysis. First, the results are specific to the datasets and summarization models that were used. However, TAC'08 and '09 are benchmark datasets for evaluating content quality and have been widely used to measure the performance of different metrics. Further, because the results from §3.5.2 are consistent across two rather different datasets (TAC and CNN/DailyMail), we believe these results are likely to hold for other datasets.

Then, the predicate-argument based information categories from §3.5.2 do not capture all of the information from a summary. A phrase like "the Turkish journalist" expresses the nationality of the journalist, but this information would not be represented by the tuples included in our analysis. However, we do not believe the addition of more tuples that express information outside of predicate-argument relationships would significantly change the experimental results.

3.8. Related Work

Most of the work that reasons about how to interpret the scores of evaluation metrics does so indirectly through correlations to human judgments (Dang and Owczarzak, 2009; Owczarzak and Dang, 2011). However, a high correlation is not conclusive evidence about what a metric measures since it is possible for the metric to directly measure some other aspect of a summary, which is in turn correlated with the ground-truth judgments (see §3.7). Our work can be viewed as more direct evidence about what ROUGE and BERTScore measure.

Recent work by Wang et al. (2020) argues that many of the same evaluation metrics covered in this work do not successfully measure the faithfulness of a summary based on low correlations to ground-truth judgments. The results from our experiments offer an explanation for why this is the case: The metrics do not compare summaries based on their information, therefore they cannot determine if a summary is factually consistent with its input.

Metrics which do attempt to directly measure information overlap between summaries are based on the gold-standard comparison technique, the Pyramid Method (Nenkova and Passonneau, 2004; Nenkova et al., 2007). Although it relies heavily on annotations by experts, there have been attempts to crowdsource (Shapira et al., 2019) or automate all or parts of the Pyramid Method (Passonneau et al., 2013; Yang et al., 2016; Hirao et al., 2018) including PyrEval (Gao et al., 2019), which we

analyzed in §3.6. These metrics have been met with less success than the text overlap-based ones covered by this work, potentially because measuring information overlap is more difficult than comparing summaries by their topics, and topic-based evaluations strongly correlate to responsiveness judgments (see §3.7).

3.9. Summary

In this Chapter, we argued that, ideally, reference-based evaluation metrics should estimate the content quality of a summary by comparing its information to that of the reference. However, we experimentally showed that ROUGE, BERTScore, and many other proposed metrics for evaluating the content quality of summaries largely do not compare summaries based on their information overlap. The implications of this result are that the summarization community does not have a reliable metric that aligns with its research goal: to generate summaries with high-quality information. This motivates the need to develop evaluation metrics which address the limitations of existing approaches by aiming to directly evaluate a summary's information.

CHAPTER 4 : Question Answering-Based Representations for Summary Evaluation

In the previous Chapter, we presented an argument that reference-based evaluation metrics should ideally calculate how similar candidate and reference summaries are based on the information they express. Then, we demonstrated that two reference-based evaluation metrics, ROUGE and BERT-Score, largely fail to evaluate summaries in this way.

To that end, this Chapter explores how question-answer (QA) pairs can be used to represent and evaluate the information in summaries through a proposal of a QA-based evaluation metric called QAEval. We experimentally demonstrate that QAEval is already very effective at evaluating summarization systems. Although it currently falls behind other metrics at the individual summary-level, we show that its potential future performance approaches the gold-standard for manual comparison of summary information. Further, we show experimental evidence that QAEval is indeed a better measure of how much information the candidate and reference summaries have in common than either ROUGE or BERTScore.

The work presented in this Chapter was originally described in Deutsch et al. (2021a), with the exception of §4.10, which was described in Deutsch and Roth (2021).

4.1. Introduction

The reference-based evaluation metric which we propose, QAEval (§4.3.1), along with other metrics that use QA to evaluate generated text (Eyal et al., 2019; Wang et al., 2020; Durmus et al., 2020) can be cast in a QA-based evaluation framework in which the information of a piece of text is represented by QA pairs, and the presence of that information in a second text is determined by calculating what proportion of those QA pairs the text correctly answers. Because the questions can only be answered if the candidate summary contains the corresponding information, QA-based metrics directly measure the amount of common information between the two texts. Therefore, they can be used to compare the information in the candidate and references summaries, providing a summary quality signal that is not effectively captured by existing evaluation metrics.

We show that with current question-generation and question-answering models, QAEval achieves

state-of-the-art system-level correlations to human judgments on benchmark datasets, outperforming all other automatic metrics and equalling the gold-standard Pyramid Method (Nenkova and Passonneau, 2004, §4.8). When evaluated at the summary-level, the metric is equal or better to other metrics on summaries that are very similar to the ground-truth and is competitive on others due to shortcomings of current state-of-the-art models (§4.7).

Through a careful analysis of each component of QAEval (§4.5-§4.7), we identify both the QA model and verifying if the predicted answer is correct as the performance bottlenecks (§4.7), whose noise likely explains the lower summary-level performance in some scenarios. Based on a manually annotated set of 2.9k QA pairs, we show that with human-level QA and answer verification performance, the summary-level upper-bound correlations of QAEval are better than all other automatic metrics and approach the gold-standard Pyramid Method. In combination with state-of-the-art correlation results, this strongly indicates that value of QA-based representations for summarization evaluation.

The contributions presented in this Chapter include (1) a proposal of QAEval, a more general QA-based metric for evaluating the content of summaries, (2) experimental evidence that demonstrates QAEval's state-of-the-art performance on benchmark datasets, (3) an analysis that identifies the QA model and answer verification as the performance bottlenecks, (4) an estimate that QAEval's upper-bound summary-level performance in scenarios in which it currently lags behind is high, approaching that of the gold-standard manual evaluation metric, the Pyramid Method, and (5) evidence that QAEval more effectively measures the information quality of a summary than either ROUGE (Lin, 2004) or BERTScore (Zhang et al., 2020).

4.2. Related Work

By far the most popular automatic methods for evaluating the content of a summary do so by comparing the tokens of the candidate and the reference. The de facto metric ROUGE (Lin, 2004) calculates a precision and recall score on the summaries' lexical overlap. Recent methods BERTScore (Zhang et al., 2020) and MoverScore (Zhao et al., 2019) instead compare tokens based on the similarity of their contextual word embeddings.

Because these text overlap metrics do nothing to specifically measure how much information is common between two summaries, their scores are polluted by spurious matches between tokens that do not express the same information, shown in Chapter 3. In contrast, QA-based evaluation metrics do directly compare summaries based on their information.

The gold-standard for manually comparing two summaries' information overlap is the Pyramid Method (Nenkova and Passonneau, 2004; Nenkova et al., 2007). It uses a domain-expert to identify spans of text between the candidate and reference summaries that express the same information, known as summary content units (SCUs). Because the Pyramid Method's final score is calculated exclusively on the number of common SCUs, it is a purely information-based evaluation. See Chapter 2 for more deatils about the Pyramid Method as well as some attempts to automate it, including PyrEval, which we compare our metric against in this Chapter.

Several recent works also use QA to evaluate summaries. Narayan et al. (2018b) use QA as part of a human evaluation to measure how much important document information was maintained by the summary. FEQA (Durmus et al., 2020) and QAGS (Wang et al., 2020) automate evaluating the faithfulness of a summary. Faithfulness and content quality are related, yet distinct, concepts. Content quality is a measure of whether the summary contains the correct information, whereas faithfulness measures whether the information is consistent with the input, regardless of its importance. FEQA and QAGS compare summaries to the input documents, whereas we compare summaries to references. Because the datasets used in our experiments are extractive summaries or have relatively high faithfulness ratings (Fabbri et al., 2021), we assume faithfulness is not an issue for simplicity.

Then, the most closely related work to ours is Eyal et al. (2019), who also use QA to evaluate the content of summaries via their metric APES. They create fill-in-the-blank questions by removing named entities from the reference summary and use a reading comprehension model to predict which entity was removed using the candidate summary.

There are several differences between their work and ours. Our proposed metric QAEval is more general than APES because QAEval asks and answers questions about noun phrases, whereas APES

is restricted to named entities. APES may fail to accurately score summaries which do not have a sufficient number of named entities. Then, our evaluation of QAEval is more comprehensive: The experiments in Eyal et al. (2019) were limited to evaluating APES on 8 input instances from TAC 2011, whereas our experiments are run on 92 instances from benchmark content quality datasets TAC 2008 and 2009 as well as 100 instances from the CNN/DailyMail dataset (Nallapati et al., 2016; Fabbri et al., 2021). Since our evaluation is more comprehensive and we demonstrate our metric has a high upper-bound performance, we believe it is a more convincing argument of the utility of using QA to evaluate summary content. Further, we perform an extensive evaluation on the individual components of the metric. We compare our metric's performance to APES' in §4.8 and §4.9.

4.3. QA-Based Evaluation

The standard line of research for evaluating the content quality of a summary is based on comparing the text of a candidate summary to a reference summary. Metrics that follow this approach include ROUGE, Basic Elements (Hovy et al., 2006), AutoSummENG (Giannakopoulos et al., 2008), METEOR (Denkowski and Lavie, 2014), BERTScore (Zhang et al., 2020), MoverScore (Zhao et al., 2019), and many more.

It is desirable to evaluate a summary based on the quality of the summary's information. For reference-based metrics, this means measuring the overlap in information between the candidate and reference summary. However, in Chapter 3, we showed evidence that suggests text overlap metrics do not successfully accomplish this. They match tokens which do not express the same information and end up comparing the similarity of two summaries based on the topics they discuss.

We argue that a much better method of comparing the information content of two summaries is through QA. In an ideal QA-based evaluation framework, all of the reference summary's information is represented by a set of QA pairs, and the candidate summary's recall of this information is measured by answering the questions against the candidate. The questions should only be answerable if the information necessary to answer them is present in the candidate. Therefore, this approach is fundamentally different from text overlap methods because it explicitly measures how

much of the reference's information is contained in the candidate.

While we cannot yet achieve this ideal QA-based metric (our QA-based representations may be incomplete, our QA models are imperfect, etc.), we next propose a specific instantiation of this framework that represents our best effort at reaching this goal with today's state-of-the-art models.

4.3.1. QAEval

At the core of this Chapter is a reference-based summarization evaluation metric that estimates the content quality of a summary, which we call QAEval. The metric represents the information of a reference summary by a set of QA pairs that are automatically generated from the reference. Then, QAEval estimates how much of this information is in a candidate summary by using a learned QA model to answer the questions against the candidate. The predictions from the QA model are verified as correct or incorrect, then the final score of the metric calculates what proportion of the questions were answered correctly.

Below, we describe the individual steps of the evaluation metric in more detail. Then, each component of QAEval is analyzed individually in Sections 4.5, 4.6, and 4.7 in order to identify any performance bottlenecks, followed by an overall evaluation of the metric in Section 4.8, a reproduction of the experiments of Eyal et al. (2019) in Section 4.9, and an application of the SCU-based analysis from Section 3.4 to QAEval in Section 4.10.

Answer Selection. The first step in generating questions from the reference summary is to pick a set of phrases that represents answers to questions that will later be generated. The answers should be chosen such that they will generate questions that cover as much of the information of the summary as possible. We evaluate how much semantic content is represented by several different answer selection strategies in §4.5.

Question Generation. Once the answers have been selected, a learned model is used to generate a question for each answer. The input to the question-generation model is a sentence which contains an answer phrase that is demarcated by special tokens. The output is a question which is answerable by that phrase.

Following Durmus et al. (2020), the generation model is a fine-tuned BART model (Lewis et al., 2020) trained on 55k human-written question-answer pairs collected by Demszky et al. (2018). The quality of the generated questions and the impact of using model-generated questions instead of human-written questions on downstream correlations is measured in §4.6.

Question Answering. Given a set of QA pairs generated from the reference summary, a QA model is used to answer the questions against the candidate summary. Since there are no summarization datasets with labeled QA pairs, the QA model must be trained on a different dataset. Further, because it is almost always the case that the candidate summary will not contain some reference summary information, it is necessary for the model to decide whether a question is answerable to reduce noise from spurious answers.

The QA model is a pre-trained ELECTRA-Large model (Clark et al., 2020) fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018), which contains unaswerable questions. The input to the model is the candidate summary and a question. The output is a span of text which contains the answer or a null string if the question is not answerable, depending on which is more probable under the model. We estimate the answering performance of the QA model on the summarization data and estimate the improvement in downstream correlations that would be expected if the QA model had human-level performance in §4.7.

Answer Verification & Scoring. Finally, once the QA model has output predictions for all of the questions generated from a reference summary, they are verified as being correct or incorrect with respect to the ground-truth answers that were used to generate the questions. We employ the two standard answer verification methods used by SQuAD, exact match (EM) and F_1 (Rajpurkar et al., 2016). If the QA model outputs the null string, the score for that answer is 0. We estimate whether these imperfect answer comparison strategies negatively impact downstream correlations in §4.7.

Finally, the metric produces two final scores that are the total EM and F_1 scores divided by the number of questions, thus calculating the proportion of questions answered correctly. If multiple reference summaries are available, the scores are macro-averaged. We refer to the metrics as QAEval-EM and QAEval-F_1.

4.4. Experimental Setup

We briefly overview the experimental setup used to evaluate metrics in this Chapter, but the details about the meta-evaluation procedure and datasets can be found in §2.3.

We meta-evaluate the metrics using the system- and summary-level correlations on three English summarization datasets: the benchmark TAC 2008 and 2009 datasets (Dang and Owczarzak, 2008, 2009) as well as the subset of the CNN/DM dataset (Nallapati et al., 2016) which was annotated by Fabbri et al. (2021).

The TAC datasets consist of 48/44 multi-document summarization instances, each with 4 reference summaries written by human annotators. Domain-expert judges rated the summaries output by 58/55 extractive models for each input on a scale of 1 to 5 based on how well they respond to an information need included in the task description. Each summary is also assigned a Pyramid Score (Nenkova and Passonneau, 2004) using a Pyramid constructed from the 4 reference summaries. Our experiments on TAC calculate the correlations of the metrics to the responsiveness score for the 58/55 model summarizers and 48/44 instances.

The annotations provided by Fabbri et al. (2021) on the single-document summarization CNN/DM dataset score the outputs of 16 models across 100 instances. The models are a mixture of extractive and abstractive approaches, and each instance has 1 reference summary. Fabbri et al. (2021) collected relevance scores from 3 expert annotators that captures if the summary contains important content from the input document. Our experiments report the correlation between the metrics' scores and the expert relevance judgments.

4.5. Answer Selection

In order for a QA-based evaluation metric to be successful, the QA pairs it uses to probe the candidate summary must represent a significant proportion of the reference summary's information. Therefore, in this Section, we aim to understand how much information the QA pairs in QAEval do represent and whether that may limit the metric's performance.

We explore three different answer selection strategies which pick phrases that are (1) named-entities,

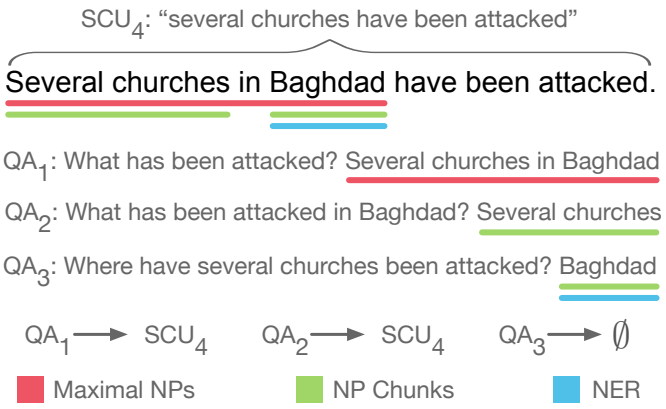

Figure 8: Example answers selected by the three strategies. The *only* SCU marked by annotators for this sentence is SCU_4, which does not include information about the location of the attacks. Therefore, an answer selection strategy that chooses “Baghdad” enables generating a QA pair such as QA_3, which probes for information not included in the Pyramid annotation.

(2) noun phrase chunks, (3) or maximally-sized noun phrases. The maximally-sized noun phrases in a sentence are identified by traversing the dependency tree down from the root until a noun is reached, then selecting the entire subtree for that noun. Example answers selected by each strategy are presented in Figure 8.

Since there is no well-established method of measuring how much semantic content is represented by a set of QA pairs, we instead compare the content covered by the QA pairs to that of another semantic representation, the Pyramid Method SCUs (see §2.2.1 for details). This approach allows us to compare answer selection strategies to a common point of reference as well as understand what types of information are represented by each formalism.

In order to compare the content covered by QA pairs and SCUs, each QA pair is manually mapped to an SCU based on whether the information that is being probed by the QA pair is included in the SCU description. For instance, in Figure 8, QA_1 and QA_2 map to SCU_4 because they target what was attacked, which is included in the SCU description, whereas QA_3 would not because the SCU does not describe the location of the attacks. This mapping allows us to calculate the proportion of QA pairs that map to some SCU, called the *QA precision*, and the proportion of SCUs that are mapped to by some QA pair, called the *SCU coverage*.

Strategy	Avg #QA Pairs	QA Precision	SCU Coverage
NER	11.7	83%	57%
NP Chunks	28.8	79%	91%
Maximal NPs	17.3	82%	77%

Table 7: The NP chunks answer selection strategy covers 91% of the information represented by the Pyramid Method (SCU Coverage) with 21% of the questions representing new information. From this, we conclude that the QA pairs generated from selecting noun chunk answers provides a semantic representation of the reference summary with very high-coverage.

To ensure the generated questions are of high-quality, we manually wrote questions for every answer selected by each strategy for 20 reference summaries across 10 input document sets from TAC 2008, totaling 801 questions. Every QA pair was further mapped to SCUs. The results (averaged over reference summaries) are presented in Table 7.

The most significant result we find is that the NP chunks strategy covers 91% of the semantic information included in the Pyramid Method, with an additional 21% of the questions targeting new information the Pyramid Method does not represent. The other two strategies have much lower SCU coverages, likely because they result in fewer generated questions since their QA precisions are approximately equal to that of NP chunks.

This result is very promising for QA-based evaluation metrics because it indicates that the QA pairs cover nearly all of the information that is used by the Pyramid Method, the best-performing manual content quality evaluation. Further, they even cover information the Pyramid Method does not, suggesting the potential for even better downstream correlations. Therefore, we conclude that the information represented by the QA pairs generated from selecting noun chunk answers is unlikely to be a factor which limits QAEval's performance, and we subsequently use that selection strategy for the rest of our experiments.

Comparing QA Pairs & SCUs. Upon comparing the information that is represented by one formalism and not the other, there are some key differences. The QA pairs miss information represented by nominal and adjectival modifiers because that information is contained within the answer noun phrase. For instance, for sentence *[A Turkish novelist] was arrested*, the question asks about

Input: On Jan. 7, 2005, with inauguration scheduled for Jan. 12, [Rossi] filed a lawsuit seeking a new election.

Expert: Who filed a lawsuit seeking a new election?

Model: On Jan. 7, 2005, with inauguration scheduled for Jan. 12, who filed a lawsuit seeking a new election?

Figure 9: A typical example of expert-written and model-generated questions answerable by the phrase in red. The model questions are often significantly more verbose than the expert questions, typically copying the majority of the input sentence.

who was arrested, and not about the nationality of the novelist, which the SCUs do include.

In contrast, the SCUs often miss specific details and generalize over information that the QA pairs do not. For instance, in Figure 8, although the SCUs do represent that the church attacks happened, it does not include information about their location, whereas this information is targeted by the QAs pairs.

4.6. Question Generation

An ideal question generation model should generate questions that are high enough quality that they do not impact the overall performance of the metric. In this Section, we compare questions generated by the learned model to expert-written questions, both empirically and extrinsically through downstream correlations to human judgments.

Empirical Analysis. Upon comparing the expert-written questions from §4.5 to model-generated questions for the same set of answers, we observe that a major difference between questions written by an expert versus a model is the level of verbosity. The model-generated questions often copy most of the input sentence over to the question, including parts of the sentence which may not be relevant to answering the question. In contrast, the questions written by an expert are more concise and remove the irrelevant details. Examples of this difference can be seen in Figure 9.

Despite the verbosity, nearly all of the model-generated questions are understandable to the authors of this work. However, because they are rather formulaic, the questions sometimes sound unnatural and could be confusing to a layman. We did not find any examples in which the answer was included in the question.

Downstream Correlation. Ideally, a QA-based evaluation metric would use an expert to write the questions to ensure they are all high-quality. Unfortunately, this does not scale and is very expensive and time consuming, so the questions must be model-generated. However, it is important to quantify any drop in performance caused by generating questions from a model rather than a domain-expert to understand the impact of using a less-than-ideal approach.

In order to measure any potential drop in performance, we compared the downstream correlations of the QA-based metrics to responsiveness judgments when using expert-written and model-generated questions. In both cases, the question-answering component was done using the learned model described in §4.3.1.

This experiment was performed on the subset of the TAC 2008 dataset for which we collected expert-written questions (see §4.5). That is, the summaries from 58 different systems across 10 input instances with 2 references each were scored using the two setups, and the respective correlations were computed.[9] We further simulated having a smaller number of input instances by downsampling the data to observe any emerging trends. The results are plotted in Figure 10.

The downstream summary-level correlations appear near-identical between the two approaches. However, surprisingly, the model-generated questions appear to result in better downstream correlations at the system-level than the expert-written questions. As soon as around 6 input instances are available, the two curves separate from each other's margins of error, with the model-generated questions clearly trending with a Spearman correlation of at least 0.05 higher.

It is not clear from examining the data why this is the case; there is no clear pattern that emerges which could explain why the model-generated questions result in higher correlations. Our best hypothesis is that the verbosity of the generated questions helps the QA model by including more keywords that can be matched against the summaries to find an answer.

From these unexpected results, we can conclude that the model-generated questions do not harm the downstream correlations of QAEval at either the summary- or system-levels. The rest of the

[9] Since we do not have expert-written questions for all 4 references across all 48 input clusters, these results are not strictly comparable to later experiments (e.g., §4.8).

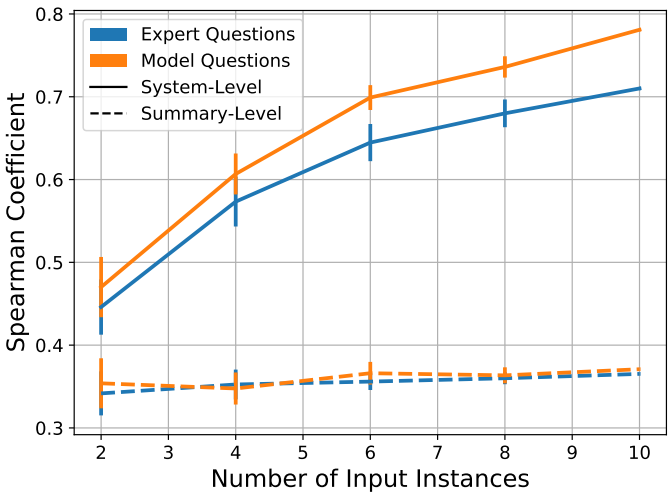

Figure 10: A comparison of the correlations of QAEval-F_1 on a subset of TAC 2008 using expert-written and model-generated questions. Each point represents the average correlation calculated using 30 samples of $\{2, 4, 6, 8, 10\}$ instances, plotted with 95% error bars. System-level correlations were calculated against the summarizers' average responsiveness scores across the entire TAC 2008 dataset. We hypothesize the model questions perform better due to their verbosity, which causes more keywords to be included in the question that the QA model can match against the summary.

experimentation in this Chapter will only use model-generated questions.

4.7. Question Answering & Verification

The task of the QA model and answer verification step are to determine whether a question is answerable against a summary, predict an answer if it is, then compare the prediction to the ground-truth answer to determine if it is correct. In this Section, we evaluate the performance of both components on the summarization data, first by calculating the QA performance (§4.7.1) and then by estimating the downstream correlation of QAEval if both components had human-level performance (§4.7.2).

4.7.1. Question-Answering Model Performance

Since the QA model is trained on Wikipedia articles in the SQuAD 2.0 dataset and used to answer questions generated from the summarization data, it is expected that the QA performance on the summarization data will be worse than on the original training data due to the domain shift.

In order to quantify the size of such a drop, one of the authors manually answered 2.9k generated questions from 20 reference summaries across 10 input clusters against 4 different summarizers

Dataset	%IsAns	IsAns-F_1	Given IsAns		
			EM	F_1	Acc
SQuAD 2.0	50.0%	92.0	88.0	94.5	-
TAC 2008	14.2%	52.4	56.5	69.5	84.3
CNN/DM	36.3%	75.3	73.8	83.6	86.3

Table 8: The QA performance on the summarization datasets drops significantly compared to its performance on SQuAD, especially for TAC 2008. This is expected due to the domain shift, however we suspect the drop is smaller for CNN/DailyMail because the generated and reference summaries are far more similar than for TAC, thus making it easier to answer questions.

on TAC 2008 and 2.3k generated questions across 10 input documents against all 16 summarizers on CNN/DM. For each question and summary pair, it was first determined whether the summary contained the answer to the question, then if it did, a span of text was selected as the answer. Then, the selected answer was later manually verified as correctly or incorrectly answering the question.

We compare the QA model's ability to both identify if a question is answerable and to select the correct answer if one exists separately on SQuAD 2.0 and the summarization datasets. This is done to measure any performance decrease on each problem in isolation. We calculated the F_1 score on the model's predictions on whether the question is answerable, plus the standard SQuAD EM and token F_1 metrics on only the subset of QA pairs for which the ground-truth and model agree that the question is answerable. We do not want to measure the quality of the predicted answer if the question is not answerable or the model outputs no answer.

In addition to EM and F_1, we also report the correct answer accuracy according to the human annotator. EM and F_1 are imperfect answer comparison strategies because they may fail to identify an answer as correct if it is a paraphrase of the ground-truth. Unlike SQuAD, the ground-truth answer and model prediction come from different source texts, increasing the likelihood that both answers will be expressed differently (see Fig. 11). Comparing the human annotator accuracy to EM and F_1 will quantify how well the automatic answer verification methods work on the summarization data.

The results are presented in Table 8. In general, the QA performance drops for both datasets, but the

Summary: The killing of Lebanon's former PM Rafiq Hariri renewed calls for Syria to abide by UN Security Council Resolution 1559 and end its dominance of Lebanon…

Question: What event put Syria under renewed pressure from the international community to abide by UN Security Council Resolution 1559 and withdraw its troops from Lebanon?

Answer: The February assassination

Prediction: The killing of Lebanon's former PM Rafiq Hariri

Figure 11: An example correct answer predicted by the model that is scored poorly by the EM and F_1 QA metrics (both would assign a score of 0 or near 0). This occurs because the answer and prediction are drawn from two different summaries, and the same event is referred to in different ways in each one.

decrease is more extreme for TAC'08. Specifically, we see that the drops in IsAns-F_1 are significant, amounting to decreasing by nearly 40 points from 92.0 on SQuAD to 52.4 on TAC'08 and almost 17 points to 75.3 on CNN/DM. This result indicates that identifying if a question is answerable is very challenging for the model, especially on TAC.

The EM and F_1 results across datasets also see a rather significant drop of around 25-30 points for TAC and 10-14 for CNN/DM, pointing to a much worse answering performance by the model when the model correctly predicts that an answer exists. However, the accuracy according to the human annotations is closer to the performance on SQuAD, implying the actual drop in performance is actually not as significant. For TAC, the discrepancy between the EM/F_1 scores and human accuracy judgments means the model's predictions are frequently correct, but EM and F_1 fail to identify them as such in a significant number of cases, thereby implying they are noisy answer verification methods. This problem has been observed for QA models before (Wang et al., 2020; Chen et al., 2020), but the issue seems particularly apparent for TAC. In the case of CNN/DM, the same differences exist but are smaller, especially for F_1. This means F_1 may be a good-enough answer verification method for this dataset.

We suspect that the QA model fares better on CNN/DM than TAC because the CNN/DM generated summaries are far more similar to the reference summaries than those in TAC. This is likely due

to several factors: (1) The CNN/DM task is in some sense easier than the TAC task. The lead-3 baseline is very strong, so the models can more easily generate high quality summaries; (2) The models included in the annotation are more recent state-of-the-art models compared to those from TAC and are likely better summarizers; (3) The task is single-document, so the information in the reference and generated summaries is more likely to be expressed the same way.

Since the two summaries being compared are similar to each other, the generated questions have a large token overlap with the target summary. This likely results in the QA model being more effective at identifying when an answer exists in the summary and then subsequently correctly identifying it. We expect this result to hold for other popular single-document summarization datasets.

From this experiment, we conclude that identifying whether a question is answerable is a potential performance bottleneck for both datasets. Further, for tasks in which the generated summaries are less similar to the references, the EM/F_1 answer verification methods may also be a limiting factor for QAEval.

4.7.2. Human-Level Performance Comparison

After identifying QA and answer verification as potentially problematic for QAEval's performance, we now estimate the size of any potential drop in downstream correlation compared to using human-level performance for both of those components.

Using the same human-annotated QA pairs from the previous Section, we calculated the summary-level correlations of QAEval when it uses either human annotations for the QA model, human annotations for the answer verification, or both. The correlations for these QAEval variants and several other metrics (discussed in §4.2) are in Table 9.

Since this experiment only uses a relatively small amount of data, none of the correlations differ by a statistically significant margin, so coming to definitive conclusions is difficult. However, some trends do emerge from the data.

For TAC 2008, QAEval is competitive to the other evaluation metrics when it uses a learned QA model and F_1 verification. Then, human-level performance for both QA and answer verification

provide large improvements in the downstream correlations, both independently and when combined. For instance, human QA annotations improve QAEval on TAC by 0.12 and 0.14 Pearson with F_1 and human verification, respectively. Human annotations for answer verification improve QAEval with model and human QA components by 0.17 and 0.29 Spearman, respectively. When both components use human annotations, the correlations are significantly better than any of the other automatic metrics and approach those of the Pyramid Method.

The results on CNN/DM are less clear. There is no obvious pattern in the data and all of the model/human combinations result in roughly the same performances. We suspect that because the drop in QA performance is less significant (§4.7.1), the differences in model and human-level QA performance is not reflected on CNN/DM as it is on TAC. Further, we empirically observed that the content of this dataset's summaries are more similar in content across models than the TAC summaries, making them harder to rank (as demonstrated by the lower correlations), which would also introduce more variance to the correlations.

Overall, this is a promising result for the future potential of QA-based evaluations, especially for more complex multi-document summarization tasks which are in some sense harder for metrics to evaluate than single-document summaries. While the current summary-level results on both datasets may be competitive to other metrics, the metric's upper-bound performance is very high on TAC and is approaching the gold-standard manual evaluation, the Pyramid Method.

4.8. Overall Metric Analysis

After analyzing each component of QAEval, we now turn to calculate the metric's correlations to human responsiveness/relevance judgments on TAC 2008, 2009, and CNN/DM (see §4.4 for more details about the experimental methodology; An additional experiment that varies the number of available references is included in Appendix A.1). For this experiment, QAEval uses the NP chunks answer selection strategy and learned question-generation and question-answering models and is therefore a fully automatic metric.

In addition to the QAEval correlations, we also report those of several baselines and state-of-the-art metrics, including the Pyramid Score, several variants of ROUGE, PyrEval, and MoverScore (which

System		TAC'08			CNN/DM		
		r	ρ	τ	r	ρ	τ
Pyramid Score		.63	.69	.65	-	-	-
ROUGE-1		.27	.27	.26	.25	.21	.18
ROUGE-2		.34	.40	.38	.13	.09	.06
ROUGE-L		.20	.22	.21	.13	.12	.08
ROUGE-SU4		.29	.22	.22	.16	.16	.12
MoverScore		.42	.28	.28	.27	.23	.18
APES		.35	.38	.37	.08	.09	.07
QAEval							
QA	Ans. Verif.						
Model	F_1	.31	.28	.26	.21	.23	.18
Human	F_1	.43	.33	.30	.15	.14	.12
Model	Human	.44	.45	.42	.25	.24	.20
Human	Human	.58	.62	.59	.22	.21	.17

Table 9: Summary-level correlations calculated using 4 systems across 10 inputs on TAC and 16 systems across 10 inputs on CNN/DailyMail compared using answers from a model or a human and verifying if the answer is correct using F_1 or a human. Because the results are on a small sample of the dataset, the results are not statistically significant. However, the trend on TAC is that human-level performance greatly improves the results, approaching correlations equal to the Pyramid Method's. On CNN/DailyMail, we suspect the same trend does not appear because the QA model performs much better than on TAC.

Metric	TAC 2008 System-Level			TAC 2008 Summary-Level		
	r	ρ	τ	r	ρ	τ
Pyramid Score	.90	.88	.70	.59	.59	.50
ROUGE-1	.79	.80	.60	.49	.48	.39
ROUGE-2	.83	.87	.67	.48	.48	.39
ROUGE-L	.74	.77	.57	.46	.45	.36
ROUGE-SU4	.80	.83	.63	.49	.48	.39
PyrEval	.81	.79	.59	.31	.31	.25
MoverScore	.83	.82	.63	**.50**	**.49**	**.40**
APES	.74	.82	.60	.25	.25	.21
QAEval-EM	**.93**	**.91**	**.76**	.33	.33	.27
QAEval-F_1	.90	.88	.71	.46	.45	.36

Metric	TAC 2009 System-Level			TAC 2009 Summary-Level		
	r	ρ	τ	r	ρ	τ
Pyramid Score	.90	.87	.70	.59	.57	.48
ROUGE-1	.83	.78	.60	**.54**	.47	.38
ROUGE-2	.76	.84	.67	.50	.50	.40
ROUGE-L	.82	.72	.54	**.54**	.47	.37
ROUGE-SU4	.77	.81	.63	.52	.50	.39
PyrEval	.86	.82	.64	.39	.35	.28
MoverScore	.82	.80	.63	.51	**.52**	**.42**
APES	**.87**	.80	.63	.41	.35	.28
QAEval-EM	.70	.87	.69	.42	.38	.30
QAEval-F_1	.81	**.89**	**.72**	.50	.45	.36

Table 10: The Pearson r, Spearman ρ, and Kendall τ correlation coefficients calculated between the metrics' scores and expert responsiveness judgments on the TAC 2008 (left) and TAC 2009 (right) datasets. QAEval has the highest system-level correlations, even better than the fully manual Pyramid Score, whereas the summary-level correlations are lower (EM) or competitive (F_1) with other metrics. We believe this supports our hypothesis that the QA model and answer verification are noisy (causing lower summary-level correlations) but average out to a high-quality metric given enough QA pairs (causing high system-level correlations). On TAC 2009, the QA r values are much lower because of an outlier, and r is sensitive to outliers. If the outlier is removed, the r values become 0.92 and 0.93 for EM and F_1.

reports better correlations than BERTScore), and APES. See §4.2 for descriptions of these metrics. Results in bold are the highest among the automatic metrics. Those underlined are statistically indistinguishable from the highest under a single-tailed permutation test for correlations with $\alpha = 0.05$, discussed in Chapter 7.

TAC 2008 & 2009. The correlations for TAC are presented in Table 10. First, we see that the summary-level correlations for the QAEval metrics are lower than or comparable to some of the other automatic metrics. For example, the TAC 2008 Pearson's r for QAEval-EM is 0.33, whereas the r values for QAEval-F_1 and ROUGE-2 are 0.46 and 0.48. Given that the QA model and answer verification components introduce noise into the metric, this result is consistent with the analysis in §4.7.2 and unsurprising.

However, the system-level results are quite surprising. The QAEval metrics achieve state-of-the-art system-level performance on nearly every correlation coefficient across both datasets, reaching correlations comparable to the Pyramid Method itself. For instance, on TAC 2008, QAEval-EM has a

Kendall's τ of 0.76 compared to 0.70 for the Pyramid Method and 0.67 for the next-highest automatic metric, ROUGE-2. This pattern largely holds for TAC 2009, with the exception of Pearson's r due to an outlier.[10]

It is unexpected that QAEval should achieve both state-of-the-art system-level results and lower summary-level results simultaneously and that the system-level results are even better than the Pyramid Method's.

We believe the discrepancy between the summary- and system-level results can be explained by the number of questions that is used by each evaluation. QAEval estimates the quality of an individual summary using around 110 questions. In contrast, the system-level scores are based over 5,000 QA pairs across 48 or 44 instances. We suspect that when QAEval's scores are averaged over such a large number of questions, the metric is able to overcome any noise introduced by the QA model or answer verification, resulting in a high-quality evaluation. APES, the other QA-based metric, also exhibits a similar pattern, supporting this hypothesis.

Then, it is likely that QAEval's system-level performance rivals the Pyramid Method's because the QA pairs probe for more semantic content than is represented by the SCUs (§4.5). The QA model and answer verification largely perform the same task as the Pyramid Method annotators: identify a span of text in the candidate summary which expresses a specific piece of information. It is unlikely the models do this better than a human, even after the noise is averaged out across thousands of examples. Therefore, it must be the case that the semantic representation of the QA pairs provides better coverage of the reference summary than the SCUs do, resulting in comparable overall performance.

CNN/DM. The results on the CNN/DM dataset are shown in Table 11. Compared to TAC, the improvement in system-level correlations is significantly larger. For instance, both QAEval variants achieve a system-level Spearman 0.91, whereas the next highest metrics APES and ROUGE-1 reach 0.73 and 0.62. Unlike for TAC, the summary-level correlations are either higher or statistically indistinguishable from the other metrics.

[10]Once removed, the r values are 0.92 and 0.93 for QAEval-EM and QAEval-F_1, higher than any other metric.

SummEval (Fabbri et al., 2021)						
Metric	**System-Level**			**Summary-Level**		
	r	ρ	τ	r	ρ	τ
ROUGE-1	.61	.62	.50	.28	.26	.20
ROUGE-2	.64	.60	.43	.23	.19	.14
ROUGE-L	.61	.48	.32	.21	.18	.14
ROUGE-SU4	.62	.56	.38	.23	.19	.15
MoverScore	.56	.54	.42	.28	.24	.18
APES	.68	.73	.58	.10	.09	.07
QAEval-EM	.80	**.91**	**.77**	.23	.23	.19
QAEval-F_1	**.82**	**.91**	**.77**	**.30**	**.29**	**.22**

Table 11: The QAEval metrics on the CNN/DailyMail annotations provided by Fabbri et al. (2021) achieve significantly higher correlations than the other automatic metrics, likely due to the relatively good QA model performance on this dataset compared to on TAC.

We hypothesize that the improved performance on CNN/DM compared to TAC is due to the QA model's quality on this dataset. In §4.7.1, we demonstrated that the QA performance did drop on CNN/DM with respect the model's results on the SQuAD data, however that performance decrease was not nearly as large on CNN/DM as on TAC. Since the QA model and answer verification are the performance bottlenecks and both suffer less on CNN/DM, the QAEval metrics achieve strong correlations.

This result is evidence to support that QAEval is a very effective metric for evaluating current state-of-the-art systems on today's popular summarization datasets.

Comparison to APES. Across all three datasets, QAEval achieves higher or comparable correlations than the other QA-based metric, APES, at both the summary- and system-levels. We suspect this is due to at least two reasons. First, their reading comprehension model likely has lower performance than the QA model used in QAEval. The QAEval pretrained model leverages recent state-of-the-art models that use contextual word embeddings, which the model of Eyal et al. (2019) does not use. Second, APES targets named entities in the summaries, which we demonstrated does not probe for as much information as using all noun phrases (§4.5). If the summaries do not contain a sufficient number of entities, APES may fail to accurately score it.

Human Score	R1	R2	RL	RSU4	APES	QAEval-EM	QAEval-F1
Pyramid Score	.73	.73	.70	.74	.61	.47	.61
Responsiveness	.62	.65	.60	.63	.50	.46	.56

Table 12: Summary-level Pearson correlations of ROUGE, APES, and QAEval to overall responsiveness and the Pyramid Score on the 8 instances from TAC 2011 that were used in Eyal et al. (2019). These numbers differ from those reported by Eyal et al. (2019) because they directly calculate the correlation between the scores for all of the summaries across all instances (personal communication with the authors). This differs from the standard definition of the summary-level correlation, which calculates a correlation per input document set then averages the correlations (see §2.3).

Overall. Since the performance of QAEval using EM and F_1 is roughly equal at the system-level, but F_1 is clearly better at the summary-level, we recommend that future work which evaluates with QAEval use the F_1 variant.

Overall, since evaluation metrics are most commonly used in the summarization community to rank summarization systems, these experimental results suggest that QAEval is one of the most effective evaluation metrics to date.

4.9. APES Experiments

To further compare QAEval to APES, we reproduce some of the experiments reported by Eyal et al. (2019) and compare the results of the two metrics.

4.9.1. TAC 2011 Comparison

First, we compare the summary-level correlations of the two metrics and ROUGE to human judgments on a subset of the TAC 2011 dataset. TAC 2011 contains extractive summaries produced by 51 models on 44 input document sets. However, Eyal et al. (2019) evaluate on the 8 input document sets about "Investigations and Trials" for which there were a sufficient number of named entities. This is because the QA model used by APES is only trained to predict named entities as answers. Similarly to TAC 2008 and 2009, each summary has an overall responsiveness score and a Pyramid score that were annotated by domain experts.

Table 12 contains the summary-level Pearson correlations of ROUGE, APES, and QAEval to the human judgments on the subset of TAC 2011. Although it is difficult to come to conclusions on

this dataset due to its relatively small size, we observe that APES out-performs QAEval-EM and under-performs QAEval-F_1 using the responsiveness score as the ground-truth. Using the Pyramid score as the ground-truth, APES and QAEval-F_1 are equal. However, both QA-based metrics are lower than the ROUGE variants, which is consistent with both APES and QAEval achieving lower correlations than ROUGE on TAC 2008 and 2009 at the summary-level. The APES correlations here are much higher on this subset of TAC 2011 than on the whole of TAC 2008 and 2009, supporting that its performance is higher when the summaries have a sufficient number of named entities.

4.9.2. Complementary Signals

Then, Eyal et al. (2019) demonstrate that APES and ROUGE are less correlated to each other than ROUGE variants are to themselves, suggesting they offer complementary signals of summary quality. In Fig. 12 we show the Pearson correlations between several different variants of ROUGE, APES, and QAEval on the TAC 2008 summaries.

Our results suggest similar conclusions to Eyal et al. (2019). Specifically, each of the ROUGE variants is very highly correlated to each other ($\geq .80$), whereas the correlations to the QA-based metrics are lower ($\approx .47$ for QAEval-EM, $.62$ for QAEval-F_1, and $.26$ for APES). Interestingly, APES and QAEval are as correlated to each other as APES is to ROUGE. We hypothesize that because the QA models are trained on different corpora (CNN for APES versus Wikipedia for QAEval), they learn different signals to answer questions and are more effective at scoring different summaries. Future work could explore combining lexical overlap and QA-based methods into a single metric.

4.10. SCU-Based Analysis, Revisited

In §3.4, we argued that ROUGE and BERTScore largely fail to evaluate the information of a summary because they score summaries using token alignments that are most often not between tokens which expert judges have marked as expressing the same meaning.

Like ROUGE and BERTScore, QAEval can be viewed as creating an alignment between the candidate and reference summaries. In this formulation, the mapping between the noun phrase in the reference summary that was used to generate a question and the predicted answer span in the candi-

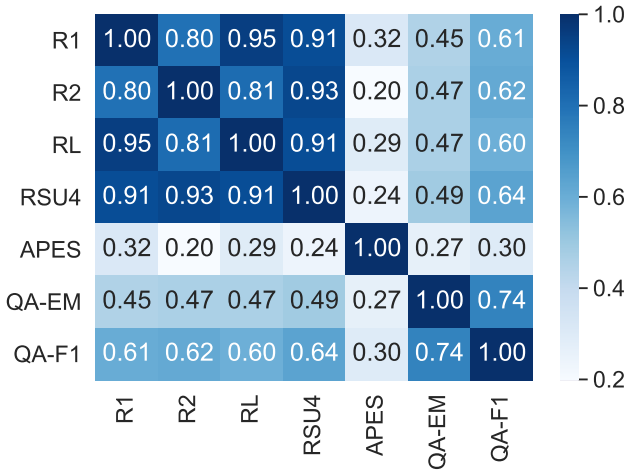

Figure 12: The Pearson correlations between the scores of several ROUGE variants, APES, and QAEval variants on TAC'08. The results support similar findings of Eyal et al. (2019), namely that the ROUGE metrics are highly correlated to each other but have low correlation to the QA-based metrics, suggesting the two types of metrics offer complementary signals.

date summary creates a token-level alignment between the two spans. The final metric score is the total weight of the alignment, where the weight of an edge is the F_1 score between the two spans, normalized by the number of questions.

Since QAEval can be viewed as a weighted alignment, we are able to repeat the analysis from §3.4 for QAEval and measure what proportion of its score can be explained by matches between SCUs.[11]

The distribution of the proportion of QAEval-F_1 that can be explained by SCU matches on TAC 2008 compared to ROUGE and BERTScore is shown in Fig. 13. The average proportion, 42%, is higher than ROUGE (25%) and BERTScore (15%), indicating QAEval captures information similarity better than the other two metrics. Table 13 summarizes this distribution for QAEval, ROUGE, and BERTScore, demonstrating that 46% of summaries have at least 50% of their QAEval score explained by SCU matches, whereas this is true for less than 4% and 0% of ROUGE and BERTScore.

Overall, this experiment provides evidence that QAEval is indeed measuring information quality more than either ROUGE or BERTScore.

[11] We do not repeat the category-based analysis from §3.5 because QAEval induces a mapping between noun phrases, so analyzing information-based categories, which include predicates, would not produce an interesting result.

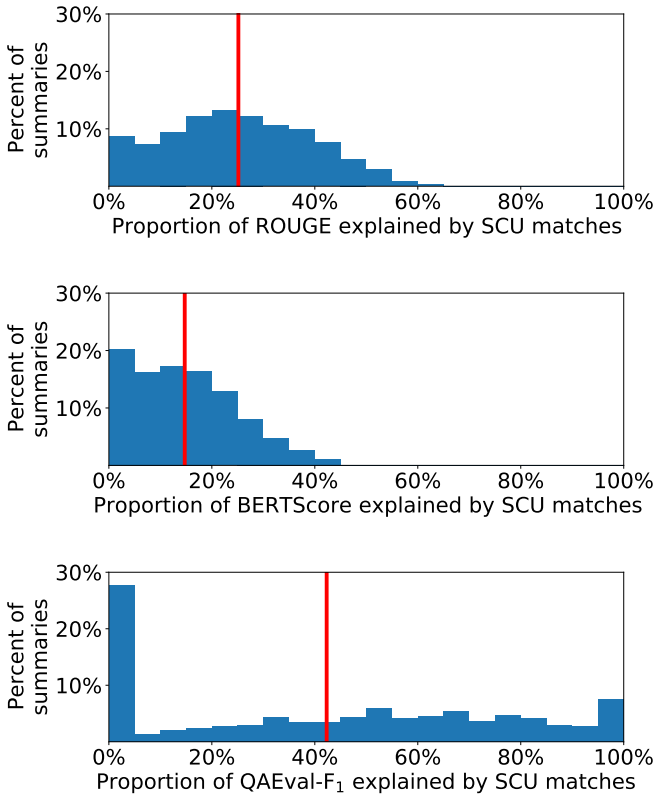

Figure 13: The distribution of the proportion of the ROUGE (top), BERTScore (middle), and QA-Eval-F_1 (bottom) scores that can be explained by SCU matches on TAC 2008 (top two plots taken from Fig. 5). Although its variance is higher, we find that QAEval can be explained by SCU matches far more than ROUGE or BERTScore on average.

Metric	Proportion of Metric			
	0-25%	25-50%	50-75%	75-100%
ROUGE	51%	45%	4%	0%
BERTScore	83%	17%	0%	0%
QAEval-F_1	36%	18%	24%	22%

Table 13: The percentage of summaries with a score explained by a given proportion of SCU matches on TAC 2008. QAEval has more summaries that have scores which can be mostly explained by SCU matches.

4.11. Discussion

Limitations. Overall, QAEval is limited by its dependence on using predicate-argument relations throughout each component of the metric. QAEval represents summaries with QA pairs that target nouns as answers, which is insufficient for representing all of the summary's information (as pointed out in §4.5). The question generation model is limited to producing questions that reason about the arguments of predicates and cannot generate more abstract questions (e.g., *What types of conflict have there been?* for Fig. 8). Likewise, QA models trained on SQuAD-style questions can only reason about matches between predicate-argument relations and cannot answer more abstract questions even if the generation model could produce them.

Because of this dependence on predicate-argument relations, any similarity between summaries that cannot be represented by matching predicates and arguments can also not be captured by QAEval. Although this does not appear to be an issue in our experiments, we anticipate that using generation and answering models which are capable of a more sophisticated level of reasoning will be necessary in the future.

QA-Based versus Text Overlap. Although QAEval has superior or comparable system-level correlations on the datasets included in our experimentation, it still lags behind text overlap-based method ROUGE at the summary-level in some settings. Therefore, we do not recommend completely replacing text overlap metrics with QAEval, nor do we believe that this should be done even if a QA-based metric achieves summary-level parity.

Both Eyal et al. (2019) and our work clearly show evidence that QA-based metrics provide a sum-

mary quality signal that is complementary to ROUGE (§4.9.2), yet both ROUGE and QAEval achieve strong correlations in our experiments. The quality signals captured by these metrics are clearly both valuable and different. Evaluating a summarization model with only one type of metric would miss out on summary quality signals captured by the other. Therefore, we recommend future work use both a text overlap metric as well as a QA-based metric to evaluate their summarization models.

Further Answer Verification Analysis. In follow up work, we performed a more in-depth analysis into different answer verification techniques (Deutsch and Roth, 2022a), which is not included in this thesis.

We explored using BERTScore as well as a trained model called LERC (Chen et al., 2020) to perform answer verification in addition to EM and token F_1 for QAEval and another QA-based metric, FEQA (Durmus et al., 2020). The experiments showed that LERC was indeed more successful at actually determining whether or not the expected and predicted answer spans were semantically equivalent, but this improved verification performance did not always translate into a better downstream metric score according to correlations with human judgment. We attribute this result to the fact that naive lexical methods of answer verification perform decently well on CNN/DailyMail (see Table 8) and that minor verification improvements over token F_1 on CNN/DailyMail or TAC may be washed out during the meta-evaluation of the metrics.

4.12. Summary

In this Chapter, we proposed a QA-based evaluation metric called QAEval. We demonstrated that QAEval already achieves state-of-the-art system-level correlations, and we estimate its upper-bound summary-level performance on multi-document summaries is quite high. Through a careful analysis of each component of QAEval, we identified that the performance bottlenecks are both the QA model and verifying whether or not the QA model's predicted answer is correct. Further, we showed evidence that QAEval approximates human annotations of common information in summaries better than either ROUGE or BERTScore, demonstrating that our proposed metric indeed is a step in the right direction toward a better evaluation metric. We believe that these results are strong evidence that QA-based evaluation metrics are a promising direction for future research on summarization

evaluation.

CHAPTER 5 : The Limitations of Reference-Free Evaluations of Generated Text

Thus far, the automatic evaluation metrics we have primarily discussed have all been reference-based in that they estimate the quality of a candidate summary by comparing it to a human-written reference summary. The inherent drawback to reference-based evaluations is that they rely on the existence of the reference summaries. References can be expensive and difficult to collect in large amounts, which limits the development of summarization models in new domains.

Reference-free evaluation metrics aim to address this limitation by estimating the content quality of a summary without access to a reference at all. Although the metric which we proposed in Chapter 4, QAEval, derives the set of salient questions from the reference summary, one natural extension of the metric could be to build a learned model which predicts a set of salient questions based on only the input document, which would remove its reliance on the reference summary.

However, in this Chapter, we present an argument which highlights the limitations of such an extension as well as of reference-free metrics as a whole. Central to our argument is that reference-free metrics are functionally equivalent to using one text generation model to evaluate another. As such, we show that simple inference procedures can be defined to select the approximate best possible output according to the metrics during inference and that they are bias toward outputs which are more similar to their own and against higher-quality outputs. These flaws are inherent to reference-free metrics, and we recommend that researchers who use them do so with an understanding of their limitations.

The work from this Chapter was originally presented in Deutsch et al. (2022a).

5.1. Introduction

Since many text generation tasks rely on references for evaluation, summarization is far from the only task which would benefit from developing high-quality reference-free evaluation metrics. For instance, there has been interest in building reference-free metrics for machine translation (MT; Fonseca et al., 2019; Rei et al., 2021), dialog generation (Mehri and Eskenazi, 2020; Honovich et al., 2021), image captioning (Hessel et al., 2021), and more. To that end, our analysis in this Chapter

focus on three reference-free evaluation metrics, QuestEval (Scialom et al., 2021) for summarization as well as Prism-src (Thompson and Post, 2020) and COMET-QE (Rei et al., 2021) for MT.

We find there are several implications of the fact that reference-free metrics are equivalent to using models to evaluate other models. First, the metrics' underlying models will achieve the best possible metric score by definition. Therefore, the "perfect" model is already known, and we show that it is possible to define simple approximate inference algorithms which use these models to find the approximate best output according to the metrics (§5.4, §5.5.1).

Then, the metrics have inherent, undesirable biases that originate from their underlying models. Not only do they favor the underlying models' outputs, but they are also biased toward outputs from models which are similar to their own, and biased against higher-quality outputs, such as those written by humans (§5.5.2, §5.5.3). Thus, if they were used as primary evaluation methods for a task, they would encourage other models to be more similar to their own and less human-like, an undesirable property of an evaluation metric.

Our recommendation is that reference-free metrics should not be used as methods for measuring progress on generation tasks such as MT, in which the goal is to achieve the highest possible value of the metric. Instead, they are better suited to be diagnostic statistics for analyzing model behavior with the understanding that they are inherently limited and biased (§5.6).

The contributions of this Chapter include: (1) insight on the equivalence of reference-free metrics and generation models, (2) a demonstration that reference-free metrics' values can be optimized at test time to achieve high-scoring outputs, and (3) an analysis that reveals reference free metrics' inherent biases and limitations.

5.2. Reference-Free Metrics as Models

Conditional text generation models can be viewed as a function $\theta(\cdot)$ which scores an output text $\mathbf{y} \in \mathcal{Y}$ for some input text $\mathbf{x}$. Then $\theta(\cdot)$ is used in conjunction with an inference procedure $f_\theta(\cdot)$ to

find the best output at test time.

$$\theta(\mathbf{x}, \mathbf{y}) \rightarrow \mathbb{R} \tag{5.1}$$

$$f_\theta(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \ \theta(\mathbf{x}, \mathbf{y}) \tag{5.2}$$

For instance, $\theta(\cdot)$ could be a learned sequence-to-sequence model and $f_\theta(\cdot)$ could be beam search.

The output of $f_\theta(\cdot)$, denoted $\hat{\mathbf{y}}$, is typically evaluated by some automatic metric $\mathcal{M}$. Reference-based metrics do this by scoring $\hat{\mathbf{y}}$ using some gold-standard text $\mathbf{y}^*$ (which is not available to the model during inference) and the input $\mathbf{x}$ (which is not always used). For instance, $\mathcal{M}_{\text{Ref-Based}}$ could calculate a BLEU score (Papineni et al., 2002) between the output translation $\hat{\mathbf{y}}$ and the gold translation $\mathbf{y}^*$.

$$\mathcal{M}_{\text{Ref-Based}}(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{y}^*) \rightarrow \mathbb{R} \tag{5.3}$$

In contrast, reference-free metrics calculate a score for $\hat{\mathbf{y}}$ without $\mathbf{y}^*$:

$$\mathcal{M}_{\text{Ref-Free}}(\mathbf{x}, \hat{\mathbf{y}}) \rightarrow \mathbb{R} \tag{5.4}$$

Such metrics include the three analyzed in this work, namely, Prism-src (Thompson and Post, 2020), COMET-QE (Rei et al., 2021), and QuestEval (Scialom et al., 2021).

Because $\theta(\cdot)$ and $\mathcal{M}_{\text{Ref-Free}}$ are both functions of only $\mathbf{x}$ and $\mathbf{y}$ (equivalently $\hat{\mathbf{y}}$), $\mathcal{M}_{\text{Ref-Free}}$ itself can be viewed as a conditional generation model. For some metrics, such as Prism-src, this is explicitly stated, whereas others are implicitly making this assumption. This is not the case for reference-based metrics since they additionally require $\mathbf{y}^*$ as input.

Since reference-free metrics are equivalent to generation models, there must exist some inference procedure which finds the best output text under the metric, denoted $g_{\mathcal{M}_{\text{Ref-Free}}}(\cdot)$:

$$g_{\mathcal{M}_{\text{Ref-Free}}}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{M}_{\text{Ref-Free}}(\mathbf{x}, \mathbf{y}) \tag{5.5}$$

Computing $g_{\mathcal{M}_{\text{Ref-Free}}}(\cdot)$ may be computationally expensive because $\mathcal{M}_{\text{Ref-Free}}$ may not support efficient inference. However, the inference procedure does always exist, and will return the best possible output according to the reference-free metric by definition.

We explore the implications of using a model to evaluate other models by analyzing the behavior of three different reference-free evaluation metrics on two text generation tasks, MT and summarization.

5.3. Analysis Setup

Datasets. Our MT experiments are run on the data collected for the WMT'19 metrics shared task (Ma et al., 2019), which includes reference translations and human-judged model outputs for 10 to 20 translation systems across 18 language pairs.

The summarization experiments use the SummEval (Fabbri et al., 2021) and REALSumm (Bhandari et al., 2020) datasets, which consist of reference summaries and human-judged model outputs for 16 and 25 summarization models, respectively, collected from the CNN/DailyMail dataset (Nallapati et al., 2016). More details on SummEval and REALSumm can be found in Section 2.3.

Prism-src. Prism-src is a reference-free evaluation translation metric that scores a translated text according to the log-probability of the translation conditioned on the original source text under a learned sequence-to-sequence translation model (Thompson and Post, 2020). The model is a multilingual MT model, meaning it was trained using many different language pairs, so the same learned parameters can be used to score translations in various languages.

COMET-QE. COMET-QE (Rei et al., 2021) is a modification of the learned reference-based MT evaluation metric COMET (Rei et al., 2020). COMET embeds the candidate translation, source text, and reference translation using a cross-lingual encoder, creates a pooled featured representation using the three encodings, and trains the model end-to-end to predict human judgments of the quality of the candidate translation. COMET-QE uses the same architecture to predict a score for the candidate translation but only uses the candidate translation and source text to create the pooled feature representation, and is therefore reference-free.

QuestEval. Scialom et al. (2021) proposed a reference-free summarization metric called QuestEval which generates QA pairs from both the source document and generated summary then scores the summary based on the proportion of those pairs which are answered correctly in the opposite text. The metric optionally includes a step in which the QA pairs generated from the source document are weighted based on a learned query weighting model. The query weighter was trained to predict the probability that a question is answered in the CNN/DailyMail reference summaries using a pre-trained QA model. We use the query weighter in our experiments since it improved the performance of QuestEval in Scialom et al. (2021).

In some sense, QuestEval could be viewed as a natural extension of QAEval (Chapter 4) which has a model of question salience for the questions derived from the input document instead of assuming the reference summary's content contains the salient information.

Reference-Based Metrics. We analyze the reference-free metrics with respect to various reference-based metrics which have been demonstrated to have strong correlations to human judgments of translation/summary quality. BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) compare the two texts using n-gram overlap statistics. BERTScore calculates a quality score based on how similar the reference and candidate texts' BERT (Devlin et al., 2019) embeddings are (Zhang et al., 2020). QAEval is a QA-based metric for summarization, which generates wh-questions from the reference summary and calculates a score for the candidate summary based on the proportion of questions answered correctly (described in detail in Chapter 4). Finally BLEURT is a learned MT metric which predicts a translation quality score using encoded BERT representations of the reference and candidate translations (Sellam et al., 2020).

5.4. Metric Optimization

Since reference-free metrics are equivalent to models, then it is possible to define inference procedures which produce the best-possible outputs according to the metrics. Here, we discuss three such (approximate) inference procedures. Importantly, they can all be run at test time because they do not rely on a reference text.

5.4.1. Direct Optimization

If a reference-free metric scores a candidate output in a way that an efficient approximate inference procedure can be defined, then finding the best possible output under the metric is straightforward.

Among the metrics analyzed in this paper, only Prism-src falls into this category. Because Prism-src assigns a score to a translation equal to its average log-probability under a learned sequence-to-sequence MT model, the approximate best translation under Prism-src can be found by running beam search with the MT model conditioned on the source text.

5.4.2. Greedy Optimization for Extractive Summarization

Summarization models are generally categorized as being either extractive or abstractive. Extractive systems create a summary by selecting k salient document sentences, whereas abstractive systems typically autoregressively generate a summary with a sequence-to-sequence model.

The best possible extractive summary according to a reference-free metric can be found by enumerating all possible summaries of k sentences, scoring them with the metric, and selecting the summary with the highest score. Since the number of k sentence summaries may be large, this may be computationally expensive. However, an approximate inference procedure can be used instead.

Rather than enumerate all possible extractive summaries, the approximate inference algorithm constructs a summary by greedily selecting one sentence that increases the score of the metric the most. This is repeated until a target summary length of k sentences is reached, resulting in an approximation of the best possible summary under the reference-free metric.

An identical procedure is commonly used for creating sentence-level labels for training extractive summarization models, except a reference-based evaluation metric, such as ROUGE, is typically used for scoring the sentences instead of a reference-free metric (Nallapati et al., 2017). The key difference is that the output summary from the reference-based procedure is used to train a model which later predicts k salient sentences during inference, whereas the reference-free procedure can be directly used during inference (i.e., without training) to pick the approximately best summary under the reference-free metric.

5.4.3. Reranking

Exact inference for any reference-free metric can be performed by enumerating all possible outputs, calculating the score of each one, and selecting the output with the highest score. However, it is almost certainly true that this is computationally intractable for any practical application of text generation due to the size of the output space.

To that end, we propose to use reranking as an approximate inference procedure in which a pre-trained model for the task at hand is used to restrict the search space to a small set of high-quality candidate outputs. These outputs are then scored and reranked using the reference-free metric to identify an approximately best output under the metric.

In practice, we identify a set of k high-quality outputs using standard beam search with pre-trained sequence-to-sequence summarization and MT models and a beam size of k. The top-k partial outputs sorted by their log-likelihood under the pre-trained models are kept at each step of beam search. The final outputs are then reranked by a reference-free metric. For summarization, we use BART (Lewis et al., 2020) trained on the CNN/DailyMail dataset. For MT, we use Facebook's submission to the WMT'19 translation shared task (Ng et al., 2019). The model is available for en→de, de→en, en→ru, and ru→en.

5.5. Analysis

5.5.1. Approximate Inference Effectiveness

Although inference methods for the reference-free metrics can be defined, it is possible that they fail to find high-scoring outputs due to the complexity of the search problem. However in this analysis, we show that the simple approximate inference procedures defined in §5.4 are effective at optimizing the metrics' scores.

We compared the outputs obtained by the inference algorithms to those from systems included in the WMT'19, SummEval, and REALSumm datasets. Fig. 14 evaluates using the direct optimization procedure (§5.4.1) to select the best Prism-src output, Fig. 15 shows the results of using reranking (§5.4.3) to pick the best outputs according to COMET-QE, and Fig. 16 contains the results of using the greedy extractive procedure (§5.4.2) to optimize QuestEval. The Figures also include the sys-

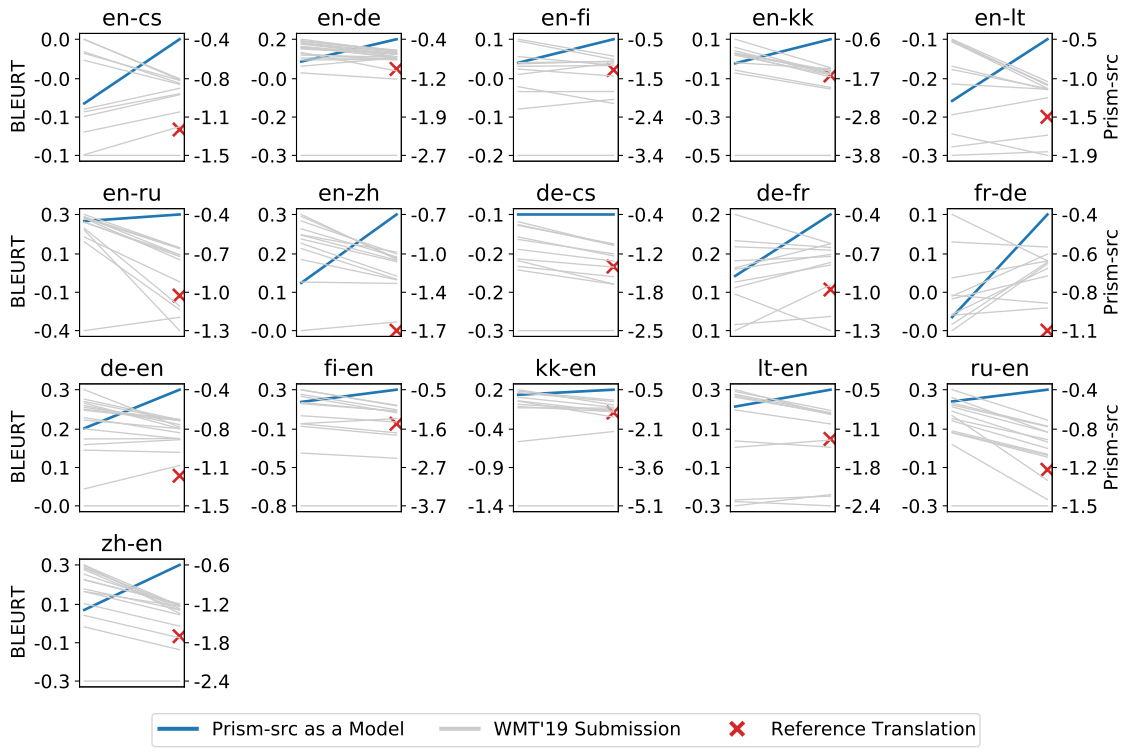

Figure 14: Directly optimizing Prism-src (blue line; §5.4.1) yields the highest Prism-src performance (right y-axis) but only an average system as evaluated by BLEURT (left y-axis). The reference translation (red "x") has a lower Prism-src score compared to many systems across all language pairs, demonstrating Prism-src's biases toward learned model output and against human-written translations.

tems' scores under the reference-based metrics BLEURT for MT and ROUGE for summarization.

In all MT language pairs and both summarization datasets, the inference algorithms produce the highest scoring outputs under the reference-free metrics, often by a large margin. For example, reranking translations according to their COMET-QE scores on de→en results in a relative 38% improvement in COMET-QE over the WMT'19 submission with the best COMET-QE score (0.347 → 0.478). Clearly, the simple inference procedures can be used to find very high scoring outputs under the reference-free metrics even if the metric does not directly support efficient inference.

Despite the improvements in reference-free scores, it does not appear as if these outputs are as high-quality as the reference-free metric scores would indicate. Ideally, the outputs would be rated by humans to establish a ground-truth quality score, but a fair comparison to the other systems'

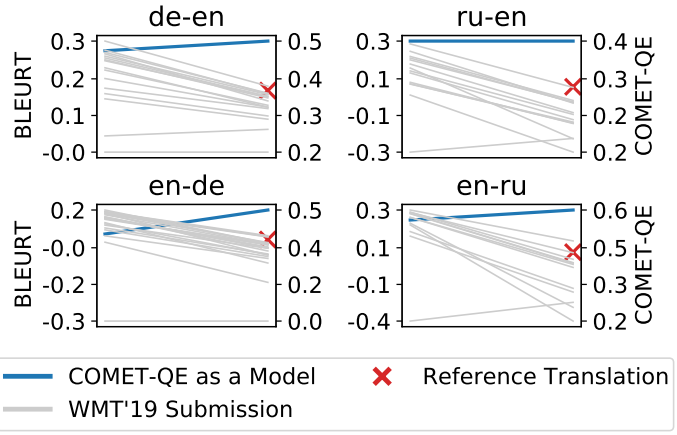

Figure 15: Reranking the output of a pre-trained model results in COMET-src scores which are far higher than the reference translations' scores (red "x"), demonstrating that a better COMET-src value means the translation is more similar to the metric's underlying model instead of human-written text.

outputs included in the datasets would require re-judging their translations or summaries, which is prohibitively expensive. Instead, we use reference-based metrics as indicators of quality.

When the outputs from our inference algorithms are compared to other systems using reference-based metrics (also shown in Figs. 14, 15, and 16), we see that they are often of average quality or worse. For example, the greedy extractive summaries obtained by optimizing QuestEval on REAL-Summ are among the lowest-performing in the dataset according to ROUGE-2 (Fig. 16). Thus, directly optimizing the reference-free metrics does not always yield a high-quality system.

5.5.2. Undesirable Metric Biases

Ideally, evaluation metrics should score human-written text higher than learned model outputs since it is very likely that the human references are of higher-quality. However, we see that this does not always happen with reference-free metrics.

Figs. 14, 15, and 16 additionally contain the scores of the reference texts under the reference-free metrics (marked with a red "x"). In all settings, the inference algorithms' outputs score higher than the references. This is unsurprising because they select their outputs to optimize the metrics' values. However, it demonstrates that as models continually improve their reference-free scores, they will begin to converge onto the metrics' underlying models and not onto human-quality text. As the

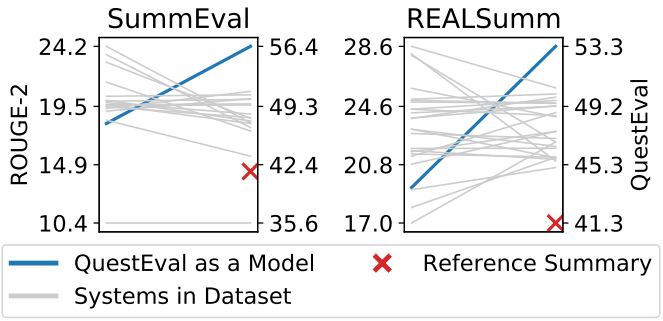

Figure 16: Nearly all models in the SummEval and REALSumm datasets have better QuestEval scores than the reference summaries, demonstrating the metrics bias toward learned model output over human-written text.

research goal of text generation is to build models which produce human-like text, the reference-free metrics' scores do not align with this goal.

Further, the same Figures show that it is often the case that other models in the datasets—which did not directly optimize the reference-free metrics—also are scored higher than human text by the reference-free metrics. This is especially true for Prism-src and QuestEval, but less so for COMET-QE. For example, on language pair de-en, only one system has a lower Prism-src score than the reference translations (Fig. 14). These metrics appear to have a bias for outputs from learned models over human-written text.

We suspect this bias for outputs from learned models is due to how the internal models used by Prism-src and QuestEval to score text are trained. Prism-src scores translations using an MT model that was trained on standard MT data. QuestEval uses a question-weighting model that predicts how likely a question is answered in a reference summary, which was trained on the same CNN/DailyMail dataset that the summarization models in the summarization datasets were also trained on. Thus, these metrics internally use models which are directly or indirectly trained to perform the generation task (MT or summarization).

Generation models which are trained on the same datasets are known to exhibit similar behavior and make similar mistakes, and their outputs often look markedly different from human-written text. Therefore, we suspect that the signals the internal MT/question-weighting models have learned to

identify high-quality text are similar to those which the task-specific models have learned to produce their output, and thus receive high scores by the metrics. In contrast, the human-written text likely does not rely on these signals, and is thus perceived as low-quality by the metrics.

This bias toward learned model outputs is potentially less severe for COMET-QE because unlike Prism-src and QuestEval, it is specifically trained to predict translation quality using manually collected human judgments. Thus, the signals it learns to identify high-quality text are likely different than what is learned by the translation models in the WMT'19 dataset.

In summary, the metrics' biases toward their own outputs (or other learned model outputs) and against human texts demonstrates they reward outputs which look more like their own instead of human-quality text, an undesirable property of an evaluation metric.

5.5.3. Reference-Free Metrics as Pseudo-References

Thus far, we have argued that the underlying model of a reference-free metric is the theoretical best model according to the metric. It would intuitively follow that the more similar another model is to the metric's underlying model, the higher metric score that model would receive. To that end, in this analysis we demonstrate that scoring a system with a reference-free metric is roughly equivalent to evaluating that system's outputs against the metric's best possible output using a reference-based metric. This further demonstrates the limitations of reference-free metrics, including their biases toward their own underlying models' outputs.

A pseudo-reference is a piece of text which is used in place of a human-written reference to evaluate some candidate text with a reference-based metric (Louis and Nenkova, 2013; Gao et al., 2020). For the reference-free metrics, we define the pseudo-reference to be the output from the inference procedures defined in §5.4 (i.e., those evaluated in §5.5.1). For example, the QuestEval pseudo-reference is the extractive summary which was selected to greedily maximize the QuestEval score.

Once the pseudo-references have been defined, they can be used in conjunction with any reference-based metric, such as BLEURT or QAEval, to evaluate other translations or summaries. To quantify the similarity between evaluating a system with a pseudo-reference and a reference-free metric, we

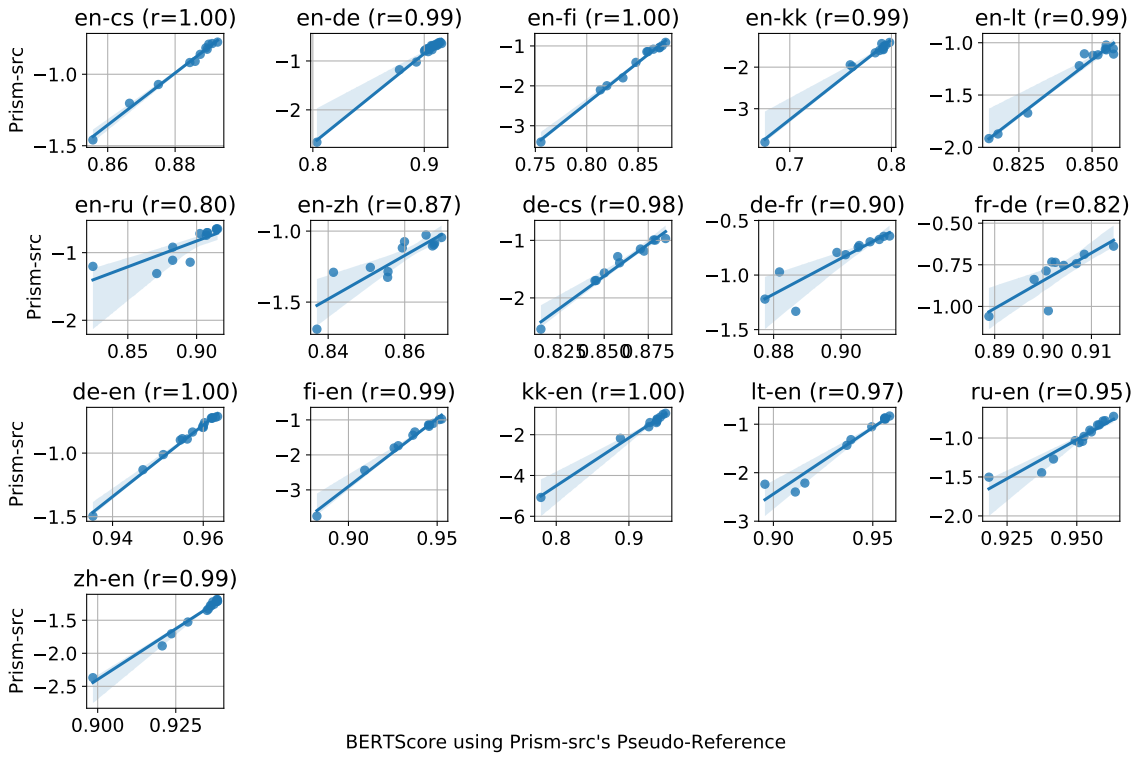

Figure 17: A system's BERTScore in which the output from directly optimizing Prism-src is used as the reference (x-axis) is strongly correlated to that same system's Prism-src score (y-axis). This demonstrates Prism-src is roughly equivalent to evaluating systems with a pseudo-reference translation which is generated by a model. Pearson's r shown in the title of each plot.

calculated the Pearson correlation between the system-level scores between the two methods. These correlations are show in Figs. 17, 18, and 19.

For MT, we find that the BERTScore of a translation that uses pseudo-references instead of the human-written reference has an average Pearson correlation of 0.95 and 0.92 to the Prism-src (Fig. 17) and COMET-QE scores (Fig. 18), respectively. The summarization correlations for QuestEval to QAEval using a pseudo-reference (Fig. 19) are also rather strong at 0.88 on average. The correlations using other reference-based metrics are slightly weaker on average due to low values on specific datasets or language pairs, but there are many instances in which the correlations are ≥ 0.9.

Overall, these correlations are very strong, suggesting that the reference-free metrics are roughly

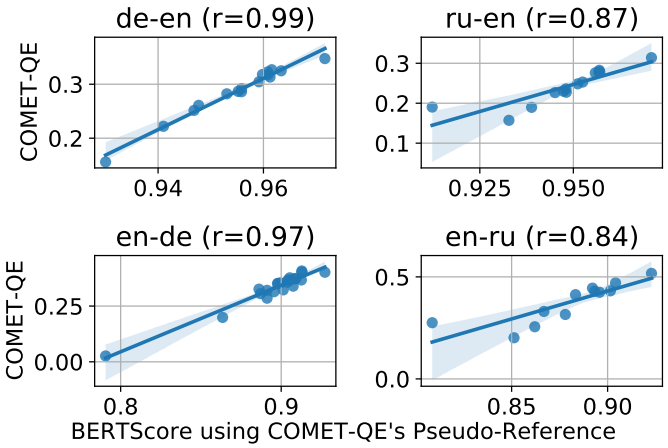

Figure 18: Scoring systems with BERTScore against psudeo-references obtained by optimizing COMET-src strongly correlates to the systems' COMET-src scores.

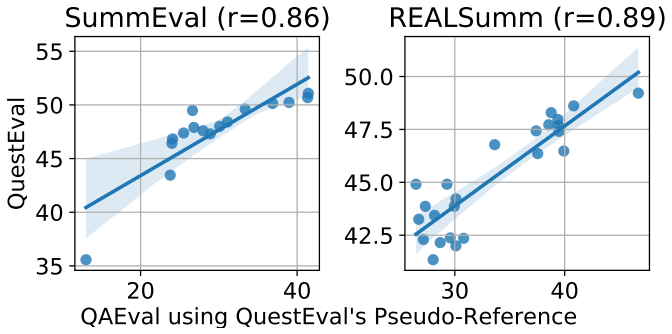

Figure 19: Calculating a system's QAEval score against the psuedo-reference chosen to maximize its QuestEval score is strongly correlated with that same systems' QuestEval score on SummEval and REALSumm.

equivalent to using pseudo-references to evaluate other models. Once the metrics are viewed this way, their limitations become clear. The metrics' outputs are the gold-standard against which all other outputs should be evaluated. Thus, the metrics favor their own outputs (the pseudo-references) and other outputs which are more similar (where similarity is measured using reference-based metrics). Further, their ability to evaluate other models is inherently limited by the qualities of their pseudo-references. If a system outputs a translation or a summary which is higher-quality than the pseudo-reference, it will be incorrectly penalized because it is different than the pseudo-reference even though those differences are actually improvements. Thus, the metrics' scores of systems which are better in quality than their own underlying models will be misleading.

5.6. Discussion

5.6.1. Reference-Free Evaluation

In this work, we have argued that reference-free metrics are equivalent to using generation models to evaluate other models (§5.2). These underlying models achieve the maximum metric score, which can be approximated using simple inference procedures even when the metrics do not support efficient inference (§5.4, §5.5.1). Because reference-free metrics are models, they have undesirable biases in which they favor their own models' outputs and can systematically underestimate the quality of systems which are better than their own models (§5.5.2, §5.5.3).

Therefore, we argue that reference-free metrics should not be used to measure progress on a task, for instance, by concluding that an MT or summarization model is better than another because it has a higher reference-free metric score. If they are used this way, the model which will achieve the best performance is already known (the metric's underlying model), and simple inference procedures are effective at using those models to generate high-scoring outputs. Improving the metric's score is no longer about creating better models on the training data for the task; rather, it is about coming up with better procedures to optimize the metric during inference. In the end, the quality of the system which is produced by this evaluation methodology is limited by the quality of the metric itself.

Instead, we recommend investing resources into collecting human-written references that can be used for evaluation purposes. Although individual reference-based metrics have their own flaws

(Graham, 2015; Bhandari et al., 2020; Deutsch et al., 2021b), the class of reference-based metrics still ultimately encourages systems to generate text which is similar to humans, which is the goal of text generation research as it is defined nowadays.

5.6.2. What about Inherently Reference-Free Evaluations?

Although we argue that measuring system quality with reference-free metrics is flawed and misleading, abandoning reference-free evaluations completely is not an option and one which we do not advocate for. There are many aspects of generated text that inherently do not rely on the presence of a reference to be evaluated, such as the fluency of a translation or how faithful a summary is to its input document. There is no obvious benefit of including a reference text in these evaluations. As such, they suffer from the same issues as the metrics analyzed in this work, yet the motivation for being able to automatically evaluate these aspects of text is clear.

In these scenarios, we recommend using reference-free metrics as diagnostic tools for better understanding the behavior of models instead of a method for measuring progress on a task. For instance, the perplexity of a summary under a large-scale language model is a useful statistic to report in order to approximate its fluency, but the value should only be interpreted as exactly what it measures—how likely the observed text is under the language model—with the understanding that the measure is inherently biased towards the underlying language model. The perplexity should not be used to drive research on how to generate more fluent summaries because the most fluent summarization model is the language model itself.

Such a recommendation can also be applied for the task of "quality estimation" within the context of machine translation (Callison-Burch et al., 2012). Similarly to as described above, we emphasize that reference-free metrics that are used to estimate the quality of a translation at test time are biased, can be exploited, and will systematically identify higher-quality translations as worse than they actually are.

This recommendation applies not only to metrics which measure inherently reference-free aspects of text, but also to the metrics that evaluate aspects which we argue should use references, such as those analyzed in this work. They are certainly useful statistics to report, but improving their values

as much as possible should not be the goal.

5.7. Related Work

Various other reference-free metrics have been proposed for MT (Fonseca et al., 2019; Rei et al., 2021), summarization (Louis and Nenkova, 2013; Scialom et al., 2019; Xenouleas et al., 2019; Vasilyev et al., 2020; Scialom et al., 2021), dialog generation (Mehri and Eskenazi, 2020; Honovich et al., 2021), image captioning (Hessel et al., 2021), and simplification (Martin et al., 2018; Kriz et al., 2020). Many of these works argue that their reference-free metrics have stronger correlations to ground-truth human judgments than their reference-based counterparts, thereby implying that reference-free metrics could be used instead of reference-based metrics. It is worth noting, however, that some acknowledge the limitations of evaluation without references and suggest that reference-free evaluations should instead complement existing reference-based evaluations (Louis and Nenkova, 2013).

Prism-src was further explored by Agrawal et al. (2021), who experimented with its model capacity, scoring methods, and more. They perform an analysis using pseudo-references similar to ours in §5.5.2. In their experiment, they argue that they could not find evidence that Prism-src is biased toward outputs which are similar to the metric's underlying model's outputs. However, we argue that the high correlations between a system's Prism-src's score and its similarity to the metric's pseudo-reference (measured by reference-free metrics) does demonstrate such a bias exists, but Agrawal et al. (2021) did not find evidence this bias negatively impacted the rankings of the systems.

In the WMT shared tasks on quality estimation (Specia et al., 2020, 2021), not all of the tasks were aimed at predicting a translation's direct assessment quality score without a reference. For instance, one task objective was to create metrics that identify "catastrophic errors" in translations, which would include translations that introduce hateful or violent text that was not present in the source. Such a reference-free metric fits our recommendation for using the metrics as diagnostic tools to better understand model behavior rather than a metric in which the goal is to improve its value as much as possible.

5.8. Summary

In this Chapter, we have argued that reference-free metrics are inherently limited in their ability to evaluate generated text. Because they are equivalent to generation models, they can be directly optimized at test time, they are biased toward their own models' outputs and outputs from similar models, and they can be biased against higher-quality text, such as human-written references. Therefore, we recommend against using reference-free metrics as measures of progress on tasks and instead advocate for them to be used as useful statistics to calculate in order to better understand model behavior.

CHAPTER 6 : Question Answering-Based Representations for Summary Generation

Thus far, the Chapters of this thesis have focused on the problem of evaluating summaries. Specifically, they centered around a proposal of a reference-based metric called QAEval that represents and evaluates the information content of a summary using question-answer (QA) pairs in Chapter 4.

In contrast, this Chapter pivots away from evaluation to the actual task of summary generation. We explore how the same QA-based representations of information used by QAEval have value beyond evaluation by demonstrating that they can be incorporated into a summarization model to generate higher-quality summaries. A question-generation and answering procedure, similar to the one used in QAEval, is used to identify salient phrases in the input document, which are then incorporated as an inductive bias into a summarization model. We experimentally show how our QA-based method of identifying salient content results in better end-to-end summaries than lexical baselines and improves the controllability of the content of the summaries, demonstrating the value of QA-based representations of information.

The research presented in this Chapter was originally investigated in Deutsch and Roth (2022b).

6.1. Introduction

Abstractive sequence-to-sequence summarization models have become very effective methods of easily generating summaries of input documents (Rush et al., 2015; Nallapati et al., 2016; Lewis et al., 2020). Previous work has demonstrated that conditioning the summary generation on salient document sentences results in higher-quality summaries and more controllable summarization models (Chen and Bansal, 2018; Dou et al., 2021). Salient sentences are typically identified during training by lexical overlap with the gold summaries (Nallapati et al., 2017) and predicted during inference.

Although marking different sentences as salient allows for some controllability over the content of the summary, desired summary content cannot be specified at the sub-sentence level. Further, labeling sentences as salient via n-gram overlap does not directly take the predicate-argument structure of the text into account, which could result in a lower-quality supervision signal that misidentifies

Figure 20: Salient spans identified by QA-based signals (shown in color) more precisely identify salient document content than those that identify salient sentences based on lexical overlap (shown in bold). Our method classifies the salient spans, marks them in the input document, and then generates a summary.

which particular instance of an n-gram is salient.

In this work, we propose to condition the summary generation on salient sub-sentence level spans which are identified by reasoning about the predicate-argument relations in the text.

We mark noun phrases (NPs) in the input document as salient if the predicate-argument relation they participate in is present in the gold summary (§6.2). This idea is implemented using automatic question generation (QG) and answering (QA). For each NP, a wh-question that is answered by the NP is generated from the text. Then, the NP is marked as salient if the generated wh-question is correctly answered in the gold summary according to a learned QA model, resulting in a more precise, sub-sentence level supervision signal (see Fig. 20).

The QA-based salience signal is incorporated into a two-stage summarization model (§6.3). First, a phrase salience classifier is trained to identify which NPs in the document are salient. Then, the predicted salient spans are marked in the input document with special tokens and used to conditionally generate a summary of the document with a fine-tuned BART model (Lewis et al., 2020).

While we show that marking NPs as salient controls the summary content, the model often outputs extra, undesired information. To that extent, we propose a data augmentation procedure that removes sentences unsupported by any salient span and generates new training examples based on what content should be able to be generated by subsets of the salient spans (§6.4).

Our experiments on three different summarization datasets show that the two-stage model trained with QA-based salient span supervision generates higher-quality summaries than lexical baseline methods of identifying salient spans on more extractive datasets according to several automatic evaluation metrics (§6.6.1). Further, our data augmentation procedure results in summaries that are significantly shorter with only a small reduction in the percent of target content covered, demonstrating it successfully eliminates undesired summary content (§6.6.2).

The contributions of this Chapter include: (1) a novel method of including QA-based signals into summarization generation; (2) a two-stage model for incorporating phrase-level supervision into a summarization system; and (3) a data-augmentation procedure which results in more controllable summarization models.

6.2. Question-Based Salience

We begin by describing how we use QA to identify salient spans of text in the input document and discuss the advantages of this approach.

We define a document NP as salient if its corresponding predicate-argument relation also appears in the gold summary. To identify such NPs automatically, we employ question-generation and question-answering models as follows.

For each NP in the source document, we use the sentence it appears in to automatically generate a wh-question for which the NP is the answer. This QA pair represents the predicate-argument relation that the NP participates in. Then, we assume if a second text can be used to correctly answer that question, it contains the same predicate-argument relation. Thus, we use a QA model to automatically answer the question against the gold summary and mark the NP as salient if the QA model predicts the question is answerable and the predicted answer is correct. In practice, we

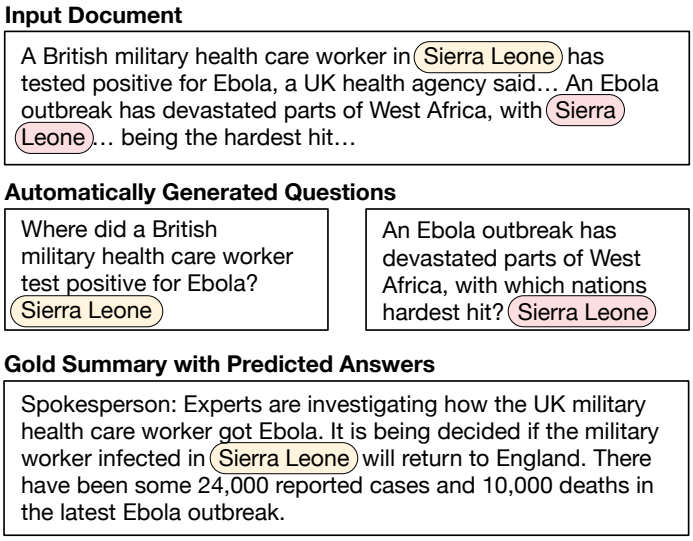

Figure 21: We define a document noun phrase as salient if the wh-question it answers is also answered in the gold summary. Here, the first (yellow) instance of "Sierra Leone" is salient and the second (red) is not because the gold summary answers the automatically generated question for the first instance but not the second.

assume a predicted answer is correct if it shares at least one token in common with the NP which was used to generate the question.

An example of this procedure is illustrated in Fig. 21 for two occurrences of the NP "Sierra Leone." Questions for each phrase are automatically generated from the input document and answered against the gold summary. Since the QA model correctly answered the first question but predicted the second question is not answerable, only the first occurrence of "Sierra Leone" is marked as salient.

We refer to the NPs identified by this procedure as "silver spans." Specific implementation details of the generation and answering models can be found in §6.5.

6.2.1. Advantages of a QA-Based Approach

Using QA to identify salient spans of text has several advantages. First, because our QA approach operates at the phrase-level, it is able to be more precise about what specifically is salient in the document in contrast to sentence-level approaches. For example, in the second sentence of Fig. 20, the QA-based salience signal identifies "Jonathan" and "his defeat" as salient but not "written state-

ment." A sentence-level approach would mark the entire sentence as salient and thus cannot make that distinction.

Second, because the QA-based approach reasons about the predicate-argument structure of the text, it is able to distinguish between which specific instances of the same NP are salient and which are not. This is illustrated in Fig. 21 in which the first occurrence of "Sierra Leone" is marked as salient but the second is not because the gold summary does say the health care worker was infected in Sierra Leone, but it does not say it is one of the hardest hit countries. A salience signal that uses a bag-of-n-grams approach (e.g., ROUGE-based methods) cannot easily decide which instance "Sierra Leone" is salient.

6.3. A Two-Stage, Span-Based Model

Next, we propose a two-stage, span-based model called SPANSUM that can incorporate the QA-based salience signals into the learning procedure. The first of the two stages, the span selection component, classifies salient spans within the text. The second stage, the generation component, generates the summary given the document and the salient spans. The details of each component are detailed next.

6.3.1. Salient Span Classifier

Given an input document $d = [x_1, \ldots, x_n]$ and a set of spans S, in which each span $s_{i,j}$ represents a sequence of tokens $x_i, \ldots, x_j$ in d, the span classifier outputs a score for each span based on how salient it is in the document. Our definition of salience is discussed in §6.2.

Concretely, the input tokens are first encoded using BART. Then, the representation of a span is created by concatenating the BART encodings of the first and last tokens in the span. Finally, a linear classifier is trained using this encoding to predict the salience of each span.

A set of silver spans $S^* \subseteq S$ is used to train the model using a binary cross-entropy loss. When using the QA-based approach, S is the set of NPs in the document and S^* is the subset that our QG-QA algorithm identified as salient. We reweight the loss term of each span such that positive and negative spans contribute equally. During inference, a score is predicted for each span in S and the top-k sorted by highest score are passed to the generation component. We choose the k spans

independently, although they could also be selected jointly.

6.3.2. Generation Component

Given an input document and set of salient spans, the generation component produces a summary of the document. The salient spans are represented by inserting special tokens directly into the document's sequence of tokens before and after the spans. For example, if span $s_{4,5}$ was marked as salient, the document's tokens would be represented as

```
... x3 [SS] x4 x5 [SE] x6 ...
```

where `[SS]` and `[SE]` mark the start and end of the span.

Since the salient spans are represented in the document tokens, we are able to directly train a sequence-to-sequence model to generate the gold summary from the modified document representation without any changes to the model's architecture.

During training, we use silver spans and the ground-truth summary to fine-tune BART using a standard cross-entropy loss function. For inference, the predicted salient spans from the span classifier are used instead of the silver spans.

6.4. Improving Controllability via Data Augmentation

Although there is nothing to directly force the generation model to learn to include content based on the supervision provided by the salient spans, if the supervision is of high enough quality, we expect the model will learn to do so. Indeed, we later show in §6.6.2 that this is true, thus the content of the summary can be controlled by which spans are marked as salient. However, it is also desirable for a controllable summarization model to also not include content which is not marked as salient. The generation models may learn to include extra information for at least two reasons.

First, the gold summaries may include content which cannot be generated based on only the silver salient spans that were used to train the generation model, so it may learn to output extra, unmarked information. This could happen if the QG/QA models are imperfect (resulting in a noisy supervision signal) or if the gold summary contains information that cannot be mapped to the document. Second, if the model is trained to generate summaries of a certain length and the length of the summary

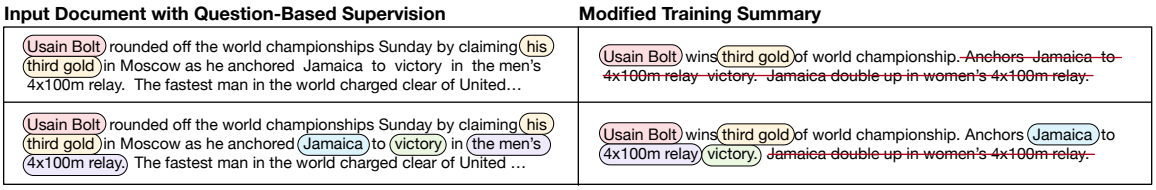

Figure 22: An example of our data augmentation procedure. The colors represent the mapping between document and summary spans. The document spans are given to the generation model during training. In this example, no span maps to the third summary sentence, so it is removed entirely. Then, new training instances are generated using the first summary sentence and first two summary sentences with their corresponding salient document spans.

necessary to include all of the information marked by the spans is smaller than those used for training — for example, because the number of marked spans is small — the model could generate additional information simply to increase the summary length.

An artifact of our silver span annotation procedure enables us to address this controllability issue. If a document span is marked as salient, that means it has a corresponding phrase in the gold summary which expresses the same content. Therefore, the QG-QA procedure creates a mapping between which parts of the gold summary should be able to be generated by marking different parts of the input document.

We propose to leverage this mapping to augment the training data in two ways. First, we remove any gold summary sentence which has no phrase mapped to the document. These sentences would encourage the model to generate additional content based on unmarked spans.

Second, we generate new pairs of salient spans and gold summaries for training by selecting the first k remaining gold summary sentences and the subset of salient document spans which map to them. For instance, if $k = 2$, only the salient spans which are mapped to the first two summary sentences are marked in the input document, and the model is trained to generate only those sentences. We generate new examples for each original training instance using all possible values of k. By training on these new pairs, the model should learn to better control the length of the output summary based on the number of marked salient spans. An example of these augmentations is included in Fig. 22.

Although this procedure is described within the context of the QA-based supervision, it can be

implemented with any such mapping between the document and gold summaries.

6.5. Experimental Setup

Datasets. Our experiments use three popular English single-document summarization datasets: CNN/DailyMail (Nallapati et al., 2016), XSum (Narayan et al., 2018a), and NYTimes (Sandhaus, 2008). Specific details on the sizes of the datasets can be found in Appendix B.1.

Baselines & Other Work. We compare the salient spans selected by our QA-based method against three baseline span selection methods. The first marks salient sentences by greedily selecting k sentences that maximize the ROUGE-2 score calculated against the gold summary, a popular method that is frequently used to train extractive summarization models (Nallapati et al., 2017) as well as other two-step abstractive systems (Chen and Bansal, 2018; Dou et al., 2021). The other two mark entities and NPs as salient if they appear in the gold summaries as determined by lexical matching. We only mark the first occurrence of the phrases as salient since we found that worked better than marking all occurrences.

Additionally, we compare our results to BART (the original implementation and our own; Lewis et al., 2020) since our models are built on top of it. We also compare to GSum (Dou et al., 2021), which uses salient sentence guidance that is similar to our baseline salient sentence method. GSum encodes the additional guidance signal separately from the input document and uses the document and guidance encodings to generate the summary.

Summarization Evaluation Metrics. The models are automatically evaluated using three metrics which calculate a similarity score between the generated and gold summaries. ROUGE (Lin, 2004) compares the two summaries based on their lexical overlap. BERTScore (Zhang et al., 2020) calculates a similarity score between the summaries based on their tokens' BERT embeddings (Devlin et al., 2019). We also evaluate with QAEval (Chapter 4), which generates questions from the gold summaries and answers them against the generated summaries. Its similarity score is equal to the average token F_1 score calculated between the predicted and expected answers.

We additionally perform a human evaluation of summary quality on Mechanical Turk. We ask 3 Turkers to rate the quality of 50 summaries per model from the CNN/DailyMail dataset on a scale

from 1 to 5 based on the importance of the information, faithfulness, fluency, and coherence. Details on the manual evaluation can be found in Appendix B.6.

Controllability Evaluation Metrics. The controllability of our model is evaluated using the *question recall*. Given k marked spans, we define the question recall to be equal to the percent of the corresponding k wh-questions that are answered by the summary according to the QA model. This approximates the recall on the desired predicate-argument structures in the summary. We additionally report the ratio between k and the length of the generated summary in tokens to measure the precision of the generated information. A larger value means the summary is more concise.

Implementation Details. The QG/QA models are the same as used by QAEval. The generation model is initialized with BART-Large and fine-tuned on data collected by Demszky et al. (2018). The answering model is initialized with ELECTRA-Large (Clark et al., 2020) and fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018).

The span classification and generation models are both initialized with BART-Large and fine-tuned on the respective datasets. They were trained for three and five epochs, respectively, and the model with the best precision@1 and ROUGE-2 F_1, respectively, on the validation set were selected as the final models. See Appendix B.2 for more specific implementation details.

6.6. Results

6.6.1. Summarization Evaluation

Automatic Evaluation. Table 14 contains the models' performances as evaluated by automatic metrics, both using the spans predicted by the classifier ("end-to-end") and the silver spans (i.e., assuming a "perfect" classifier).

Interestingly, we find a somewhat surprising result. On CNN/DailyMail and NYTimes, the end-to-end QA-based model performs the best among the different span labeling methods and the baseline BART. On NYTimes, it is also better than GSum. However, if the silver span labels are used, the lexical NP-based model out-performs the rest by a somewhat large margin. It is surprising that a seemingly better generation model would result in worse end-to-end performance.

Method	CNN/DailyMail					NYTimes				
	R1	**R2**	**RL**	**BSc**	**QAE**	**R1**	**R2**	**RL**	**BSc**	**QAE**
Baselines & Other Work										
BART	44.2	21.3	40.9	-	-	-	-	-	-	-
BART (ours)	44.1	21.0	40.9	88.3	23.5	54.0	35.2	50.7	89.5	27.3
GSum	46.0†	22.3†	42.6†	88.6†	22.9	54.3	35.4	47.6	-	-
Silver Spans										
Sentences	51.7	29.9	48.8	89.4	28.6	62.7	46.0	59.8	91.2	33.5
Entities	51.5	27.6	48.0	89.6	30.0	60.9	42.8	57.6	90.8	32.0
Lexical NPs	**59.6**	**34.6**	**55.8**	**90.6**	**36.2**	**68.2**	**50.7**	**64.8**	**92.0**	**36.6**
QAs	55.3	31.4	51.9	90.0	33.7	65.7	48.7	62.6	91.6	35.8
End-to-End										
Sentences	45.0	**21.8**	41.8	88.2	23.2	54.6	35.9	51.4	89.6	27.6
Entities	43.5	20.3	40.4	88.3	23.2	53.5	34.6	50.3	89.4	27.0
Lexical NPs	44.8	21.0	41.6	88.4	23.2	54.6	35.4	51.3	89.6	27.1
QAs	**45.5**	**21.9**	**42.4**†	**88.5**	**24.4**†	**55.2**†	**36.3**†	**51.9**†	**89.7**†	**28.0**†

Table 14: The automatic metric results for the baselines and other work (top), models that use silver spans (middle), and end-to-end models (bottom) evaluated with ROUGE (R1, R2, RL), BERTScore (BSc), and QAEval (QAE). Values in bold are statistically the best in each section and † marks the best values overall (excluding silver labels) using a permutation test with $\alpha = 0.05$.

To better understand this result, we manually labeled all of the NPs in 50 CNN/DailyMail documents as salient or not salient based on whether the corresponding predicate-argument relation was present in the reference summary (see Appendix B.4 for details). These spans, which we call the gold spans, can be used to evaluate the precision and recall of the silver spans as well as the output from the salient span classifiers.

Table 15 shows that the QA-based labels are more precise but have lower recall than the lexical NP labels. Because the lexical NP method aggressively marks the first occurrence of any NP in the document which is present in the reference as salient, it is unsurprising that its recall would be high. Since it cannot distinguish between instances of the same NP due to its bag-of-words representation, its precision is low. In contrast, the QA-based approach can reason about which occurrence of an NP is salient (resulting in higher precision), but the recall is lower likely due to noise in the QG/QA models. This same pattern appears in Table 15 with the outputs from the salient span classifiers, although the precisions and recalls are notably lower than the silver span labels'.

Method	Precision	Recall	F_1
Silver Labels			
Lexical NPs	32.7	**66.3**	41.8
QAs	**43.8**	51.5	**45.3**
Predicted Spans			
Lexical NPs@25	23.8	**54.4**	**32.0**
QAs@20	**27.3**	49.1	**33.8**

Table 15: The average summary-level precision, recall, and F_1 scores of the silver labeling methods (top) and the output from the span classifiers (bottom) evaluated against the human-annotated gold labeling. Results in bold are statistically higher (or tied) under a single-tail pairwise permutation test with $\alpha = 0.05$. The @k values were selected based on validation set performance.

We believe that the discrepancy between the end-to-end and silver span-based models' performances can be explained by these results. The lexical NP generation model was trained with a high recall silver supervision at 66.3, allowing the generation model to achieve good performance when the silver spans are provided. Yet during inference the model is provided with spans that only have around 54.4 recall, 12 points lower. We suspect the generation model learned to rely heavily on the marked salient spans — and empirically we observed that it copied very heavily from them — thus when the quality of the span signal was reduced, the resulting summaries similarly got worse. In contrast, the difference between the QA-based model's recall during training and inference is only estimated to be around 2.4, so this issue is less severe, resulting in better end-to-end summaries.

To test this hypothesis, we artificially ablated the lexical NP-based generation model's silver span supervision's recall by removing $k\%$ of the salient spans uniformly at random — thus making the training spans look more similar to the spans during inference — and retrained the model. We would expect the silver span-based model's performance to decrease while the end-to-end model's increases. Indeed, in Table 16 we find that this does happen. These results suggest that the relationship between the classifier's performance and generation model's supervision is important for good end-to-end results and could be explored in future work.

Although the end-to-end lexical NP results begin to approach the QA-based model's performance, they do not quite reach it. Further, the QA-based silver spans maintain an F_1 advantage over the lexical NP method (Table 15). While the QA-based approach can be improved with better question

Method	CNN/DailyMail		
	R1	**R2**	**RL**
Silver Spans			
Lexical NPs	**59.6**	**34.6**	**55.8**
+10% Noise	57.8	32.8	54.0
+20% Noise	56.3	31.5	52.6
+30% Noise	55.0	30.4	51.4
+35% Noise	54.1	29.6	50.6
QAs	55.3	31.4	51.9
End-to-End			
Lexical NPs	44.8	21.0	41.6
+10% Noise	45.0	21.3	41.8
+20% Noise	45.2	21.6	42.0
+30% Noise	45.3	21.7	42.1
+35% Noise	45.1	21.6	41.9
QAs	**45.5**	**21.9**	**42.4**

Table 16: The ablated lexical NP supervision shows as the noise increases, the silver span performance decreases but end-to-end performance improves.

generation and answering models, the lexical NP labeling method is inherently limited. Therefore, the QA-based method does appear to be the best method of incorporating additional supervision into the summarization models based on the automatic metrics.

XSum. Table 17 contains the automatic summarization evaluation results on the XSum dataset. Unlike for CNN/DailyMail and NYTimes, we see that incorporating the span-based supervision does not improve end-to-end results over the baseline BART model. This is also a conclusion reached by GSum in their experiments.

We suspect this is due to the abstractive nature of XSum compared to the more extractive NYTimes and CNN/DailyMail. Since the methods for identifying salient spans rely on the document and gold summary explicitly stating the salient content, we suspect the abstractiveness of XSum would result in this happening less frequently and thus be less beneficial to a summarization model trained on XSum. This would impact our model as well as GSum equally.

Human Evaluation. Table 18 contains the results of evaluating BART and the sentence- and QA-based models on CNN/DailyMail (the best performing) using human summary quality annotations

Method	XSum				
	R1	**R2**	**RL**	**BSc**	**QAE**
Baselines & Other Work					
BART	45.1	21.3	40.9	-	-
BART (ours)	45.7†	22.4†	37.2†	91.3†	18.9†
GSum	44.9	21.2	36.0	90.4	17.9
Silver Spans					
Sentences	47.3	24.2	38.7	91.5	19.9
Entities	48.1	24.2	39.1	91.7	21.3
Lexical NPs	**54.3**	**29.3**	**44.1**	**92.4**	**26.1**
QAs	47.9	24.1	39.2	91.6	21.4
End-to-End					
Sentences	**45.0**	**21.7**	**36.6**	**91.2**	**18.6**
Entities	44.1	20.9	35.9	91.0	17.6
Lexical NPs	42.5	19.2	34.2	90.8	16.4
QAs	**45.1**	**21.8**	**36.7**	91.2	17.9

Table 17: The results of the models trained on the XSum dataset as evaluated with the automatic evaluation metrics. The span-based models do not improve over the baseline BART, potentially due to the abstractive nature of the XSum dataset.

	BART	**Sentences**	**QA**
Quality Score	3.76	**3.86**	**4.00**

Table 18: Summary quality scores according to humans. Results in bold are statistically tied for the best score.

from Mechanical Turk. On average, our span-based methods have higher quality summaries than the baseline method of BART. After collecting annotations for 50 summaries on CNN/DailyMail, we were unable to obtain statistical significance between the two span-based models, however, doing so may be prohibitively difficult (Wei and Jia, 2021).

6.6.2. Controllability Evaluation

Automatic Evaluation. The controllability of the QA-based generation model is evaluated in Fig. 23 using the original training data as well as the augmented data described in §6.4. We plot the question recall and the ratio between k and the length of the generated summaries for the top k most salient spans output by the QA-based salient span classifier for various values of k on CNN/DailyMail. The data augmentation procedure is split into only removing sentences that do

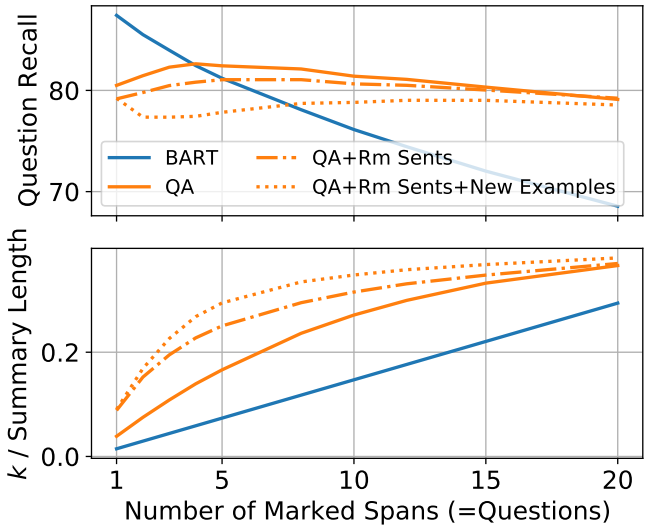

Figure 23: The percent of questions which correspond to the marked spans answered by the generated summaries (top) and the summary lengths in tokens (bottom). The QA methods have higher question recall than BART and are far more concise, demonstrating that marking input spans controls the summary content.

not answer a question ("+Rm Sents") plus also generating new training examples ("+New Examples"). We also include the results for BART (for which the summary is constant for all k) for relative comparisons.

Although BART's question recall is initially higher than the QA models' recalls, as k increases it falls lower. We suspect this is because BART has learned to include the same content that the span classifier also identifies as salient when k is small and the length of its summaries allows it to cover more content. However, when k increases, the span classifier potentially predicts different spans as salient than what BART learned, resulting in divergent content and a lower recall for BART. The higher recall of the QA models demonstrates that their summary content is indeed being controlled via the input spans. Further, the QA models have far better k-to-length ratios, meaning their summaries are shorter than BART's even when their recalls are higher, suggesting they generate far less content which is unrelated to the marked spans.

Among the QA-based models, we do observe a small drop in recall when the model is trained with data augmentation. However, the data-augmented summaries express that information far more

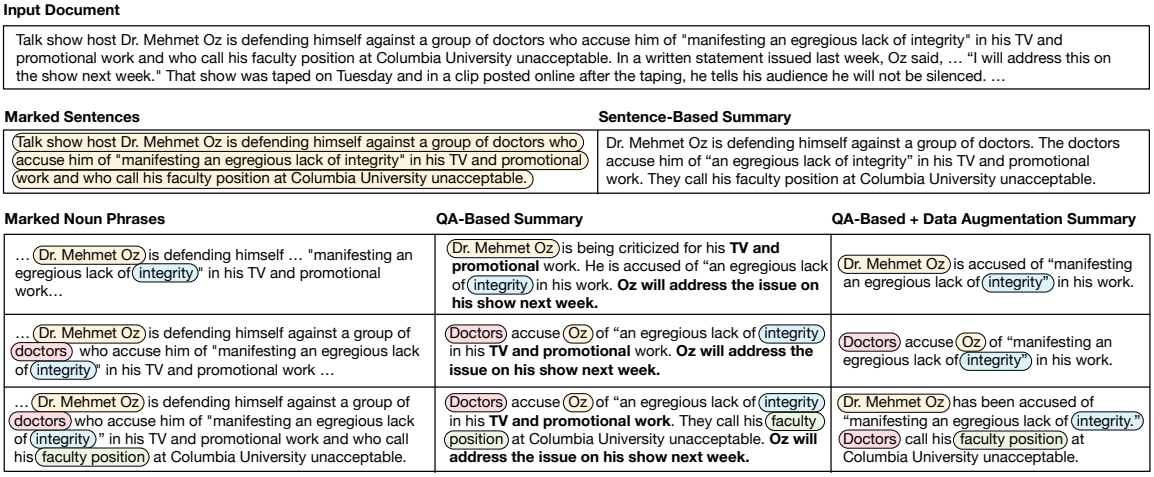

Figure 24: Example summaries generated by the sentence-based model (middle), QA-based model (bottom center) and QA-based model trained on the augmented data (bottom right). The QA-based models allow for much more control over the summary content than the sentence model by marking different combinations of phrases. The augmented-data summaries better eliminate unmarked content from the input than the standard model (extra information generated by the standard model shown in bold).

concisely (because the ratio between k and the summary length is higher). For example, when 10 input spans are marked, there is a relative 0.9% and 3.2% drop in recall for removing sentences and the full data augmentation procedure, respectively, but the summary lengths are 14% and 22% shorter. Therefore, the data augmentation procedures do result in models which have learned to not generate extra content.

Controllability Example. Example summaries from the QA models and sentence-based model with different marked input spans are shown in Fig. 24. Because the sentence-based model is limited to marking full sentences, the content which is taken from the marked sentence cannot be further controlled. In contrast, the figure shows how the QA models' summaries can be altered by marking different NPs within the sentence, thus demonstrating the benefits of phrase-level controllability.

The example in Fig. 24 also shows how the data augmentation procedure improves controllability. The phrases which the standard model includes but the augmented model does not are marked in bold. The augmented model does a better job at excluding content which was not marked in the input document.

QA-Based Signals. QA-based signals have been used for evaluating summaries (Eyal et al., 2019; Durmus et al., 2020; Wang et al., 2020; Deutsch et al., 2021a), including Scialom et al. (2021), who explore a similar notion of document salience. They have also been used to align content across documents (Weiss et al., 2021) as well as train summarization models (Arumae and Liu, 2018, 2019; Scialom et al., 2019). The models which incorporate QA-based signals typically do so using reinforcement learning. In contrast, our approach is simpler. We incorporate the QA-based signal by marking spans in the document, and our models are trained using easier-to-optimize cross-entropy objective functions.

Incorporating Additional Supervision. Recent work by Dou et al. (2021) proposes a framework for incorporating additional guidance into summarization models, called GSum. They separately encode the input document and the supervision signal, whereas we directly mark spans in the text. This allows for our generation component to have a simpler architecture than theirs. While they are able to encode any natural language string, our model provides more direct supervision by identifying which specific tokens are salient.

Other work has included predicate-argument structure into summarization but with the goal of producing more faithful summaries (Cao et al., 2018; Jin et al., 2020; Zhu et al., 2021). They represent the predicate-arguments either using dependency trees or OpenIE tuples, whereas we represent them via QA pairs. These works include that information to try and generate faithful summaries, whereas our goal is to identify salient document content.

Controllable Summarization. Work on controllable summarization has focused on aspects such as the length of the summary (Fan et al., 2018) and the content in an interactive setting (Shapira et al., 2017) or via prompting (He et al., 2020). Incorporating our QA-based signal via prompting may be difficult given the number of questions which would need to be concatenated onto the input.

Other approaches control content via planning as in entity templates (Narayan et al., 2021) or marking records in a data-to-text approach (Puduppully et al., 2019). The marked salient spans in our work could be viewed as a content plan as well.

Data Augmentation. Previous work has proposed methods for removing sentences or full summaries from the training data in order to discourage the summarization model from learning to generate unfaithful information (Matsumaru et al., 2020; Nan et al., 2021; Narayan et al., 2021). In addition to removing sentences, we generate new training instances in order to learn to exclude content which is not marked as salient in the input, resulting in more controllable models.

6.8. Summary

In this Chapter, we proposed a method for incorporating QA-based signals into a summarization model by automatically marking document NPs as salient based on whether a NP's corresponding wh-question is answered correctly in the summary. We showed that incorporating this signal into our two-stage summarization model results in higher quality summaries than baseline methods of identifying salient spans that are based on lexical matching and do not reason about the predicate-argument structure of the text. This provides evidence that the higher-level QA-based signals do indeed provide a better inductive bias for summarization systems. Finally, we demonstrated that our data augmentation algorithm, which is applied on top of the QA-based supervision, improves controllability by eliminating unmarked content from the output summaries.

CHAPTER 7 : Resampling Methods for Metric Meta-Evaluation

In Chapter 4 of this thesis, we meta-evaluated several summarization evaluation metrics and argued that certain metrics had stronger correlations to human judgments than others by a statistically significant margin. Using hypothesis testing to support claims of differences between correlations is critically important for understanding how likely it is that the observed result can be explained by a real difference between the two metrics rather than a result obtained by random chance.

In this Chapter, we discuss our proposals of the permutation tests that were used to compare the metrics' correlations in Chapter 4 as well as some closely related bootstrapping methods that can be used to better understand the true strength of the metrics' correlations. Because hypothesis testing and confidence interval estimation is largely not performed for summarization evaluation metrics, our proposals provide new insights into the metrics' performances which were not previously known. This includes the result that the community is very uncertain how well automatic metrics truly correlate to human judgments of summary quality, which has wider implications for how much we should trust conclusions reached by automatic evaluations.

The work discussed in this Chapter was originally published in Deutsch et al. (2021b).

7.1. Introduction

The methods we propose for improving the methodology for meta-evaluating metrics are based on the resampling techniques of bootstrapping (Efron and Tibshirani, 1994) and permutation (Noreen, 1989). Resampling techniques are advantageous because, unlike parametric methods, they do not make assumptions which are invalid in the case of summarization (§7.3.1; §7.4.1). Bootstrapping and permutation techniques use a subroutine that samples a new dataset from the original set of observations. Since the correlation of an evaluation metric to human judgments is a function of *matrices* of values (namely the metric's scores and human annotations for multiple systems across multiple input texts; §7.2), this subroutine must sample new *matrices* in order to generate a new instance, in contrast to standard applications of bootstrapping and permutation that sample vectors of numbers. To that end, we propose three different bootstrapping (§7.3.2) and permutation (§7.4.2)

techniques for resampling matrices, each of which makes different assumptions about whether the systems or inputs are constant or variable in the calculation.

In order to evaluate which resampling methods are most appropriate for summarization, we perform two simulations. The first demonstrates that the bootstrapping resampling technique which assumes both the systems and inputs are variable produces CIs that generalize best to held-out data (§7.5.1). The second shows that the permutation test which makes the same assumption has more statistical power than the equivalent bootstrapping method and Williams' test (Williams, 1959), a parametric hypothesis test that is popular in machine translation (§7.5.2).

Finally, we analyze the results of estimating CIs and applying hypothesis testing to a set of summarization metrics using annotations on English single- and multi-document datasets (Dang and Owczarzak, 2008; Fabbri et al., 2021; Bhandari et al., 2020). We find that the CIs for the metrics' correlations are all rather wide, indicating that the summarization community has relatively low certainty in how similarly automatic metrics rank summaries with respect to humans (§7.6.1). Additionally, the hypothesis tests reveal that our metric, QAEval (see Chapter 4), and BERTScore (Zhang et al., 2020) emerge as the best metrics in several of the experimental settings, whereas no other metric consistently achieves statistically better performance than ROUGE (§7.6.2; Lin, 2004).

Although we focus on summarization, the techniques we propose can be applied to evaluate automatic evaluation metrics in other text generation tasks, such as machine translation or structure-to-text. The contributions of this Chapter include (1) a proposal of methods for calculating CIs and running hypothesis tests for summarization metrics, (2) simulation experiments that provide evidence for which methods are most appropriate for summarization, and (3) an analysis of the results of the statistical analyses applied to various summarization metrics on three datasets.

7.2. Preliminaries: Evaluating Metrics

The details of how metrics are meta-evaluated are described in §2.3, but the procedure is summarized again in this Section with some additional formalism that is required for this Chapter.

Summarization evaluation metrics are typically used to either argue that a summarization system

generates better summaries than another or that an individual summary is better than another for the same input. How similarly an automatic metric does these two tasks with respect to humans is quantified as follows.

Let $\mathcal{X}$ be an evaluation metric that is used to approximate some ground-truth metric $\mathcal{Z}$. For example, $\mathcal{X}$ could be ROUGE and $\mathcal{Z}$ could be a human-annotated summary quality score. The similarity of $\mathcal{X}$ and $\mathcal{Z}$ is evaluated by calculating two different correlation terms on a set of summaries. First, the summaries from summarization systems $\mathcal{S} = \{S_1, \ldots, S_N\}$ on input document(s) $\mathcal{D} = \{D_1, \ldots, D_M\}$ are scored using $\mathcal{X}$ and $\mathcal{Z}$. We refer to these scores as matrices $X, Z \in \mathbb{R}^{N \times M}$ in which x_i^j and z_i^j are the scores of $\mathcal{X}$ and $\mathcal{Z}$ on the summary output by system S_i on input D_j. Then, the correlation between X and Z is calculated at one of the following levels:

$$r_{\text{SYS}}(X, Z) = \textsc{Corr}\left(\left\{\left(\frac{1}{M}\sum_j x_i^j, \frac{1}{M}\sum_j z_i^j\right)\right\}_{i=1}^{N}\right) \tag{7.1}$$

$$r_{\text{SUM}}(X, Z) = \frac{1}{M}\sum_j \textsc{Corr}\left(\left\{\left(x_i^j, z_i^j\right)\right\}_{i=1}^{N}\right) \tag{7.2}$$

where $\textsc{Corr}(\cdot)$ typically calculates the Pearson, Spearman, or Kendall correlation coefficients.[12]

These two correlations quantify how similarly $\mathcal{X}$ and $\mathcal{Z}$ score systems and individual summaries per-input for systems $\mathcal{S}$ and documents $\mathcal{D}$. The system-level correlation r_{SYS} calculates the correlation between the scores for each system (equal to the average score across inputs), and the summary-level correlation r_{SUM} calculates an average of the correlations between the scores per-input.

The correlations r_{SYS} and r_{SUM} are also used to reason about whether $\mathcal{X}$ is a better approximate of $\mathcal{Z}$ than another metric $\mathcal{Y}$ is, typically by showing that $r(X, Z) > r(Y, Z)$ for either r.

[12]For clarity, we will refer to r_{SUM} and r_{SYS} as correlation levels and Pearson, Spearman, and Kendall as correlation coefficients.

7.3. Correlation Confidence Intervals

Although the strength of the relationship between $\mathcal{X}$ and $\mathcal{Z}$ on one dataset is quantified by the correlation levels r_{SYS} and r_{SUM}, each r is only a point estimate of the true correlation of the metrics, denoted ρ, on inputs and systems distributed similarly to those in $\mathcal{D}$ and in $\mathcal{S}$. Although we cannot directly calculate ρ, it is possible to estimate it through a CI.

7.3.1. The Fisher Transformation

The standard method for calculating a CI for a correlation is the Fisher transformation (Fisher, 1992). The transformation maps a correlation coefficient to a normal distribution, calculates the CI on the normal curve, and applies the reverse transformation to obtain the upper and lower bounds:

$$z_r = \text{arctanh}(r) \tag{7.3}$$

$$r_u, r_\ell = \tanh\left(z_r \pm z_{\alpha/2} \cdot c \,/\sqrt{n-b}\right) \tag{7.4}$$

where r is the correlation coefficient, n is the number of observations, $z_{\alpha/2}$ is the critical value of a normal distribution, and b and c are constants.[13]

Applying the Fisher transformation to calculate CIs for ρ_{SYS} and ρ_{SUM} is potentially problematic. First, it assumes that the input variables are normally distributed (Bonett and Wright, 2000). The metrics' scores and human annotations on the datasets that we experiment with are, in general, not normally distributed (see Appendix C.1). Thus, this assumption is violated, and we expect this is the case for other summarization datasets as well. Second, it is not clear whether the transformation should be applied to the summary-level correlation since its final value is an average of correlations, which is not strictly a correlation.[14]

7.3.2. Bootstrapping

A popular nonparametric method of calculating a CI is bootstrapping (Efron and Tibshirani, 1994). Bootstrapping is a procedure that estimates the distribution of a test statistic by repeatedly sam-

[13] $b = 3, 3, 4$ and $c = 1, \sqrt{1 + r^2/2}, \sqrt{.437}$ for Pearson, Spearman, and Kendall, respectively (Bonett and Wright, 2000).

[14] Correlation coefficients cannot be averaged because they are not additive in the arithmetic sense, however it is standard practice in summarization.

pling with replacement from the original dataset and calculating the test statistic on each sample. Unlike the Fisher transformation, bootstrapping is a very flexible procedure that does not assume the data is normally distributed nor that the test statistic is a correlation, making it appropriate for summarization.

However, it is not clear how to perform bootstrap sampling for correlation levels. Consider a more standard bootstrapped CI calculation for the mean accuracy of a question-answering model on a dataset with k instances. Since the mean accuracy is a function of the k individual correct/incorrect labels, each bootstrap sample can be constructed by sampling with replacement from the original k instances k times. In contrast, the correlation levels are functions of the matrices X and Z, so each bootstrap sample should also be a pair of matrices of the same size that are sampled from the original data.

There are at least three potential methods for sampling the matrices:

1. **BOOT-SYSTEMS:** Randomly sample with replacement N systems from $\mathcal{S}$, then select the sampled system scores for all of the inputs.

2. **BOOT-INPUTS:** Randomly sample with replacement M inputs from $\mathcal{D}$, then select all of the system scores for the sampled inputs.

3. **BOOT-BOTH:** Randomly sample with replacement M inputs from $\mathcal{D}$ and N systems from $\mathcal{S}$, then select the sampled system scores for the sampled inputs.

Once the samples are taken, the corresponding values from X and Z are selected to create the sampled matrices. An illustration of each method is shown in Figure 25.

Each sampling method makes its own assumptions about the degrees of freedom in the sampling process that results in different interpretations of the corresponding CIs. BOOT-INPUTS assumes that there is only uncertainty on the inputs while the systems are held constant. CIs derived from this sampling technique would express a range of values for the true correlation ρ between $\mathcal{X}$ and $\mathcal{Z}$ for the *specific* set of systems $\mathcal{S}$ and inputs from the same distribution as those in $\mathcal{D}$. The opposite

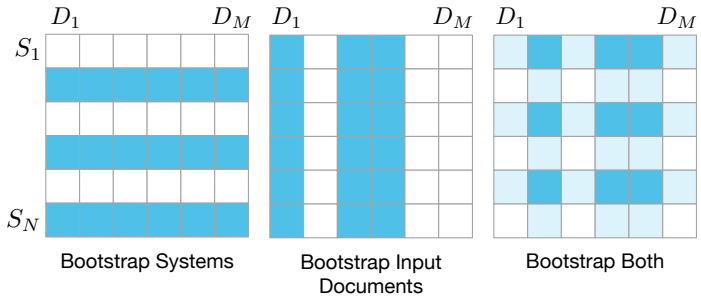

Figure 25: An illustration of the three methods for sampling matrices during bootstrapping. The dark blue color marks values selected by the sample. Only 3 system and input samples are shown here, when N and M are actually sampled with replacement.

Algorithm 1 BOOT-BOTH Confidence Interval

Input: $X, Z \in \mathbb{R}^{N \times M}, k \in \mathbb{N}, \alpha \in [0, 1]$
Output: $(1 - \alpha) \times 100\%$-confidence interval

samples $\leftarrow$ an empty list
for k iterations **do**
    $S \leftarrow$ samp. $\{1, \ldots, N\}$ w/ repl. N times
    $D \leftarrow$ samp. $\{1, \ldots, M\}$ w/ repl. M times
    $X_s, Z_s \leftarrow$ empty $N \times M$ matrices
    for $(i, j) \in \{1, \ldots, N\} \times \{1, \ldots M\}$ **do**
        $X_s[i, j] \leftarrow X[S[i], D[j]]$
        $Z_s[i, j] \leftarrow Z[S[i], D[j]]$
    end for
    Append $r(X_s, Z_s)$ to samples
end for
$\ell, u \leftarrow (\alpha/2) \times 100$ and $(1 - \alpha/2) \times 100$ percentiles of samples
return ℓ, u

assumption is made for BOOT-SYSTEMS (uncertainty in systems, inputs are fixed). BOOT-BOTH, which can be viewed as sampling systems followed by sampling inputs, assumes uncertainty on both the systems and the inputs. Therefore the corresponding CI estimates ρ for systems and inputs distributed the same as those in $\mathcal{S}$ and $\mathcal{D}$.

Algorithm 1 contains the pseudocode for calculating a CI via bootstrapping using the BOOT-BOTH sampling method. In §7.5.1 we experimentally evaluate the Fisher transformation and the three bootstrap sampling methods, then analyze the CIs of several different metrics in §7.6.1.

7.4. Significance Testing

Although CIs express the strength of the correlation between two metrics, they do not directly express whether one metric $\mathcal{X}$ correlates to another $\mathcal{Z}$ better than $\mathcal{Y}$ does due to their shared de-

pendence on $\mathcal{Z}$. This statistical analysis is performed by hypothesis testing. The specific one-tailed hypothesis test we are interested in is:

$$H_0 : \rho(\mathcal{X}, \mathcal{Z}) - \rho(\mathcal{Y}, \mathcal{Z}) \leq 0 \tag{7.5}$$

$$H_1 : \rho(\mathcal{X}, \mathcal{Z}) - \rho(\mathcal{Y}, \mathcal{Z}) > 0 \tag{7.6}$$

7.4.1. Williams' Test

One method for hypothesis testing the difference between two correlations with a dependent variable that is used frequently to compare machine translation metrics is Williams' test (Williams, 1959). It uses the pairwise correlations between X, Y, and Z to calculate a t-statistic and a corresponding p-value. The t-statistic is calculated as the following:

$$t = (r_{XZ} - r_{YZ}) \cdot \sqrt{\frac{(n-1)(1+r_{XY})}{2K\left(\frac{n-1}{n-3}\right) + \left(\frac{r_{XZ}+r_{YZ}}{2}\right)^2 (1-r_{XY})^3}} \tag{7.7}$$

with $n-3$ degrees of freedom, where n is the number of pairs used to calculate the correlations, r_{XY}, r_{YZ}, and r_{XZ} are the sample correlations between the respective matrices, and K is defined as the determinant of the matrix of pairwise correlations:

$$K = \begin{vmatrix} 1 & r_{XY} & r_{XZ} \\ r_{XY} & 1 & r_{YZ} \\ r_{XZ} & r_{YZ} & 1 \end{vmatrix} \tag{7.8}$$

$$= 1 - {r_{XY}}^2 - {r_{XZ}}^2 - {r_{YZ}}^2 + 2 r_{XY} r_{XZ} r_{YZ} \tag{7.9}$$

Williams' test is frequently used to compare machine translation metrics' performances at the system-level (Mathur et al., 2020b, among others).

However, the test faces the same issues as the Fisher transformation: It assumes the input variables are normally distributed (Dunn and Clark, 1971), and it is not clear whether the test should be applied at the summary-level.

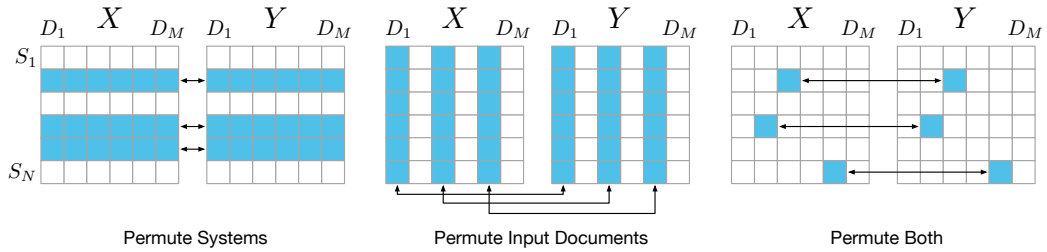

Figure 26: An illustration of the three permutation methods which swap system scores, document scores, or scores for individual summaries between X and Y.

7.4.2. Permutation Tests

Bootstrapping can be used to calculate a p-value in the form of a paired bootstrap test in which the sampling methods described in §7.3.2 can be used to resample new matrices from X, Y, and Z in parallel. However, an alternative and closely related nonparametric hypothesis test is the permutation test (Noreen, 1989). Permutation tests tend to be used more frequently than paired bootstrap tests for hypothesis testing because they directly test whether any observed difference between two values is due to random chance. In contrast, paired bootstrap tests indirectly reason about this difference by estimating the variance of the test statistic.

Similarly to bootstrapping, a permutation test applied to two paired samples estimates the distribution of the test statistic under H_0 by calculating its value on new resampled datasets. In contrast to bootstrapping, the resampled datasets are constructed by randomly permuting which sample each observation in a pair belongs to (i.e., resampling without replacement). This relies on assuming the pair is exchangeable under H_0, which means H_0 is true for either sample assignment for the pair. Then, the p-value is calculated as the proportion of times the test statistic across all possible permutations is greater than the observed value. A significant p-value implies the observed test statistic is very unlikely to occur if H_0 were true, resulting in its rejection. In practice, calculating the distribution of H_0 across all possible permutations is intractable, so it is instead estimated on a large number of randomly sampled permutations.[15]

For example, a permutation test applied to testing the difference between two QA models' mean

[15]This is known as an approximate randomization test.

Algorithm 2 Permutation Hypothesis Test

Input: $X, Y, Z \in \mathbb{R}^{N \times M}$, $k \in \mathbb{N}, \alpha \in [0, 1]$ **Output:** p-value

Standardize X and Y
c $\leftarrow 0$
$\delta \leftarrow r(X, Z) - r(Y, Z)$
for k iterations **do**
    $X_s, Y_s \leftarrow$ empty $N \times M$ matrices
    for $(i, j) \in \{1, \ldots, N\} \times \{1, \ldots, M\}$ **do**
        if random Boolean is true **then** ▷ swap
            $X_s[i, j] \leftarrow Y[i, j]$
            $Y_s[i, j] \leftarrow X[i, j]$
        else ▷ do not swap
            $X_s[i, j] \leftarrow X[i, j]$
            $Y_s[i, j] \leftarrow Y[i, j]$
        end if
    end for
    $\delta_s \leftarrow r(X_s, Z) - r(Y_s, Z)$
    if $\delta_s > \delta$ **then**
        $c \leftarrow c + 1$
    end if
end for
return c/k

accuracies on the same dataset would sample a permutation by swapping the models' outputs for the same input. Under H_0, the models' mean accuracies are equal, so randomly exchanging the outputs is not expected to change their means. In the case of evaluation metrics, each permutation sample can be taken by randomly swapping the scores in X and Y. There are at least three ways of doing so:

1. **PERM-SYSTEMS**: For each system, swap its scores for all inputs with probability 0.5.

2. **PERM-INPUTS**: For each input, swap its scores for all systems with probability 0.5.

3. **PERM-BOTH**: For each summary, swap its scores with probability 0.5.

To account for differences in scale, we standardize X and Y before performing the permutation. Fig. 26 contains an illustration of each method, and the pseudocode for a permutation test using the PERM-BOTH method is provided in Alg. 2.

Similarly to the bootstrap sampling methods, each of the permutation methods makes assumptions about the system and input document underlying distribution. This results in different interpretations

of how the tests' conclusions will generalize. Since PERM-SYSTEMS randomly assigns system scores for all documents in $\mathcal{D}$ to either sample, we only expect the test's conclusion to generalize to a system distributed similarly to those in $\mathcal{S}$ evaluated on the *specific* set of documents $\mathcal{D}$. The opposite is true for PERM-INPUTS. The results for PERM-BOTH (which can be viewed as first swapping systems followed by swapping inputs) are expected to generalize for both systems and documents distributed similarly to those in $\mathcal{S}$ and $\mathcal{D}$.

In §7.5.2 we run a simulation to compare the different hypothesis testing approaches, then analyze the results of hypothesis tests applied to summarization metrics in §7.6.2.

7.5. Simulation Experiments

We run two sets of simulation experiments in order to determine which CI (§7.5.1) and hypothesis test (§7.5.2) methods are most appropriate for summarization metrics.

The datasets used in the simulations are the multi-document summarization dataset TAC'08 (Dang and Owczarzak, 2008) and two subsets of the single-document summarization CNN/DM dataset (Nallapati et al., 2016) annotated by Fabbri et al. (2021) and Bhandari et al. (2020). These datasets have $N = 58/16/25$ summarization models and $M = 48/100/100$ inputs, respectively. The summaries were assigned overall responsiveness, relevance, or Lightweight Pyramid (Shapira et al., 2019) scores, respectively, by human annotators. The scores of the automatic metrics are correlated to these human annotations.

7.5.1. Confidence Interval Simulation

In practice, evaluation metrics are almost always used to score summaries produced by systems $\mathcal{S}'$ on inputs $\mathcal{D}'$ which are disjoint (or nearly disjoint) from and assumed to be distributed similarly to the data that was used to calculate the CI, $\mathcal{S}$ and $\mathcal{D}$. It is still desirable to use the CI as an estimate of the correlation of a metric on $\mathcal{S}'$ and $\mathcal{D}'$, however this scenario violates assumptions made by some of the bootstraping sampling methods (e.g., BOOT-SYSTEMS assumes that $\mathcal{D}$ is fixed). This simulation aims to demonstrate the effect of violating these assumptions on the accuracy of the CIs.

Setup. The simulation works as follows. The systems $\mathcal{S}$ and inputs $\mathcal{D}$ are each randomly partitioned into two equally sized disjoint sets $\mathcal{S}_A$, $\mathcal{S}_B$, $\mathcal{D}_A$, and $\mathcal{D}_B$. Then the submatrices X_A, Z_A, X_B,

CI Method	TAC'08		Fabbri et al.		Bhandari et al.	
	ρ_{SYS}	ρ_{SUM}	ρ_{SYS}	ρ_{SUM}	ρ_{SYS}	ρ_{SUM}
Fisher	0.72	1.00	0.87	1.00	0.85	1.00
BOOT-SYSTEMS	0.76	0.72	0.81	0.73	0.80	0.72
BOOT-INPUTS	0.58	0.70	0.70	0.73	0.68	0.62
BOOT-BOTH	**0.82**	**0.92**	**0.98**	**0.93**	**0.94**	**0.88**

Table 19: The proportion of times the 95% confidence interval for the true correlations ρ of QAEval-F_1 calculated using Pearson contains the sample correlation of a held-out set of systems and inputs for the different methods of calculating confidence intervals. Values in bold are closest to 0.95 (and less than 1.0) and significantly different under a one-tailed difference of proportions z-test at $\alpha = 0.05$.

and Z_B are selected from X and Z based on the system and input partitions. Matrices X_A and Z_A are used to calculate a 95% CI using one of the methods described in §7.3, and then it is checked whether sample correlation $r(X_B, Z_B)$ is contained by the CI. The entire procedure is repeated 1000 times, and the proportion of times the CI contains the sample correlation is calculated.

It is expected that a CI which generalizes well to the held-out data should contain the sample correlation 95% of the time under the assumption that the data in A and B is distributed similarly. The larger the difference from 95%, the worse the CI is at estimating the correlation on the held-out data.

The results of the simulation calculated on TAC'08 and CNN/DM using both the Fisher transformation and the different bootstrap sampling methods to CIs for QAEval-F_1 (Deutsch et al., 2021a) are shown in Table 19.[16]

BOOT-BOTH generalizes the best. Among the bootstrap methods, BOOT-BOTH produces CIs that come closest to the ideal 95% rate. Any deviations from this number reflect that the assumption that all of the inputs and systems are distributed similarly is not true, but overall violating this assumption does not have a major impact.

The other bootstrap methods, which sample only systems or inputs, captures the correlation on the held-out data far less than 95% of the time. For instance, the CIs for ρ_{SYS} on Bhandari et al. (2020)

[16]The Fisher transformation was directly applied to the averaged summary-level correlation.

only successfully estimate the held-out correlation on 80% and 68% of trials. This means that a 95% CI calculated using BOOT-INPUTS is actually only a 68% CI on the held-out data. This pattern is the same across the different correlation levels and datasets. The lower values for only sampling inputs indicates that more variance comes from the systems rather than the inputs.

Fisher analysis. The Fisher transformation at the system-level creates CIs that generalize worse than BOOT-BOTH. The summary-level CI captures the held-out sample correlation 100% of the time, implying that the CI width is too large to be useful. We believe this is due to the fact that as the absolute value of $r(X, Z)$ decreases, the width of the Fisher CI increases. Summary-level correlations are lower than system-level correlations (see §7.6.1), and therefore Fisher results in a worse CI estimate at the summary-level.

Conclusion. This experiment presents strong evidence that violating the assumptions that either the systems/inputs are fixed or that the data is normally distributed does result in worse CIs. Hence, the BOOT-BOTH method provides the most accurate CIs for scenarios in which summarization metrics are frequently used.

7.5.2. Power Analysis

The power of a hypothesis test is the probability of accepting the alternative hypothesis given that it is actually true (equal to 1.0 – the type-II error rate). It is desirable to have as high of a power as possible in order to avoid missing a significant difference between metrics. This simulation estimates the power of each of the hypothesis tests.

Setup. Measuring power requires a scenario in which it is known that ρ is greater for one metric than another (i.e., H_1 is true). Since this is not known to be true for any pair of proposed evaluation metrics, we artificially create such a scenario by adding randomness to the calculation of ROUGE-1.[17] We define $\mathcal{R}_k$ to be ROUGE-1 calculated using a random $k\%$ of the candidate summary's tokens. We assume that since $\mathcal{R}_k$ only evaluates a summary with $k\%$ of its tokens, it is quite likely that it is a worse metric than standard ROUGE-1 for $k < 100$.

To estimate the power, we score summaries with ROUGE-1 and $\mathcal{R}_k$ for different k values and

[17]We use the recall variant of ROUGE for experiments on TAC'08 and Bhandari et al. (2020) and the F_1 variant on Fabbri et al. (2021) throughout the paper.

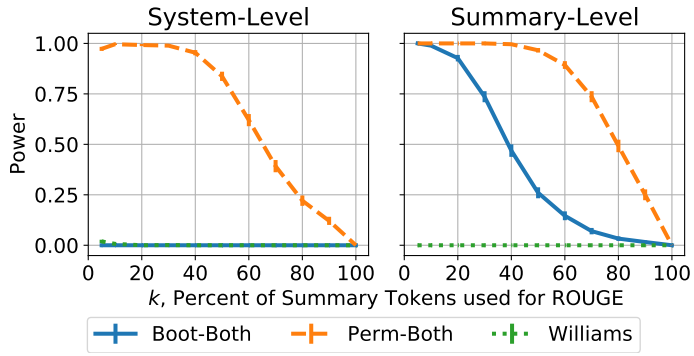

Figure 27: The system- and summary-level Pearson estimates of the power of the BOOT-BOTH, PERM-BOTH, and Williams hypothesis test methods calculated on the annotations from Fabbri et al. (2021). The power for BOOT-BOTH and Williams at the system-level is ≈ 0 for all values.

count how frequently each hypothesis test rejects H_0 in favor of identifying ROUGE-1 as a superior metric. This trial is repeated 1000 times, and the proportion of significant results is the estimate of the power.

Since the various hypothesis tests make different assumptions about whether the systems and inputs are fixed or variable, it is not necessarily fair to directly compare their powers. Because the assumptions of BOOT-BOTH and PERM-BOTH most closely align with the typical use case of summarization, we compare their powers. We additionally include Williams' test because it is frequently used for machine translation metrics and it produces interesting results, discussed below.

PERM-BOTH has the highest power. Fig. 27 plots the power curves for various values of k on the CNN/DM annotations by Fabbri et al. (2021). We find that PERM-BOTH has the highest power among the three tests for all values of k. As k approaches 100%, the difference between ROUGE-1 and $\mathcal{R}_k$ becomes smaller and harder to detect, thus the power for all methods approaches 0.

BOOT-BOTH has lower power than PERM-BOTH both at the summary-level and system-level, in which it is near 0. This result is consistent with permutation tests being more useful for hypothesis testing than their bootstrapping counterparts. We believe the power differences in both levels are due to the variance of the two correlation levels. As we observe in §7.6.1, the system-level CIs have significantly larger variance than at the summary-level, making it harder for the paired bootstrap to reject the system-level H_0.

Williams' test has low power. Interestingly, the power of Williams' test for all k is ≈ 0, implying the test never rejects H_0 in this simulation. This is surprising because Williams' test is frequently used to compare machine translation metrics at the system-level and does find differences between metrics. We believe this is due to the strength of the correlations of ROUGE-1 to the ground-truth judgments as follows.

The p-value calculated by Williams is a function of the pairwise correlations of X, Y, and Z and the number of observations. The closer both $r(X, Z)$ and $r(Y, Z)$ are to 0, the higher the p-value. The correlation of ROUGE-1 in this simulation is around 0.6 and 0.3 at the system- and summary-levels. In contrast, the system-level correlations for the metrics submitted to the Workshop on Machine Translation (WMT) 2019's metrics shared task for de-en are on average 0.9 (Ma et al., 2019). Among the 231 possible pairwise metric comparisons in WMT'19 for de-en, Williams' test yields 81 significant results. If the correlations are shifted to have an average value of 0.6, only 3 significant results are found. Thus we conclude that Williams' test's power is worse for detecting differences between lower correlation values.

Because this simulation is performed with summarization metrics on a real summarization dataset, we believe it is faithful enough to a realistic scenario to conclude that Williams' test does indeed have low power when applied to summarization metrics. However, we do not expect Williams' test to have 0 power when used to detect differences between machine translation metrics.

Conclusion. Since PERM-BOTH has the best statistical power at both the system- and summary-levels, we recommend it for hypothesis testing the difference between summarization metrics.

7.6. Summarization Analysis

We run two experiments that calculate CIs (§7.6.1) and run hypothesis tests (§7.6.2) for many different summarization metrics on the TAC 2008 and CNN/DM datasets (§7.5). Each experiment also includes an analysis which discusses the implications of the results for the summarization community.

The metrics used for experimentation are the following: AutoSummENG (Giannakopoulos et al., 2008), BERTScore (Zhang et al., 2020), BEwT-E (Tratz and Hovy, 2008), METEOR (Denkowski

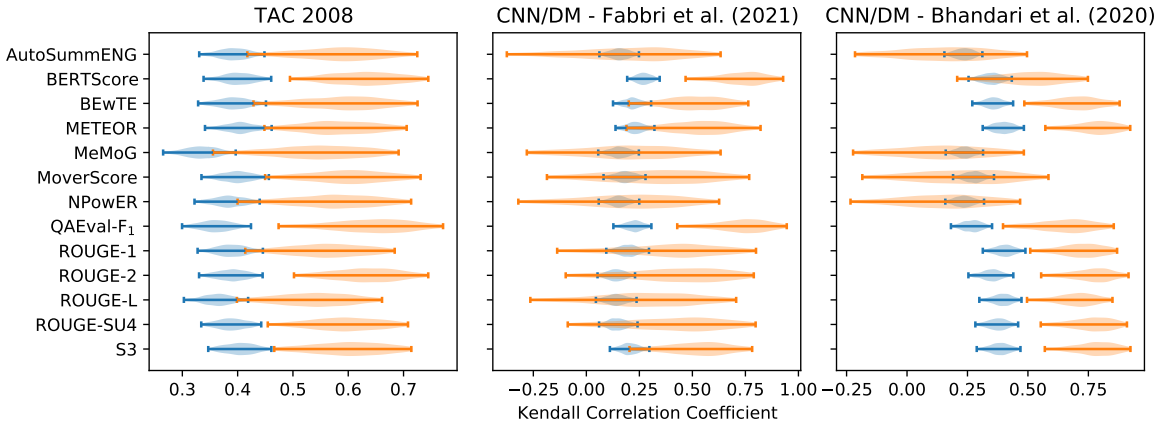

Figure 28: The 95% confidence intervals for ρ_{SUM} (blue) and ρ_{SYS} (orange) calculated using Kendall's correlation coefficient on TAC 2008 (left) and CNN/DM summaries (middle, SummEval; right, REALSumm) are rather large, reflecting the uncertainty about how well these metrics agree with human judgments of summary quality.

and Lavie, 2014), MeMoG (Giannakopoulos and Karkaletsis, 2010), MoverScore (Zhao et al., 2019), NPowER (Giannakopoulos and Karkaletsis, 2013), QAEval (Deutsch et al., 2021a), ROUGE (Lin, 2004), and S^3 (Peyrard et al., 2017). We use the metrics' implementations in the SacreROUGE library (Deutsch and Roth, 2020).

7.6.1. Confidence Intervals

Fig. 28 shows the 95% CIs calculated via BOOT-BOTH for ρ_{SUM} and ρ_{SYS} for each metric calculated using Kendall's τ. Since ROUGE is the most commonly used metric, the following discussion will mostly focus on its results, however the conclusions largely apply to other metrics as well.

Confidence intervals are large. The most apparent observation is that the CIs are rather large, especially for ρ_{SYS}. The ROUGE-2 ρ_{SYS} CIs are $[.49, .74]$ for TAC 2008 and $[-.09, .84]$ on CNN/DM using the annotations from Fabbri et al. (2021). The wide range of values demonstrates that there is a large amount of uncertainty around how precise the correlations reported in the literature truly are.

The size of the CIs has serious implications for how trustable existing automatic evaluations are. Since Kendall's τ is a function of the number of pairs of systems in which the automatic metric and ground-truth agree on their rankings, the metrics' CIs can be translated to upper- and lower-bounds on the number of incorrect rankings. Specifically, ROUGE-2's system-level CI on Fabbri

et al. (2021) implies it incorrectly ranks systems with respect to humans 9-54% of the time. This means that potentially more than half of the time ROUGE ranks one summarization model higher than another on CNN/DM, it is wrong according to humans, a rather surprising result. However, it is consistent with similar findings by Rankel et al. (2013), who estimated the same result to be around 37% for top-performing systems on TAC 2008-2011.

We suspect that the true ranking accuracy of ROUGE (as well as the other metrics) is not likely to be at the extremes of the confidence interval due to the distribution of the bootstrapping samples shown in Fig. 28. However, this experiment highlights the uncertainty around how well automatic metrics replicate human annotations of summary quality. An improved ROUGE score does not necessarily mean a model produces better summaries. Likewise, not improving ROUGE should not disqualify a model from further consideration. Consequently, researchers should rely less heavily on automatic metrics for determining the quality of summarization models than they currently do. Instead, the community needs to develop more robust evaluation methodologies, whether it be task-specific downstream evaluations or faster and cheaper human evaluation.

Comparing CNN/DM annotations. The CIs calculated on the annotations by Bhandari et al. (2020) (REALSumm) are in general higher and more narrow than on Fabbri et al. (2021) (SummEval). We believe this is due to the method of selecting the summaries to be annotated for each of the datasets. Bhandari et al. (2020) selected summaries based on a stratified sample of automatic metric scores, whereas Fabbri et al. (2021) selected summaries uniformly at random. Therefore, the summaries in Bhandari et al. (2020) are likely easier to score (due to a mix of high- and low-quality summaries) and are less representative of the real data distribution than those in Fabbri et al. (2021).

7.6.2. Hypothesis Testing

Although nearly all of the CIs for the metrics are overlapping, this does not necessarily mean that no metric is statistically better than another since the differences between two metrics' correlations could be significant.

In Fig. 29, we report the p-values for testing $H_0 : \rho(\mathcal{X}, \mathcal{Z}) - \rho(\mathcal{Y}, \mathcal{Z}) \leq 0$ using the PERM-BOTH permutation test at the system- and summary-levels on TAC 2008 and CNN/DM for all

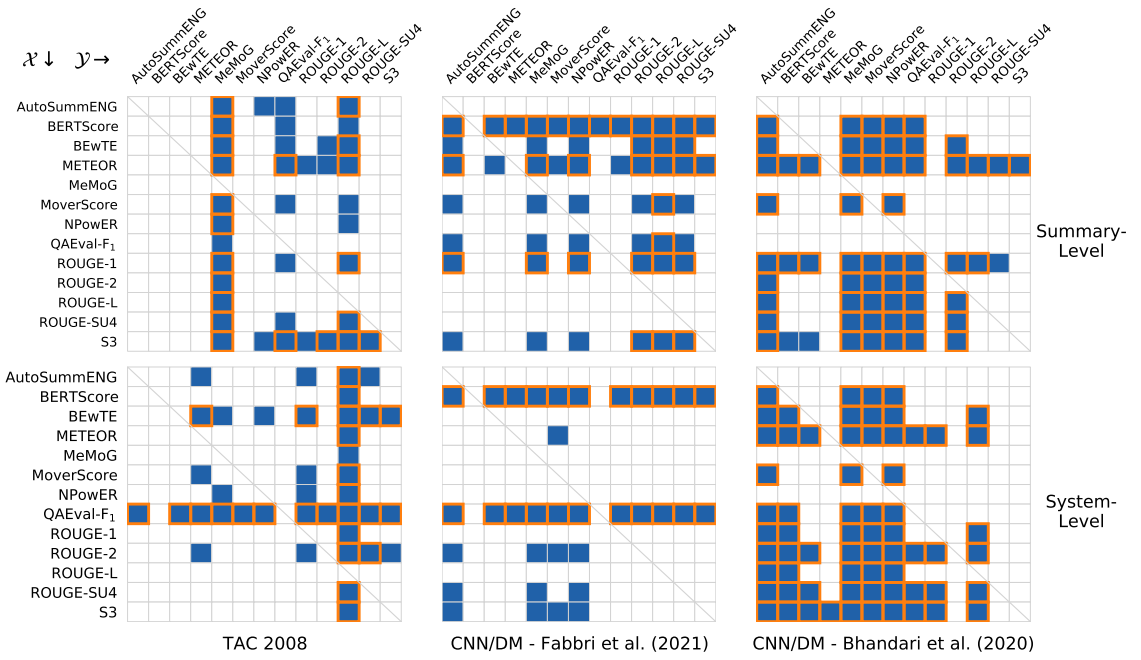

Figure 29: The results of running the PERM-BOTH hypothesis test to find a significant difference between metrics' Pearson correlations. A blue square means the test returned a significant p-value at $\alpha = 0.05$, indicating the row metric has a higher correlation than the column metric. An orange outline means the result remained significant after applying the Bonferroni correction. Summary-level correlations are the top row and system-level on the bottom. TAC 2008 is in the left column, SummEval in the middle, and REALSumm on the right.

possible metric combinations (see Azer et al. (2020) for a discussion about how to interpret p-values). The Bonferroni correction (which lowers the significance level for rejecting each individual null hypothesis such that the probability of making one or more type-I errors is bounded by α; Bonferroni, 1936; Dror et al., 2017) was applied to test suites grouped by the $\mathcal{X}$ metric at $\alpha = 0.05$.[18] A significant result means that we conclude that $\rho(\mathcal{X}, \mathcal{Z}) > \rho(\mathcal{Y}, \mathcal{Z})$.

The metrics which are identified as being statistically superior to others at the system-level on TAC 2008 and CNN/DM using the annotations from Fabbri et al. (2021) are QAEval and BERTScore. Although they are statistically indistinguishable from each other, QAEval does improve over more metrics than BERTScore does on TAC 2008. At the summary-level, BERTScore has significantly better results than all other metrics. Overall, none of the other metrics consistently outperform

[18] A version of the results when the correction is applied to p-values grouped by the dataset and correlation level pair is included in Appendix C.2.

all variants of ROUGE. Results using either the Spearman or Kendall correlation coefficients are largely consistent with Fig. 29, although QAEval no longer improves over some metrics, such as ROUGE-2, at the system-level on TAC 2008.

The results on the CNN/DM annotations provided by Bhandari et al. (2020) are less clear. The ROUGE variants appear to perform well, a conclusion also reached by Bhandari et al. (2020). The hypothesis tests also find that S^3 is statistically better than most other metrics. S^3 scores systems using a learned combination of features which includes ROUGE scores, likely explaining this result. Similarly to the CI experiment, the results on the annotations provided by Bhandari et al. (2020) and Fabbri et al. (2021) are rather different, potentially due to differences in how the datasets were sampled. Fabbri et al. (2021) uniformly sampled summaries to annotate, whereas Bhandari et al. (2020) sampled them based on their approximate quality scores, so we believe the dataset of Fabbri et al. (2021) is more likely to reflect the real data distribution.

7.7. Limitations

The large widths of the CIs in §7.6.1 and the lack of some statistically significant differences between metrics in §7.6.2 are directly tied to the size of the datasets that were used in our analyses. However, to the best of our knowledge, the datasets we used are some of the largest available with annotations of summary quality. Therefore, the results presented here are our best efforts at accurately measuring the metrics' performances with the data available. If we had access to larger datasets with more summaries labeled across more systems, we suspect that the scores of the human annotators and automatic metrics would stabilize to the point where the CI widths would narrow and it would be easier to find significant differences between metrics.

Although it is desirable to have larger datasets, collecting them is difficult because obtaining human annotations of summary quality is expensive and prone to noise. Some studies report having difficulty obtaining high-quality judgments from crowdworkers (Gillick and Liu, 2010; Fabbri et al., 2021), whereas others have been successful using the crowdsourced Lightweight Pyramid Score (Shapira et al., 2019), which was used in Bhandari et al. (2020).

Then, it is unclear how well our experiments' conclusions will generalize to other datasets with dif-

ferent properties, such as documents coming from different domains or different length summaries. The experiments in Bhandari et al. (2020) show that metric performance depends on which dataset you use to evaluate, whether it be TAC or CNN/DM, which is supported by our results. However, our experiments also show variability in performance within the same dataset when using different quality annotations (see the differences in results between Fabbri et al. (2021) and Bhandari et al. (2020)). Clearly, more research needs to be done to understand how much of these changes in performance is due to differences in the properties of the input documents and summaries versus how the summaries were annotated.

7.8. Related Work

Summarization. CIs and hypothesis testing were applied for summarization evaluation metrics over the years in a relatively inconsistent manner – if at all. To the best of our knowledge, the only instances of calculating CIs for summarization metrics is at the system-level using a bootstrapping procedure equivalent to BOOT-SYSTEMS (Rankel et al., 2012; Davis et al., 2012). Some works do perform hypothesis testing, but it is not clear which statistical test was run (Tratz and Hovy, 2008; Giannakopoulos et al., 2008). Others report whether or not the correlation itself is significantly different from 0 (Lin, 2004), which does not quantify the strength of the correlation nor allow for comparisons. Some studies apply Williams' test to compare summarization metrics. For instance, Graham (2015) use it to compare BLEU (Papineni et al., 2002) and several variants of ROUGE, and Bhandari et al. (2020) compares several different metrics at the system-level. However, our experiments demonstrated in §7.5.2 that Williams' test has lower power than the suggested methods due to the lower correlation values.

As an alternative to comparing metrics' correlations, Owczarzak et al. (2012) argue for comparison based on the number of system pairs in which both human judgments and metrics agree on statistically significant differences between the systems, a metric also used in the TAC shared-task for summarization metrics (Dang and Owczarzak, 2009, *i.a.*). This can be viewed similarly to Kendall's τ in which only statistically significant differences between systems are counted as concordant. However, the differences in discriminative power across metrics was not statistically tested itself.

More broadly in evaluating summarization systems, Rankel et al. (2011) argue for comparing the performance of summarization models via paired t-tests or Wilcoxon signed-rank tests (Wilcoxon, 1992). They demonstrate these tests have more power than the equivalent unpaired test when used to separate human and model summarizers.

Machine Translation. The summarization and machine translation (MT) communities face the same problem of developing and evaluating automatic metrics to evaluate the outputs of models. Since 2008, the Workshop on Machine Translation (WMT) has run a shared-task for developing evaluation metrics (Mathur et al., 2020b, among others). Although the methodology has changed over the years, they have converged on comparing metrics' system-level correlations using Williams' test (Graham and Baldwin, 2014). Since Williams' test assumes the input data is normally distributed and our experiments show it has low power for summarization, we do not recommend it for comparing summarization metrics. However, human annotations for MT are standardized to be normally distributed, and the metrics have higher correlations to human judgments, thus Williams' test will probably have higher power when applied to MT metrics. Nevertheless, the methods proposed in this work can be directly applied to MT metrics as well.

7.9. Summary

In this Chapter, we proposed several different methods for estimating CIs and hypothesis testing for summarization evaluation metrics using resampling methods. These analyses largely did not exist for summarization metrics before, and we demonstrated how these new techniques provided insights into the performance of evaluation metrics — including our own QAEval — which were not previously known. For instance, we found that the CIs for the correlations are very wide, meaning the community is very uncertain how well metrics actually correlate to human judgments, and that QAEval and BERTScore do indeed out-perform ROUGE in some evaluation settings.

CHAPTER 8 : Re-Examining Metric Meta-Evaluation

In the previous Chapter, we proposed a set of statistical methods for improving our analyses of the system- and summary-level correlations of evaluation metrics, thus making metric meta-evaluation more robust. These provided insights into the behavior of the evaluation metrics which were previously not known.

However, in this Chapter, we question whether the standard definition of the system-level correlation that is used throughout summarization actually meta-evaluates metrics the way we want them to. We argue that there are two disconnects between how metrics are meta-evaluated and how they are used in practice to evaluate summarization systems, thus the current meta-evaluation methodology does not provide the most accurate portrait of the performance of evaluation metrics in realistic use cases. To address this problem, we propose modifications to the definition of the system-level correlation to make the evaluation of metrics better aligned to how they are actually used in practice.

The work in this Chapter was originally presented in Deutsch et al. (2022b).

8.1. Introduction

The changes we propose to the system-level correlation definition are based on two observations about how evaluation metrics are used in practice versus how they are meta-evaluated. First, the metrics' scores which are used in practice are not the ones which are evaluated in system-level correlations: Researchers compare systems based on metric scores calculated on the entire test set but calculate scores for system-level correlations when evaluating metrics on a much smaller subset of judged summaries. Second, metrics are evaluated in a setting that is much easier than how they are actually used. Metric correlations are calculated using systems that vary greatly in quality, whereas researchers compare new systems to recent work, which are likely to be very close in quality. Discriminating between two systems of similar quality is much harder than doing so between low and high quality systems.

To that end, we propose two changes to how system-level correlations are calculated. First, we propose to modify the system-level correlation definition to use the entire test set to calculate the

system scores for automatic metrics instead of only the subset of summaries judged by humans (§8.3). With this change, the scores which are used to compare systems are directly evaluated, and we further demonstrate how the precision of our estimate of system-level correlations improves as a result. Calculating system scores over a larger number of instances reduces the variance of the scores, which results in confidence intervals (CIs) for the correlations that are 16-51% more narrow on average (§8.3.2).

Second, we redefine a high quality metric to be one for which a small difference in score reliably indicates a difference in quality (§8.4). Then, instead of calculating the correlation with all available system pairs, we only evaluate with pairs of systems whose automatic metric scores differ by some threshold. This allows us to show that a ROUGE-1 score difference of less than 0.5 between systems has almost no correlation to how humans would rank the same two systems according to our best estimates (§8.4.2). For two other metrics, BERTScore (Zhang et al., 2020) and QAEval (Deutsch et al., 2021a), we show their correlations calculated on system pairs of similar quality are much worse than under the standard correlation definition. These results cast doubt on how reliable automatic evaluation metrics are for measuring summarization system quality in realistic scenarios.

Our analyses point to the need to collect more high-quality human judgments of summaries in order to have more accurate estimates of metric correlations as well as the need to improve the ability of automatic metrics to discriminate between similarly performing systems.

8.2. Background

This Section describes some background required for this Chapter. Much of it was covered in Chapter 2, however we review some of the core concepts that are re-examined in this work and make necessary changes to the notation.

Automatic evaluation metrics are most commonly used to argue that one summarization system is better than another, typically by showing that the value of a metric improves with the "better" system. How similarly automatic metrics replicate human judgments of system quality is quantified by system-level correlations as follows.

The summaries from N systems on M_{jud} input documents are judged by humans $\mathcal{Z}$ and scored with an automatic metric $\mathcal{X}$. Then, the system-level correlation between $\mathcal{X}$ and $\mathcal{Z}$ is calculated as

$$r_{\text{SYS}} = \textsc{Corr}\left(\left\{\left(\frac{1}{M_{\text{jud}}}\sum_{j}^{M_{\text{jud}}} x_i^j, \frac{1}{M_{\text{jud}}}\sum_{j}^{M_{\text{jud}}} z_i^j\right)\right\}_{i=1}^{N}\right) \tag{8.1}$$

where x_i^j and z_i^j are the scores of $\mathcal{X}$ and $\mathcal{Z}$ for the summary produced by the i-th system on the j-th input document and CORR is some correlation function.

In this work, we use Kendall's τ (the "b" variant[19]) as the correlation function because we are most concerned with a metric's ability to correctly determine whether one system is better than another since that is how metrics are used in practice. Kendall's τ is computed based on the number of system pairs out of $\binom{N}{2}$ which are ranked the same by $\mathcal{X}$ and $\mathcal{Z}$. It is defined as

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) \cdot (P + Q + U)}} \tag{8.2}$$

where P and Q are the number of pairs ranked the same or different by $\mathcal{X}$ and $\mathcal{Z}$, respectively, and T and U are the number of ties only in $\mathcal{X}$ or $\mathcal{Z}$, respectively.

Because the computation of r_{SYS} involves randomness — its value depends on which M_{jud} input documents (and even which N systems) were used — it is only an approximation of the true correlation between $\mathcal{X}$ and $\mathcal{Z}$. As such, Deutsch et al. (2021b) proposed various methods for calculating confidence intervals for r_{SYS}. For instance, their BOOT-INPUTS method uses bootstrapping to repeatedly resample the M_{jud} input documents used to calculate r_{SYS}, thereby calculating a confidence interval for the true r_{SYS} value for $\mathcal{X}$ and $\mathcal{Z}$.

Datasets. The datasets that are used in this paper's analyses are SummEval (Fabbri et al., 2021) and REALSumm (Bhandari et al., 2020), two recently collected datasets with human annotations for summary quality collected from the CNN/DailyMail dataset (Nallapati et al., 2016). SummEval has $M_{\text{jud}} = 100$ summaries annotated with a summary relevance score for $N = 16$ systems. REAL-

[19] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html

Summ has $M_{\text{jud}} = 100$ summaries annotated with a Lightweight Pyramid score (Shapira et al., 2019) for $N = 25$ systems. We correlate the scores of the automatic metrics to these annotations. The CNN/DailyMail test split has $11,490$ instances.

Automatic Metrics. Our experiments will analyze three different reference-based automatic evaluation metrics which were chosen because they were demonstrated to have the best correlations with human judgments on the SummEval and REALSumm datasets (Deutsch et al., 2021b). ROUGE-n (Lin, 2004) evaluates a generated summary by calculating an F_1 score on the number of n-grams it has in common with a human-written reference summary. BERTScore (Zhang et al., 2020) aligns the generated and reference summaries' tokens based on their BERT embeddings (Devlin et al., 2019) and calculates a score based on the similarity of the aligned tokens' embeddings. QAEval (Deutsch et al., 2021a) compares the two summaries by automatically generating questions from the reference and calculating what proportion of those questions are answered correctly by the generated summary.

8.3. Evaluating with All Available Instances

Although the above definition of the system-level correlation has been used by recent meta-evaluation studies of metrics (Bhandari et al., 2020; Fabbri et al., 2021; Deutsch et al., 2021b), there is a disconnect between how the automatic metrics are evaluated and how they are used in practice.

Researchers who develop summarization systems evaluate those systems with automatic metrics on all M_{test} test instances, not just the subset of M_{jud} instances which were judged by humans. Evaluating a system on a larger number of summaries may end up changing the system's score, which could potentially alter the overall ranking of a set of systems. Therefore, the rankings that are used by practitioners to determine system quality are not the ones which are being evaluated in the standard definition of system-level correlation.[20]

To that end, we propose to modify the correlation definition to use all M_{test} instances to calculate

[20] We suspect this methodology is an artifact of how system-level correlations were first calculated for summarization in the DUC shared tasks when the dataset sizes were small enough that $M_{\text{jud}} = M_{\text{test}}$ (e.g., Dang and Owczarzak, 2008).

the system scores for the automatic metrics. That is (differences in bold):

$$r_{\text{SYS}} = \text{CORR}\left(\left\{\left(\frac{1}{\mathbf{M_{test}}}\sum_{j}^{\mathbf{M_{test}}} x_i^j, \frac{1}{M_{\text{jud}}}\sum_{j}^{M_{\text{jud}}} z_i^j\right)\right\}_{i=1}^{N}\right) \tag{8.3}$$

In practice with modern, large-scale datasets, this minor change could mean estimating system quality based on ≈10k inputs instead of around 100. This new definition now properly evaluates the way metrics are actually used by researchers.

We expect that scoring systems with M_{test} inputs instead of M_{jud} should lead to a better estimate of the true automatic metric score, which would in turn result in a lower-variance estimate of the correlation between $\mathcal{X}$ and $\mathcal{Z}$ in the form of smaller confidence intervals for r_{SYS}. In the next sections, we carry out analyses to demonstrate that this is true.

8.3.1. Reducing Automatic Metric Variance

First, we empirically show that scoring systems with M_{test} instances instead of M_{jud} does indeed reduce the variance of the estimate of the automatic metric scores and subsequently increases the stabilities of the system rankings.

Ideally, the $\mathcal{X}$ score for a system would be its "oracle" $\mathcal{X}$ score, equal to the expected value of $\mathcal{X}$ for a document sampled from the latent distribution over documents defined by the dataset (e.g., a system's ROUGE score on an infinite number of examples from a dataset). Since this cannot be calculated, it is approximated by averaging the $\mathcal{X}$ score on a sample (i.e., either the M_{jud} or M_{test} input documents). Because $M_{\text{test}} \gg M_{\text{jud}}$, we expect that the variance of this estimate using M_{test} inputs should be lower than when using M_{jud}.

To quantify this, we calculated the variance of estimating the oracle $\mathcal{X}$ score using both M_{jud} and M_{test} input documents via bootstrapping. We randomly sampled M input documents with replacement, recomputed the system scores, and calculated the variance of those scores over 1k iterations. For all three metrics on both datasets, we found around a 99% reduction in the variance when M_{test} inputs were used instead of M_{jud}, clearly demonstrating that evaluating systems with M_{test} inputs results in a better estimate of the system scores. In Fig. 30, this is visualized for BERTScore on the

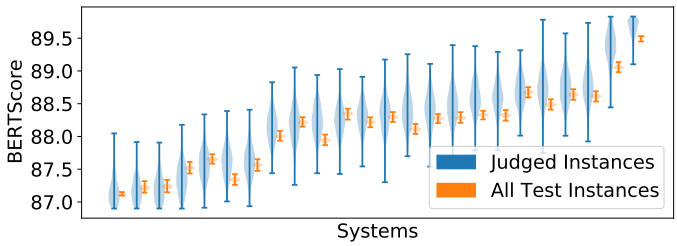

Figure 30: The bootstrapped 95% confidence intervals for the BERTScore of each system in the REALSumm dataset using M_{jud} judged instances in blue and M_{test} instances in orange. Evaluating systems with M_{test} instances leads to far better estimate of their true scores.

REALSumm dataset.

However, because we are interested in evaluating the metrics' rankings, we also quantify how much of an effect this reduction in variance has on the stability of the system rankings induced by $\mathcal{X}$. Similarly to the system scores, there is an oracle ranking of systems for $\mathcal{X}$, equal to the ordering of systems by their respective oracle $\mathcal{X}$ scores (e.g., systems sorted by their ROUGE scores calculated on an infinite number of examples from a dataset). As the variance of the system score estimates decreases, the computed ranking of systems should begin to converge to the oracle $\mathcal{X}$ ranking. We aim to understand to what extent this happens if M_{test} instances are used for evaluation instead of M_{jud}.

To quantify this notion, we calculate the Kendall's τ between two system rankings for $\mathcal{X}$ that were based on two sets of M input documents, each sampled with replacement from the set of available documents. This simulates how much the system rankings would change if the evaluation procedure was run twice, each time with M random input documents. This quantity is calculated 1k times for various values of M and plotted in Fig. 31.

As M approaches M_{test}, the automatic metrics' τ values approach 1, which is significantly higher than the respective values at M_{jud}, typically around 0.6-0.8. A value near 1 means that the rankings calculated using M_{test} inputs are almost constant, implying the rankings have converged to the oracle ranking. Therefore, the reduction in variance from evaluating on M_{test} instances does indeed greatly stabilize the system rankings.

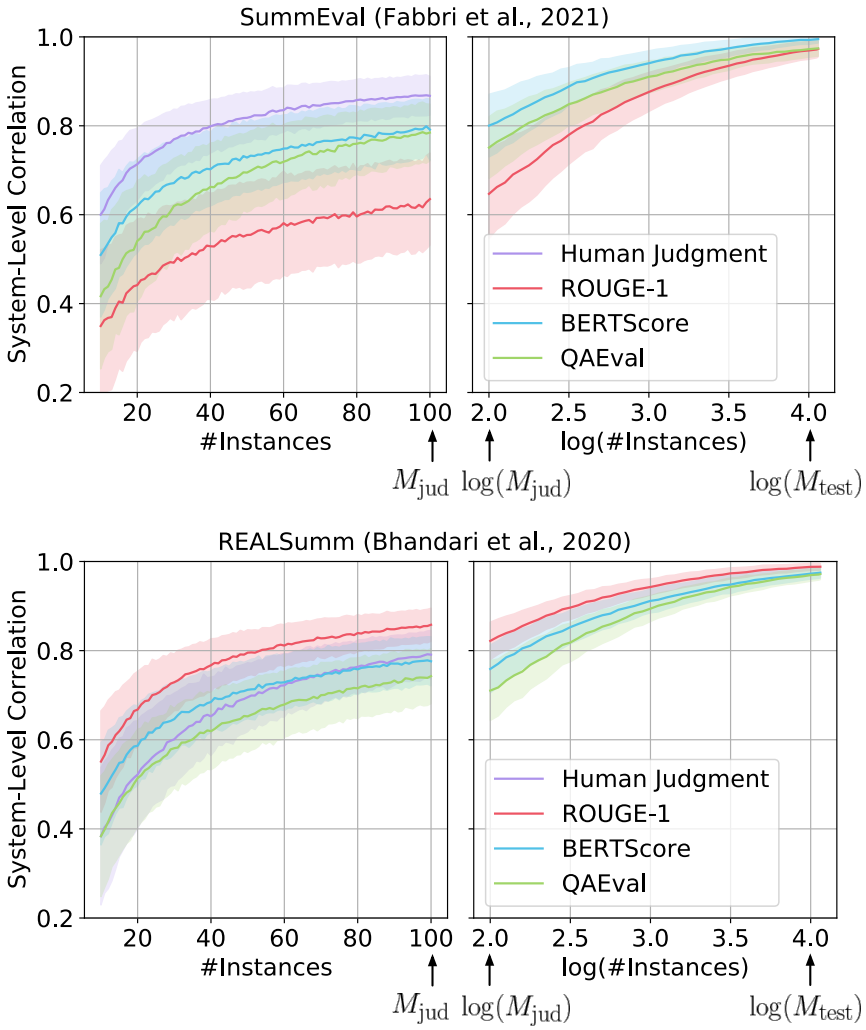

Figure 31: Bootstrapped estimates of the stabilities of the system rankings for automatic metrics and human annotations on SummEval (top) and REALSumm (bottom). The τ value quantifies how similar two system rankings would be if they were computed with two random sets of M input documents. When all M_{test} test instances are used, the automatic metrics' rankings become near constant. The error regions represent ± 1 standard deviation.

Fig. 31 also contains the same analysis performed for the human judgments $\mathcal{Z}$ in both datasets, although it is limited to a maximum of M_{jud} input documents. We see that on both datasets the judgments' rankings are still quite variable, reaching a maximum of around 0.8-0.85 τ.

8.3.2. Confidence Interval Analysis

Next, we show that the improved estimate of system scores leads to a more precise estimate of r_{SYS} by demonstrating the widths of the confidence intervals for r_{SYS} decrease.

The confidence intervals for r_{SYS} calculated using bootstrapping methods proposed by Deutsch et al. (2021b) are rather wide. For instance, the 95% CI for ROUGE-2 on SummEval is $[-.09, .84]$, demonstrating a rather high level of uncertainty in its value. This is problematic because it means we do not have a good picture of how reliable automatic evaluation metrics are. Reducing the width of the CIs will help us better understand the true metric quality.

We suspect that the large width of the confidence interval is due to the variance of the system rankings of the automatic metrics and human judgments. The more unstable the rankings are with respect to the M inputs, the larger the variance of the estimate of r_{SYS} should be since very different system rankings would be compared on each bootstrapping iteration. Deutsch et al. (2021b) used M_{jud} input documents to calculate their CIs. Therefore, we expect the improved stability of the automatic metric system rankings from evaluating on M_{test} instances should result in a more narrow confidence interval for r_{SYS} since some noise has been removed from this computation.

To demonstrate this, we calculated 95% CIs for r_{SYS} using the BOOT-INPUT method on SummEval and REALSumm using both M_{jud} and M_{test} input documents, shown in Fig. 32. We find that the widths of the CIs shrank on average by 51% on SummEval and 16% on REALSumm. The largest decrease in width is in the ROUGE family of metrics on SummEval, likely because that metric and dataset combination saw the biggest improvement in ranking stability (see Fig. 31). Thus, the improved estimate of the system scores did result in more precise estimates of r_{SYS}. We repeated this analysis using the other bootstrapping methods proposed by Deutsch et al. (2021b), and the results are discussed in Appendix D.1.

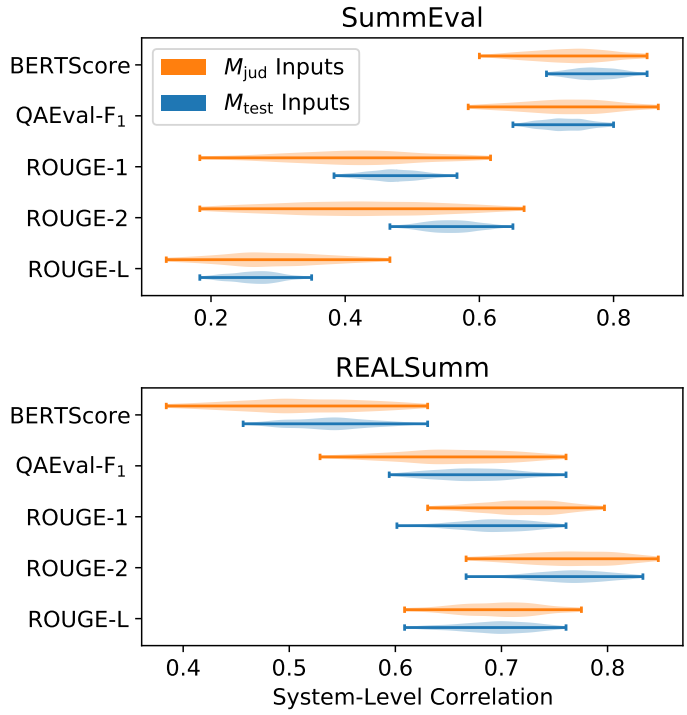

Figure 32: 95% confidence intervals for r_{SYS} calculated with the BOOT-INPUTS resampling method when the system rankings for the automatic metrics are calculated using only the judged data (orange) versus the entire test set (blue). Scoring systems with more summaries leads to better (more narrow) estimates of r_{SYS}.

8.3.3. Conclusions & Recommendations

By estimating system quality using automatic metrics on all available instances instead of only those which were judged, we showed that the variances of the system scores and subsequent rankings reduce significantly, resulting in better estimates of r_{SYS}. Because this methodology additionally directly evaluates the system scores used by researchers, we recommend future work do the same.

In order to continue to improve the estimate of r_{SYS}, as much variance as possible needs to be removed from the system rankings. Evaluating systems using M_{test} instances removed a large amount of variance from the automatic metric rankings, but as demonstrated in Fig. 31, the human judgments still have a large amount of variance.

The human rankings' variances can either be reduced by judging more summaries per system or making the judgments more consistent. Since the human rankings' stabilities in Fig. 31 are mostly

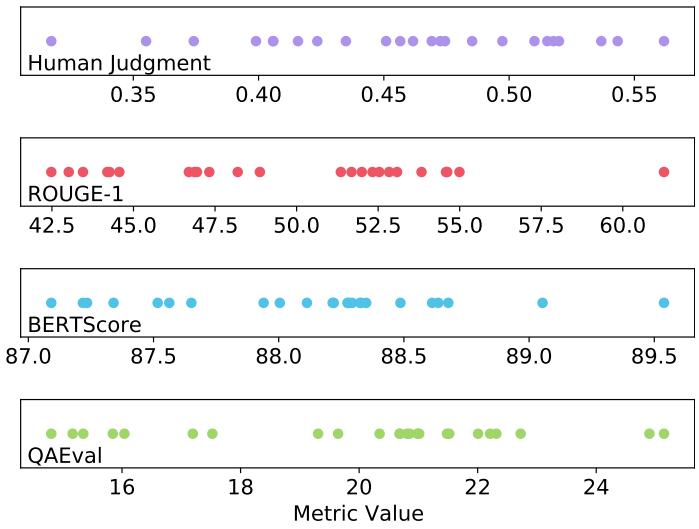

Figure 33: The systems (each represented by a point) on the two datasets (shown here for REAL-Summ) are rather diverse in quality as measured by both human judgments and automatic metrics.

beginning to plateau — especially for SummEval — it may be prohibitively expensive to collect a sufficient number of judgments to better stabilize the rankings (Wei and Jia, 2021). Therefore, we expect the more feasible solution is to improve the consistency of the human judgments, for example by better training the annotators or improving the annotation protocol.

8.4. Evaluating with Realistic System Pairs

Next, we argue that the set of systems used to evaluate metrics is not reflective of how metrics are used in practice and propose a new system-level correlation variant to address this problem.

8.4.1. Evaluating with All System Pairs

The N systems which are used for calculating system-level correlations are typically those which participated in a shared task, as in DUC/TAC (Dang and Owczarzak, 2008, among others), or those which have been published in the previous 3-4 years (Bhandari et al., 2020; Fabbri et al., 2021). As such, they are typically rather diverse in terms of their qualities, both as rated by human annotators and automatic metrics.

The system scores of all of the systems in the REALSumm dataset as evaluated by humans and automatic metrics are shown in Fig. 33. Clearly, the scores are rather diverse. For example, the systems cluster into low, medium, and high quality groups (with an additional outlier) as evaluated

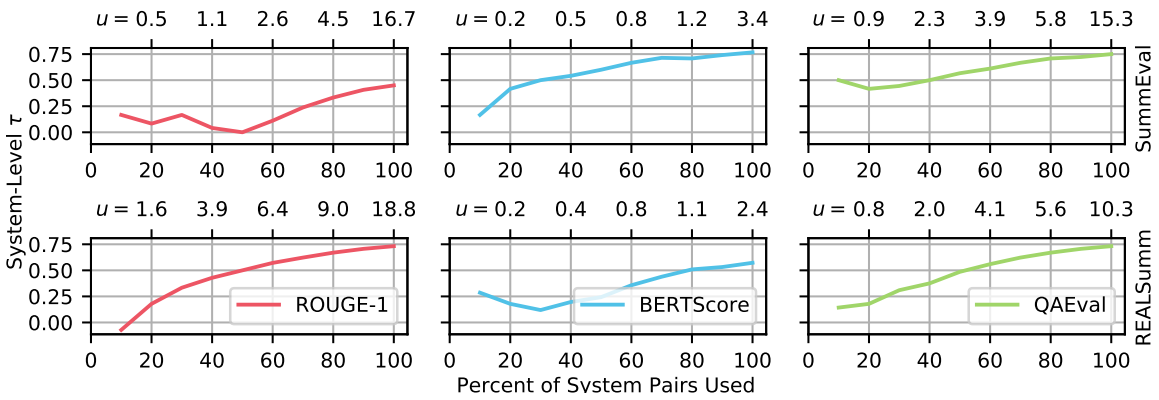

Figure 34: The $r_{\text{SYS}}\Delta(\ell, u)$ correlations on the SummEval (top) and REALSumm (bottom) datasets for $\ell = 0$ and various values of u (additional combinations of ℓ and u can be found in Appendix D.2). The u values were chosen to select the $10\%, 20\%, \ldots, 100\%$ of the pairs of systems closest in score. Each u is displayed on the top of each plot. For instance, 20% of the $\binom{N}{2}$ system pairs on SummEval are separated by < 0.5 ROUGE-1, and the system-level correlation on those pairs is around 0.08. As more systems are used in the correlation calculation, the allowable gap in scores between system pairs increases, and are therefore likely easier to rank, resulting in higher correlations.

by ROUGE. A difference of around 5 ROUGE points between them is a rather large gap for ROUGE scores.

The standard definition for a high quality evaluation metric is one which correctly ranks a set of systems with respect to human judgments. As such, the implementation of the system-level correlation calculated with Kendall's τ will rank all N systems according to the human judgments and an automatic metric, then count how many pairs were ranked the same out of all $\binom{N}{2}$ pairs (see §8.2). As a consequence, even pairs of systems which are separated by a large margin according to the automatic metric — likely systems with a clear difference in quality — are included in the evaluation. Therefore, automatic metrics are rewarded for correctly ranking such "easy" system pairs.

8.4.2. Evaluating with Realistic Pairs

This standard evaluation setting does not reflect how summarization metrics are actually used by researchers. New systems are typically only slightly better than previous work. Based on a survey of summarization papers in *ACL conferences over the past few years, we found that the average improvement over baseline/state-of-the-art models that was reported on the CNN/Dailymail dataset was on average 0.5 ROUGE-1. It is rarely the case that the improvement in automatic metrics is very

large. Therefore, evaluating metrics using pairs of systems which *are* separated by a large margin does not reflect the reality that metrics are very frequently used to compare those separated by a small margin. Including "easy" system pairs in the system-level correlation likely overestimates the quality of the metrics in settings which occur in practice.

To that extent, we redefine a high quality evaluation metric to be one for which a small difference in scores reliably indicates a difference in quality. We quantify this by proposing a variant of the system-level τ which is calculated between system pairs which are separated by a pre-defined automatic metric score margin. Instead of using all $\binom{N}{2}$ system pairs, only pairs whose difference in scores falls within the margin are used to calculate the system-level correlation. We denote this correlation variant as $r_{\text{SYS}}\Delta(\ell, u)$ where ℓ and u are the lower- and upper-bounds of the allowable differences in automatic metrics' scores. This would enable, for example, evaluating how well ROUGE correlates to human judgments on system pairs that are separated by 0.0-0.5 ROUGE points, thereby directly evaluating the scenario in which ROUGE is used to make decisions about system quality.

In Fig. 34 we report the $r_{\text{SYS}}\Delta(\ell, u)$ correlations for $\ell = 0.0$ and various values of u on both the SummEval and REALSumm datasets (more combinations of ℓ and u are included in Appendix D.2). That is, we evaluate r_{SYS} only on system pairs which are separated by at most an automatic score of u. The values of u were selected by picking the minimum u which would result in evaluating on $10\%, 20\%, \ldots, 100\%$ of the $\binom{N}{2}$ possible system pairs closest in score to be consistent across all three metrics.

The correlations for each metric on the system pairs closest in score are far lower than the correlations evaluated on all of the system pairs. For instance, the correlation of BERTScore on SummEval with the closest 20% of system pairs ($u \approx 0.2$) is only 0.42 compared to 0.77 under the standard definition of r_{SYS}. Thus, it is clear that the metrics are much less reliable approximations of human judgments when the system scores are close than was previously known. Evaluating on all possible system pairs leads to an overly optimistic view of automatic metric quality.

The $r_{\text{SYS}}\Delta(\ell, u)$ correlation of ROUGE for $\ell = 0.0$ and $u = 0.5$ — a typical improvement reported by researchers — is 0.08 and 0.0 on the SummEval and REALSumm datasets. Therefore, these results suggest the most popular summarization evaluation metric agrees with human judgments of system quality in realistic scenarios only slightly better than or equal to random chance.

This result also offers an explanation for why a naive metric such as ROUGE achieves moderately strong correlations under the standard definition of the system-level correlation (0.45 and 0.73 on SummEval and REALSumm) despite well known flaws and criticisms (Passonneau et al., 2005; Conroy and Dang, 2008; Deutsch and Roth, 2021, among others): It has benefited from an easy evaluation protocol. Despite its simplicity, it is not too surprising that a large gap of 5-10 ROUGE points actually does correctly rank system pairs. Most of its positive correlation comes from such easy examples.

8.4.3. Conclusions & Recommendations

If it is assumed that we have enough high-quality judgments to accurately discriminate between two similarly performing systems, then the results in Fig. 34 show that the correlations in realistic settings are trending very low, meaning automatic metrics are not nearly sensitive enough to distinguish between systems with only minor differences in quality. This is problematic because this is the scenario in which metrics are most frequently used, and therefore they are not very reliable methods of evaluating summarization systems. However, it is not all bad news. Because the standard system-level τ values are moderately positive, consistent improvements in automatic metrics over time will likely result in better quality systems. Similarly to stochastic gradient descent, not every reported improvement is real, but on average over time, the quality does improve. Nonetheless, future work should focus on improving the quality of evaluation metrics when the differences in system performance are small, and researchers who compare systems should invest more effort into their human evaluations since automatic evaluations are not very reliable.

However, because the available number of system pairs to calculate the correlations in Fig. 34 is rather small — especially when evaluating on the closest system pairs — and recent work suggests we may not have enough human judgments to accurately distinguish between similarly performing

systems (Wei and Jia, 2021), it could be difficult to reach any definitive conclusions about the metrics' correlations. That being said, these are our best estimates of the correlations with the available data. Not knowing how much we can trust automatic metrics is not a good outcome. In this scenario, future work should focus on collecting more, high-quality human judgments so that we can better meta-evaluate automatic metrics. Since we argue that it is important to distinguish between similarly performing systems, new data collection efforts should consider using targeted pairwise judgments between those systems instead of direct assessments across a variety of systems of diverse quality.

We recommend that proposals of new evaluation metrics also report correlations on system pairs with various differences in scores in addition to the standard system-level correlation definition. Reporting this information would better inform users of metrics about how likely humans would agree their observed improvement is real based on its value.

8.5. Related Work

The methodology behind meta-evaluating summarization evaluation metrics was established during the DUC/TAC shared tasks (Dang and Owczarzak, 2008, among others). In addition to competitions for developing high-quality summarization systems, there were also shared tasks for creating automatic metrics that correlated well with human judgments. The benchmark datasets created during DUC/TAC were small in size by today's standards because they were manually collected multi-document summarization datasets, which are hard to create at scale. As such, all of the model-generated summaries on the full test set were judged (so $M_{\text{jud}} = M_{\text{test}}$; §8.3), unlike for current datasets which are too large to fully judge.

Recently, there has been growing interest in revisiting the meta-evaluation of automatic evaluation metrics for summarization, in part due to the large differences between currently popular summarization datasets and those used in DUC/TAC. We view our work as continuing this direction of research.

Peyrard (2019) argues that current evaluation metrics do not work as well when they are used to evaluate high-performing systems compared to those which were evaluated in DUC/TAC.

Both Fabbri et al. (2021) and Bhandari et al. (2020) re-evaluated how well existing evaluation metrics work on the popular CNN/DailyMail dataset (Nallapati et al., 2016) by collecting judgments of summary quality using recent state-of-the-art systems. These datasets were used in our analyses. While the goal of these works was to identify which metrics correlated best with human judgments, our goal is to point out the ways in which the current methodology of meta-evaluating metrics is inconsistent with how they are used.

Then, the work of Deutsch et al. (2021b) proposed statistical methods for estimating and comparing correlation values. In contrast to our work, they provide statistical tools for analyzing correlations, whereas we propose new definitions of correlations.

Finally, Wei and Jia (2021) provided a theoretical analysis of the bias and variance of automatic and human evaluations of machine translations and summaries. Among their conclusions, they argue for evaluating metrics with pairwise accuracy (Kendall's τ) and that it may be prohibitively expensive to collect enough human judgments to distinguish between two systems with very similar quality. Our work further argues that metrics should be evaluated with a variant of Kendall's τ calculated using realistic system pairs (§8.4). Unfortunately, their results suggest that collecting enough human judgments to accurately measure how well automatic metrics perform in this setting may be very difficult.

Related studies to ours have examined how the choice of which systems to include in metric evaluations impacts the correlation values. Both Mathur et al. (2020a) and Bhandari et al. (2020) identify that metrics perform worse when scoring only the top-k systems in machine translation and summarization, respectively, and examine the use of pairwise comparisons for metric evaluation. Further, Mathur et al. (2020a) demonstrate that outlier systems have an out-sized influence on the correlation values and recommend removing them from the metric evaluations. In contrast, our work proposes to change the evaluation methodology for metrics so that it more closely resembles how they are used in practice. This results in evaluating only on system pairs which are realistically compared by researchers, that is, those separated by small margins in automatic metric scores. We believe that this is a more principled approach to how to select which system pairs to evaluate on compared to

previous work.

8.6. Summary

In this Chapter, we proposed two independent changes to how the system-level correlation of metrics is calculated to better align with how they are used to evaluate systems. Our analyses showed that these modifications led to lower-variance estimates of correlations and that commonly reported improvements in metric scores may not reliably predict how humans would judge system quality. The results from the analyses point to the need for future data collection efforts of high-quality human judgments and improving automatic evaluation metrics when differences in system performance are small.

CHAPTER 9 : Conclusion

Evaluation is an essential — and often overlooked — component of developing machine learning applications. Without high-quality evaluations of models, progress in machine learning and NLP is not measurable and thus is not possible.

In this thesis, I pointed out several ways in which evaluation in summarization is flawed and lacking and proposed different solutions for improving the state of evaluation. The main areas of focus were on the actual task of evaluating summaries as well as meta-evaluating metrics.

9.1. Summary of Contributions

In Chapter 3, I argued that commonly used reference-based evaluation metrics fail to evaluate the information of summaries through an analysis of ROUGE and BERTScore. As a consequence of this, the ways in which researchers evaluate summaries is not aligned with the goal of generating summaries that contain salient information, and therefore I demonstrated that these metrics are inadequate tools for measuring true progress on summarization.

To address the shortcomings of existing evaluations, in Chapter 4 I explored how QA-based representations of semantic content could be leveraged for evaluating summaries through a proposal of a reference-based metric called QAEval. The experiments demonstrated that the metric already achieves state-of-the-art performance at evaluating summarization systems, and I showed evidence that as the pre-trained models which are used as subroutines of the metric continue to improve, the overall quality of the metric will as well. Further, I argued that QAEval does a better job at evaluating the information content of summaries than either ROUGE or BERTScore, evidence that it is better metric for measuring progress of summarization research. Thus, this work strongly demonstrated the value of QA-based representations for summarization evaluation.

In Chapter 5, I explored the idea of a natural extension of QAEval in which the set of questions used to evaluate a summary is predicted using the document by a learned salience model instead of using a reference summary. I argued that not only is such a metric flawed, but the class of reference-free evaluation metrics are inherently limited in their ability to evaluate summaries by showing how

they are equivalent to using text generation models to evaluate other text generation models, which comes with a set of undesirable biases. The experiments demonstrated evidence that reference-free metrics should not be used to measure progress on tasks like summarization, but rather as diagnostic tools for better understanding model behavior.

Then, in Chapter 6, I briefly described an extension of the idea behind QAEval and showed how QA-based representations could be incorporated into a summary generation model. The method I proposed represents document information with QA pairs and used a QA procedure similar to QAEval to select which document content is salient. The salient content was then incorporated into a two-stage summarization model as an inductive bias, which resulted in better end-to-end and more controllable summaries. Thus, the work in this Chapter demonstrates that QA-based representations can also be used for generating better summaries, not only for summarization evaluation.

Then, Chapter 7 switched to the problem of meta-evaluating metrics. I explained how the existing tools for meta-evaluating metrics, including QAEval, were lacking because the community has a very poor understanding of how effective evaluation metrics actually are and whether differences in metric performance are actual improvements or due to random chance. Through a proposal of confidence interval and hypothesis testing methods based on resampling, I improved the rigor of metric meta-evaluation and showed how these tools provided new insights into the behavior of metrics which was not known before. The contributions of this Chapter are not limited to summarization because the methods I proposed are general enough to applied to any meta-evaluation procedure used for text generation tasks. In fact, they were adopted by the Workshop on Machine Translation's metrics shared task in 2021 (Freitag et al., 2021).

Finally in Chapter 8, I questioned whether the definition of the system-level correlation used to meta-evaluate metrics is the right one to use at all. I identified two ways in which the standard calculation does not reflect how evaluation metrics are actually used in practice, thus the decisions which researchers make with evaluation metrics are not actually being evaluated. Our changes resulted in more precise estimates of the metrics' correlations, but also revealed that commonly reported differences in automatic metrics may not reliably align with human judgments of summa-

rization system quality. The latter result is especially problematic because it casts doubt over the effectiveness of the primary method that researchers use to evaluate summarization models.

9.2. Future Directions

Overall, this thesis has touched on and improved various parts of text summarization, from the evaluation of summaries, to the meta-evaluation of metrics, and to the actual task of summary generation. However, there are still many open questions and interesting directions for future research that build on the work presented here.

Moving Beyond Evaluating Predicate-Argument Structure. The wh-questions which are generated from the reference summaries and the pre-trained QA model used by QAEval largely correspond to representing and reasoning about the predicate-argument structure of the text; the questions ask about things which are explicitly stated in the summaries, and the QA model has some basic level of understanding of the syntax of sentences and paraphrasing that it learned from its SQuAD training data. Because of these limitations, QAEval as it is proposed here cannot compare summaries at a higher-level than what is explicitly said with the summaries' predicate-argument structures.

I believe that the wh-questions used by QAEval likely come close to sufficiently representing all the predicate-argument relations in the summaries; its limitation is the QA performance. Therefore, I anticipate that future metrics which continue to evaluate predicate-argument structure will only achieve minor improvements over QAEval, and those improvements will not be due to their fundamental ability to represent information. Instead, it will be driven by how well their QA model (or the model which determines whether some information is contained in a text) performs on the summarization datasets.

For example, a recent proposed work demonstrates experimental evidence of an improved reference-based evaluation metric called Lite^{3}Pyramid (Zhang and Bansal, 2021) over QAEval. They represent summary information using phrases derived from semantic role labeling (SRL) and determine whether that content is present in another summary using an entailment model. Like QAEval, I would argue their metric is also limited to evaluating the predicate-argument structure of the text

because the SRL-based phrases can only express what is explicitly stated. I hypothesize that their improved performance is potentially due to the entailment model generalizing better to the summarization dataset and the fact that they were able to fine-tune the model on labeled in-domain data, not because their representations are more expressive.

Therefore, I hypothesize that the next frontier of summarization evaluation is to find ways of evaluating summaries beyond their predicate-argument structure. It is not clear to me exactly what this looks like, but one example would be automatically coming up with entailment-style premises and reasoning about whether summaries entail those premises using some level of reasoning that is beyond comparing predicate-argument relations. However, to do this research, it is not clear whether existing datasets for metric meta-evaluation are sufficient. That is, it is not clear whether a perfect system for matching predicate-argument structure would achieve the best possible correlations to the human judgments or if a metric with a higher level of reasoning capability could actually make an improvement. Understanding the answers to these questions would be a difficult, yet exciting future research direction to explore.

Moving Beyond Modeling Predicate-Argument Structure. Like QAEval, the summarization model proposed in Chapter 6 is limited in what types of salient document information it can identify using the reference summaries. That is, it is only able to identify noun phrases that participate in some predicate-argument relationship in the reference summaries via reasoning about the syntax of the sentences and some notion of paraphrasing due to how the wh-questions are generated and the QA model is trained. As such, the salient phrase classification model is trained to also identify such salient relations in the documents.

I suspect that our proposed summarzation model performs well on some of the datasets which we used for evaluation because reasoning about predicate-argument relations is highest-level of reasoning required for generating summaries which look like the references. Summarization datasets which are popular today are large-scale, single-document, and typically drawn from the news domain. Because the salient information in a news article is often at the beginning, copying the first few sentences of the input documents verbatim is often a quite strong baseline to beat (See et al.,

2017; Kryscinski et al., 2019), suggesting the tasks are not too interesting from a perspective of building a summarization system which has to perform high-level reasoning about the information in the text and actually write summaries at a more abstract level. Likely, all that is required is to focus on identifying salient predicate-argument relations in the first few sentences, which offers a potential explanation for why our QA-based model was able to do well on these datasets.

Similarly to the proposed extension to move beyond evaluating predicate-argument structure, it would be interesting to explore ways of identifying salient document information that requires reasoning beyond predicate-argument relations. However, this would require summarization datasets in which the references summaries are also written using a higher-level of reasoning; it is unclear exactly what that looks like.

One example could be a summarization dataset in which the references truly abstract over the information in the documents, and thus identifying salient document information would require reasoning about which parts of the input combined and provided evidence for what was written in the text, which is beyond the capabilities of the methods proposed in this thesis. Understanding the level of reasoning required by existing summarization datasets and then creating new datasets which force models to perform more challenging reasoning tasks is an exciting next step for summarization.

Improving Manual Evaluations of Summaries. The proposal of the resampling methods for calculating confidence intervals for evaluation metrics' correlations in Chapter 7 revealed that the community is very uncertain how well automatic metrics can actually substitute for human evaluations of summarization systems due to how wide the confidence intervals were. Then in Chapter 8, it was further demonstrated that metrics performance in practice is likely far lower than what is reported in standard meta-evaluations. This is problematic because it casts significant doubt on how trustworthy evaluation metrics truly are.

I suspect that the ability to improve evaluation metrics is currently limited by our ability to evaluate summaries with humans. Currently, our ability to precisely estimate their correlations either in the standard definition or the modified version that we proposed in Chapter 8 relies on having a large number of high-quality manual evaluations. However, manually evaluating summaries is notori-

ously difficult for various reasons: it is difficult to describe clear criteria for scoring summaries; most human evaluation today is done with crowd workers who are difficult to train, especially for cognitively challenging tasks like summary evaluation; there is standardized evaluation methodology; collecting a sufficient amount of data for meta-evaluation can be prohibitively expensive. Future work must focus on how to do better human evaluation in order to improve automatic evaluation.

One potential path forward might be to focus on manual reference-based evaluations instead of direct assessments. Direct assessments are considered to be the preferred evaluation method because you are able to describe to the human judges exactly how the summaries should be scored. However in practice, it is incredibly challenging to achieve consistent scores across annotators — especially crowd workers (Fabbri et al., 2021) — to the point where I suspect there may be too much variance for direct assessments to be reliable. The task is also very cognitively taxing because the judge must completely understand the input text — which could be very long — in order to accurately evaluate summary.

As such, I would explore ways of improving manual reference-based evaluations because I suspect comparing the contents of two summaries is cognitively easier and more likely to get better inter-annotator agreements, as demonstrated by the Lightweight Pyramid (Shapira et al., 2019). The downside is that reference-based evaluations are restricted to incorrectly assuming that the only salient content is what is present in the reference, but I hypothesize that the improved consistency of the evaluations would prove to be more valuable than a direct assessment, even if they are not perfect evaluations. Such improvements over existing protocols could explore different solutions to the same problems faced by automatic reference-based evaluations: how to extract and represent summary content and how to identify occurrences of that content in another summary.

Reference-based evaluations also implicitly better define the task of summarization: a good summary is one which is similar to the reference. While there is likely still no explicit definition for what that summary should contain, the evaluation protocol now establishes what the "correct" summary is, and all other summaries are evaluated with respect to it. In contrast, the direct assessment

approach does not narrow the task definition down further, which allows the annotators to use their own reasons for why the summaries are written and assign them scores accordingly. This ambiguity results in less consistent annotations.

An Annual Summarization Shared Task. The field of machine translation has significantly benefited from the existence of the Workshop on Machine Translation (WMT). WMT has been running annually since 2006, and each year it runs several different shared tasks for developing translation systems and evaluation metrics. The submissions to the shared tasks are evaluated by human annotators, and as such, MT has a very large collection of manually scored translations. These translations are incredibly useful for training new systems, understanding and analyzing existing models, and building and meta-evaluating metrics. While MT still faces many of the evaluation challenges that summarization does, it benefits from access to data and benchmark evaluation datasets.

Going forward, I believe that creating an annual shared task for developing summarization systems and evaluation metrics would be very impactful for summarization. It would allow the community to begin to address some of the shortcomings and limitations of the existing summarzation task definitions and evaluation methodologies. Importantly, it could also enable collecting a large number of human evaluations of summaries and establish standard practices for how authors of papers should perform evaluations. In this thesis, I argued that the community does not know how well automatic metrics actually replicate human judgments, and our best estimates point to the fact that they do not do so well in realistic use cases. An annual summarization shared task would be a great venue for beginning to address some of the core issues within summarzation evaluation.

APPENDICES

A. Appendix for Chapter 4

A.1. Number of Available References

Previous work has argued that multiple reference summaries are necessary for metrics to achieve stable correlations to ground-truth annotations, especially at the summary-level (Nenkova and Passonneau, 2004; Louis and Nenkova, 2013). Since the TAC datasets provide four reference summaries per input, we are able to measure how much benefit the additional references provide by simulating having fewer references.

In order to simulate only having one reference summary, for each input document set from TAC 2008, we randomly sample one reference, score all of the peer summaries against only that reference, and calculate the correlation to the responsiveness scores. We collect 30 such samples and report the average correlation. This procedure is also repeated for two and three references to understand the impact of each additional reference. The results are plotted in Figure 35.

At the system level, the Pearson correlations are largely the same when the metrics are provided with one or four references. This is in agreement with Louis and Nenkova (2013), who show system-level results are relatively stable with either one or four references. Among the metrics, the QA-based metrics see the largest improvement in performance with adding additional references. QAEval-F_1 increases from 0.85 with one reference compared to 0.90 with four. Despite its drop in performance with one reference, QAEval-F_1 is still better than ROUGE even with four references at 0.79. APES improves from 0.66 to 0.74.

When the metrics are compared at the summary level, it is clear that the correlations for each metric are less stable. Nearly all of the metrics greatly benefit from additional references: Pyramid Score improves by 0.09 (+19%), ROUGE by 0.08 (+18%), and QAEval-F_1 by 0.15 (+49%). The large improvement by QAEval-F_1 is further evidence that the noisy question-answering model averages out to a high-quality responsiveness estimator when provided with a large number of QA pairs.

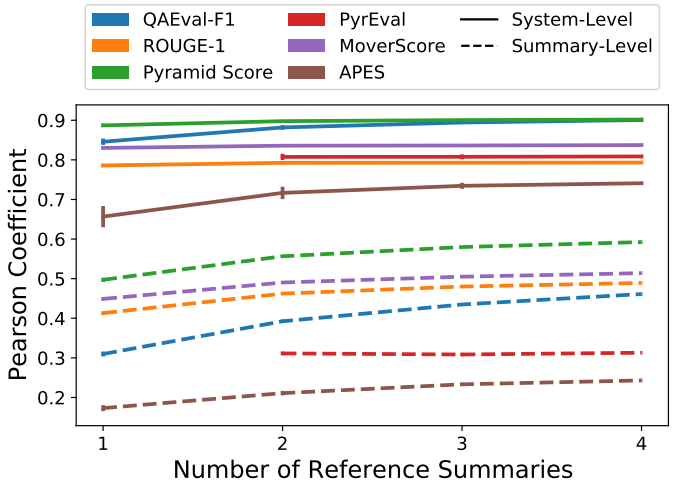

Figure 35: The system- and summary-level Pearson correlations as the number of available reference summaries increases. 95% confidence error bars shown, but may be too small to see. PyrEval is missing data because the official implementation requires at least two references. Even with one reference summary, QAEval-F_1 maintains a higher system-level correlation than ROUGE.

Overall, QAEval does incur a significant performance drop at the summary-level, but since most comparisons of summarization systems are done at the system-level, it does not appear that having multiple references per input is necessary for good results.

B. Appendix for Chapter 6

B.1. Dataset Statistics

The sizes of the CNN/DailyMail, XSum, and NYTimes datasets are included in Table 20. The Table also includes the number of spans per span type that were selected from the classification component and passed to the generation component during inference. The values were selected based on a parameter sweep on the validation set. The number of spans with the highest ROUGE-2 F_1 score was selected.

B.2. Implementation Details

All of the models were trained with the same hyperparameters for across datasets and span types which were based on those used by BART (Lewis et al., 2020).

The classification component was a BART-Large model that was fine-tuned with a binary cross-entropy classification loss. We selected the model based on which had the best precision@1 on

Dataset	#Train	#Valid	#Test	Span Type	#Spans
CNN/DM	287,113	13,368	11,490	Sentences	3
				Entities	10
				Lex. NPs	25
				QA	20
XSum	204,045	11,332	11,334	Sentences	1
				Entities	1
				NPs	5
				QA	1
NYTimes	44,382	5,523	6,495	Sentences	4
				Entities	15
				Lex. NPs	45
				QA	27

Table 20: The number of instances in the training, validation, and test splits of the three datasets used in our experiments as well as the number of spans selected by the classification component that were passed as input to the generation component.

the validation dataset. The generation models were also fine-tuned BART-Large models, but they instead use a cross-entropy loss function.

Both the components were trained using Adam (Loshchilov and Hutter, 2019) with weight decay and learning rate 3e-5. The classification component was trained for 3 epochs, and the final model was selected based on the precision@1 on the validation set. The generation component was trained for 5 epochs, and the final model was selected based on the ROUGE-2 F_1 score on the validation set.

B.3. Salient Span Classifier Evaluation

Fig. 36 contains the precision@k and recall@k of the span based classifiers calculated against the corresponding silver spans. These plots should be interpreted as how well the span classifiers were able to learn from their respective supervision, not necessarily the true quality of the output span labels (which would require evaluating against human-annotated gold labels, as in Table 15). The "x" symbols denote the operating points used in the end-to-end model, which were chosen based on the number of spans that resulted in the highest ROUGE-2 F_1 score on the validation data.

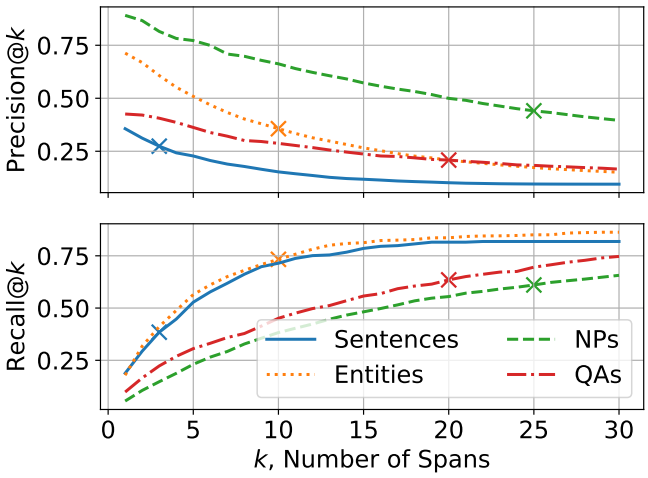

Figure 36: The performances of the salient span classifiers using the different types of salient phrase labeling evaluated against the silver spans. The "x" marks the operating points used in the end-to-end models.

B.4. Gold Span Annotation Protocol

We selected 50 test instances from the CNN/DailyMail dataset uniformly at random and labeled each of the document NPs as salient or not salient based on whether the corresponding predicate-argument relation also appears in the gold summary. We did not mark instances in which the NP's predicate-argument relation could be inferred from the gold summary via entailment as salient since our silver span labeling methods aim to mark phrases as salient if the content is explicilty included in the gold summary.

In general, this procedure was straightforward due to the extractive nature of the dataset in which the gold summaries copy heavily from the input document. If information was repeated in the input document, we tried to label the occurrence which contained the most predicate-argument relations which also matched the gold summary. That is, we selected the "best match." Otherwise, the first occurrence was selected.

Although our labeling procedure may be noisy, we do not have reason to believe that the labels may be biased in favor of either the lexical NP or QA labeling methods. Therefore, the statistics calculated from these labels should only be used as diagnostic tools to make relative comparisons

Model	R1	R2	RL	BSc	QAE
Silver Spans					
QAs	55.3	31.4	51.9	90.0	33.7
QAs + Data Aug.	55.2	31.3	51.7	89.9	33.4
End-to-End					
QAs	45.5	21.9	42.4	88.5	24.4
QAs + Data Aug.	45.3	21.8	42.1	88.4	24.3

Table 21: The automatic evaluation metrics for summary quality are nearly the same for the QA-based model and the QA-based model trained on the augmented data.

between the different labeling methods rather than precise estimates of their exact values. 50 documents were sufficient to achieve statistically different results.

Our annotations will be released after publication.

B.5. Data-Augmentation Automatic Evaluation

Table 21 contains the comparison between the standard and data-augmented training procedures based on the automatic metrics. The scores are nearly the same. The benefit of the model trained on the augmented data is in its controllability, which is not captured by this evaluation because the models trained with the standard and augmented training data receive the same spans as input supervision.

B.6. Human Evaluation Details

Fig. 37 contains a screenshot of the tool we used for annotating summary quality on MTurk. The annotators were instructed to rate the summaries from "Very Poor" to "Very Good" based on whether the summary contained important information, was faithful to the input document, was fluent, and was cohesive. The ratings were converted to a Likert scale from 1-5 and averaged across all of the ratings for a system.

In order to encourage the annotators to pay attention to the task, we also required that they write a very brief explanation of how they made their decision, inspired by Narayan et al. (2021).

The MTurk annotators were paid at a rate of around $15 USD per hour.

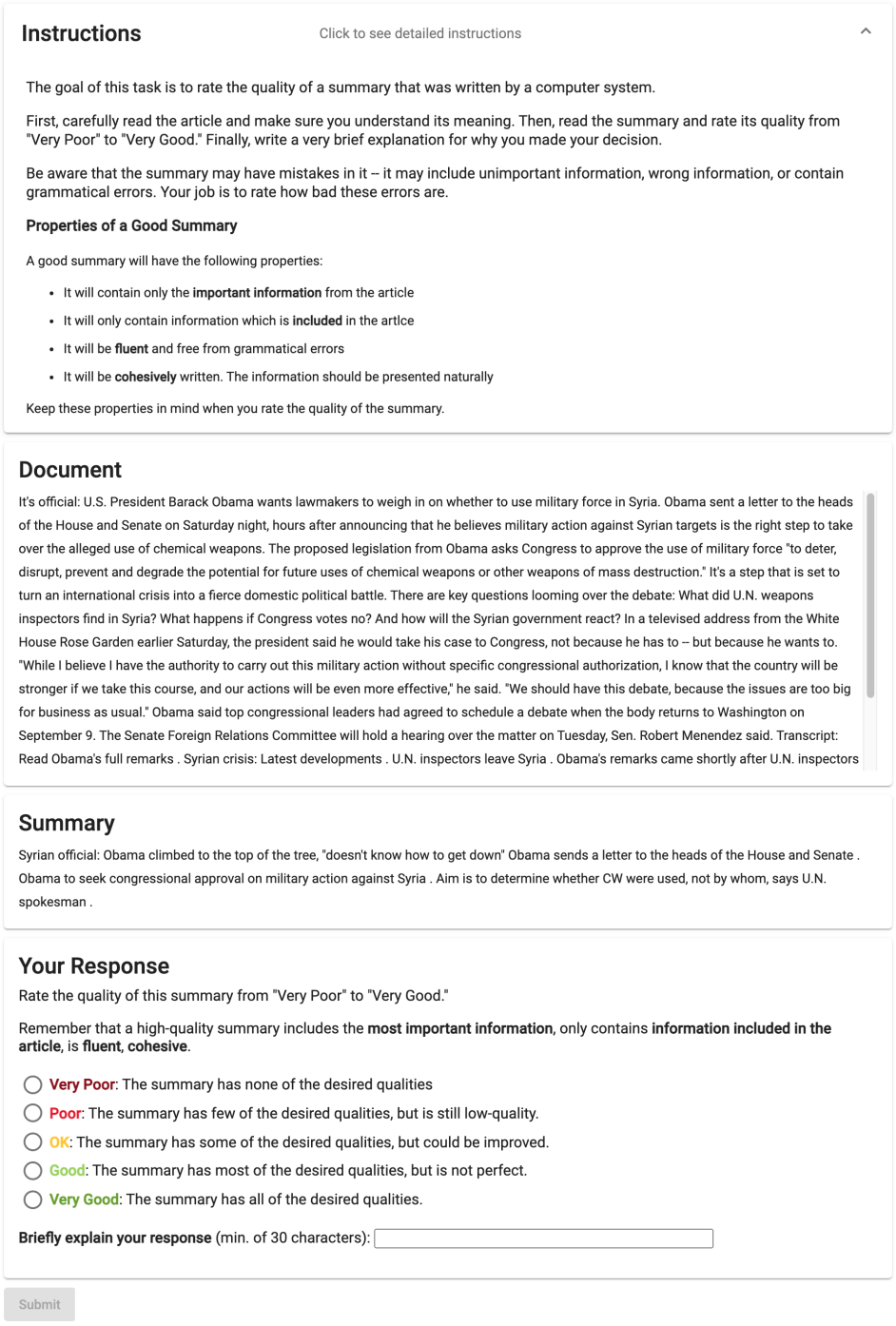

Figure 37: A screenshot of the tool we used for annotating summary quality on Mechanical Turk.

Metric	TAC'08		Fabbri et al.		Bhandari et al.	
	r_{SUM}	r_{SYS}	r_{SUM}	r_{SYS}	r_{SUM}	r_{SYS}
Resp/Rel/Pyr	100.0	0.00	32.0	0.52	75.0	0.84
AutoSummENG	18.8	0.26	33.0	0.01	28.0	0.55
MeMoG	37.5	0.53	33.0	0.01	28.0	0.55
NPowER	29.2	0.36	33.0	0.01	28.0	0.55
BERTScore	35.4	0.00	26.0	0.15	28.0	0.18
BEwTE	22.9	0.06	37.0	0.00	33.0	0.68
METEOR	27.1	0.15	27.0	0.00	30.0	0.61
MoverScore	47.9	0.25	35.0	0.00	31.0	0.50
QAEval-F_1	58.3	0.00	40.0	0.01	45.0	0.21
ROUGE-1	33.3	0.06	32.0	0.00	30.0	0.91
ROUGE-2	31.2	0.71	34.0	0.00	61.0	0.62
ROUGE-L	25.0	0.13	26.0	0.13	37.0	0.12
ROUGE-SU4	29.2	0.44	32.0	0.00	44.0	0.84
S3	20.8	0.32	26.0	0.00	47.0	0.66

Table 22: For r_{SYS} the p-value of the Shapiro-Wilk test. For r_{SUM}, the percent of the per-input document tests which had a significant result at $\alpha = 0.05$. A significant p-value means H_0 (the data is distributed normally) is rejected. For r_{SUM}, the larger the percentage the more the data appears to be not normally distributed.

C. Appendix for Chapter 7

C.1. Normality Testing

To understand if the normality assumption holds for summarization data we ran the Shapiro-Wilk test for normality (Shapiro and Wilk, 1965), which was reported to have the highest power out of several alternatives (Razali and Wah, 2011; Dror et al., 2018, 2020). The results of the tests for the ground-truth responsiveness scores and automatic metrics are in Table 22. Most of the p-values are significant, i.e., applying a statistical test which assumes normality is incorrect in general.

C.2. Extended Bonferroni Correction

Fig. 38 contains the results from the pairwise hypothesis tests (§7.6.2) when then Bonferroni correction is applied to set of p-values grouped by the dataset and correlation level pair instead of each dataset, correlation level, and metric shown in Fig. 29. The results are overall very similar with only a handful of results now becoming not significant.

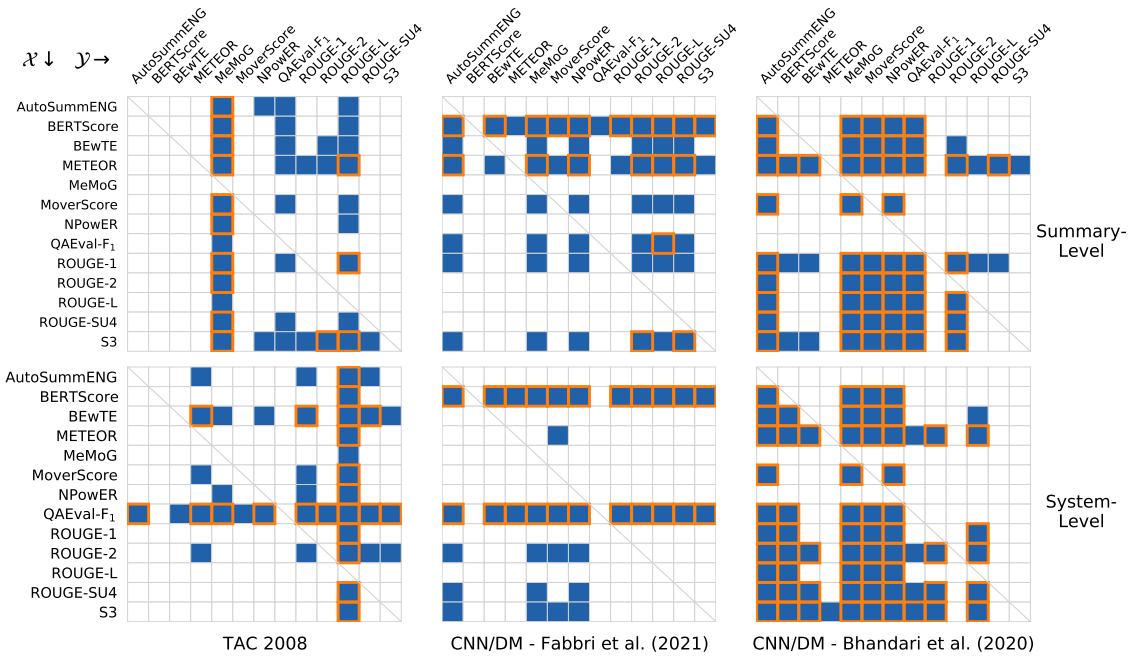

Figure 38: The results of running the PERM-BOTH hypothesis test to find a significant difference between metrics' Pearson correlations with the Bonferroni correction applied per dataset and correlation level pair instead of per metric (as in Fig. 29). A blue square means the test returned a significant p-value at $\alpha = 0.05$, indicating the row metric has a higher correlation than the column metric. An orange outline means the result remained significant after applying the Bonferroni correction.

D. Appendix for Chapter 8

D.1. Additional Confidence Interval Results

In addition to the BOOT-INPUTS CI method proposed by Deutsch et al. (2021b), the authors also proposed BOOT-SYSTEMS and BOOT-BOTH. Each of the three methods makes assumptions about whether the set of N systems and M input documents are fixed or variable during the bootstrapping calculation. For instance, BOOT-INPUTS assumes the N systems are always the same and the M input documents are random, then subsequently resamples M input documents on each bootstrapping iteration to calculate the confidence interval. BOOT-SYSTEMS does the opposite by resampling which N systems are used while holding the original M input documents fixed. BOOT-BOTH assumes both the systems and inputs are variable.

Figs. 39 and 40 contain the 95% CIs for ROUGE, BERTScore, and QAEval on the SummEval and

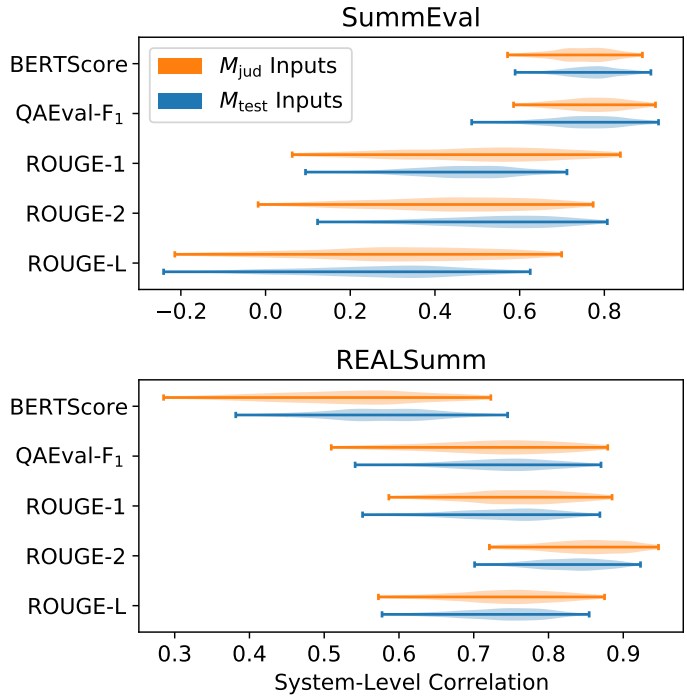

Figure 39: The 95% CIs calculated using the BOOT-SYSTEMS bootstrapping method with M_{jud} summaries in orange and M_{test} in blue.

REALSumm datasets using the BOOT-SYSTEMS and BOOT-BOTH methods calculated using all M_t test instances and only the M_a annotated instances (BOOT-INPUTS included in the main body of the paper, Fig. 32). The widths of the BOOT-BOTH CIs decreased by 14% and 12%, whereas the BOOT-SYSTEMS CIs only decreased by 1% and 6%.

The BOOT-SYSTEMS widths likely decreased less because its estimation of r_{SYS} is not dependent on the variance of the system score estimates. Since the set of M input documents is fixed, the system scores do not change at all during bootstrapping, so increasing the number of summaries used to estimate those scores should not have a major effect on the estimation of r_{SYS}.

D.2. Additional $r_{\text{SYS}}\Delta(\ell, u)$ Results

Fig. 41 contains the $r_{\text{SYS}}\Delta(\ell, u)$ correlations for when $\ell = 0$ for ROUGE-1, ROUGE-2, and ROUGE-L, equivalent to those shown in Fig. 34 in the main body of the paper (ROUGE-1 is shown in both). The ROUGE-2 and ROUGE-L results are largely consistent with those of ROUGE-1. The metrics' correlations to human annotations are low (or even negative) when the differences between

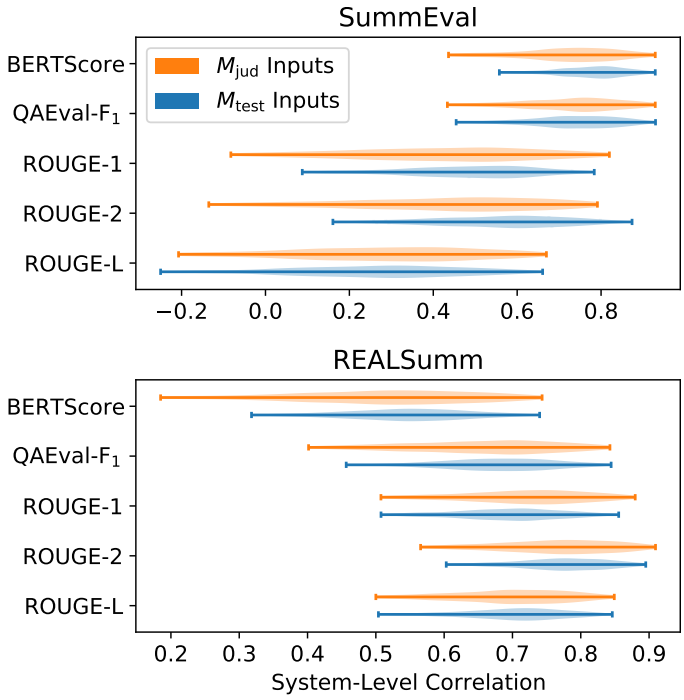

Figure 40: The 95% CIs calculated using the BOOT-BOTH bootstrapping method with M_{jud} summaries in orange and M_{test} in blue.

system scores are small. As more pairs are added that differ by larger margins, the correlations increase.

Figs. 42 and 43 contain the $r_{\text{SYS}}\Delta(\ell, u)$ correlations for ROUGE, BERTScore, and QAEval for various combinations of ℓ and u on both the SummEval and REALSumm datasets. The first rows of each heatmap are plotted in Figs. 34 and 41.

We see that as the allowed score gap between system pairs is allowed to increase (i.e., adding "easier" pairs to rank), the correlation increases by a large margin over the correlation on pairs close in score. All of the metrics have nearly perfect correlation when the system pairs are separated by large margins.

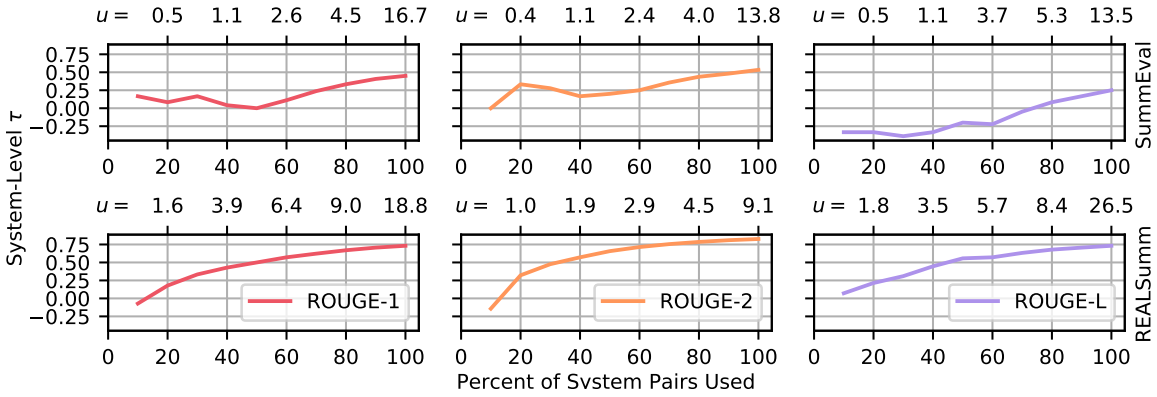

Figure 41: The $r_{\text{SYS}}\Delta(\ell, u)$ correlations on the SummEval (top) and REALSumm (bottom) datasets for $\ell = 0$ and various values of u for ROUGE-1, ROUGE-2, and ROUGE-L. The u values were chosen to select the 10%, 20%, $\ldots, 100\%$ of the pairs of systems closest in score. Each u is displayed on the top of each plot.

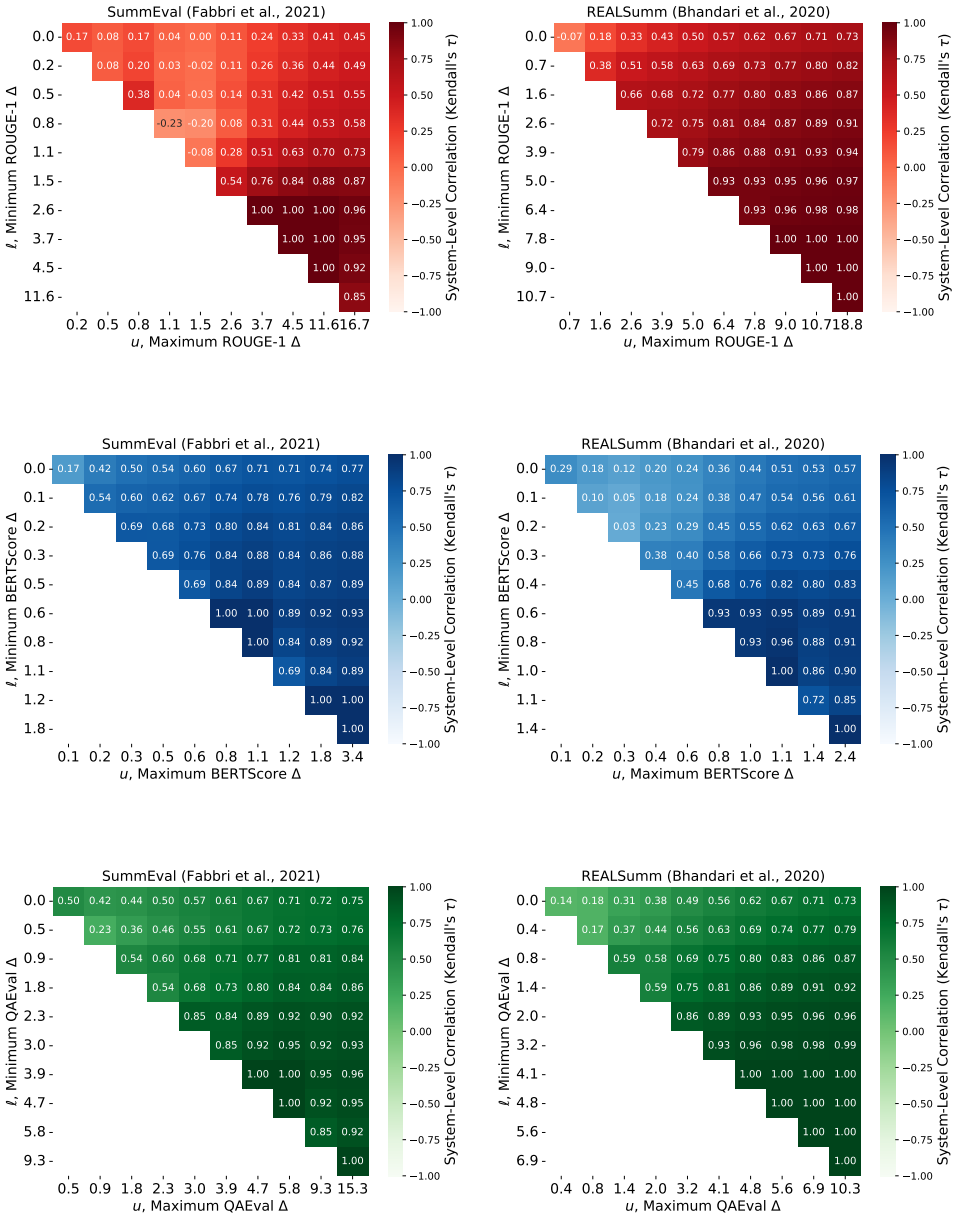

Figure 42: $r_{\text{SYS}}\Delta(\ell, u)$ correlations for various combinations of ℓ and u (see §8.4.2) for ROUGE (top), BERTScore (middle), and QAEval (bottom) on SummEval (left) and REALSumm (right). The values of ℓ and u were chosen so that each value in the heatmaps evaluates on 10% more system pairs than the value to its left. For instance, the first row evaluates on $10\%, 20\%, \ldots, 100\%$ of the system pairs. The second row evaluates on $10\%, 20\%, \ldots, 90\%$ of the system pairs, never including the 10% of pairs which are closest in score. The first row of each of the heatmaps is plotted in Fig. 34. The correlations on realistic score differences between systems are in the upper left portion of the heatmaps and contain the lowest correlations overall. Evaluating on all pairs is the top-rightmost entry, and the "easiest" pairs (those separated by a large score margin) are in the bottom right.

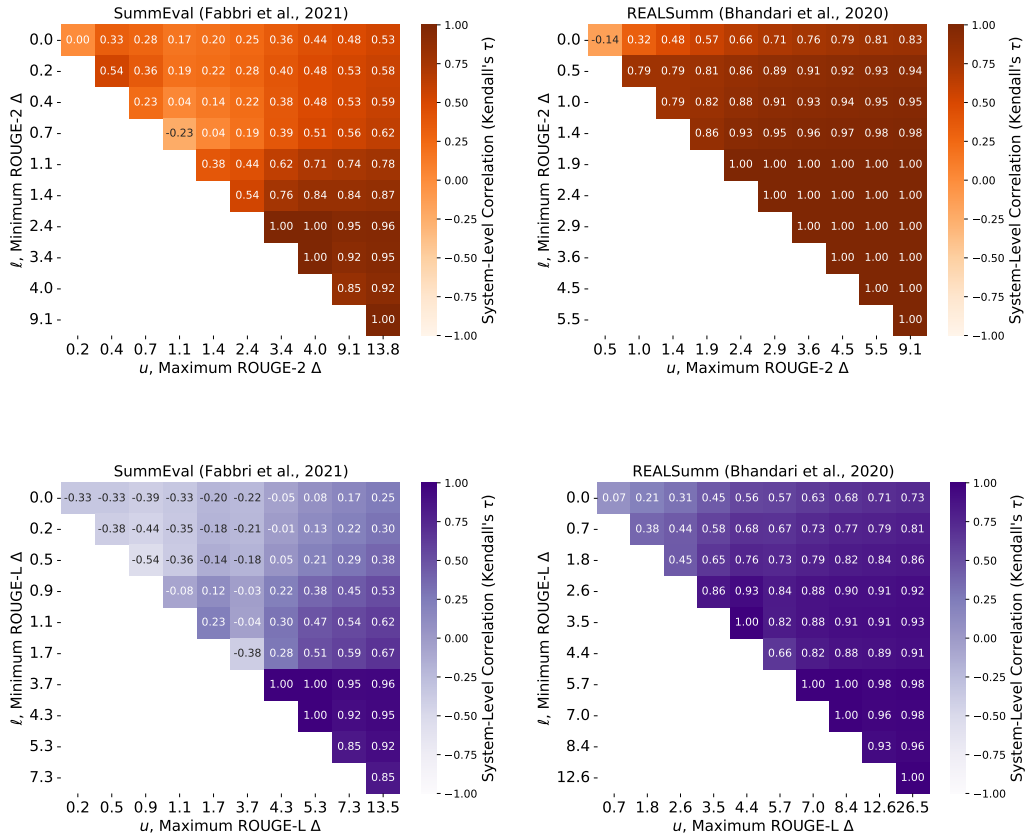

Figure 43: See Fig. 42 for a description of the heatmaps, shown here for ROUGE-2 (top) and ROUGE-L (bottom).

BIBLIOGRAPHY

Sweta Agrawal, George Foster, Markus Freitag, and Colin Cherry. 2021. Assessing Reference-Free Peer Evaluation for Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1158–1171, Online. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina Espa na Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Kristjan Arumae and Fei Liu. 2018. Reinforced Extractive Summarization with Question-Focused Rewards. In *Proceedings of ACL 2018, Student Research Workshop*, pages 105–111, Melbourne, Australia. Association for Computational Linguistics.

Kristjan Arumae and Fei Liu. 2019. Guiding Extractive Summarization with Question-Answering Rewards. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2566–2577, Minneapolis, Minnesota. Association for Computational Linguistics.

Erfan Sadeqi Azer, Daniel Khashabi, Ashish Sabharwal, and Dan Roth. 2020. Not All Claims are Created Equal: Choosing the Right Statistical Approach to Assess Hypotheses. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5715–5725, Online. Association for Computational Linguistics.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating Evaluation in Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. 2011. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. 2016. Ten Years of WMT Evaluation Campaigns: Lessons Learnt. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

D. Bonett and T. A. Wright. 2000. Sample Size Requirements for Estimating Pearson, Kendall and Spearman correlations. *Psychometrika*, 65:23–28.

Carlo E. Bonferroni. 1936. *Teoria Statistica Delle Classi e Calcolo Delle Probabilita*. Libreria internazionale Seeber.

Chris Callison-Burch. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, Singapore. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532, Online. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

John M. Conroy and Hoa Trang Dang. 2008. Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 145–152, Manchester, UK. Coling 2008 Organizing Committee.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 Update Summarization Task. In *Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008*. NIST.

Hoa Trang Dang and Karolina Owczarzak. 2009. Overview of the TAC 2009 Summarization Track. In *Proceedings of the Text Analysis Conference*.

Sashka T Davis, John M Conroy, and Judith D Schlesinger. 2012. OCCAMS–An Optimal Combinatorial Covering Algorithm for Multi-Document Summarization. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 454–463. IEEE.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming Question Answering Datasets Into Natural Language Inference Datasets. *ArXiv*, abs/1809.02922.

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021a. Towards Question-Answering as an Automatic Metric for Evaluating the Content Quality of a Summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021b. A Statistical Analysis of Summarization Evaluation Metrics Using Resampling Methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022a. On the Limitations of Reference-Free Evaluations of Generated Text. In *In Submission*.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022b. Re-Examining System-Level Correlations of Automatic Summarization Evaluation Metrics. In *In Submission*.

Daniel Deutsch and Dan Roth. 2020. SacreROUGE: An Open-Source Library for Using and Developing Summarization Evaluation Metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2021. Understanding the Extent to which Content Quality Metrics Measure the Information Quality of Summaries. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 300–309, Online. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2022a. Benchmarking Answer Verification Methods for Question Answering-Based Summarization Evaluation Metrics. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Dublin, Ireland. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2022b. Incorporating Question Answering-Based Signals into Abstractive Summarization via Salient Span Selection. In *In Submission*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability Analysis for Natural Language Processing: Testing Significance with Multiple Datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical Significance Testing for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 13(2):1–116.

Olive Jean Dunn and Virginia Clark. 1971. Comparison of Tests of the Equality of Dependent Correlation Coefficients. *Journal of the American Statistical Association*, 66(336):904–908.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Bradley Efron and Robert J Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC press.

Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of artificial intelligence research*, 22:457–479.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question Answering as an Automatic Eval-

uation Metric for News Article Summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander Fabbri, Wojciech Kryscinski, Bryan McCann, R. Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Ronald Aylmer Fisher. 1992. Statistical Methods for Research Workers. In *Breakthroughs in Statistics*, pages 66–70. Springer.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 Shared Tasks on Quality Estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards New Frontiers in Unsupervised Evaluation Metrics for Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics.

Yanjun Gao, Chen Sun, and Rebecca J. Passonneau. 2019. Automated Pyramid Summarization Evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 404–418, Hong Kong, China. Association for Computational Linguistics.

George Giannakopoulos and Vangelis Karkaletsis. 2010. Summarization System Evaluation Variations Based on N-Gram Graphs. *Text Analysis Conference*.

George Giannakopoulos and Vangelis Karkaletsis. 2013. Summary Evaluation: Together We Stand NPowER-ed. In *Computational Linguistics and Intelligent Text Processing*, pages 436–450, Berlin, Heidelberg. Springer Berlin Heidelberg.

George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization System Evaluation Revisited: N-Gram Graphs. *ACM Trans. Speech Lang. Process.*, 5(3).

Dan Gillick and Yang Liu. 2010. Non-Expert Evaluation of Summarization Systems is Risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151, Los Angeles. Association for Computational Linguistics.

Yvette Graham. 2015. Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Lisbon, Portugal. Association for Computational Linguistics.

Yvette Graham and Timothy Baldwin. 2014. Testing for Significance of Increased Correlation with Human Judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 172–176, Doha, Qatar. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. CTRLsum: Towards Generic Controllable Text Summarization.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tsutomu Hirao, Hidetaka Kamigaito, and Masaaki Nagata. 2018. Automatic Pyramid Evaluation Exploiting EDU-based Extractive Reference Summaries. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4186, Brussels, Belgium. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eduard H. Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006*, pages 899–902. European Language Resources Association (ELRA).

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. SemSUM: Semantic Dependency Guided Neural Abstractive Summarization. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Reno Kriz, Marianna Apidianaki, and Chris Callison-Burch. 2020. Simple-QE: Better Automatic Quality Estimation for Text Simplification. *arXiv preprint arXiv:2012.12382*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Adam Lopez. 2012. Putting Human Assessments of Machine Translation Systems in Order. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 1–9, Montréal, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proc. of the International Conference on Learning Representations*.

Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):267–300.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot. 2018. Reference-less Quality Estimation of Text Simplification Systems. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 29–38, Tilburg, the Netherlands. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 Metrics Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725.

Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. 2020. Improving Truthfulness of Headline Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1335–1346, Online. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. 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*, pages 2727–2733, Online. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. 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*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Shashi Narayan, Yao-Dong Zhao, Joshua Maynez, Gonccalo Simoes, Vitaly Nikolaev, and Ryan T. McDonald. 2021. Planning with Learned Entity Prompts for Abstractive Summarization.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Eric W Noreen. 1989. Computer Intensive Methods for Hypothesis Testing: An Introduction. *Wiley, New York*, 19:21.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An Assessment of the Accuracy of Automatic Evaluation in Summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9, Montréal, Canada. Association for Computational Linguistics.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the TAC 2011 Summarization Track: Guided Task and AESOP Task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated Pyramid Scoring of Summaries using Distributional Semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–147, Sofia, Bulgaria. Association for Computational Linguistics.

Rebecca J Passonneau, Ani Nenkova, Kathleen McKeown, and Sergey Sigelman. 2005. Applying

the Pyramid Method in DUC 2005. In *Proceedings of the document understanding conference (DUC 05), Vancouver, BC, Canada*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Maxime Peyrard. 2019. Studying Summarization Evaluation Metrics in the Appropriate Scoring Range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy. Association for Computational Linguistics.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to Score System Summaries for Better Content Selection Evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Mark Przybocki, Kay Peterson, and Sebastian Bronsart. 2008. Official Results of the NIST 2008 "Metrics for MAchine TRanslation" Challenge. In *AMTA-2008 Workshop on Metrics for Machine Translation*, Honolulu, Hawaii.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-Text Generation with Content Selection and Planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Peter Rankel, John Conroy, Eric Slud, and Dianne O'Leary. 2011. Ranking Human and Machine Summarization Systems. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 467–473, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2013. A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–136, Sofia, Bulgaria. Association for Computational Linguistics.

Peter A. Rankel, John M. Conroy, and Judith D. Schlesinger. 2012. Better Metrics to Automatically Predict the Quality of a Text Summary. *Algorithms*, 5(4):398–420.

Nornadiah Mohd Razali and Yap Bee Wah. 2011. Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests. *Journal of Statistical Modeling and Analytics*, 2(1):21–33.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André FT Martins, and Alon Lavie. 2021. Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task. In *Proceedings of the Sixth Conference on Machine Translation, Online. Association for Computational Linguistics*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient Elicitation of Annotations for Human Evaluation of Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA. Association for Computational Linguistics.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization Asks for Fact-based Evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. 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)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing Lightweight Pyramids for Manual Summary Evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Ori Shapira, Hadar Ronen, Meni Adler, Yael Amsterdamer, Judit Bar-Ilan, and Ido Dagan. 2017. Interactive Abstractive Summarization for Event News Tweets. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 109–114, Copenhagen, Denmark. Association for Computational Linguistics.

Samuel Sanford Shapiro and Martin B Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52(3/4):591–611.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 Shared Task on Quality Estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Stephen Tratz and Eduard Hovy. 2008. BEwT-E: Basic Elements with Transformations for Evaluation. In *TAC 2008 Workshop*.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The Statistical Advantage of Automatic NLG Metrics at the

System Level. 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)*, pages 6840–6854, Online. Association for Computational Linguistics.

Daniela Brook Weiss, Paul Roit, Ayal Klein, Ori Ernst, and Ido Dagan. 2021. QA-Align: Representing Cross-Text Content Overlap by Aligning Question-Answer Propositions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9879–9894, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Frank Wilcoxon. 1992. Individual Comparisons by Ranking Methods. In *Breakthroughs in Statistics*, pages 196–202. Springer.

Evan James Williams. 1959. *Regression Analysis*, volume 14. Wiley.

Stratos Xenouleas, Prodromos Malakasiotis, Marianna Apidianaki, and Ion Androutsopoulos. 2019. SUM-QE: a BERT-based Summary Quality Estimation Model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6005–6011, Hong Kong, China. Association for Computational Linguistics.

Qian Yang, Rebecca J. Passonneau, and Gerard de Melo. 2016. PEAK: Pyramid Evaluation via Automated Knowledge Extraction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2673–2680. AAAI Press.

Shiyue Zhang and Mohit Bansal. 2021. Finding a Balanced Degree of Automation for Summary Evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. 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*, pages 718–733, Online. Association for Computational Linguistics.